Utrecht University

U.S.E. Master Thesis

From School to Work:

The Effects of Secondary Vocational Education on Employment and Earnings in Low- and Middle-Income Countries

By: Xavier J.J.C. Friesen | S8113860 | x.j.j.c.friesen@students.uu.nl

Supervised by Dr. S. Matthewes

Abstract: This paper examines the impact of secondary vocational education (TVE) on labour outcomes in low and middle-income countries. Using data from the Young Lives study across Peru, India, Vietnam, and Ethiopia, this paper estimates the effects of TVE through propensity score-weighted regressions, controlling for selection biases using an unprecedented range of baseline characteristics. I find that attending TVE provides no benefits over attending general secondary education. However, I argue the unique value of TVE lies in its ability to provide a different educational pathway, predicting that without TVE over 54% of vocational students would have dropped out after primary. This is an important effect, as attending TVE compared to dropping out after primary school brings major labour market advantages, although only for female students: female vocational students are 50% more likely to be employed, 68% more likely to have a formal job than dropouts, and work 8 hours more per week. These large effects may justify investments in secondary vocational education. For practitioners, this implicates that the general procedure of evaluating TVE impact by directly comparing general with vocational secondary students is shortsighted. Instead, the more critical question is how effective a vocational secondary school is in helping potential dropouts continue their secondary education: for those students TVE is of greatest value.

JEL: I25: Education and Economic Development | I26: Returns to Education | C21: Treatment Effect Model

Keywords: Secondary Vocational Education; Returns to Education; Low- and Middle-Income Countries; Technical and Vocational Education; Propensity Score Weighting; Average Marginal Effects

The author would like to explicitly thank dr. S. Matthewes and dr. K. Moreno for their feedback and guidance on earlier drafts.

The author received no financial support for this paper. The author is currently an advisor to a foundation which intends to support the construction of a tertiary vocational school in Africa. This side-activity had no influence on the findings in this paper.

List of appendices

List of Tables

List of Figures

1. Introduction

This research paper provides an empirical analysis of the effects of secondary vocational education on labour market outcomes and transitioning to tertiary education in four low- and middle-income countries: India, Ethiopia, Vietnam, and Peru. More than two-thirds of the world's youth lack the basic skills necessary for effective economic participation, significantly contributing to global inequality (Gust et al., 2024). Technical and Vocational Education (TVE) has historically been a major tool for narrowing this skill divide (Bennell, 2023). Unlike traditional general education, which broadens general knowledge, vocational training equips students with specific, employer-sought skills, making them ready to enter the workforce. This potential has led international organisations, including the World Bank, to invest heavily in TVE across developing countries (World Bank, 2023). The perceived importance of TVE is further exemplified by its inclusion in two sub-goals of Sustainable Development Goal 4: quality education (United Nations, 2024). One popular type of TVET, and the focus of this study, is secondary vocational schools. Within such an educational system, secondary students can choose between attending a general school or investing extra time in learning vocational skills at a vocational secondary school.

The effectiveness and cost-efficiency of vocational secondary schools remain subjects of significant debate (World Bank, 2023). Providing vocational education is notably more expensive for governments than general education (Patrinos & Psacharopoulos, 2020). Impact evaluations attempt to justify these large investments by demonstrating that attending secondary vocational education improves labour outcomes more than if the same students had attended general education. However, the limited studies on this topic offer mixed results—some indicating marginal benefits and others showing no additional advantages (Campuzano et al., 2016; Borkum et al., 2017; Field, 2019; Acevedo, 2020).

This study's primary objective is to evaluate the impact of secondary vocational education on labour outcomes in India, Ethiopia, Vietnam, and Peru—countries where the effects of the TVE system have not been systematically studied. Utilizing a uniquely comprehensive longitudinal dataset, this study will be the first to control for motivation, cognitive and non-cognitive skills, as well as household and community characteristics, thereby enhancing the robustness of the selection-on-observables assumption, allowing for causal inferences. The analysis compares the impact of secondary vocational education relative to general secondary education, as well as to those who dropped out after primary school. Previous observational research only used general secondary students as the counterfactual, but that leaves out key information. I show that many vocational students, up to 54.6% in the sample, would likely have dropped out if vocational secondary education had not been available. This is because vocational secondary requires a different skill set and motivation, with many students being demotivated for general education anticipating they would fail their final exams anyway. For these students, dropping out is a more realistic counterfactual. This insight is crucial for future impact evaluations of vocational schools: vocational schools offer an alternative pathway to secondary education, meaning both counterfactuals should be considered.

Initial research from the 1970s and 1980s was sceptical about the efficiency of vocational secondary, finding similar rates of return compared to general education despite higher costs (Tilak, 1988; Psacharopoulos, 1987, 1993; IBRD, 1995). However, these studies failed to account for selection bias, which can now be tackled using advanced statistical techniques (see Bennell, 1996 for an extensive discussion). Yet research on the effectiveness of formal TVE is limited, and with most articles focussing on tertiary education, research specifically estimating the effects of secondary vocational education is even more rare. The studies that do exists suggests very moderate positive impacts on employment and earnings on average, although many studies still fail to find a positive significant effect (e.g., Camargo et al., 2018; Field et al., 2019; World Bank, 2023). Particularly noteworthy are the large effects among vulnerable groups, such as women in countries like Nepal, Liberia, and Uganda, underscoring the potential of vocational training to transform lives (Adoho et al., 2014; Camargo et al., 2018; Bandera et al., 2020). This paper supplements the limited existing research by evaluating formal, vocational secondary programs in four countries where effectiveness has not been analysed. Furthermore, it gives a different perspective by also considering the benefits associated with vocational secondary as an alternative educational pathway. Finally, I investigate the treatment heterogeneity, analysing how time, personal characteristics and community characteristics impact vocational education.

This study finds no significant differences between general and vocational secondary in wages, likelihood of employment or type of jobs. Only in India, vocational students work slightly more hours and have a marginally higher employment rate. However, this does not provide the full picture. Despite the higher costs of TVE, vocational secondary can be justified by its role in reducing dropout rates. The analysis predicts that 54.6% of vocational students would likely have dropped out if TVE was not existent. I show attending TVE can have significant benefits over dropping out after primary. On average, attending TVE makes one 28% more likely to get a formal job and 34% less likely to be self-employed. However, the effects are highly heterogeneous, being much more effective across certain sub-groups. Most importantly, vocational education is particularly effective among females, where effect sizes are much larger. Females attending secondary vocational schools work 0.3 standard deviations more hours per week, earn a 0.14 standard deviations higher hourly wage, are 50% more likely to have a paying job, and are 68% more likely to have a formal job. In contrast, TVE has no significant advantages for males. Similarly, significant effect sizes are found for TVE in Ethiopia. These are the magnitude of benefits that may very well outweigh the extra costs of vocational secondary education.

Previous studies evaluating TVE impact would have underestimated these effects, as they did not use dropouts as a counterfactual, despite many of those students likely having dropped out if vocational secondary schools were not available. TVE's role as an alternative educational pathway should be emphasised by practitioners evaluating TVE's impact. Higher secondary enrolment not only improves private labour outcomes but also may provide broader societal benefits. To accurately evaluate the impact of TVE, practitioners should also assess how likely their students were to drop out had there not been a secondary vocational school. To improve the impact of TVE, practitioners could consider focusing on how TVE can support and attract those at risk of dropping out, especially females, where TVE has the most potential. Vocational education is not a silver bullet, but if properly used to supplement general education, it can be a powerful tool. By emphasizing TVE's role in reducing dropout rates and targeting at-risk students, practitioners can maximize its effectiveness and ensure that the benefits outweigh the costs.

This paper will continue by discussing the most important literature on the impact of vocational schools. Then it will describe the Young Lives survey data, the operationalisation of the variables and balance in the data. The following chapter will outline the empirical strategy employing propensity score weighting with multiple imputed data. Then, I will estimate the results, starting with linear regressions and moving on to propensity score-based methods, also including analyses of heterogeneous effects and effects over time. Finally, I will summarize the implications of this paper for further research and development agendas. The Appendices include supporting tables and figures and robustness checks using an machine-learning algorithm (GBM) to calculating the propensity score.

2. A Review of the Literature

This paper focuses exclusively on formal vocational programmes, referred to as TVE. For the purposes of this review, I define TVE as formal education at a secondary or tertiary level, that is predominantly taught in a classroom rather than at a firm/through an internship and aims to equip individuals with the essential skills and knowledge required for specific sectors. It is important to clearly distinguish formal TVE from short-term, active labour market interventions targeting unemployed youth, which are often confusingly included under the broader TVE umbrella. Formal TVE differs in several keyways from short-term training programmes. Formal TVE is integrated into the regular educational system at both secondary and tertiary levels and awards a recognised diploma upon graduation. Unlike short-term TVE programmes, which typically target the unemployed, formal TVE is open to all students, generally attracting a younger population who have never worked before. Additionally, formal TVE programmes typically last much longer, often spanning several years, compared to short-term training programmes that last only a few months. The curricula also differ significantly: while short-term training programmes focus exclusively on job-specific skills, formal TVE can incorporate a broader range of general skills within its curriculum due to its longer duration. Unless explicitly mentioned, the evidence from this review refers to formal TVE secondary and tertiary schools, as there is insufficient research available to only focus on secondary level programs.

2.1. Average Effects of TVE

Across impact evaluations of formal TVE programs in LMICs, the estimated returns to vocational education still differed widely. Four out of six randomised controlled trials (RCTs) report large positive effects on employment and earnings, although sometimes only for women, while two others found no difference with attending general education (derived from World Bank, 2023).[1] Three studies specifically focussed on secondary vocational. In Mongolia, TVE increased the likelihood of having a paid job after one year by four percentage points and the chance of keeping that job by nine percentage points (Field, 2019). In Brazil, attending technical education boosted employment among women by twenty-one percentage points and earnings by over 50%, whereas it did not yield significant benefits for men (Camargo et al., 2018). The effects are likely magnified due to the vulnerable target

[1] Large positive impacts: Field et al., 2019 (Mongolia), Chakravarty et al., 2019, especially among women (Nepal), Camargo et al., 2018, but only for women (Brazil), Hicks et al. 2016 (Kenya), but only among wage earners. No effects: Borkum et al., 2017 (Namibia), Campuzano et al., 2016 (El Salvador)

group of low-educated women without alternative employment options. However, success is not universally guaranteed. TVE did not increase employment or earnings one year after graduation in El Salvador, potentially due to very low labour demand for qualified workers (Campuzano et al., 2016). Three more RCTs were done focussing on adult or tertiary education. TVE-impacts were extraordinarily large in Nepal, increasing non-farming employment levels by 31% among compliers (Chakravarty et al., 2019), but were negligible in Namibia (Borkum et al., 2017). Or perhaps the positive impacts had yet to materialize, as was the case in Kenya, where impacts were initially negative and then turned positive after a year (Hicks et al., 2016).

Expanding the analysis to include observational studies gives equivalent results. Four papers find vocational secondary to be much more effective than general secondary, three find vocational to have a small edge, and seven find no difference or a small advantage for general secondary (derived from World Bank, 2023).[2] However, if only picking modern studies that explicitly control for selection bias one argues vocational secondary is much more effective, one study finds only positive effects among vocational students that did not continue to tertiary education and four argue there is no difference between vocational and general education. Still, even if there is no difference between general and vocational education, the private returns of attending secondary education in developing countries remains high: estimated at 18.7%, double the rate seen in advanced economies (Patrinos & Psacharopoulos, 2020).

2.2.Effects over Time

Experiments suggest that the effects of vocational education on labour outcomes grow from the short to medium term (Card, Kluve & Weber, 2010; Chakravarty et al., 2019). Training programs are generally ineffective in the short term (under one year) but start having meaningful effects between one and two years after the intervention (Card et al., 2010; Hicks,

[2] **Controlling for selection bias:** ***Large differences:*** Guo and Wang, 2020 (China). ***Mixed results:*** Vandenberg and Laranjo, 2020, only positive effects among those who did not continue to tertiary education (Philippines). ***No differences:*** Malamud and Pop-Eleches, 2010 (Romania), Kahyarara and Teal, 2008 (Tanzania). S***tudies finding general education more effective than vocational education***: Kraft, 2018 (Egypt)

Not controlling for selection bias: ***Large differences*** between vocational and general education defined as 8% additional positive effect on employment or wages, or more: El-Hamidi, 2006 (Egypt); Moenjak and Worswick, 2003 (Thailand); Almeida et al., 2007 (Brazil) and Guo and Wang, 2020 (China). ***Small positive significant difference or mixed results***: Patrinos, Psacharopoulps and Tansel, 2019 (Turkey), Newhouse and Suryadarma, 2011, only positive effects for women (Indonesia), ***Studies finding no significant difference***: Mahirda and Wahyuni, 2016 (Indonesia) ***Studies finding general education more effective than vocational education***: Lassibille and Tan, 2005 (Rwanda), and Horowitz and Schenzler, 1999 (Suriname)

2016; Ibarrarán, 2019). This delay might be attributed to vocational training making trainees, especially men, pickier while searching for a job. It then takes a while before they adjust their beliefs (Acevedo et al., 2020; Banerjee & Sequeira, 2020).

Looking at the long-term perspective, evidence from developed economies suggests that the disparities between TVE and general education tend to diminish in the long-term (Golsteyn & Stenberg, 2017; Hanushek et al., 2017; Choi et al., 2019). The immediate labour market advantages provided by TVE, such as higher initial wages due to occupation-specific skills, gradually lose their edge as these skills become less relevant with technological evolution (Montenegro & Patrinos, 2014; Hanushek et al., 2017). However, the limited evidence from informal job-training programs suggests TVE-skills remain relevant longer in developing countries. For instance, in the Dominican Republic, training effects were still observable six years later (Ibarrarán, 2019), and Attanasio (2017) reported that even after ten years, trainees were earning 11.8% more than their counterparts. The long-term impacts made both programs cost-effective. Both programs are non-formal, so more research is needed before generalizing these conclusions to formal TVE.

2.3. The Gap: Heterogeneous Effects of TVE

The diversity of impacts found in previous studies signals heterogenous returns to vocational education. This underscores the importance of tailoring vocational training to the specific demands and contexts in which they are implemented, aligning closely with the labour market's needs and the characteristics of the student population (McKenzie, 2020). To do so, research into these heterogenous returns and their moderators is required, and that is currently lacking in the literature. Drawing from research insights from the wider educational literature and research in advanced countries, I aim to understand for whom and under which conditions TVE is most beneficial. When data is available, these arguments are later formally tested.

2.3.1. Individual Characteristics

Some studies find TVE to be more beneficial for females, while others report no significant gender difference (Newhouse & Suryadarma, 2011; Camargo, 2018; Field, 2019; Chakravarty, 2019). This may suggest gender serves as a proxy for earning potential, with larger returns to education for individuals from disadvantaged groups (Fasih et al., 2012; Arias et al., 2019; Chakravarty, 2019). For example, Nepalese women generally have low educational backgrounds and suffer from restrictive gender roles, suggesting they have much more to gain from TVE, and indeed Chakravarty (2019) found much larger impacts among women.

Cognitive abilities, particularly foundational skills like literacy and mathematics, may be critical moderators of TVE's impact. The World Bank (2023) and various studies (Psacharopoulos, 1993; Loyalka et al., 2016; Jakubowski, 2016) emphasise the need for students to be “school ready.” Following their reasoning, vocationals should have a minimum of foundational skills to be able to learn from vocational education. Studies show that returns to general education in LMICs increase with cognitive skills, although which particular type of skill was most important varied between regions (Nikolov et al., 2020; Ozawa et al., 2022).

Moreover, intrinsically motivated students generally perform better at school and translate their skills better into labour outcomes (Dunifon & Duncan, 1998; Ryan & Deci, 2000; Silliman & Virtanen, 2022). It is an open question how motivation influences vocational education specifically. Moreover, non-cognitive skills such as conscientiousness and social skills are linked to academic achievements, its effect on vocational education remains unclear due to a lack of data (Heckman et al., 2006; Cameiro et al., 2007; Brunello & Schlotter, 2011; Lipnevich & Roberts, 2012; Camargo et al., 2020).

2.3.2. Impact of Economic and Labour Market Conditions

Returns to vocational education are likely to vary between specialisations, with higher returns in fields with high local demand (Grave & Goerlitz, 2012; Arcidiacono et al., 2012; Nomura et al., 2015; Aydede & Orbay, 2016; Arias et al., 2019; Tran & Van Vu, 2020). In developing countries, however, there seems to be no consistent trend on which specialisations pay the best (e.g., Nomura et al., 2015; Aydede & Orbay, 2016; Arias, 2019). Presumably, local demand plays a larger role for vocational students, given their occupation-specific skills are less versatile, necessitating employment within their field of study (Nordin et al., 2010; Zhu, 2014; World Bank, 2023). Nonetheless, the impact of such "horizontal mismatches" has yet to be thoroughly explored for vocational training in developing countries.

Moreover, the economic conditions at the time of graduation, including business cycle fluctuations, may moderate the impact of vocational education. For instance, Field (2019) observed that vocational students in Mongolia faced varying impacts based on business cycle fluctuations. This observation aligns with broader findings that "recession graduates" face long-term negative impacts, although, at least in advanced economies, the negative effects for vocational education were smaller than for general education (Kahn, 2010; Oreopoulos, 2012; Van den Berge, 2018; Liu et al., 2014).

2.3.3. *Impact of TVE systems and Quality of Education*

Formal TVE-systems can broadly be classified into school-centred and dual systems that integrate education with apprenticeships. Currently, many low- and middle-income countries want to transition towards a dual system to capitalise on the benefits of workplace learning (Deissinger, 2014; Hagos Baraki & van Kemenade, 2013; Caicedo, 2022). Despite the clear theoretical advantages of dual systems, such as smoother transitions into the workforce, challenges in implementation persist (Deissinger, 2014; 2015). This may be why research finds no comparative advantage of dual TVE over school-centred TVE (Deissinger, 2015; Valiente et al., 2020; Vanderhoven et al., 2024).

Instead, the quality of education within developing countries tends to vary more within countries rather than between them (World Bank, 2023). Attracting enough high-quality teachers is a widespread problem due to competition with the industry (ADB, 2008; ILO and UNESCO, 2018; World Bank, 2023). Additionally, vocational schools frequently suffer from inadequate equipment, limited access to technology, and insufficient infrastructure (e.g., Hagos Barak, 2013; Akanbi, 2017; World Bank, 2023). These challenges likely diminish the effectiveness of vocational education as a lack of resources and insufficient number of teachers have been shown to lower general educational outcomes in LMICs (Kunter et al., 2013; Sirait, 2016; Canales & Maldonado, 2018). This may also explain why Bettinger et al., (2017) suggest that private schools outperform public ones due to their greater flexibility in adapting their curricula and better infrastructure. However, the evidence remains mixed: for example, studies on general education find no evidence of larger returns to private schools (Hicks et al., 2013; Glewwe & Muralidharan, 2016).

3. Data, Operationalisation and Validity

3.1.Description of the Dataset and its External Validity

I use longitudinal data from the Young Lives Survey, an international study on childhood poverty conducted by the University of Oxford's Department of International Development. The Young Lives study tracks the impact of poverty on children's well-being over a 15-year period in two cohorts across four developing countries: Ethiopia, India (specifically the regions of Andhra Pradesh and Telangana), Peru, and Vietnam (Boyden, 2016). The choice of countries included this study was determined by the availability of Young Lives data. In each country, the survey included approximately 1,000 individuals in the older cohort, aged 7-8 in 2002, and 2,000 children in the younger cohort, aged between 6 and 18 months in 2002

Table 1: Characteristics of the TVE-Systems Across the Four Countries

	Ethiopia	India	Peru	Vietnam
Vocational Education at an Upper-Secondary Level (ISCED 3)	**ISCED 3:** • Level 1 (one year, hairdressing, cooking, midwives, knitting) • Level 2 (two years, electrician, plumbing) • Level 3 (Nursing, Business Accountants) **Alternative** is general upper-secondary education. Afterwards it is possible to continue to tertiary TVE-education at level 4 and 5 in polytechnic colleges (4 years)	**ISCED 3:** • Senior Secondary Vocational (two years, wide range of skills) • Polytechnic Diploma (three years, wide range of skills) **Alternative** is Senior Secondary Academic. Afterwards it is possible to continue to tertiary TVE-education with a bachelor's in vocational education or advanced polytechnics diploma (4 years)	Technically no vocational education at **ISCED 3 level** exists, but in practice students enrol in 'Technician' (ISCED 4 level) without having completed secondary education. I treat thus Technician (2 years) the same as ISCED-level courses in other countries. Afterwards it is possible to continue to tertiary TVE at college training (2 or 3 years)	**ISCED 3:** • Secondary Vocational Education (three of four years). **Alternative** is general upper-secondary education. Afterwards it is possible to continue to tertiary TVE named college training (2 or 3 years)
Vocational Education at a lower-Secondary Level (ISCED 2)	Does not exist.	It is possible to enrol in vocational at ISCED 2 (Craftsman).	It is possible to enrol in vocational at ISCED 2 (CETPRO).	It is possible to enrol in vocational at ISCED 2, and then continue into ISCED 3 (four years)
Enrolment Rates (Official by the World Bank	7% of students enrolled in secondary school studies TVE in 2015 (World Bank, 2024a) Net secondary enrolment rates in any form of secondary are 31% (World Bank, 2024b)	3.5% of students enrolled in secondary school studies TVE in 2022 (World Bank, 2024a) Net secondary enrolment rate in any form of secondary is 62% in 2013 (World Bank, 2024b)	1.9% of students enrolled in secondary school studies TVE in 2022 (World Bank, 2024a) Net secondary enrolment rate in any form of secondary is 89% in 2018 (World Bank, 2024b)	9.4% of students enrolled in secondary school studies TVE in 2022 (World Bank, 2024a) Net secondary enrolment rate in any form of secondary is unknown (World Bank, 2024b)
Type of TVE	Dual TVE system, 70% of time should be spent in school, 30% as an apprenticeship, but there are insufficient apprenticeship spots	Dual TVE system, although students show little demand for apprenticeships, leaving many trainees spots empty.	Dual TVE system, with the last year mostly being an internship.	Dual TVE system, with the last year mostly being an internship.

Supply or demand driven	Supply-driven: The government determines the curriculum, specializations offered, and the number of students allocated to TVE-institutions based on the expected labour demand. Grades determine if you may access TVE-education	Demand-driven: students can choose whether they want to go to TVE and can choose their specialisation. There is mismatch between the specializations offered and jobs available due to weak linkage between TVE and industry.	Demand-driven: students can choose their own specialisations, and whether they want to go to TVE.	Mostly demand-driven, students can choose their own specialization. There is little cooperation between industry and TVE, but there are financial incentives to choose high-demanded jobs.
Private Costs	Public institutions are free of charge and are generally attended by students with lower educational outcomes. Private institutions are considered of higher quality but are costlier.	Student must pay for school fees and stationaries	Public schools are free, conditional on performance. Underperforming students may have their exemption for tuition fees can be waived.	Students must pay tuition fees, but these may be waived in public schools for students with lower socio-economic backgrounds.

Notes: Data from vocational educational levels available from TVET country profiles (UNESCO, 2024). Data from vocational enrolment from World Bank Gender data (World Bank, 2024a). Data on secondary enrolment rates from (World Bank 2024b). The other information is derived from Krishnan (2013) for Ethiopia, Agragal & Agragal (2017) for India, Tuan & Cuong (2019) for Vietnam and Gaentzsch & Zapata-Román (2020) for Peru.

Table 2: Proportion of Students Attending Each Educational Level per Country and Cohort

	Vocational Secondary		**General Secondary**		**Dropped Out Post Primary**		**Dropped Out Before Primary**	
	OC	YC	OC	YC	OC	YC	OC	YC
Ethiopia	140 (19.6%)	39 (5.4%)	111 (15.6%)	18 (2.5%)	258 (36.3%)	424 (59.1%)	202 (28.4%)	236 (32.9%)
India	118 (13.0%)	61 (6.9%)	437 (48.4%)	343 (38.8%)	212 (23.5%)	342 (38.7%)	136 (15.1%)	137 (15.5%)
Peru	36 (6.4%)	16 (1.43%)	361 (64.1%)	911 (81.9%)	23 (4.1%)	170 (15.3%)	143 (25.4%)	15 (1.3%)
Vietnam	47 (5.2%)	10 (1.05%)	486 (53.6%)	433 (45.7%)	295 (32.6%)	474 (15.3%)	78 (8.6%)	31 (3.7%)
Total	341	126	1395	1705	788	1410	559	419

Note: % is the proportion of observations within that specific country and cohort with this level of education, students that had not yet completed their last education by the last call in Wave 6 are excluded from these calculations.

(Barnett et al., 2013). To date, six waves of surveys have been conducted, with the latest administered via phone in 2020 and 2021. For this study, I gathered baseline characteristics in Waves 1, 2, and 3 for the older cohort and in Waves 1, 3, and 4 for the younger cohort. Outcome data were collected during the final two waves, after the students had completed their education.

The Young Lives data is uniquely suited for an observational study on the (heterogeneous) impacts of vocational education due to its comprehensive collection of baseline variables. This extensive array of variables, including proxies for ability and motivation, allows for effective control of selection biases. Another advantage of the Young Lives data is its coverage of four different countries. Grouping these countries together enhances the statistical power of the analysis. Additionally, a comparative approach allows for cross-country comparisons of TVE systems, strengthening external validity if similar effects are found across countries. The main limitation is the significant amount of missing data, which is discussed in the next section.

Table 1 describes the main characteristics of the TVE system in each country. Notably, official enrolment rates from the World Bank show that vocational enrolment is highest in Vietnam and Ethiopia, followed by India, and is much lower in Peru. Strikingly, these official numbers, especially for Ethiopia and India, are much lower than the enrolment rates reported in the Young Lives data, see Table 2. The latter is self-reported. Four potential explanations may account for this discrepancy. Firstly, Andhra Pradesh in India has historically been a frontrunner with much higher investments in vocational secondary education, leading to higher vocational enrolment. This could explain why regional enrolment rates are far above the country's average (Sanwal, 2019). Secondly, model estimates by the World Bank tend to be lower than those from country-specific research, suggesting systematic underestimation, but such country-specific research is lacking for the four countries discussed here (e.g., Fukunishi & Machikita, 2017; Vandenberg and Laranjo, 2020). Thirdly, it is likely that Young Lives approach of oversampling "poorer" households also resulted in a sample with more people attending vocational education, which is often associated with having a lower social-economic background. However, even together these are unlikely to explain the major discrepancy in Ethiopia. Instead, it is possible that some general "secondary" students, who opted for a school with a highly competence-based curriculum are counted as vocational secondary in the Young Lives survey. Indeed, Ethiopian general education already has a significant focus on vocational skills (Krishnan, 2013; Fukunishi & Machikita, 2017). While this problem cannot be fully resolved, it is not a major issue, as general vs. vocational education is a spectrum, and the impact of competence-based general education is still of major interest. Importantly, there seems to be little reason to suggest that respondents would be more likely to dishonestly claim they attended vocational education, considering general education is seen as more prestigious. Otherwise, there would be significant bias, but this appears to be highly unlikely.

Back to Table 1, the structure of the TVE system is largely similar across the four countries. All four have vocational secondary schools, generally lasting a minimum of two to three years. Additionally, except for Ethiopia, there is also vocational education at the lower-secondary level. The Young Lives survey does not distinguish between these levels, so lower-secondary vocational education is considered equivalent to upper-secondary. All four countries use a dual TVE system combining classroom instruction and internships. However, in India, classroom instruction is more dominant since internships are unpopular (Agragal & Agragal, 2017). India is also the only country where secondary vocational education is not free for those of lower socio-economic status. In Peru, waivers for school fees in Peru are conditional on satisfactory performance, and for Ethiopia and Vietnam vocational education is free. Ethiopia's TVE system is unique in being supply-driven, with the government determining the number of students allowed to choose a certain specialisation based on expected labour demand. In all other countries, students can choose their own specialisation.

3.2. Operationalisation of Variables

Firstly, I estimate the impact of enrolment in vocational secondary education (0 = no, 1 = yes). Enrolment is chosen as the variable because there is no reliable data on whether students successfully graduated. This likely decreases the treatment effects, as the benefits of vocational education cannot be signalled to prospective employers without a diploma. Still, an intention-to-treat estimate provides more valuable information for assessing effectiveness, as providing education to non-graduates also results in significant costs. In general, vocational secondary education is defined as attending vocational secondary at the ISCED-2 (lower-secondary) or ISCED-3 (upper-secondary) level. The Young Lives Data makes no distinction between the two. Specifically in Peru, students also tend to enrol in ‘Technician’ (defined as ISCED-4, or tertiary education) without first having completed a general secondary track (Gaentzsch & Zapata-Román, 2020). Thus, that specific track is also considered secondary vocational education.

For more detailed analysis, I divide the sample in two. First, following the majority of previous articles I compare outcomes of vocationals to those enrolled in the last grade of upper-secondary education.[3] Secondly, contrary to previous observational research, this data also explicitly compares vocationals with those dropped out after primary school, since dropping out is a likely counterfactual for many of the vocational students.[4]

I consider the following seven indicators as outcomes, collectively measuring the impact of vocational education on job type and quality. Firstly, to measure the quantity of work, I assess the average hours worked per week during the last month. Work is defined as any activity generating income, either monetary or in-kind, including informal work. Secondly, the Any IGA (Income-Generating Activity) indicator measures unemployment by determining whether the person participated in any income-generating activity in the last month. Thirdly, I measure the self-reported hourly wage, converted to US dollars using the exchange rate at the interview date. If paid in kind, the assumed value of the product is considered salary. Both hourly wage and hours worked per week are susceptible to measurement errors as they are continuous, self-reported, and unverified. However, generally, the distribution of both variables seemed realist. Around ten clear data errors were identified and removed and later imputed for. To reduce the bias of any remaining, less explicit data errors, these variables were winsorised at the 99% level. In supplementary material II, the exact operationalisation is discussed in more detail.

Underemployment is rare in low- and middle-income countries (LMICs), with many individuals instead engaging in small-scale farming or informal micro-businesses. Thus, I construct three additional binary indicators to assess shifts in the sectoral composition of employment, following Chakravarty (2019). First, the formal IGA indicator assesses whether someone has a formal job, defined as waged work with a formal contract. A formal job is preferred as it provides more stability, rights, and access to other benefits. Secondly, the self-employment indicator shows whether someone is self-employed and likely a micro-entrepreneur. If there is a large discrepancy between formal IGA and self-employment, it reflects effects on informal waged workers. Finally, I construct an indicator for pursuing tertiary education to assess the likelihood of continuing education.

[3] Enrolled in upper-secondary education is defined as enrolled in grade 12 in Ethiopia, Vietnam and India and grade 11 in Vietnam. If a person was both enrolled in the last grade of upper-secondary and later vocational secondary they are coded as attended vocational secondary, since it is most likely they failed to complete their general education and then switched to vocational secondary.

[4] Dropped out after primary is defined as being enrolled in the final year of primary school, and then not having been enrolled in vocational secondary education or in the final year of general secondary. If a dropout would later attend any form of adults' education, which is very rare, this does not change their treatment status.

To control for selection effects, I need to control for a comprehensive set of predetermined characteristics that influence both the likelihood of enrolling in vocational education and subsequent labour market outcomes. Appendix A names the variables, and Supplementary Materials II.3 discusses the computation in detail, with offering in-depth explanations of their computation. The variables were chosen manually based on their assessed relevance and a review of the literature. All variables were collected before the cohort commenced vocational education to avoid the use of 'bad' controls.

The selected variables are grouped across three levels. At the individual level, variables include assessments of cognitive skills, extensive self-reported measurements of health during childhood, time spent during the day, including at school, study, and work, as well as non-cognitive measures of sociability, leadership, self-esteem, and self-efficacy, and indicators of later educational and job aspirations. At the household level, detailed controls include parents' education, socio-economic status, family size, perception and valuation of education, expectations for their children, and the occurrence of sudden financial shocks. At the community level, controls include the accessibility of the community, types of available jobs, and the availability of different types of education.

3.3. Validity, Attrition and Missing Data

Considering generalisability, it is important to note that the Young Lives survey was designed to over-represent households from lower socio-economic backgrounds. In India, the sample is specific to two regions, while in the other countries, the entire nation was included. Still, follow-up analysis has confirmed that the samples are broadly representative in Peru, slightly poorer in Vietnam, and marginally better off in Ethiopia and India (Barnett et al., 2013). Therefore, while country-specific estimates cannot exactly be generalised to the entire population, the differences between sample and population will be very small. Of course, India is special, with results being not generalisable beyond the two regions sampled.

Furthermore, internal validity is strengthened by low attrition across waves: 83% (N = 9,753) of the originally sampled participants were still successfully contacted in either 2017 or 2021.[5] Young Lives invested significantly in actively tracking respondents over time. Non-response rates in Ethiopia were especially high, as several waves were cancelled in particular regions due to violent conflicts (Young Lives, 2023). Common reasons for attrition include

[5] An extra thirty observations are missing for outcomes formal IGA, self-employment, and non-farming IGA, since their type of job was unknown, and thus these outcome variables could not be coded. Multiple imputation was used to impute these values, using the same techniques as for the covariates.

migration, new partners forbidding participation after marriage, and dissatisfaction with the study's impact (Young Lives, 2024). Importantly, poorer, urban households are more likely to drop out (Sánchez & Escobal, 2020). Despite this, subsequent analyses by Young Lives show that attrition is overwhelmingly random, and the magnitude of bias is negligible, making it "highly unlikely to bias research inferences" (Young Lives, 2024b). Therefore, after controlling for socio-economic status and region, the limited attrition does not threaten the study's internal validity.

Finally, Appendix B shows the item-response rate: none of the observations have a complete set of baseline characteristics, with an average of 9.6% of data missing.[6] A systematic reason for missing data is that some questions were not asked in specific countries or cohorts. For those variables, the data is missing at random conditional on cohort and fixed effects. Including extra covariates only increases the plausibility of the missing at random assumption (Li, 2013). The same argument applies to community-level data, which will be missing for all individuals within that community. Excluding those variables, an average of 5.5% of data is missing. Running a Little's test shows convincing evidence that this data is missing completely at random ($\chi 2$ = 32,527, df = 66,974, p = 1). With less than 10% of data missing and a very credible missing at random assumption, multiple imputation is the optimal method for causal inference, resulting in unbiased estimates of coefficients and standard errors (Rubins, 1996; Newman, 2014).[7] `The R-package Mice was used to impute five different datasets (van Buuren & Groothuis-Oudshoorn, 2011). While a higher number than five datasets would slightly increase statistical power, this would come at significant computational expense (White, Royston & Wood, 2011).

3.4. Descriptive Data and Balance in Original Data

As shown in Table 2, vocational secondary education is most popular in Ethiopia and India and much less so in Peru and Vietnam. Additionally, the relative rate of Older Cohort (OC) students attending vocational secondary education is much higher than that of the Younger Cohort. This is because vocational students generally take longer to complete their secondary

[6] Within construct missingness was rare, and if some of the variables within a construct were missing, I still computed the construct with the remaining data (e.g. self-esteem index), following Newman (2014).
[7] As recommended, all outcome variables, treatments and covariates are included in the prediction model. Quickpred's algorithm in the mice package is used to select which predictor variables are used to impute a variable, as is recommended in high-dimensional models, with parameters tuned to achieve the optimal average of 25 predictors per covariate (van Buuren & Groothuis-Oudshoorn, 2011). Cohort, country and gender are always included as predictors. Lastly, to impute I use predictive mean matching for continuous variables, proportional odds logistic regression for ordinal categories and categorical variables by logistic regression.

Table 3: Descriptive Statistics

	Vocational Secondary		General Secondary		Dropped Out Post Primary	
	OC	YC	OC	YC	OC	YC
N:	341	126	1395	1705	788	1410
Outcomes:						
Hours per week worked	32.92	25.13	34.83	29.2	39.53	28.77
Hourly income (us $)	0.41	0.33	0.63	0.44	0.24	0.16
Any IGA (binary)	0.60	0.44	0.68	0.51	0.69	0.44
Formal IGA (binary)	0.43	0.26	0.53	0.34	0.39	0.26
Self-employed (binary)	0.09	0.21	0.10	0.15	0.15	0.22
Not employed in farming (binary)	0.48	0.35	0.60	0.44	0.46	0.35
Attended higher education	0.54	0.23	0.82	0.43	0.00	0.00
Survey Characteristics						
Age at measurement outcome variables	22.08	18.8	23.48	19.05	23.97	19.54
Work experience in years	6.71	3.48	5.17	1.42	2.83	0.92
Individual Characteristics						
Gender: male	0.54	0.55	0.49	0.49	0.53	0.53
Cognitive scores: Math test at 13	508.58	473.13	541.76	511.45	471.82	455.25
Language score at 13	2.65	2.55	2.91	2.73	2.37	2.36
Non-cognitive skills: Number of friends	7.89	6.42	10.23	7.09	8.26	5.81
Trust in others (1-100 scale)	77.84	67.23	71.99	64.46	78.54	68.82
Self-efficacy (1-100 scale)	75.86	64.04	79.45	64.43	76.11	62.87
Self-esteem (1-100 scale)	79.14	64.63	85.19	68.36	76.82	63.52
Expectations: self-expected grade: technical/vocational college	0.06	0.02	0.07	0.08	0.06	0.03
university/college	0.72	0.8	0.83	0.76	0.56	0.62
upper-secondary	0.14	0.1	0.05	0.07	0.19	0.19
Dream job is vocational: Yes	0.11	0.15	0.12	0.23	0.2	0.21
not known	0.06	0.03	0.1	0.05	0.09	0.05
Household Characteristics						
Household size at age 13	5.61	5.02	5.16	4.94	5.67	5.27
Household primary job: agriculture	0.34	0.25	0.27	0.18	0.48	0.41
Mom attended formal education: Yes	0.53	0.65	0.74	0.8	0.47	0.56
Dad attended formal education: Yes	0.58	0.67	0.77	0.79	0.58	0.65
Wealth index (0-1)	0.45	0.54	0.55	0.6	0.39	0.5
Community Characteristics						
Time to provincial capital (hours)	13.91	14.69	11.23	10.65	12.63	12.55
Urban locality	0.38	0.37	0.42	0.44	0.21	0.2
Public secondary available: no. but there is one in a nearby locality	0.4	0.47	0.23	0.22	0.35	0.34
yes	0.5	0.44	0.65	0.63	0.54	0.57
Public higher vocational available: no. and not in a nearby locality	0.16	0.1	0.24	0.24	0.34	0.3
no. but there is one in a nearby locality	0.58	0.67	0.5	0.48	0.53	0.57
yes	0.14	0.13	0.12	0.13	0.05	0.06

Notes a): all descriptives are on non-imputed data, b) Vocational secondary is defined as once having been enrolled in non-tertiary TVET, even if earlier enrolled in general secondary, dropped-out is defined as having been enrolled in the final grade of primary, but never have been enrolled in the final grade of upper-secondary. c) For categorical variables values are proportions, for numerical variables values are the non-standardized differences d) not known encompasses missing data, refusals to answer and "I do not know answers" education and are thus still enrolled during the last survey wave. They are more prone to study delays and often initially unsuccessfully attempt to complete general secondary education before switching. Ethiopia is a special case where many general students also suffer from lengthy study delays. As a result, 3,010 members of the Younger Cohort were still enrolled and are excluded from the sample due to a lack of outcome data. This imbalance has two inherent consequences: it reduces the sample size, making estimates of vocational education in the Younger Cohort, especially in Peru and Vietnam, uncertain. Additionally, it may introduce bias, as slower students are less likely to be included in the sample, as are the rare group of postgraduate students. Thus, cohort fixed effects are required to achieve valid estimates. Additionally, the treatment estimate of the Young Cohort specifically may not be generalisable to the full population, even though controlling for among others (non)-cognitive skills and socio-economic status will address the majority of the beforementioned bias.

In total, 467 persons attended vocational secondary education within the sample. This, compared with the large control groups, provides sufficient power to identify modest and strong effects of vocational education. However, an exact calculation of statistical power is not possible, due to the complex weighted model with multiple imputation. The sample size is too small to identify weak to modest heterogeneous treatment effects, especially if the number of groups increases.

Table 3 provides descriptive statistics on outcomes and selected covariates by cohort and type of school attended, with descriptives on the remaining covariates shown in Appendix C. Considering the outcome variables, vocational students work slightly fewer hours per week than dropouts and general students. However, while dropouts work the most hours per week and are most likely to be employed, their jobs are of much lower quality, as evidenced by lower hourly wages. Compared to dropouts, vocational students benefit from higher salaries and are more likely to have formal jobs. Still, general students earn the most on average, over 50% more than vocational students. They are also more likely to have formal jobs and pursue further education beyond secondary school. Also, differences between cohorts are substantial. Across all treatments, the Older Cohort studied longer, worked more often, and earned more. This difference can be attributed to work experience and the fact that many members of the Younger Cohort finished studying in 2020/2021, joining the workforce during the uncertainty of COVID-19.

Considering baseline characteristics, general students typically score the highest, followed by vocational students, and then dropouts. This suggests selection effects will positively bias outcomes for the general students and negatively bias outcomes among dropouts. General students have the highest math and reading scores, report higher self-esteem and self-efficacy, and are more likely to aspire to attend college or university. Vocational students fall in between. They are more likely to be male but are not more likely to aspire to have a vocational career. At the age of eight, vocational students and dropouts are smaller, more underweight, and generally judge themselves to be less healthy than prospective general students. Vocational students also spent less time studying outside of school. The same trends are seen at the household level. General students come from smaller, wealthier households with better-educated parents. Vocational students live in larger, less wealthy households with typically less-educated parents. However, as shown in Appendix C, vocational students' parents value education equally and often hope that their child will reach higher academic levels. Dropouts live in less-wealthy, larger households with parents with less formal education. At the community level, vocational students are more likely to live in places with worse access to public secondary schools and better access to public higher vocational education. Perhaps the possibility of continuing their education increases the attractiveness of TVE. Finally, dropouts are more likely to live in communities with no access to vocational secondary schools or similar institutions, suggesting a lack of access to TVE might have influenced their decision to drop out.

After standardisation, the differences in means between treatment groups remain substantial, indicating significant imbalance. Figure 1 shows that over half of the standardised differences between covariates exceed 0.1, and 13 to 25% of differences exceed 0.25. The

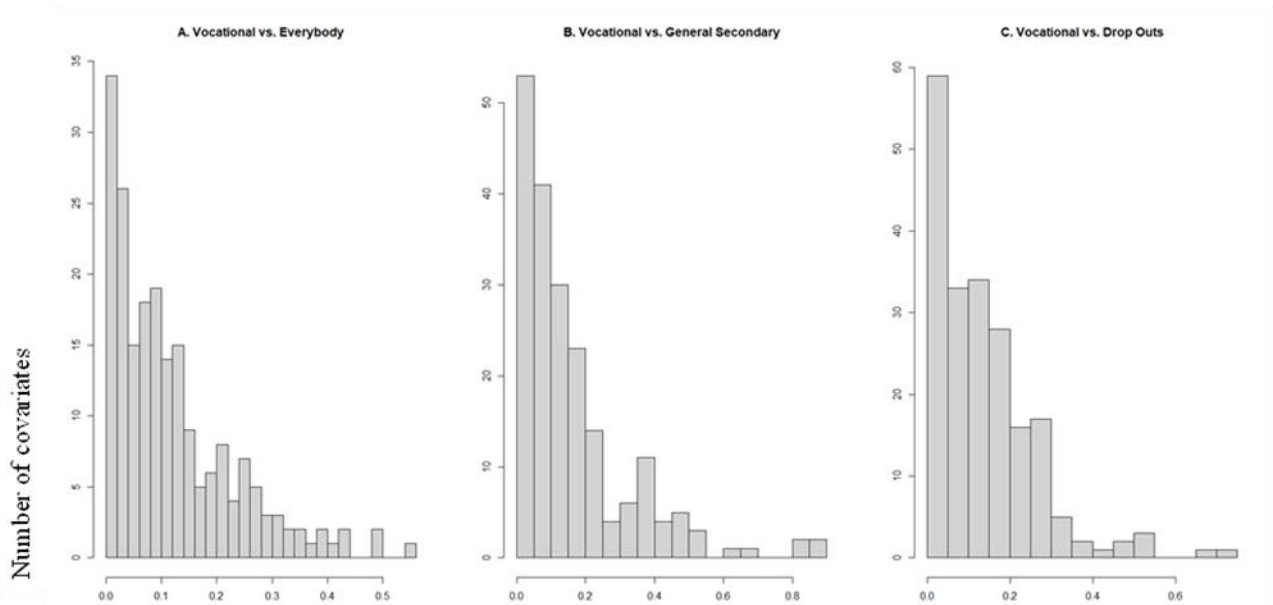


Figure 1: Covariate Balance in Raw Sample

balance is worse in the sub-samples. These differences include important indicators such as test scores, non-cognitive skills, parental education, and the availability of secondary schools. Under such conditions, simple regression adjustment cannot fully account for pre-baseline differences, leading to ineffective and biased estimates (Rubin & Imbens, 2015). Instead, more advanced techniques, such as propensity score weighting, are required to achieve a more balanced sample.

4. Empirical Strategy

4.1.Selection on Observables

Applying causal inference on observational data using propensity score methods hinge on the critical unconfoundedness assumption: no unobserved factors exist that simultaneously affect both the likelihood of undergoing vocational education and the outcomes (Rosenbaum & Rubin, 1983; Guo & Wang, 2020). These unobserved factors, discussed in the literature review, include motivation, cognitive and non-cognitive skills, socio-economic factors, parental pressure, and the type of community. Only if this assumption holds can the treatment and control groups be considered truly equivalent, free from selection bias or bias from omitted variables. However, this assumption is inherently unverifiable. Previous studies have used limited characteristics, focusing on basic demographics, household socio-economic status, household size, and parental education as selection variables (Moenjak & Worswick, 2003; Guo & Wang, 2020). To my knowledge, Farias, and Sevilla (2015) implemented the most thorough set of controls to date, also incorporating test scores and attendance rates as proxies for cognitive skills and motivation, as well as measuring parents' willingness to invest in education and their expectations for their children's futures.

The scope of pre-observed characteristics in this analysis significantly expands upon that of Farias and Sevilla (2015). The main advantage of the rich longitudinal Young Lives dataset is its ability to control for a much wider range of covariates. The absence of obvious unobserved factors, coupled with high-quality, repeatedly measured proxies, strengthens the internal validity of this study. It suggests that any residual selection bias will be minimal, thereby allowing for a careful causal interpretation of the finding. Variables include assessments of the child’s responsibilities at home, health, multiple measurements of cognitive and non-cognitive skills, and educational and job aspirations. At the household level, it includes basic demographics, socio-economic indices, the sector of employment by parents, and extensive data on parental education, their perception of the usefulness and quality of education, and the occurrence of financial shocks or natural disasters. Community-level measures cover the type of community, the proximity to general secondary and vocational centres, and the types of jobs available. This set of variables addresses all unobserved factors influencing the choice for vocational education mentioned in previous papers (e.g., Moenjak & Worswick, 2003; Meer, 2007; Farias & Sevilla, 2015; Guo & Wang, 2020).

4.2.Propensity Score Estimation

Nonetheless, addressing self-selection in observational data through multivariate regressions presents several challenges (Baser, 2007; Adelson, 2019; Amoah et al., 2020). As shown in Figure 1, the data is unbalanced to such an extent that multivariate regression cannot fully control for selection effects. Additionally, multivariate regressions are highly sensitive to the model's functional form (Baser, 2007). Correctly specifying the functional form is difficult, particularly without prior research on this topic. Moreover, multivariate regression becomes less efficient with the inclusion of many covariates, which can create multicollinearity issues and give undue weight to outliers (Adelson, 2019; Amoah et al., 2020).

Instead of relying solely on multivariate regressions, I will primarily use propensity score matching. Although instrumental variables have been used, they face criticism due to the questionable validity of the instruments, which can result in overestimations (Farias & Sevilla, 2015). Propensity scores estimate the probability of an individual receiving a treatment, creating a quasi-experimental design that mimics random assignment (Rosenbaum & Rubin, 1983). Within vocational research, propensity score methods are commonly used since random assignment is often impractical or ethically infeasible (e.g., Moenjak & Worswick, 2003; Meer, 2007; Guo & Wang, 2020). The correct application of propensity score techniques addresses several limitations associated with multivariate regressions by reducing sensitivity to the model's functional form, enhancing efficiency in managing multiple covariates, and minimizing the impact of outliers (Baser, 2007; Adelson, 2019; Amoah et al., 2020).

The wide range of selection variables presents a new challenge due to the lack of guidance from the literature on the weighting and functional forms of these variables; no previous studies have controlled for such an extensive range of baseline characteristics. Correct modelling is important, as the weighting and functional form affect the bias and precision of propensity scores (Brookhart et al., 2006). To address this, propensity weights were calculated using three different methods: a) using additive probit regression with all baseline characteristics, b) lasso selection and c) generalised boosted regression models.[8] All pre-determined baseline characteristics were included.[9] While these characteristics should correlate with both the outcome and the treatment, the correlation does not have to be causal for the variable to improve the model (Austin, 2011). The generalised boosted regression model, in particular, is a machine-learning algorithm that selects which covariates, which functional form, and which higher level interaction should be included in the estimation. Thereby, GBM controls for complex and nonlinear relationships and is generally preferred in contexts with abundant selection variables and complex relationships (McCaffrey, 2013; Zhu, 2014; Setodji et al., 2018).

The simple probit model yielded the best balance across baseline characteristics and thus should be preferred (Imbens & Rubin, 2015). Balance means that after conditioning on the propensity score, there should be no relationship between baseline characteristics and treatment assignment. The above was determined pre-treatment. That a simple probit model outperforms generalised boosted regression models suggests that interaction and non-linear effects are not important in this particular context. While the probit model is used exclusively in the main paper, Appendix G shows that the main findings are also robust to generalised boosted regressions. This is an important robustness check, showing that the results are robust to a different model for propensity score estimation, which also includes higher level interaction.

4.3.Propensity Score Weighting, Balance, and Overlap

Propensity scores have been applied to weight, match, or stratify data (Farias & Sevilla, 2015). While stratifying is considered sub-optimal, existing evidence does not favour weighting or matching (Austin, 2011). For this data set, weighting yielded significantly better balance than matching procedures and is thus used. Weights are estimated separately for the main sample and the two sub-samples. Since the sample size of the treated group is limited, only weights for the average treatment effect on the treated (ATT) can be computed.

[8] Package Weightthem was used to compute propensity scores (Pishgar et al., 2020). For GBM a Bernoulli distribution was used, 12.000 trees were calculated, shrinkage was set at .05, maximisation criteria were the average standardised mean effect and interaction depth was allowed to vary between 1 and 3.

[9] Cohort and Country fixed effects (and their interactions) were included as factors in the propensity score calculation. Sample size was not sufficient to perform calculations stratified per country and/or cohort. However, including fixed effects in the model will lead to unbiased estimates if balance across countries is also achieved (See Li, Zaslavsky & Landrum, 2013; DuGoff, Schuler & Stuart, 2014).

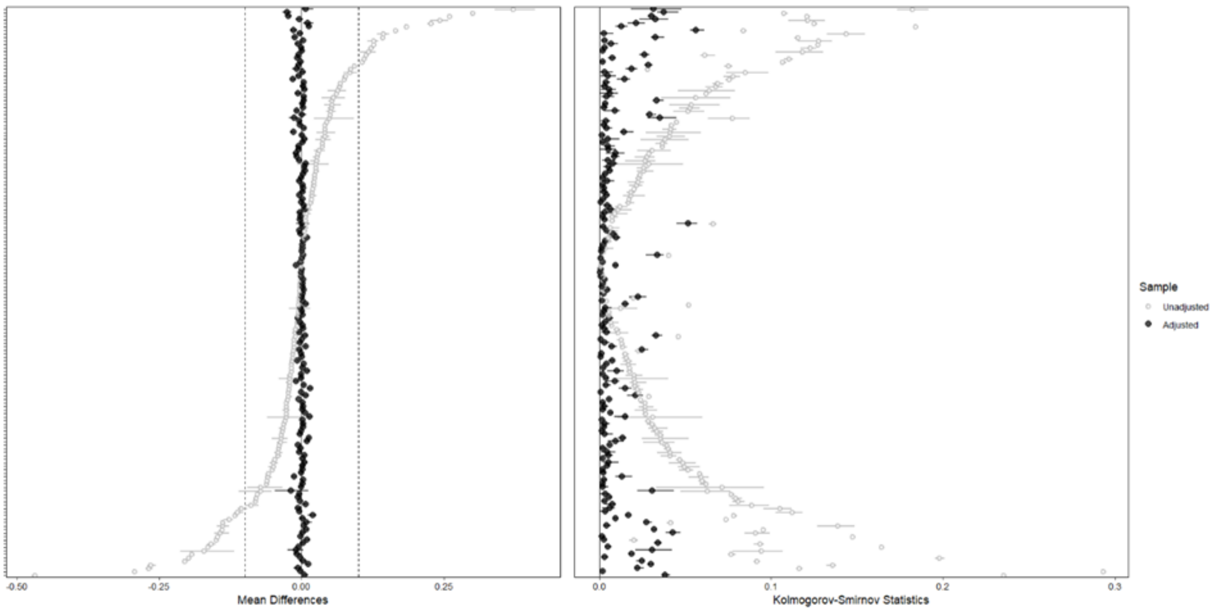


Figure 2: Covariate Balance in Full Sample after Propensity Weighting

In this method, treated observations get a weight of one, with untreated observations being weighted to create a balanced sample (Desai & Franklin, 2019). In the results, an ATT-coefficient represents the average effect of attending vocational education among vocational students. Importantly, ATT only equals ATE if there is no heterogeneity in treatment effects, which may be close to true when comparing to general secondary, but clearly not when comparing to dropouts.

Causal interpretation hinges on two additional assumptions apart from unconfoundedness. After weighting, covariates should be balanced and there should be sufficient overlap in propensity scores. The weighting procedure successfully balanced the dataset. Austin (2009, 2011) and Imbens & Rubin (2015) define a standardised mean difference of 0.1 as a conservative threshold for balance.[10] As shown in Figure 2 and Appendix D, this criteria was comfortably met; across the main sample and both sub-samples, all standardised mean differences were below 0.1, with all differences in the main sample being below 0.025.[11] The balance of the sample is further confirmed by the KS-statistic, which tests the similarity of distributions rather than means. All KS-statistics are well below the threshold of 0.1 (as mentioned in Markoulidakis et al., 2021). Additionally, similar balance is achieved across countries and cohorts, allowing for sub-group analyses. Thus, the

[10] The R-package Cobalt was used to test for balance, see Greifer (2020).
[11] An index for trust in Panel C: Vocational vs. dropout is the only exception with .11 mean standardised difference.

sufficient overlap in the data and excellent balance make valid inferences on the average effect of vocational education on the treated (ATT) possible.

To test overlap, Appendix E shows the area of common support of propensity scores per imputed dataset. The area of common support covers almost the full range of treated observations. However, the control sample size is much larger for lower propensity scores. Still, the approximately fifteen vocational students with the highest propensity scores (3.2% of treated group) fall just outside of the area of common support. In response, the propensity scores of these approximately fifteen students are winsorised, giving them the maximum propensity score within the area of common support, as recommended by Imbens & Rubin (2015). Winsorizing is possible since the difference between those fifteen and the area of support is very small. Compared to discarding these observations, this method keeps the sample size intact and allows for full ATT-estimations. However, any ATT-estimates are not fully reflective of the entire treated population, as the top 3% of students most likely to attend vocational education are slightly underweighted, but in practice this distinction has little impact.

4.4. Model Specification

Equation one summarizes the main model. Here, β_1 represents the average effect on the treated (ATT) of graduating from TVE. C_i denotes country fixed effects: although regional fixed effects would have been preferred, there would be too few (n < 5) treated subjects remaining in some clusters for estimation. Furthermore, OC_i is a cohort fixed effect, which controls for important systematic differences. Additionally, to account for unique time trends within each country, I include fixed effects of the interaction between country and the year the survey outcomes were measure. This is critical, especially because of the impact of COVID-19 on labour markets. A simple COVID-19 dummy would not be sufficient, as the virus had different effects in different years across countries.

$$y_{i,c} = \beta_0 + \beta_1 Attended\ Vocational\ Secondary_i + \gamma X_i + C_i + OC_i + FE * (Country_i \times Year_i) + \epsilon_i \quad eq\ 1$$

X_i is the set of observed baseline characteristics directly included in the regression to correct for any remaining minor discrepancies, a technique known as double-robust regression (Farias & Sevilla, 2015). A few less important covariates used in the propensity estimation are excluded to prevent multicollinearity, for example separate educational dummies for caretaker, which are highly multicollinear with the mother's education.[12] After this selection,

---

[12] Excluded are every variable for caretaker, since these were highly correlated with values for mothers. Additionally, family size now only includes household size and the number of children born before and after the

no significant multicollinearity is detected. These multicollinear variables are still included in the propensity score estimation, since they likely contain useful information, and do not bias the results (McMurry et al., 2015). A double-robust regression is important, since it is unbiased if either the propensity score or the outcome regression is correctly specified, instead of requiring both to be well specified (Funk et al., 2011). All models are estimated using linear regression with propensity score weights, even if the outcome variable is binary. Finally, heteroskedasticity was consistently detected using a Breusch-Pagan test, so all standard errors are robust.

I use Equation 2 to test for heterogeneous treatment effects. Equation 2 is exactly equal to Equation 1 but includes an interaction between treatment and the moderator of interest.[13] This is used to test for heterogeneity based on country, time since graduation, personal characteristics, and community characteristics. In all these cases, I present the average marginal effect of treatment (AME). For each treated individual, the individual marginal treatment effect is calculated, denoting the predicted difference in outcome between attending vocational secondary education or not, considering the observed characteristics of that individual. Individual treatment effects for untreated individuals cannot be included since I still estimate only average effects on the treated. Then, the AME is simply given by the average of these individual effects. Using AME over general ATT's has no methodological up- or downsides but is preferred because it makes interpreting the treatment by covariate interactions much cleaner, with a clear treatment effect per subgroup (see Onukwugha, Bergtold & Jain, 2015, and Esarey & Sumner, 2018 for further discussion).

$$\begin{aligned} y_{i,c} = \beta_0 + \beta_1 * Attended\ Vocational\ Secondary_i + \beta_2 * Attended\ Vocational\ Secondary_i \\ * Moderator_i + \beta_3 * Moderator_i + \gamma X_i + C_i + OC_i + FE \\ * (Country_i \times Year_i) + \epsilon_i \quad eq\ 2 \end{aligned}$$

surveyed child. Also, the expected age parents expect their child to leave school is removed (correlating too much with the other parental expectations). Furthermore, the factor variable for dream job by both the individual and parents are removed, they are replaced by dummies for whether their dream job is vocational or academic. Lastly, the availability of private secondary school is also removed, being too highly correlated with other measurements of availability.

[13] In total, fifteen moderators are used:
a) Time since graduation to check for effects over time,
b) Country to check for heterogeneity across countries,
c) Propensity to attend general education to check if treatment effects are similar for those who otherwise would likely have dropped out,
d) Gender, cognitive test scores, self-reported non-cognitive scores, and motivation to evaluate for heterogeneous effects based on personal baseline characteristics, and
e) Size of town and type of jobs available to test for heterogeneous effects based on community baseline characteristics.

To allow for causal interpretation, I must assume homogeneity within a subgroup after weighting; in other words, covariates have to be balanced within subgroups (Brand & Xi, 2011; Brand & Thomas, 2013). Appendix F provides balance plots across covariates for all fifteen moderators tested. In all cases, balance is not perfect and does not meet the conservative standard of 0.1 standardised mean difference. Two moderators are severely unbalanced: propensity scores to attend general education, and time since graduation. These are severely unbalanced since they were not included in the propensity score calculation. This means these results should be interpreted as correlations, considering there may be significant residual confounding even after propensity weighting.

Gender meets the less conservative boundary of a 0.25 standardised mean difference. This leads to some residual confounding if only using propensity score weighting, but this can be effectively managed with double robust regression, making it possible to causally interpret gender effects (Rubin & Imbens, 2015). Country heterogeneity is balanced, except for Peru, which has a few outliers. There, all estimates can be interpreted causally, except for Peru. The personal and community characteristics also meet the 0.25 threshold except for a few outliers. There is no trend in which variables become outliers. For these moderators, residual confounding will be very limited but not non-existent. Thus, these coefficients are not technically causal but will be void of almost all selection bias and provide strong evidence for causal relationships. Sample size was insufficient to use more advanced weighting procedures to achieve perfect balance across sub-groups.

5. Results

5.1. Average Impact of Vocational Education: Linear Regressions

Table 4 reports different model specifications for hourly wage and hours worked per week, comparing vocational with general education, all using linear regression without propensity weights. Since estimates are only slightly affected by the introduction of covariates, this signals selection bias is limited, despite the clear theoretical arguments suggesting otherwise. Still, as expected, the selection bias negatively affects vocational treatment estimates relative to general secondary. Much more impactful is the introduction of country * year fixed effects, which especially helps to control for the effects of COVID-19. Comparing the models fit, the complete model with all covariates tends to have the highest adjusted R^2 and tend to have the lowest AIC criteria, suggesting that Model 6 fits the data best. Although community and household characteristics do not add much in the hours worked per week model, there are

Table 4: Model Specification with OLS

ATE	Reduced Form	Cohort and Country FE	Country*Year FE	+ Individual Characteristics	+ Household Characteristics	+ Community Characteristics
	(1)	(2)	(3)	(4)	(5)	(6)
Panel A: Hours Worked per Week						
Treatment	-0.035 (0.048)	0.118 (0.052)**	0.096 (0.052)*	0.067 (0.051)	0.061 (0.051)	0.071 (0.051)
Adjusted R^2	0	0.084	0.104	0.138	0.141	0.149
AIC-criterion	10024.808	9714.957	9649.246	9551.322	9574.707	9565.627
Panel B: Hourly Wage						
Treatment	-0.158 (0.048)***	-0.039 (0.054)	-0.062 (0.056)	-0.066 (0.055)	-0.068 (0.055)	-0.074 (0.056)
Adjusted R^2	0.002	0.218	0.239	0.26	0.261	0.265
AIC-criterion	11196.267	10329.741	10242.73	10185.86	10215.48	10218.072
Individual level	No	No	No	Yes	Yes	Yes
Family level	No	No	No	No	Yes	Yes
Household level	No	No	No	No	No	Yes
Cohort FE	No	Yes	Yes	Yes	Yes	Yes
Country FE	No	Yes	Yes	Yes	Yes	Yes
Country*Year FE	No	No	Yes	Yes	Yes	Yes
MI	Yes	Yes	Yes	Yes	Yes	Yes

Notes: a)*** p < .01, ** p < .05, * p < .1. b) new covariates are added additively, thus community characteristics is the full sample. c) all estimates use the full set of individual, family and sector covariates b) all estimates are pooled from OLS regressions of five multiple imputed datasets using the Mice Package, and standard errors are robust of type HC1.

strong economic arguments to control for factors as household wealth with their direct effect on access to education. Model six is also preferred for almost all other outcomes and across subsamples (tables not reported). Thus, this model, which includes all covariates at the personal, household, and community levels, as well as country, cohort, and country * year fixed effects, is preferred and is used exclusively moving forward, as discussed in section 4.4.

It is worth reiterating that simple covariate adjustment does not address all imbalances in the data, resulting in likely biased coefficients and standard errors. Still, these results are valuable as a comparison for the propensity score-weighted regression to assess bias. In Table 5, Panel A, I compare the impact of vocational secondary education to everyone else in the sample, including those studying in general secondary, dropouts, and those who have never attended formal education. Three effects are significant. Vocational students are five percentage points (p.p.) less likely to continue to higher education, 5.7 p.p. less likely to be self-employed, and earn 0.08 standard deviations (sd.) less in income, the latter being significant, but practically a very modest effect. Vocational students are also slightly more

Table 5: Average Treatment Effects without Propensity Score Weighting

ATT	Hours per Week Worked	Hourly Income	Any IGA	Formal Work	Self Employed	Non Farming IGA	Attending Higher Education
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Panel A: Full Sample							
Treatment	0.016 (0.047)	-0.081 (0.049)*	0.011 (0.024)	0.031 (0.024)	-0.057 (0.017)***	0.017 (0.024)	-0.051 (0.022)**
Adjusted R^2	0.171	0.272	0.196	0.153	0.093	0.201	0.423
Mean outcome	0 1SD = 26.0 hours	0 1SD = $0.86	0.554	0.365	0.164	0.443	0.313
Observations	6743	…	…	…	…	…	…
Panel B: Vocational vs. General Secondary							
Treatment	0.069 (0.051)	-0.073 (0.056)	0.012 (0.026)	0.003 (0.027)	-0.03 (0.019)	-0.002 (0.027)	-0.231 (0.024)***
Adjusted R^2	0.148	0.267	0.166	0.138	0.058	0.186	0.346
Mean outcome	-0.120 SD = 25.6	0.137 SD = $1,00	0.582	0.426	0.126	0.506	0.587
Observations	3567	…	…	…	…	…	…
Panel C: Vocational vs. Drop Out							
Treatment	-0.032 (0.065)	0.074 (0.062)	0.023 (0.031)	0.084 (0.032)***	-0.073 (0.024)***	0.029 (0.033)	0.329 (0.026)***
Adjusted R^2	0.205	0.201	0.214	0.13	0.083	0.179	0.45
Mean outcome	0.015 SD = 26.6	-0.193 SD = $0.57	0.531	0.324	0.183	0.402	0.084
Observations	2665	…	…	…	…	…	…
Cohort FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Country FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Country*Year FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MI	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Notes: a)*** p < .01, ** p < .05, * p < .1. b) all estimates use the full set of individual, family and sector covariates c) all estimates are pooled from OLS regressions of five multiple imputed datasets using the Mice Package, and standard errors are robust of type HC1

likely to have a formal waged job (3 p.p.), although this effect is not significant.

Comparing attending general versus vocational secondary education in Panel B, the differences are negligible. Vocational students earn 0.07 (0.07$) standard deviations less and work 0.06 sd. or 1.5 more hours a week. Both effects are non-significant and effect sizes are small. Furthermore, unemployment rates are about the same, as are formal job rates and the likelihood of working as a farmer. The only difference is that students choosing for TVE are much less likely to attend higher education (-22.7 p.p.). These results align with observational studies finding little to no differences between general and vocational education but contrast sharply with the major employment benefits of TVE found in some recent randomised trials.

Comparing attending vocational education versus dropping out after primary education in Panel C shows some clear advantages of attending vocational education. Vocational students are much less likely to be self-employed, which is associated with economic uncertainty. Instead, vocationals are 8.4 percentage points more likely to have a formal job. Vocational students are also slightly more likely to have an income-generating activity (3.3 p.p.), but this effect is insignificant. However, counterintuitively, these benefits do not translate into significantly higher wages or more hours worked. The findings indicate that while vocational education does not significantly improve early-career job quality or earnings compared to general education, it does offer advantages over dropouts, primarily by increasing the likelihood of formal employment.

5.2. Average Impact of Vocational Education: Weighted Propensity Scores

Table 6 reports the results of estimating equation 1, this time including propensity score weighting to improve balance. In contrast to linear regression, these coefficients can be interpreted causally under the assumption of unconfoundedness, considering overlap and balance is sufficient. This model estimates the average treatment effect on the treated (ATT) rather than the average effect on the general population. Quantitatively, the findings are very comparable to the simple linear regressions in Table 5, differing by only around one percentage point on average. The higher adjusted R^2 indicates that the weighted regressions explain the data better. Model fit parameters prefer the full covariates model, which also has more efficient standard errors, and are thus reported moving forward.

In Panel A, the full sample, the most substantial change relative to Table 5 is that attending vocational education decreases the likelihood of being self-employed by only 3.7 percentage points, rather than 5.4 percentage points. Additionally, attending TVE no longer has a significant negative impact on hourly wage. In Panel B, weighting reinforces the conclusion that attending vocational versus general education has little to no effect on labour outcomes, apart from general students being much more likely to continue into higher education (26.4 p.p.). Differences in formal work, unemployment, wage, or hours worked per week are negligible. In Panel C, the same story holds true as after the linear regression. Vocational students are much more likely to get a formal job (9.1 p.p. or 28%) and much less likely to be self-employed (6.3 p.p. or 33%) than dropouts and are slightly, yet insignificantly,

Table 6: Average Effects on the Treated with Propensity Score Weighting

ATT	Hours per Week Worked	Hourly Income	Any IGA	Formal Work	Self Employed	Non Farming IGA	Attending Higher Education
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Panel A: Full Sample							
Treatment	0.016 (0.047)	-0.081 (0.049)*	0.011 (0.024)	0.031 (0.024)	-0.057 (0.017)***	0.017 (0.024)	-0.051 (0.022)**
Adjusted R^2	0.171	0.272	0.196	0.153	0.093	0.201	0.423
Mean outcome	0 SD = 26.0 hours	0 SD = 0.86$	0.554	0.365	0.164	0.443	0.313
Observations	6743	…	…	…	…	…	…
Panel B: Vocational vs. General Secondary							
Treatment	0.069 (0.051)	-0.073 (0.056)	0.012 (0.026)	0.003 (0.027)	-0.03 (0.019)	-0.002 (0.027)	-0.231 (0.024)***
Adjusted R^2	0.148	0.267	0.166	0.138	0.058	0.186	0.346
Mean outcome	-0.120 SD = 25.6	0.137 SD = 1.00$	0.582	0.426	0.126	0.506	0.587
Observations	3567	…	…	…	…	…	…
Panel C: Vocational vs. Drop Out							
Treatment	-0.032 (0.065)	0.074 (0.062)	0.023 (0.031)	0.084 (0.032)***	-0.073 (0.024)***	0.029 (0.033)	0.329 (0.026)***
Adjusted R^2	0.205	0.201	0.214	0.13	0.083	0.179	0.45
Mean outcome	0.015 SD = 26.6	-0.193 SD = 0.57$	0.531	0.324	0.183	0.402	0.084
Observations	2665	…	…	…	…	…	…
Cohort FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Country FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Country*Year FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MI	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Notes: a)*** p < .01, ** p < .05, * p < .1. b) all estimates use the full set of individual, family and sector covariates c) all estimates are pooled from OLS regressions of five multiple imputed datasets using the Mice Package, and standard errors are robust of type HC1

more likely to have a job (3.5 p.p. or 6.8%). However, this does not translate into significantly higher earnings or more hours worked. Methodologically, the limited changes in Table 6 relative to the linear regressions indicate that imbalances in the data is a smaller problem than anticipated. Appendix G shows that using GBM-propensity scores the effects are very similar, except that TVE now in Panel C also has a very modest significant effect on hourly wage (0.11 sd.) and much higher effect employment (8.7 p.p.).

Just studying the average effects on vocational students, this study finds no significant differences between attending TVE vs general secondary. Comparing to drop-outs, there is a major change in job composition with more people having formal waged work. These benefits are far from the success stories reported by Camargo et al. (2018) and Chakravarty et al. (2019), where employment increased by around twenty percentage points. They are more in line with Field's (2019) randomised trial in Mongolia, which found a 4-percentage point increase in any income-generating activity (IGA), and with observational studies controlling for selection bias finding no effect on earnings. Both Camargo and Chakravarty's projects targeted extra-vulnerable groups, with more room for improvement, likely resulting in higher estimates. The shift from self-employment to formal employment is seldom addressed in the literature on formal TVE. However, McKenzie's (2017) review indicates that informal short-term vocational training programmes for the unemployed have minimal impact on overall employment but do lead to a 3.6 percentage point increase in formal employment. Thus, the effect on formal wages observed is consistent with, but more pronounced than, findings from shorter vocational training programmes. This may be attributable to vocational secondary education lasting three to four years, compared to short informal training courses of typically six months.

5.3. A Different Impact of Vocational Education: Keeping Students in Education

When assessing the effectiveness of TVE, it is important to consider that TVE might serve as a crucial pathway for students who would otherwise have dropped out. These students may

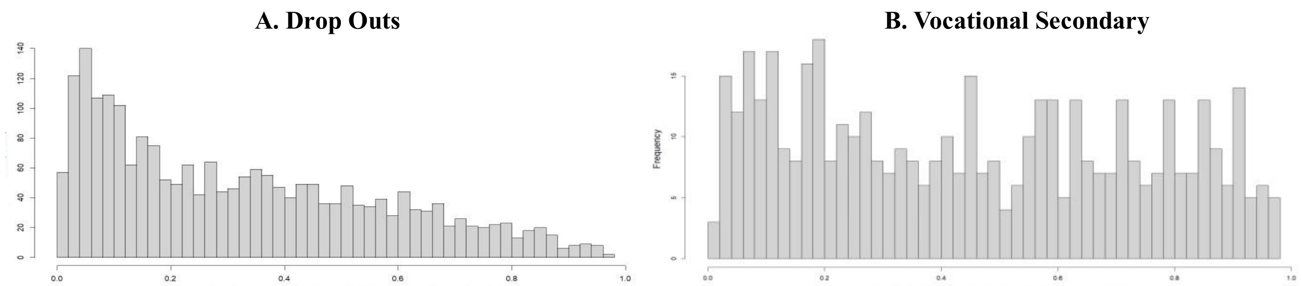


Figure 3: Histograms of Propensity Scores to Attend General Education

Table 7: Predicted number of vocational students who without TVE would have dropped

	Number of vocational students	Average propensity score to attend general secondary among vocational students	Predicted % of vocational students dropping-out if no TVE exists
Ethiopia	179	0.175	82,5%
India	179	0.579	42,1%
Peru	52	0.841	15,9%
Vietnam	57	0.590	41,0%
Total	467	0.681	54.6%

Note: The number of vocationals who would have dropped out is calculated by the sum of propensity scores per country across vocational students divided by the number of vocational students.

have had insufficient grades, lacked access to general education, or felt unsatisfied with theoretically focused education. To provide an indication of the extent of this effect, I calculate propensity scores for transitioning to general secondary education versus dropping out after completing primary in a hypothetical world without secondary vocational education.[14] Qualitatively, these propensity scores denote the likelihood of transitioning into general secondary education. Table 7 and Figure 3 show that a sizeable proportion of vocational students have a low propensity to attend higher education. On average, these propensity scores predict that 54.6% of vocational students would have dropped out had vocational education not existed. This effect is especially major in Ethiopia, where the propensity score predicts that 82.5% of vocational students would have dropped out after primary if there was no TVE. But also, in Vietnam and India TVE helped over 40% of students to continue studying. Of course, this is not a perfect causal what-if analysis but does provide an indication that TVE is important as an alternative educational pathway.

The implications of this result should not be underestimated; on itself it is a major positive outcome of TVE, contributing towards achieving Sustainable Development Goal 4: Quality Education, particularly in terms of full secondary enrolment. At the same time, it highlights the academic importance of also using dropouts as a counterfactual. As shown, the positive impacts of vocational education are larger when comparing to dropouts than to general secondary students. Therefore, using only the latter as a counterfactual will inevitably result in an underestimation of TVE-impact. Especially in Ethiopia, where dropouts are a much more realistic counterfactual than general secondary students.

This also means that the treatment effect on vocational students who would otherwise have dropped out is of particular interest. If these treatment effects are positive, it underscores the value of TVE in improving labour market outcomes for a population that would otherwise remain unreached. Therefore, I stratify the propensity scores into five equal strata, interact them with treatment, and graph the average marginal effects. For, hourly income, any IGA, and formal IGA, the lack of trends in Figure 4 shows that the propensity to attend general education does not significantly affect the impact of TVE. Only for hours worked per week is there an upwards slope, suggesting that vocational students with higher propensities to attend general secondary work more hours a week, but this effect is insignificant. The absence of a clear upward trend suggests that any benefits of vocational education extend to those likely to

[14] Propensity scores were calculated using a simple additive logit model. The sample included all students who at least completed primary education. The propensity score thus denotes the propensity to transition to secondary after primary. Successful transition was defined as being enrolled in the final year of upper-secondary education.

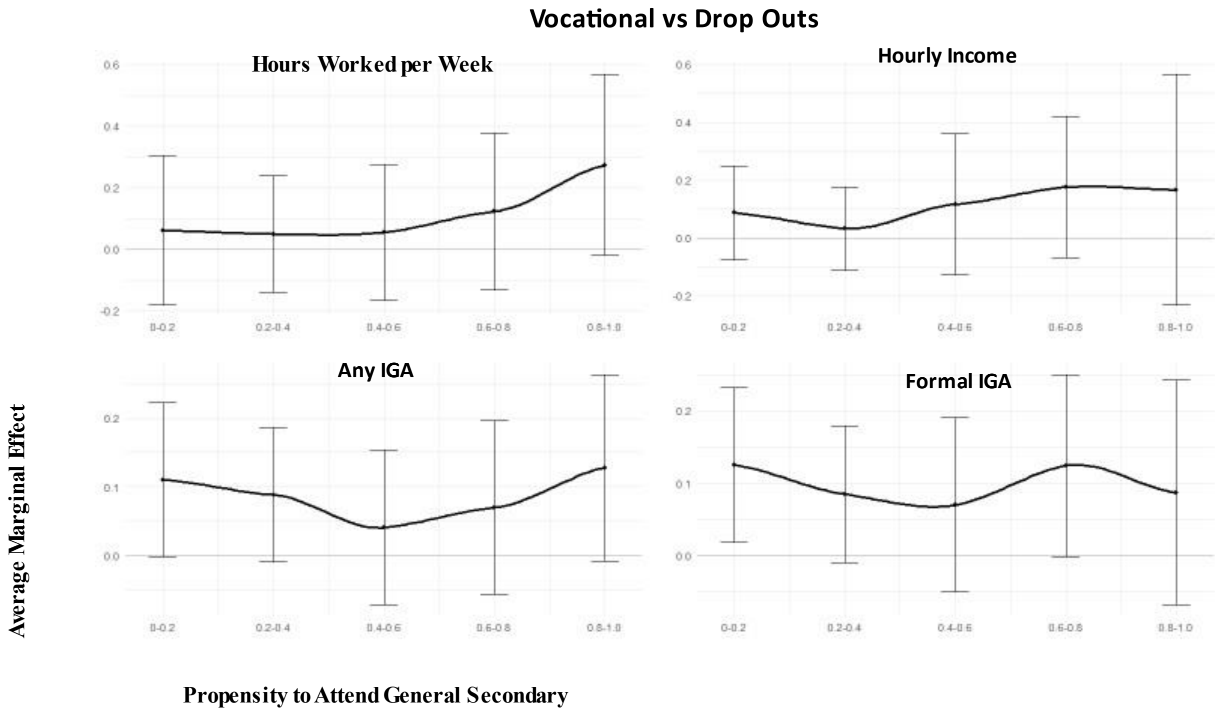


Figure 4: Average Marginal Treatment Effect by Propensity to Attend General Education

have dropped out, indicating that TVE can act as an equaliser, as long as attending TVE brings significant benefits. Using GBM-propensity scores in Appendix G delivers similar results, except that the upwards trend is slightly steeper for hours worked per week. The following sections will demonstrate that TVE has much larger heterogeneous impacts among women and in Ethiopia. In these sub-groups, the equalising factor plays a crucial role. Without TVE, many of these vocational students would have dropped out rather than attending general secondary education, thereby missing these educational benefits.

5.4. Heterogeneous Impacts of Vocational Education

5.4.1. Personal Characteristics

As discussed extensively in the literature review, it is highly likely that treatment effects are heterogeneous. Gender effects are debated extensively in the TVE literature, with some studies finding significant advantages for females, while others report no significant differences. Lessons from the broader educational literature also highlight the importance of cognitive skills, intrinsic motivation, and non-cognitive skills—such as social skills, self-efficacy, and conscientiousness—in contributing to larger returns to education. Previous literature did not yet analyse these factors for vocational education in LMICs, likely due to a lack of data. The extensive baseline controls available in the Young Lives data allow for testing these hypotheses. I find that TVE is much more effective for females, but cognitive skills, motivation, and non-cognitive skills do not moderate treatment impact.

I use the same method as before: incorporating an interaction between the proxy variable and treatment, and then calculating the average marginal treatment effect for each subgroup. A separate regression was run for each proxy to prevent multicollinearity. The limited sample size reduces the power of this analysis, and p-values are not corrected for multiple hypothesis testing.[15] P-values denote the significance of the sub-group average marginal effect, not whether the difference between marginal effects is significant. As shown in Appendix F, except for gender, data is not perfectly balanced across sub-groups, so a very limited amount of confounding will remain. Gender is balanced and can be causally interpreted. The other moderators provide strong evidence for a causal relationship but describe a correlation. However, the main aim of this analysis is exploratory: to provide valuable insights into the conditions under which TVE is most effective and to pave the way for more comprehensive future research. Moderators were manually chosen pre-analysis, based on the findings in the literature review.

What is immediately obvious from Table 8 is that the chosen counterfactual strongly affects the magnitude and statistical significance of the heterogeneous effects. Heterogeneous effects are negligible when comparing to general secondary but are substantial when using dropouts as the counterfactual. This indicates that these moderators affect returns to education but do not significantly moderate the difference between TVE and general secondary education. In other words, these heterogeneous impacts would not be of added value if all vocational students would have otherwise attended general secondary. However, considering many vocationals would have otherwise dropped out, as shown in Figure 3, the positive impacts observed when using dropouts as a counterfactual highlight an important benefit of TVE.

Comparing vocational to general education, neither cognitive test scores, non-cognitive skills, nor motivation significantly moderate the impact of TVE. The only exception is gender, where females consistently experience higher positive impacts of TVE compared to males. Females work significantly more hours per week and are slightly less likely to be

[15] No other variables except those mentioned in this chapter were used to assess for heterogeneous treatment effects to prevent finding spurious results.

Table 8: Heterogeneous Impacts for Gender and other Personal Characteristics

Sample:	**Vocationals vs. General Education**				**Vocational vs. Drop Out**			
Outcome *ATT*	Hours per Week Worked	Hourly Income	Any IGA	Formal Work	Hours per Week Worked	Hourly Income	Any IGA	Formal Work
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Panel A: Gender								
Gender: Male	-0.061 (0.069)	-0.02 (0.077)	-0.031 (0.037)	-0.047 (0.037)	-0.108 (0.082)	0.068 (0.085)	-0.011 (0.04)	0.017 (0.042)
Female	0.209 (0.079)***	-0.015 (0.052)	0.06 (0.041)	0.069 (0.039)*	0.299 (0.096)***	0.134 (0.054)**	0.18 (0.046)***	0.175 (0.045)***
Panel B: Cognitive Skills								
Math score: -1 SD	0.038 (0.085)	0.015 (0.075)	0.015 (0.047)	0.000 (0.045)	0.108 (0.092)	0.102 (0.076)	0.084 (0.045)*	0.104 (0.045)**
Mean	0.059 (0.053)	-0.022 (0.05)	0.009 (0.028)	-0.001 (0.028)	0.077 (0.066)	0.094 (0.059)	0.083 (0.032)**	0.094 (0.032)***
+1 SD	0.079 (0.069)	-0.06 (0.071)	0.003 (0.035)	-0.002 (0.037)	0.047 (0.09)	0.086 (0.074)	0.082 (0.043)*	0.084 (0.045)*
PPVT-Score -1 SD	0.101 (0.083)	0.014 (0.068)	0.031 (0.044)	0.002 (0.043)	0.092 (0.094)	0.047 (0.078)	0.053 (0.046)	0.051 (0.046)
Mean	0.065 (0.052)	-0.025 (0.05)	0.01 (0.028)	-0.001 (0.028)	0.076 (0.067)	0.089 (0.059)	0.08 (0.032)**	0.09 (0.032)***
+1 SD	0.028 (0.075)	-0.063 (0.082)	-0.011 (0.037)	-0.003 (0.039)	0.06 (0.089)	0.132 (0.075)	0.107 (0.042)**	0.128 (0.043)***
Panel C: Non-Cognitive Skills								
Self-Efficacy -1SD	0.116 (0.085)	-0.015 (0.08)	0.046 (0.046)	0.041 (0.045)	0.123 (0.096)	0.062 (0.08)	0.098 (0.047)**	0.097 (0.048)**
Mean	0.073 (0.054)	-0.025 (0.051)	0.016 (0.028)	0.008 (0.028)	0.083 (0.067)	0.088 (0.058)	0.085 (0.032)***	0.094 (0.033)***
+1 SD	0.03 (0.066)	-0.036 (0.069)	-0.014 (0.035)	-0.025 (0.035)	0.043 (0.081)	0.114 (0.075)	0.072 (0.04)*	0.09 (0.042)**
Self-Esteem -1SD	-0.017 (0.077)	-0.059 (0.076)	-0.010 (0.039)	0.000 (0.04)	0.114 (0.08)	0.127 (0.074)	0.07 (0.04)	0.085 (0.042)**
Mean	0.05 (0.052)	-0.032 (0.052)	0.006 (0.027)	-0.001 (0.028)	0.077 (0.066)	0.095 (0.058)	0.082 (0.031)**	0.093 (0.032)***
+1SD	0.116 (0.068)*	-0.006 (0.063)	0.021 (0.035)	-0.001 (0.035)	0.04 (0.088)	0.062 (0.076)	0.094 (0.041)**	0.1 (0.043)**
Take the lead: Never	0.141 (0.081)*	-0.062 (0.073)	0.033 (0.042)	0.035 (0.044)	0.18 (0.095)*	0.06 (0.075)	0.112 (0.045)**	0.12 (0.046)**
Sometimes	-0.033 (0.085)	-0.043 (0.078)	-0.024 (0.045)	-0.009 (0.045)	0.002 (0.1)	0.064 (0.081)	0.023 (0.049)	0.07 (0.051)
Always	0.071 (0.116)	0.13 ‘ (0.123)	0.026 (0.061)	-0.037 (0.057)	-0.041 (0.136)	0.269 (0.147)*	0.091 (0.067)	0.046 (0.067)
Panel D: Motivation								
Vocational dreamjob: No	0.082 (0.056)	-0.023 (0.057)	0.017 (0.03)	0.007 (0.03)	0.101 (0.072)	0.115 (0.062)*	0.093 (0.034)***	0.096 (0.035)***
Yes	-0.105 (0.148)	0.019 (0.127)	-0.05 (0.076)	-0.015 (0.077)	-0.123 (0.151)	-0.026 (0.122)	-0.056 (0.073)	0.028 (0.084)

Daily hours spent on study: -1SD	-0.011 (0.079)	-0.062 (0.087)	0.004 (0.044)	0.033 (0.043)	-0.105 (0.148)	0.019 (0.127)	-0.05 (0.076)	-0.015 (0.077)
Mean	0.055 (0.052)	-0.031 (0.051)	0.008 (0.028)	0.002 (0.028)	0.023 (0.092)	0.095 (0.098)	0.074 (0.047)	0.13 (0.047)***
+1 SD	0.121 (0.071)	0 (0.076)	0.012 (0.036)	-0.028 (0.038)	0.073 (0.066)	0.093 (0.059)	0.082 (0.032)**	0.095 (0.032)***

Notes: a)*** p < .01, ** p < .05, * p < .1. b) Average marginal effects are computed using MarginalEffects package in R, for numeric variables predictor is held at mean, -1 sd. and +1 sd. deviation c) all estimates use the full set of individual, family and sector covariates, and include cohort, country and country * year fixed effects c) all estimates are pooled from OLS regressions of five multiple imputed datasets using the Mice Package, and standard errors are robust of type HC1, d) all coefficients estimate the average effect on the treated.

unemployed or have a formal job, although the latter two effects are non-significant. For all other variables, there is no indication of heterogeneous effects

The effects are much larger when comparing to dropouts. Panel A shows that TVE impacts among females are significant and substantial. Attending TVE increases the likelihood of being employed by a significant eighteen percentage points. Considering that only 39.8% of women in this subsample are employed, this coincides with an almost 50% increase in the likelihood of employment. A massive impact. It is true that those females would have benefitted similarly from attending general secondary, but for many that would not have been an option. A very similar large effect size is found for formal work (17.5 p.p. with only 26.7% of women formally employed). The effects for hours per week work (0.30 sd./8 hours) and hourly wage (0.13 sd./0.07$) are smaller, but still significant. In sharp contrast, males experience no benefits at all. These results match the effect sizes for females found in several previous studies (Newhouse & Suryadarma, 2011; Fasih et al., 2012; Chakravarty, 2019), but sharply contradict studies finding no differences (e.g., Camargo, 2018; Field, 2019). Indeed, Arias et al. (2019) suggested that earning potential, rather than gender, was the main moderator, with the absence of a gender difference simply indicating that females already had similar opportunities in those countries. My findings point to a potentially different conclusion: the choice of counterfactual may explain why studies comparing to alternative education find no effect, while those comparing to no education finding a much larger effect.

The moderating effect of cognitive skills in Panel B is ambiguous. Math scores at age 13 have little effect on how much a student benefits from TVE. This is not unexpected, as math may not be the most crucial skill for most specialisations. Higher PPVT scores at age 13, a test measuring language skill, may result in slightly larger benefits from TVE, especially concerning employment prospects and formal work. A one standard deviation increase in PPVT score corresponds to a 2.7 percentage point increase in any IGA and a 3.9 percentage point increase in formal work. The literature tends to argue that vocational education can only be effective if students meet a minimum level of cognitive skills (Psacharopoulos, 1993; Loyalka et al., 2016; Jakubowski, 2016). These empirical findings do not confirm this: those lagging behind in math and language can still benefit, although perhaps slightly less.

In Panel C, the moderating effect of the different non-cognitive skills is very limited. Most importantly, lower self-efficacy scores do not negatively affect TVE impact, suggesting that those less proficient in planning and self-management can also benefit from TVE. The same is true for self-esteem. Finally, those who report never taking the lead seem to benefit more from TVE compared to those with a greater tendency to lead. This suggests that leadership and corresponding personality traits are important skills to overcome the educational gap for dropouts. Without those traits, it may be more important to attend secondary education. Still, these non-cognitive skills do not strongly influence returns to education, as was hypothesised by, for example, Camargo et al. (2020). This is important for vocational education, because students choosing vocational education tend to have lower self-efficacy levels, and perhaps also self-esteem, than those opting for general secondary education.

Although the proxies are of lesser quality, Panel D provides some indication that a general motivation to study increases returns to TVE, but a specific motivation for vocational studies does not. Interestingly, intrinsic motivation for vocational jobs, as measured by having a "vocational" dream job at age thirteen, negatively affects TVE outcomes. Vocational students without a "vocational" dream job are significantly more likely to be (formally) employed and earn slightly more. Perhaps students with a general motivation are more willing to learn a variety of skills, giving them more flexibility in the labour market. In contrast, a narrow focus could limit their job prospects if their specific vocational field is not in high demand. Indeed, general motivation to study, as measured by the hours spent studying at home at age 13, does significantly increase the impact of TVE. Those spending above-average time studying are 12.4 percentage points more likely to have a job than those studying one standard deviation less than the mean. While these proxies measure motivation far from perfect, they do suggest that general ambition, motivation, and self-control to spend time on studying help moderate the impact of TVE.

5.4.2. Community Characteristics

When discussing where TVE is most effective, previous authors primarily focus on country-wide differences between TVE systems and the quality of education. Few researchers examine the effect of local labour demand, despite this theoretically playing a large role for vocational students. Given that their occupation-specific skills are less versatile across sectors, it is more likely they will require employment within their field of study. Therefore, local labour demand may be especially important (Nordin et al., 2010; Zhu, 2014; World Bank, 2023). There is no data available on the type of specialisation chosen by the vocational students, but the community-level baseline characteristics include indicators on the relative importance of agriculture, industry, and handicraft/small-scale manufacturing in local labour demand. Most secondary vocational programs will train students for handicraft/small-scale manufacturing, and to a lesser degree for industry. It is thus expected that TVE is more effective in places where those jobs are in higher demand. To test this, I again interact the type of jobs available with treatment and report the average marginal effects in Table 9.

Indeed, the type of jobs available in the community drastically impacts the effects of TVE relative to dropping out, but with general education as the counterfactual, the type of jobs available is no longer an important moderator. This trend is very similar to that found for personal characteristics: the type of jobs available affects the returns to secondary education but does not significantly affect the difference between vocational and general secondary education. When comparing vocational students to those in general education, the impact of TVE appears to be slightly larger, although insignificantly, in towns with smaller populations. A one standard deviation increase in population lowers TVE's impact on any income-generating activity (IGA) and formal employment by 3.6 percentage points. Additionally, Table 9 suggests that it is easier for general secondary students to find a job in a factory. Therefore, in larger, industrial towns, attending general secondary seems to be advantageous over attending vocational secondary. Conversely, in towns that depend more on the crafts and small industry sector, vocational education offers a slight comparative, insignificant advantage.

Comparing vocational students to dropouts paints a very different picture of the type of communities where TVE is most impactful. The impacts of TVE relative to dropouts are largest in large towns. A one standard deviation increase in population increases any income-generating activity (IGA) by 3.4 percentage points and formal work by 4.6 percentage points. The differences are even larger in towns with some jobs in the crafts and small industry sector, where vocational students are 9.4 percentage points (or 20%) more likely to find a job than vocational students in towns where the crafts sector is not important. The difference is even larger for formal jobs, with a 14.5 percentage point increase. A similar trend is seen in

Table 9: Heterogeneous Impacts for Local Labour Demand

Sample:	**Vocationals vs. General Education**				**Vocational vs. Drop Out**			
Outcome *ATT*	Hours per Week Worked	Hourly Income	Any IGA	Formal Work	Hours per Week Worked	Hourly Income	Any IGA	Formal Work
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Population Size -1 SD	0.15 (0.068)**	-0.024 (0.064)	0.045 (0.037)	0.035 (0.036)	0.057 (0.091)	0.016 (0.077)	0.046 (0.045)	0.043 (0.047)
Mean	0.062 (0.052)	-0.028 (0.051)	0.009 (0.027)	-0.001 (0.028)	0.074 (0.066)	0.087 (0.058)	0.08 (0.032)**	0.089 (0.033)***
+1 SD	-0.026 (0.077)	-0.031 (0.073)	-0.028 (0.04)	-0.037 (0.041)	0.09 (0.088)	0.158 (0.076)**	0.113 (0.041)***	0.135 (0.042)***
Agricultural jobs: Not important	0.042 (0.128)	-0.181 (0.146)	0.021 (0.063)	0.002 (0.065)	0.038 (0.139)	0.163 (0.112)	0.074 (0.068)	0.085 (0.069)
Somewhat important	-0.199 (0.199)	-0.057 (0.1)	-0.112 (0.086)	-0.136 (0.094)	-0.02 (0.195)	0.071 (0.079)	0.023 (0.086)	0.124 (0.083)
Very important	0.097 (0.06)	0.028 (0.058)	0.022 (0.032)	0.023 (0.032)	0.098 (0.078)	0.082 (0.067)	0.082 (0.038)**	0.083 (0.039)**
Factory jobs: Not important	0.081 (0.058)	0.023 (0.056)	0.026 (0.03)	0.03 (0.03)	0.058 (0.073)	0.114 (0.066)*	0.071 (0.035)**	0.063 (0.036)*
Somewhat important	-0.038 (0.124)	-0.193 (0.124)	-0.065 (0.065)	-0.107 (0.068)	0.129 (0.147)	0.037 (0.097)	0.087 (0.067)	0.181 (0.068)**
Crafts and small industry jobs: Not important	0.054 (0.059)	0.013 (0.062)	0.006 (0.031)	-0.018 (0.031)	0.07 (0.079)	0.094 (0.07)	0.047 (0.037)	0.046 (0.039)
Somewhat important	0.07 (0.11)	-0.102 (0.088)	0.017 (0.055)	0.063 (0.056)	0.083 (0.113)	0.105 (0.083)	0.141 (0.055)**	0.191 (0.052)***

Notes: a)*** p < .01, ** p < .05, * p < .1. b) Average marginal effects are computed using MarginalEffects package in R, for numeric variables predictor is held at mean, -1 sd. and +1 sd. deviation c) all estimates use the full set of individual, family and sector covariates, and include cohort, country and country * year fixed effects c) all estimates are pooled from OLS regressions of five multiple imputed datasets using the Mice Package, and standard errors are robust of type HC1, d) all coefficients estimate the average effect on the treated, e) in no communities were factory or crafts very important.

towns where factories are more important, with vocational students being much more likely to find a formal job (18.1 percentage points) than dropouts. In contrast, I find no moderating impact of the importance of the agricultural sector. Similarly, the type of jobs available and the size of the town do not affect hours worked per week and hourly income.

Under optimal conditions, such as in larger cities or places with more factory or crafts jobs, marginal estimates of TVE impact are significantly different from zero, especially for employment and formal employment relative to dropouts. Still, general education may slightly outperform vocational secondary education in these large towns. This does not mean that we should avoid building vocational secondary schools in large towns. To the contrary, considering that vocational secondary education helps many students continue their education, these findings suggest that larger cities with many jobs in the crafts sector may be the most effective places for TVE. Yes, these students might experience slightly more benefits had they attended general secondary education, but the point remains that this would often not been the case.

5.4.3. The Impact of Vocational Education by Country

So far, I have assumed homogeneous treatment effects of TVE across the four countries. Since this is an unlikely assumption, I add an interaction effect of treatment with country to equation one. Adding the interactions improves the model fit, with a joint F-test on the added interactions being almost always significant. However, especially for Peru and Vietnam, sample sizes are small, so power is lacking to detect modest effects. As usual, I estimate average marginal effects per country, yet since there are fewer categories I can now present the coefficients in a forest plot for easier interpretation. The table is available in Appendix H.

Figure 5 shows that Indian vocational students may experience some benefits from attending TVE rather than general secondary: they work significantly more hours (0.16 sd./4.0 hours) and are more likely to be employed (7.3 p.p. or 10.8%). They are also slightly less likely to work in a farm (6 p.p.). Finally, Indian vocational students are only 13.5 p.p. or 23% less likely to attend higher education, a much smaller difference than in other countries. In Ethiopia and Vietnam, attending TVE leads to minimally worse outcomes compared to attending general education. They work slightly fewer hours, earn slightly lower wage, and have minimally higher unemployment rates. None of these effects come close to being significant. Only in Vietnam are vocationals significantly less-likely to be self-employed (7.3 p.p.). Specifically in Ethiopia vocational students are very unlikely to continue to higher education (43.9 p.p.), signalling there may be a lack of access to vocational tertiary education. Finally, in Peru, attending TVE leads to slightly more worse outcomes than general education, although the difference is not significant. Vocationals earn less (0.23 s.d./0.23$), work slightly less often (3.2 p.p.) and have fewer formal wage opportunities (7.1. p.p.).

Holding all other variables constant, attending vocational education offers major advantages over dropping out in Ethiopia. Vocationals are 13.7 p.p. or 26% more likely to be employed, and 14.4 p.p. more likely to have a formal job (44% increase). This is mostly due to a shift from self-employment and farming to waged work. However, also in Ethiopia these effects do not translate into significantly more hours worked or a higher wage. In India, Vietnam, and Peru, attending TVE also increases the likelihood of employment by 2 to 11 p.p., but these effects are statistically not significant. The same is true for the shift from self-employed work to formal work, estimates indicate a large positive effect, but this is only significant in Vietnam. Only in Peru does attending TVE increase the hours worked per week with a very large .43 sd (11.4 hours). Finally, I find that between 20 and 40% of vocational students continue to higher education, except in Ethiopia where the transition rates are much lower.

Notes: a) 95% confidence intervals are displayed using robust HC1 standard errors. B) Unit of average marginal effect is standard deviations for hours worked and hourly wage, and percentage points for all other dummies. C) Marginal effects were estimated using package marginal effects according to equation 1.

Figure 5: Heterogeneous Impacts by Country

No quantitative research has yet tried to estimate the impact of secondary TVE in any of these countries, so data for comparison is limited. Still, relative to dropouts, the returns to vocational education limited to employment likelihood and type of employment. They are far off the often-cited private rate of return of an 18.7% increase in salaries by attending secondary education in LMICs (Patrinos & Psacharopoulos, 2020). While the paper by Patrinos and Psacharopoulos employs a Mincer equation, which differs significantly from the approach of directly comparing vocational students to dropouts using propensity scores, this distinction remains noteworthy. Several factors may explain this difference: firstly, standard errors are very large, which prevents small effects from being statistically significant. Additionally, dropouts are likely to have more work experience than the relatively new vocational students, and they could have used this time to secure a (good) job. The value of a few years of work experience may be similar to that of a vocational secondary degree. Furthermore, with high unemployment across these countries, the bargaining power for new entrants to negotiate higher hourly wages may be limited, even with additional education. A methodological reason for this disparity is that this study employs extensive controls for socio-economic status, cognitive skills, and non-cognitive skills, all of which are not included in standard Mincer equations. These controls directly affect the probability of attending vocational secondary education but are also likely to mediate labour outcomes. Considering that "better" students are more likely to attend TVE rather than drop out, as confirmed earlier by the descriptive statistics in Table 3, and assuming these "better" students tend to have better labour outcomes, controlling for selection effects will decrease the apparent rate of return to education.

It is challenging to directly link the performance of TVE in a country to their TVE system, especially considering the comparison involves only four countries. It is evident that TVE effects are heterogeneous across countries. Relative to general education, Indian TVE performed significantly better than that of the other three countries. This suggests that apprenticeships are not a necessary condition for TVE success. Although India is officially a dual system, in practice it is predominantly school-based due to a lack of interest in apprenticeships. Similarly, India is the only country where secondary TVE is not free, demonstrating that non-free secondary TVE can still be effective. Another point of consideration is the effectiveness of the supply-driven curricula in Ethiopia in reducing unemployment. In Ethiopia, the government allocates students to specialisations based on expected labour demand, whereas in the other three countries, students are free to choose their own specialisation. Indeed, in Ethiopia, vocational students are significantly more likely to be employed than dropouts. This may be due to the supply-driven TVE system, which aligns specialisations with labour demand, thereby reducing unemployment. This is further suggested by the sharp increase in the likelihood of securing a formal job after attending TVE. Most TVE specialisations aim to prepare students for formal employment, and thus a sharp increase in formal waged work may indicate greater success in job-matching in Ethiopia compared to the other countries. Finally, in Peru, the effects of TVE appear to be slightly negative compared to general education. Peru's TVE system, however, is very similar to that of Vietnam, where TVE has equal returns compared to general education. One notable difference is that enrolment rates in secondary TVE are much lower in Peru than in other countries. These low enrolment rates may indicate a negative reputation of secondary TVE and/or limited recognisability of a TVE diploma in Peru, which in turn affects how prospective employers value the diploma.

This discussion is intended as a starting point: this research design does not allow for causal determination of the linkage between TVE systems and labour markets, especially considering many other economic conditions may mediate this effect. Therefore, further research is essential. However, the limited availability of cross-country datasets makes such analyses challenging, thus policymakers will likely remain dependent on in-depth cases studies in the near future.

5.5. The Impact of Vocational Education over Time

The time between graduation and the measurement of outcomes varies across the sample. I exploit this variation to explore how the impact of TVE changes over time. To investigate this, I use potential labour market experience as a proxy for job experience, defined as the time since leaving secondary school (for dropouts, the time since their 18th birthday). This is the best proxy available due to the lack of reliable data on the exact moment of graduation from higher education. Consequently, some measurement error may arise from years spent in further education being counted as years of experience.[16] I stratify time since graduation by year and interact this with treatment, thus allowing for non-linear interactions. As before, I report marginal effects per time interval in Figure 6 for the four key outcome variables. While confidence intervals are large and specific strata effects are rarely significant, trends can still

[16] Another point to mention is that the year since graduation is also determined by the speed of study and the structure of the educational system. All survey respondents within the cohort started at a similar age, but some managed to graduate earlier than others, which is likely correlated with (non)-cognitive skills, but also socio-economic status. However, this is not likely to bias the estimates due to the extensive controls for the residual confounding using propensity weighting and the inclusion of all covariates in the outcome regression.

be carefully interpreted. However, since covariates are not balanced across sub-groups, the effects are not causal. Generally, a negative selection effect is expected when comparing to general students, and a positive selection effect is expected when comparing to dropouts. Appendix G shows that the trends over time are the same when using GBM-propensity scores.

Logical reasoning suggests that TVE trains job-specific skills, which provide immediate labour market advantages. However, research indicates that non-formal TVE

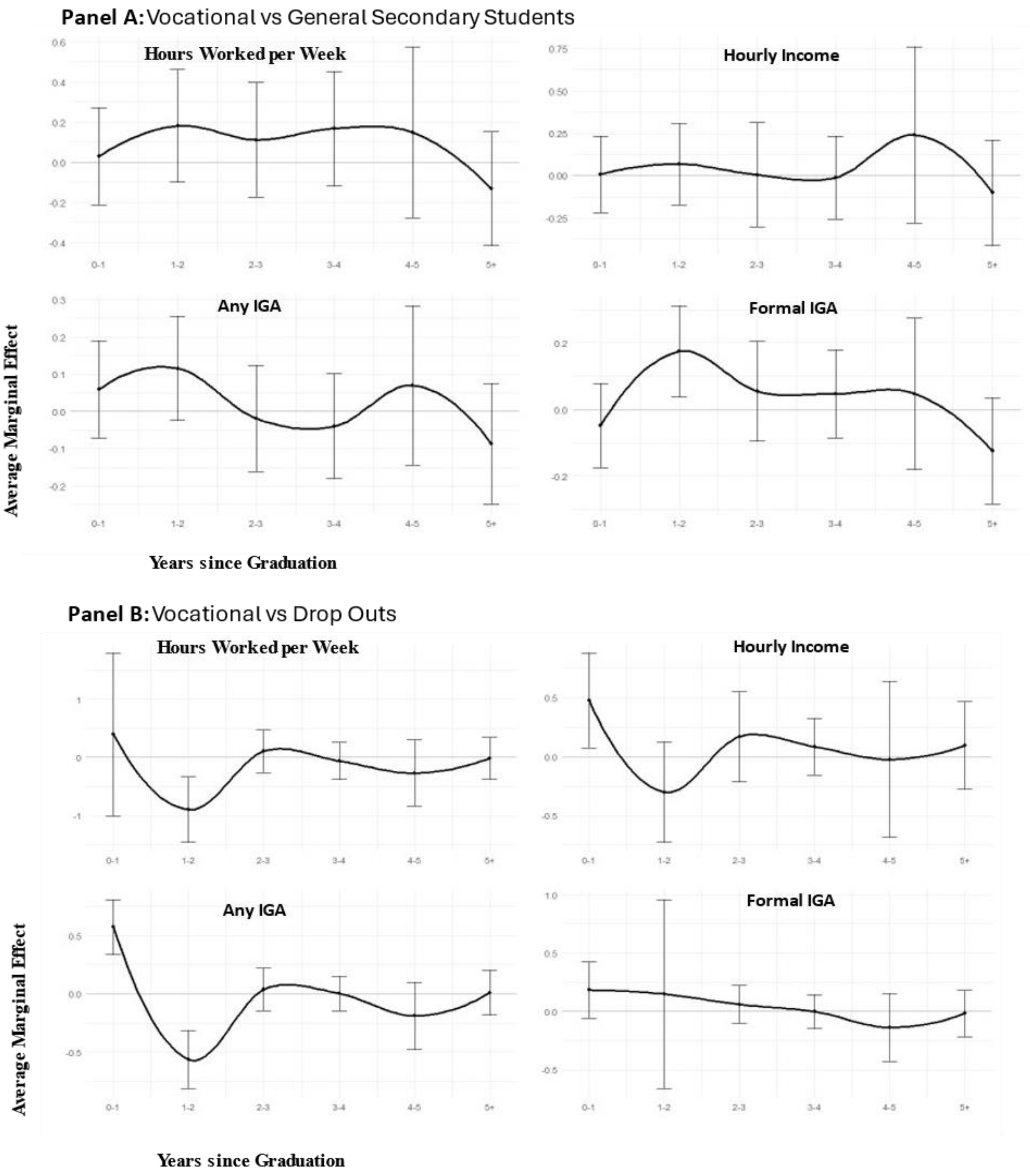


Figure 6: Treatment Effects over Time

training programmes are generally ineffective during the first year, with effects growing in the short-to-medium term, taking around two years to maximise (Card, Kluve & Weber, 2010; Hicks et al., 2016; Chakravarty et al., 2019). This delay may be because vocational students, compared to dropouts, become more selective about the types of jobs they accept, significantly increasing their job search times. However, when they do accept a job, it tends to be of better quality. Another explanation is that vocational secondary students are more likely to retain their jobs (Field, 2019). This, over time, decreases average unemployment rates among vocational students, and longer job durations may also lead to more pay raises. However, in the long term, the benefits of TVE relative to general education are likely to slowly depreciate, with technical skills becoming less relevant due to technological innovations (Golsteyn & Stenberg, 2017; Hanushek et al., 2017; Choi et al., 2019; Card et al., 2010; Ibarrarán, 2019).

Indeed, the comparative advantage of vocational students relative to general students grows during the first two years of potential work experience and then slowly decreases afterwards. These findings largely align with previous research. More specifically, between one and two years after graduation, treatment effects significantly increase the likelihood of having employment (12 percentage points) and of having a formal job (17 percentage points). Additionally, vocational students also work more hours per week (0.19 standard deviations, though this is insignificant). These effects are much more impactful than those found earlier in this paper, but they fade in the longer term. A second, albeit insignificant, bump is observed around four to five years of potential work experience, which may correspond with vocational students who graduated a year earlier from higher vocational education (typically lasting two to three years). After five years, attending TVE does not provide any meaningful advantages over general secondary education. Instead, vocational students tend to be more often unemployed, earn slightly less, work fewer hours, and have fewer formal jobs. This supports the argument that vocational skills, relative to general skills, offer a small market advantage that diminishes quickly over time. Simultaneously, it suggests that vocational students are not more likely to retain their jobs. While job retention could not be measured directly, it would be expected that treatment effects, especially for any income-generating activity (IGA), would show a continuous upward trend if job retention were higher among vocational students.

Comparing vocational students to dropouts, the marginal effects over time exhibit significant changes. The comparative advantage of vocational students is largest during the first year. They are much more likely to immediately have a job than comparable 18-19-year-old dropouts (55 percentage points), work 0.39 standard deviations more hours per week, and earn 0.47 standard deviations more per hour. The unemployment and income effects are highly significant, but the effect on hours worked per week is not. These effects are much larger than the average effects estimated earlier.

For those with one or two years of potential work experience, the reverse is true, with attending TVE leading to much more unemployment and fewer hours worked per week. In the long run, the average marginal effect balance out around zero. The volatility in the results, combined with the very large treatment effects, calls into question the reliability of the point estimates. Despite there being no indication of what could cause such bias, there is no theoretical reason to expect such a large shift between subsequent one-year periods. Still, interpreting the general trend, it suggests that attending vocational education provides an immediate advantage over dropping out after leaving school, but these benefits quickly fade to little or no comparative advantage.

This differs from previous research, which finds that the benefits of TVE take at least a year to materialise. This discrepancy is likely due to using a different counterfactual, offering a new perspective. Dropouts aged 20/21 may have gained enough practical work experience to offset the immediate market advantages of a TVE diploma, especially since they likely started working much earlier than their 18th birthday and may have accumulated more experience than their peers of the same age. I also fail to confirm long-term (after five years) impacts TVE, as were previously found for informal training programs by Attanasio (2017) and Ibarrarán (2019), but statistical power is very limited.

Finally, the validity of these findings is severely limited by the sample size and the cross-sectional nature of the data. I recommend longitudinal follow-up research, using Wave 7 of the Young Lives data (available at the start of 2025), to establish effects with stronger causal inferences.

6. Conclusion

6.1. Main Findings

This study found that secondary TVE has a nuanced impact on labour outcomes, with significant variations across countries and gender. Overall, attending secondary TVE provided no advantages over attending general secondary education. More specifically, vocational education did not result in higher wages, more hours worked, or less unemployment. Treatment effects were also rarely significant in sub-groups. Only in India did attending vocational secondary lead to a significant increase in hours worked per week and higher employment chances. Estimates of TVE impacts were also larger among women, but this impact was not always statistically significant. When considering general versus vocational secondary education, this study aligns with the group of observational studies showing no advantages of TVE over general education (e.g., Borkum et al., 2017; Campuzano et al., 2016; Kraft, 2018).

However, I argue the literature evaluating the impact of TVE tends to underestimate its major advantage: TVE functions as an alternative educational pathway for many students who would likely have dropped out without it. By calculating the propensity to attend general education, I estimated that roughly 54.6% of vocational students would have dropped out after primary education had TVE not existed. In Ethiopia, this effect is largest, with 82.5% of vocational students otherwise dropping out. For these students, individuals who dropped out after primary education are a much more realistic counterfactual than general secondary students. Previous observational research tended to exclusively focus on comparing vocational versus general secondary students, but by doing so, I argue they miss a large part of the picture. Because when comparing vocational students to dropouts, TVE's impacts are much more substantial, and thus similar to general secondary's impact vs. dropouts. Across vocational students, there is a shift from self-employment (-6.3 percentage points or 33%) to formally waged work (9.1 percentage points or 28%). This is important since formally waged work is associated with benefits in terms of job security and access to government benefits. These positive effects, however, do not translate into a significant increase in hours worked, income, or overall employment. However, there was substantial heterogeneity. Most notably, in Ethiopia attending vocational education brings major advantages relative to dropping out, with a 13.5 percentage point or 26% increase in employment. Simultaneously, there is a 16.4 percentage point or 51% increase in formal employment, shifting jobs from micro-entrepreneurship and farming to formally waged work. However, this did not translate into a significant increase in hours worked per week or hourly wage, suggesting that employed dropouts are likely working more hours. A similar trend was found in the three other countries, but with smaller effect sizes, and thus generally insignificantly. All findings are robust to using different model specifications in the outcome regression and using propensity scores calculated with a generalised boosted model.

As importantly, attending TVE was only effective for females. Among females, attending vocational instead of dropping out increased hours worked per week with eight hours, hourly income by 0.134 standard deviations, and employment by eighteen percentage points, an almost 50% increase. The likelihood of having formal work increased massively by 17.5 percentage points, corresponding to a 69% increase. In contrast, TVE had no positive impact for males. This may be because women face more barriers to entry in the labour market than men, increasing the importance of obtaining the credentials of a vocational secondary. Conversely, men may have better access to alternative pathways to employment, such as informal work or entry-level positions. These effects correspond, also in effect sizes, with studies finding very large differences between genders (Newhouse & Suryadarma, 2011; Fasih et al., 2012; Chakravarty, 2019), but sharply contrast with studies finding no differences between genders (e.g., Camargo, 2018; Field, 2019). I hypothesise that the composition of the counterfactual group may help explain these different findings, with authors comparing vocational to general education finding little to no effects.

6.2. Policy Implications

It is well established that the cost of providing secondary TVE is significantly higher than that of general secondary education. These additional costs are not justified by any significant advantage of TVE over general education. However, the investment might be worthwhile when considering that vocational education allows many more students to achieve a secondary diploma. Even when comparing to dropouts, the average effects of TVE are limited to changes in job composition. Yet, when considering heterogeneity, I find major impacts of TVE in Ethiopia and among females, massively increasing employment, formal employment, and hours worked per week. I also find no reason to suggest that vocational students at risk of dropping out benefitting any less from TVE. This means that TVE's ability to serve as an alternative educational pathway should play a much bigger role in the debate on its effectiveness in LMICs. Additionally, while this study exclusively focused on private benefits, higher secondary enrolment also brings societal benefits. Increased enrolment may contribute to a more educated and skilled population, enhancing overall productivity and economic development. Also, compared to general education, TVE reaches marginalised communities better, potentially helping to reduce social inequalities. Practitioners should explicitly consider how to value such broader societal impacts when making their investment decisions.

When evaluating a proposal for a new vocational secondary school, the primary consideration should not be the marginal advantages of vocational education over general education. Instead, the crucial question is how effective a vocational secondary school is in helping potential dropouts continue their secondary education. This effectiveness varies between contexts, countries, and possibly regions, being most pronounced in Ethiopia and to a lesser extent in Vietnam and India within this sample. In such contexts, TVE brings the largest marginal benefits. The type of benefits that may well outweigh the extra costs of vocational education. I recommend further research into what determines whether secondary TVE successfully reaches an otherwise uneducated group within the population.

At the same time, practitioners may want to maximise their impact by actively attracting dropouts to vocational secondary education. This can be done on a case-by-case basis, perhaps using awareness campaigns, recruitment events, and offering specific specialisations that are deemed attractive. A similar strategy can be used to attract females to vocational secondary, among whom the marginal impact is much larger. It must be noted that the estimates in this paper are average effects on the treated and cannot be generalised to average treatment effects for the control. However, considering the extensive controls, limited differences in descriptives and robust findings, the evidence strongly suggests dropouts would benefit in similar ways. Additionally, the small difference between the linear regression and propensity score weighted regressions suggest limited effects of selection bias.

Additionally, I found vocational secondaries to be much more effective in larger towns, and places with more jobs in factories, handcrafts, and small-scale manufacturing. The difference to small villages or towns, who predominantly rely on agriculture is significant. Thus, I recommend practitioners carefully consider local labour demand when deciding where to place a vocational school. On a larger scale, this study was unfortunately not equipped to conclude which type of TVE-system is most effective. The heterogeneity between countries underscores the importance of tailoring vocational education programs to the specific economic contexts and labour market demands of each country, but at the same time this heterogeneity did not correlate with clear differences in TVE-systems. More research is required to understand under which conditions TVE works best.

The study also explored how the impact of vocational education changes over time. Compared to general education, the benefit of attending vocational education is highest after two years. This trend aligns with findings by Card, Kluve, and Weber (2010) and Hicks et al. (2016), who noted that vocational training programs often show delayed but growing impacts in the short to medium term. A new finding is that this differs when using dropouts as the comparison group. During the first-year post-graduation, the impact of vocational education is very large relative to 18/19-year-old dropouts, but then quickly fades away. Although this analysis is underpowered, it is clear the impact of vocational education changes over time.

This has several implications for practitioners: when conducting an impact evaluation on TVE, the time between graduation and outcome measurement is likely to significantly affect the results. If possible, I recommend measuring outcomes at various times. Additionally, follow-up research identifying why TVE impacts change over time could help design supplementary low-cost interventions to increase TVE impact cost-effectively. For example, if the hypothesis that vocational students are too restrictive when accepting jobs post-graduation proves true, additional classes on expectation management might be fruitful.

6.3. Limitations

This study has several limitations that should be acknowledged. Firstly, this study estimated average effects on the treated. Thus, any reported coefficients are only generalisable across current vocational students and cannot be seen as average effects for the population, as there was insufficient overlap to estimate average treatment effects. In practical terms, this means the coefficients describe the effect of stopping with vocational secondary education, rather than the effect of expanding TVE. Additionally, the effects are only causally interpretable under the assumption of balanced covariates and unconfoundedness. The unconfoundedness assumption is very likely met considering the wide range of baseline characteristics and balance is perfectly achieved for the main analysis, but for the follow-up analysis on timed effects and the moderating effect of propensity scores to attend general education balance is lacking. This means residual confounding prevents causal interpretations in those cases.

Secondly, the study relies on self-reported data, which can be subject to measurement errors and biases. Of particular concern is the discrepancy between the much higher percentage of people attending TVE in the sample and the official UNESCO data. This suggests that, mainly in Ethiopia, some students studying at a competence-based general secondary school reported being in a vocational school. While this is not a major issue, considering a competence-based general secondary has many elements of TVE, it could still dilute the estimates. To correct this, Young Lives should ask a clarification questions in the next wave to correct the data. Additionally, the validity of results would be strengthened if Young Lives would validate some of the self-reported data across treatment groups, to check for systematic differences in measurement errors across different educational levels, even though there is no indication that this is a problem.

Finally, sub-group analyses were limited by several factors, including the limited data available. The Young Lives data was not collected with the intent to evaluate vocational secondary education, so only 467 out of 12,000 people attended vocational secondary. At the start of 2025, Young Lives will publish a new wave of data, which will now also include outcomes for the slower vocational students in the Younger Cohort. This may increase the sample size and allow for longitudinal analysis with more power. A larger dataset would also allow for a more complex model testing different sub-group analyses simultaneously, enabling the identification of more complex moderating effects, while also strengthening balance across sub-groups. Additionally, it was not possible to test several moderators named in the literature review, such as economic conditions at the time of graduation and private versus government vocational schools.

By building on the insights from this study, policymakers can better leverage vocational secondary education as a tool for economic development and social progress. As importantly, the recommendation to use dropouts as a secondary counterfactual can result in better future evaluations of TVE-impact. In the end, secondary TVE is not the silver bullet everybody once hoped it to be, but when used as an alternative to general education for dropouts, TVE is an important development tool. A well-designed vocational education system has the potential to offer significant benefits both for students who would otherwise have dropped out and for society as a whole. As Nelson Mandela once said, “Education is the most powerful weapon which you can use to change the world.”

7. Bibliography

Acevedo, P., Cruces, G., Gertler, P., & Martinez, S. (2020). How vocational education made women better off but left men behind. Labour Economics, 65, 101824.

ADB (Asian Development Bank). (2008).“Education and Skills: Strategies for Accelerated Development in Asia and the Pacific”. Manila, Philippines: ADB

Adelson, J. L. (2019). Educational research with real-world data: Reducing selection bias with propensity score analysis. Practical Assessment, Research, and Evaluation, 18(1), 15.

Adoho, F., Chakravarty, S., Korkoyah, D. T., Lundberg, M. K., & Tasneem, A. (2014). The impact of an adolescent girls employment program: The EPAG project in Liberia. *World Bank Policy Research Working Paper*, (6832).

Agrawal, T., & Agrawal, A. (2017). Vocational education and training in India: a labour market perspective. *Journal of Vocational Education & Training*, *69*(2), 246-265.

Akanbi, G. O. (2017). Prospects for technical and vocational education and training (TVET) in Nigeria: Bridging the gap between policy document and implementation. International Education Journal: Comparative Perspectives, 16(2), 1-15.

Almeida, R., Anazawa, L., Menezes Filho, N., & Vasconcellos, L. (2015). Investing in Technical and Vocational Education and Training.

Amoah, J., Stuart, E. A., Cosgrove, S. E., Harris, A. D., Han, J. H., Lautenbach, E., & Tamma, P. D. (2020). Comparing propensity score methods versus traditional regression analysis for the evaluation of observational data: a case study evaluating the treatment of gram-negative bloodstream infections. Clinical Infectious Diseases, 71(9), e497-e505.

Arias, O., Evans, D. K., & Santos, I. (2019). The skills balancing act in Sub-Saharan Africa: Investing in skills for productivity, inclusivity, and adaptability. World Bank Publications.

Attanasio, O., Guarín, A., Medina, C., & Meghir, C. (2017). Vocational training for disadvantaged youth in Colombia: A long-term follow-up. *American Economic Journal: Applied Economics*, *9*(2), 131-143.

Austin, P. C. (2009). Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in medicine*, *28*(25), 3083-3107.

Austin, P. C. (2011). An introduction to propensity score methods for reducing the effects of confounding in observational studies. Multivariate behavioral research, 46(3), 399-424.

Aydede, Y., & Orbay, B. Z. (2016). Comparison of economic benefits of university and vocational high school education among different majors in Turkey. European journal of higher education, 6(2), 176-192.

Barnett, I., Ariana, P., Petrou, S., Penny, M. E., Duc, L. T., Galab, S., ... & Boyden, J. (2013). Cohort profile: the Young Lives study. International journal of epidemiology, 42(3), 701-708.

Baser, O. (2007). Choosing propensity score matching over regression adjustment for causal inference: when, why and how it makes sense. Journal of Medical Economics, 10(4), 379-391.

Bennell, P. (1996). General versus vocational secondary education in developing countries: a review of the rates of return evidence. *The Journal of Development Studies*, *33*(2), 230-247.

Bennell, P. (2023). Practising what you preach, preaching what you practice: World Bank support for technical and vocational education and training in sub-Saharan Africa. Compare: A Journal of Comparative and International Education, 53(8), 1410-1428

Bettinger, E., Kremer, M., & Saavedra, J. E. (2007). How do vouchers work? Evidence from Colombia. Unpublished working paper.

Borkum, E., Mamun, A., & Mubeen, M. K. (2017). *Evaluation of the Vocational Training Grant Fund in Namibia* (No. c23d691613754f048ef486f035804271). Mathematica Policy Research.

Boyden, J. (2016). Young Lives: An International Study of Childhood Poverty. *Oxford Department of International Development, University of Oxford.*

Brand, J. E., & Thomas, J. S. (2013). Causal effect heterogeneity. In Handbook of causal analysis for social research (p.p 189-213). Dordrecht: Springer Netherlands.

Brand, J. E., & Xie, Y. (2010). Who benefits most from college? Evidence for negative selection in heterogeneous economic returns to higher education. American sociological review, 75(2), 273-302.

Briones, K. (2018). A guide to Young Lives rounds 1 to 5 constructed files. Young Lives Technical Note, 48, 1-31.

Brookhart, M. A., Schneeweiss, S., Rothman, K. J., Glynn, R. J., Avorn, J., & Stürmer, T. (2006). Variable selection for propensity score models. American journal of epidemiology, 163(12), 1149-1156.

Brunello, G., & Schlotter, M. (2011). Non-cognitive skills and personality traits: Labour market relevance and their development in education & training systems.

Caicedo, S., Espinosa, M., & Seibold, A. (2022). Unwilling to train?—Firm responses to the Colombian apprenticeship regulation. Econometrica, 90(2), 507-550.

Camargo, J., Lima, L., Riva, F., & Souza, A. P. (2018). *Technical education, noncognitive skills and labour market outcomes: experimental evidence from Brazil*. Escola de Economia de São Paulo da Fundação Getulio Vargas.

Campuzano, L., Blair, R., Poggio, V., Padilla, I., Morgan, S., & Brannan, M. (2016). Evaluation of the Formal Technical Education Sub-Activity of the Human Development Project, El Salvador. *Washington, DC: Mathematica Policy Research*.

Canales, A., & Maldonado, L. (2018). Teacher quality and student achievement in Chile: Linking teachers' contribution and observable characteristics. International journal of educational development, 60, 33-50.

Card, D., Kluve, J., & Weber, A. (2010). Active labour market policy evaluations: A meta-analysis. *The economic journal*, *120*(548), F452-F477.

Chakravarty, S., Lundberg, M., Nikolov, P., & Zenker, J. (2019). Vocational training programs and youth labour market outcomes: Evidence from Nepal. *Journal of Development Economics*, *136*, 71-110.

Choi, S. J., Jeong, J. C., & Kim, S. N. (2019). Impact of vocational education and training on adult skills and employment: An applied multilevel analysis. *International Journal of Educational Development*, *66*, 129-138.

Deissinger, T. (2014). TVET system research. In Areas of vocational education research (p.p 91-108). Berlin, Heidelberg: Springer Berlin Heidelberg.

Deissinger, T. (2015). The German dual vocational education and training system as ‘good practice’?. Local Economy, 30(5), 557-567.

DuGoff, E. H., Schuler, M., & Stuart, E. A. (2014). Generalizing observational study results: applying propensity score methods to complex surveys. *Health services research*, *49*(1), 284-303.

Dunifon, R., & Duncan, G. J. (1998). Long-run effects of motivation on labour-market success. Social Psychology Quarterly, 33-48.

El-Hamidi, Fatma. "General or vocational schooling? Evidence on school choice, returns, and sheepskin effects from Egypt 1998." *Journal of Economic Policy Reform* 9.2 (2006): 157-176.

Esarey, J., & Sumner, J. L. (2018). Marginal effects in interaction models: Determining and controlling the false positive rate. *Comparative Political Studies*, *51*(9), 1144-1176.

Farías, M., & Sevilla, M. P. (2015). Effectiveness of vocational high schools in students' access to and persistence in postsecondary vocational education. Research in higher education, 56, 693-718.

Fasih, T., Kingdon, G. G., Patrinos, H. A., Sakellariou, C., & Söderbom, M. (2012). Heterogeneous returns to education in the labour market. World Bank Policy Research Working Paper, (6170).

Field, E. M., Linden, L. L., Malamud, O., Rubenson, D., & Wang, S. Y. (2019). Does vocational education work? Evidence from a randomised experiment in Mongolia (No. w26092). National Bureau of Economic Research.

Fukunishi, T., & Machikita, T. (2017). Vocational education and employment outcomes in Ethiopia: displacement effects in local labour markets. *IDE Discussion Paper*, *678*.

Funk, M. J., Westreich, D., Wiesen, C., Stürmer, T., Brookhart, M. A., & Davidian, M. (2011). Doubly robust estimation of causal effects. *American journal of epidemiology*, *173*(7), 761-767.

Gaentzsch, A., & Zapata-Román, G. (2020). *Climbing the ladder: Determinants of access to and returns from higher education in Chile and Peru* (No. 2020-2). UNRISD Working Paper.

Glewwe, P., & Muralidharan, K. (2016). Improving education outcomes in developing countries: Evidence, knowledge gaps, and policy implications. In Handbook of the Economics of Education (Vol. 5, p.p 653-743). Elsevier.

Golsteyn, B. H., & Stenberg, A. (2017). Earnings over the life course: General versus vocational education. Journal of Human Capital, 11(2), 167-212.

Grave, B. S., & Goerlitz, K. (2012). Wage differentials by field of study–the case of German university graduates. Education Economics, 20(3), 284-302.

Greifer, N. (2020). Covariate balance tables and plots: a guide to the cobalt package. Accessed March, 10, 2020.

Guo, D., & Wang, A. (2020). Is vocational education a good alternative to low-performing students in China. *International Journal of Educational Development*, *75*, 102187.

Gust, S., Hanushek, E. A., & Woessmann, L. (2024). Global universal basic skills: Current deficits and implications for world development. Journal of Development Economics, 166, 103205.

Hagos Baraki, A., & van Kemenade, E. (2013). Effectiveness of Technical and Vocational Education and Training (TVET) Insights from Ethiopia's reform. The TQM Journal, 25(5), 492-506.

Hanushek, E. A., Schwerdt, G., Woessmann, L., & Zhang, L. (2017). General education, vocational education, and labour-market outcomes over the lifecycle. Journal of human resources, 52(1), 48-87.

Heckman, J. J., Stixrud, J., & Urzua, S. (2006). The effects of cognitive and noncognitive abilities on labour market outcomes and social behavior. Journal of Labour economics, 24(3), 411-482.

Hicks, J. H., Kremer, M., Mbiti, I., & Miguel, E. (2013). Vocational education in Kenya: Evidence from a randomised evaluation among youth. *Nashville, TN: Vanderbilt University*.

Hirshleifer, S., McKenzie, D., Almeida, R., & Ridao-Cano, C. (2016). The impact of vocational training for the unemployed: experimental evidence from Turkey. The Economic Journal, 126(597), 2115-2146.

Horowitz, A. W., & Schenzler, C. (1999). Returns to general, technical and vocational education in developing countries: recent evidence from Suriname. *Education Economics*, *7*(1), 5-20.

Ibarrarán, P., Kluve, J., Ripani, L., & Rosas Shady, D. (2019). Experimental evidence on the long-term effects of a youth training program. *ILR Review*, *72*(1), 185-222.

IBRD, 1995. World Bank Policy Paper: Vocational and Technical Education and Training. Washington DC: World Bank.

ILO and UNESCO. (2018). A Global Overview of TVET Teaching and Training: Current Issues, Trends and Recommendations. Geneva: ILO

Imbens, G. W., & Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press.

Jakubowski, M., Patrinos, H. A., Porta, E. E., & Wiśniewski, J. (2016). The effects of delaying tracking in secondary school: evidence from the 1999 education reform in Poland. Education Economics, 24(6), 557-572.

Kahn, L. B. (2010). The long-term labour market consequences of graduating from college in a bad economy. Labour economics, 17(2), 303-316.

Kahyarara, G., & Teal, F. (2008). The returns to vocational training and academic education: Evidence from Tanzania. *World Development*, *36*(11), 2223-2242.

Krafft, C. (2018). Is school the best route to skills? Returns to vocational school and vocational skills in Egypt. *The Journal of Development Studies*, *54*(7), 1100-1120.

Krishnan, P., & Shaorshadze, I. (2013). Technical and vocational education and training in Ethiopia. *International Growth Centre, London School of Economics and Political Science*.

Kunter, M., Klusmann, U., Baumert, J., Richter, D., Voss, T., & Hachfeld, A. (2013). Professional competence of teachers: Effects on instructional quality and student development. Journal of educational psychology, 105(3), 805.

Lassibille, G., & Tan, J. P. (2005). The returns to education in Rwanda. *Journal of African Economies*, *14*(1), 92-116.

Li, C. (2013). Little's test of missing completely at random. *The Stata Journal*, *13*(4), 795-809.

Li, F., Zaslavsky, A. M., & Landrum, M. B. (2013). Propensity score weighting with multilevel data. *Statistics in medicine*, *32*(19), 3373-3387.

Lipnevich, A. A., & Roberts, R. D. (2012). Noncognitive skills in education: Emerging research and applications in a variety of international contexts. Learning and Individual Differences, 22(2), 173-177.

Liu, K., Salvanes, K. G., & Sørensen, E. Ø. (2014). Bad times at a tender age–how education dampens the impact of graduating in a recession. Nordic Economic Policy Review, 5(1), 51-73.

Loyalka, P., Huang, X., Zhang, L., Wei, J., Yi, H., Song, Y. = & Chu, J. (2016). The impact of vocational schooling on human capital development in developing countries: evidence from China. The World Bank Economic Review, 30(1), 143-170.

Mahirda, K., & Wahyuni, H. (2016). Returning to general and vocational high-schools in Indonesia. *Review of Economic and Business Studies*, *9*(2), 9-28.

Malamud, O., & Pop-Eleches, C. (2010). General education versus vocational training: Evidence from an economy in transition. *The review of economics and statistics*, *92*(1), 43-60.

Markoulidakis, A., Holmans, P., Pallmann, P., Busse, M., & Griffin, B. A. (2021). How balance and sample size impact bias in the estimation of causal treatment effects: a simulation study. *arXiv preprint arXiv:2107.09009*.

McCaffrey, D. F., Griffin, B. A., Almirall, D., Slaughter, M. E., Ramchand, R., & Burgette, L. F. (2013). A tutorial on propensity score estimation for multiple treatments using generalised boosted models. Statistics in medicine, 32(19), 3388-3414.

McKenzie, D. (2017). How effective are active labour market policies in developing countries? a critical review of recent evidence. The World Bank Research Observer, 32(2), 127-154.

McKenzie, D. (2020). Small Business Training to Improve Management Practices in Developing Countries Reassessing the Evidence for "Training Doesn't Work." http://www.worldbank.org/prwp.

McMurry, T. L., Hu, Y., Blackstone, E. H., & Kozower, B. D. (2015). Propensity scores: methods, considerations, and applications in the Journal of Thoracic and Cardiovascular Surgery. *The Journal of thoracic and cardiovascular surgery*, *150*(1), 14-19.

Meer, J. (2007). Evidence on the returns to secondary vocational education. Economics of education review, 26(5), 559-573.

Moenjak, T., & Worswick, C. (2003). Vocational education in Thailand: a study of choice and returns. *Economics of Education Review*, *22*(1), 99-107.

Montenegro, C. E., & Patrinos, H. A. (2014). Comparable estimates of returns to schooling around the world. World Bank Policy Research Working Paper, (7020). https://doi.org/10.1596/1813-9450-7020

Newhouse, D., & Suryadarma, D. (2011). The value of vocational education: High school type and labour market outcomes in Indonesia. *The World Bank Economic Review*, *25*(2), 296-322.

Newman, D. A. (2014). Missing data: Five practical guidelines. *Organisational Research Methods*, *17*(4), 372-411.

Nikolov, P., Jimi, N., & Chang, J. (2020). The importance of cognitive domains and the returns to schooling in South Africa: Evidence from two labour surveys. Labour economics, 65, 101849.

Nomura, S., Rahman, M., Goyal, S., Nakata, S., & Al-Zayed, S. R. (2015). How does the short-term training program contribute to skills development in Bangladesh? A tracer study of the short-term training graduates (No. 103117, p.p 1-81). The World Bank.

Nordin, M., Persson, I., & Rooth, D. O. (2010). Education–occupation mismatch: Is there an income penalty?. Economics of education review, 29(6), 1047-1059.

OECD (2018) TVET Country Profile Vietnam. Accessed April 1, 2024 at https://unevoc.unesco.org/wtdb/worldtvetdatabase_vnm_en.pdf

Onukwugha, E., Bergtold, J., & Jain, R. (2015). A primer on marginal effects—Part I: Theory and formulae. *Pharmacoeconomics*, *33*, 25-30.

Oreopoulos, P., Von Wachter, T., & Heisz, A. (2012). The short-and long-term career effects of graduating in a recession. American Economic Journal: Applied Economics, 4(1), 1-29.

Ozawa, S., Laing, S. K., Higgins, C. R., Yemeke, T. T., Park, C. C., Carlson, R., & Omer, S. B. (2022). Educational and economic returns to cognitive ability in low-and middle-income countries: A systematic review. World development, 149, 105668.

Patrinos, H. A., & Psacharopoulos, G. (2020). Returns to education in developing countries. In The Economics of education (p.p 53-64). Academic Press.

Patrinos, H. A., Psacharopoulos, G., & Tansel, A. (2019). Returns to investment in education: The case of Turkey. *Available at SSRN 3358397*.

Pishgar, F., Greifer, N., Leyrat, C., & Stuart, E. (2020). MatchThem:: matching and weighting after multiple imputation. *arXiv preprint arXiv:2009.11772*.

Psacharapoulos, G. (1993). Returns to Investment in Education: A Global Update. WPS. Washington DC: World Bank.

Psacharopoulos, G. (1987). To vocationalize or not to vocationalize? That is the curriculum question. *International review of education*, *33*, 187-211.

Psacharopoulos, G., & Patrinos, H. A. (2018). Returns to investment in education: A further update. Education Economics, 26(5), 445-458. https://doi.org/10.1080/09645292.2018.1484426

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. Biometrika, 70(1), 41-55. https://doi.org/10.1093/biomet/70.1.41

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American statistical Association*, *91*(434), 473-489.

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. Contemporary educational psychology, 25(1), 54-67.

Sánchez, A., & Escobal, J. (2020). Survey attrition after 15 years of tracking children in four developing countries: The Young Lives study. Review of Development Economics, 24(4), 1196-1216.

Sanwal, S. (2019). “A study of Vocationalisation of School Education in Andhra Pradesh- A Research Report”. Accessed at https://www.researchgate.net/publication/333798060_A_study_of_Vocationalisation_of_School_Education_in_Andhra_Pradesh-_A_Research_Report/link/5d04a89c92851c90043c980f/download?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19 ,

Setodji, C. M., McCaffrey, D. F., Burgette, L. F., Almirall, D., & Griffin, B. A. (2017). The right tool for the job: choosing between covariate-balancing and generalised boosted model propensity scores. Epidemiology, 28(6), 802-811.

Silliman, M., & Virtanen, H. (2022). Labour market returns to vocational secondary education. American Economic Journal: Applied Economics, 14(1), 197-224.

Sirait, S. (2016). Does teacher quality affect student achievement? An empirical study in Indonesia. Journal of Education and Practice, 7(27).

Tilak, J. 1988. “Economics of Vocationalisation: A Review of the Evidence” Canadian and International Education 17(1).

Tran, T. Q., & Van Vu, H. (2020). Wage earning differentials by field of study: Evidence from Vietnamese university graduates. International Journal of Educational Development, 78, 102271.

Tuan, N. D., & Cuong, N. H. (2019). Technical and vocational education and training (TVET) in Vietnam. *Vocational education and training in ASEAN member states: Current status and future development*, 229-256.

UNESCO (2021). “Estudio Sobre la Situación Actual de la Docencia en la Educación y Formación Técnica Profesional en Bolivia, Colombia, Ecuador y Venezuela.” Paris: UNESCO

UNESCO (2024a). TVET Country Profile: Ethiopia. Accessed on March 27, 2024 at https://unevoc.unesco.org/home/Dynamic+TVET+Country+Profiles/country=ETH

UNESCO (2024b). TVET Country Profile: India. Accessed on March 27, 2024 at https://unevoc.unesco.org/home/Dynamic+TVET+Country+Profiles/country=IND

UNESCO (2024c). TVET Country Profile: Peru. Accessed on March 27, 2024 at https://unevoc.unesco.org/home/Dynamic+TVET+Country+Profiles/country=PER

UNESCO (2024d). TVET Country Profile: Vietnam. Accessed on March 27, 2024 at https://unevoc.unesco.org/home/Dynamic+TVET+Country+Profiles/country=VNM

United Nations (2024). Goal 4. Accessed June 20th 2024 at https://sdgs.un.org/goals/goal4#targets_and_indicators

Valiente, Oscar, Queralt Capsada-Munsech, and Jan Peter G de Otero. "Educationalisation of youth unemployment through lifelong learning policies in Europe." European Educational Research Journal 19, no. 6 (2020): 525-543.

Van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of statistical software*, *45*, 1-67.

Van den Berge, W. (2018). Bad start, bad match? The early career effects of graduating in a recession for vocational and academic graduates. Labour Economics, 53, 75-96.

Vandenberg, P., & Laranjo, J. (2020). The impact of vocational training on labour market outcomes in the Philippines. *Asian Development Bank Economics Working Paper Series*, (621).

Vanderhoven, E., Fontdevila, C., Langthaler, M., Valiente, O., Hermann, R., Marković, J., ... & Calderón, E. (2024). Realising the human development promise in dual VET. Compare: A Journal of Comparative and International Education, 1-16.

Vo, X. T. (2019). Dual or Complemental in Vocational Training: which concept would be suitable for TVET Vietnam?. TVET@ Asia, (13), 1-16.

White, I. R., Royston, P., & Wood, A. M. (2011). Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, *30*(4), 377-399

World Bank (2024a). Vocational and technical enrolment (% of total secondary enrolment). Accessed on March 28, 2024 at https://genderdata.worldbank.org/indicators/se-sec-enrl-vo-zs/?gender=total

World Bank (2024b). School enrollment, secondary (% net). Accessed on March 28, 2024 at https://data.worldbank.org/indicator/SE.SEC.NENR

World Bank, UNESCO, & ILO. (2023). Building Better Formal TVE Systems: Principles and Practice in Low-and Middle-Income Countries.

Young Lives (2024). Sampling and attrition. Accessed March 31, 2024 at https://www.younglives.org.uk/sampling-and-attrition

Young Lives. (2023). *COVID-19 phone survey report Call 5: Attrition report*. Young Lives. Retrieved from https://www.younglives.org.uk/sites/default/files/2023-04/yl-covid-19-phonesurvey5-attritionreport.pdf

Zhu, R. (2014). The impact of major–job mismatch on college graduates' early career earnings: evidence from China. Education Economics, 22(5), 511-528.

Appendix A: Baseline Characteristics for Propensity Scores

Variables in Italic are not included in outcome regressions to prevent multicollinearity. More information on the computation and exact definitions of the variables can be found in Supplementary Materials II.

Table 10: Baseline Characteristics for Propensity Scores and Outcome Regression

Level	Dimension	Variables
Individual	Basic Demographics	- Gender - *Ethnic group* - *Region*
	Child's Health	Early childhood - *Underweight* - Stunting - Thinness Teenager - Disability and long-term health problems - Serious illness since last round - Subjective wellbeing
	Time use (wave 2-5)	Average time spent daily on: - Sleeping - Caring for others - Household chores - Household tasks - Household work - School - Studying - Having fun - Did the child work while going to school? - Did you miss school for at least one month?
	Cognitive Skills	- Item response theory scores for a) mathematics, b) reading and c) Peabody picture vocabulary tests. - Improvement on math test and PPVT between ages 8 and 13 on these test
	Non-cognitive Skills	- Self-esteem, PRIDE scale - Self-efficacy - Trust in others - Sociability, friends and Extrovertness - Leadership - Helping others at school
	Motivations and Expectations	- Educational grade you would like to complete? - *Sector of job you want to do later* - Is that dream job vocational? - Does that dream job require academic study?
Household	General Household Demographics	- *Caregiver's,* father and mothers age\ - Illness of father or mother - Caregiver's relation to Young Lives Child

	Household size	- Household size - Number of children born before and after the Young Lives Child - *Number of boys/girls between 0-12* - *Number of children in the household*
	Parents education	- *Caregiver*/father/mother cannot read - *Caregiver*/father/mother attended formal education - *Caregiver*/father/mother attended education beyond primary - *Caregiver*/father/mother attended post-secondary education - *Caregiver*/father/mother attended vocational secondary/tertiary
	Perception of education	- Perceived quality of primary school - Perceived usefulness of formal education - Should child stay in school during financial hardship
	Household Economy	- *Sector of primary occupation* - Household owns the house? - Housing quality index - Access to services index - Access to consumer durables index - Is household in debt?
	Occurrence of (Economic) Shocks	- Somebody in household lost their job - Felt victim to crime - Victim of natural disasters - Damage to house
	Aspirations for child	- *In what sector do you want your child to work later?* - Is that job vocational in nature? - Does that job require academic study? - At what age should child be married? - At what age should child earn their own income? - At what age should child leave school? - Do you expect the child to meet your expectations?
Community Characteristics	Main characteristics	- Population in locality - Type of area (rural vs urban) - Distance to district capital in minutes by public transport
	Type of jobs available	- Local land used for agriculture? - Local land used for industry? - Local land used for handicraft/small scale manufacturing?
	Availability of educational institutes	- Public/*Private* Secondary schools available or nearby? - Lower-vocational schools nearby? - Post-secondary technological institutes available or nearby?

Appendix B: Item-Response Missingness

Table 11: Item Response Missingness per Country and Cohort

		Ethiopia		India		Peru		Vietnam	
Variable	Total	OC	YC	OC	YC	OC	YC	OC	YC
Average	9,6%	10,5%	11,4%	5,1%	6,7%	10,9%	11,9%	9,7%	11,0%
Average excluding missing variables per country/cohort	5,5%	6,5%	5,0%	4,3%	3,7%	8,4%	7,0%	4,2%	4,7%
chsex	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
zweight_8	1.6%	8.2%	0.3%	0.0%	0.5%	0.5%	1.3%	0.0%	2.0%
zheight_8	1.1%	3.7%	0.3%	0.1%	0.7%	0.7%	1.3%	0.0%	2.4%
zbmi_8	2.7%	13.8%	1.1%	0.3%	1.4%	0.9%	1.3%	0.0%	2.5%
chillness_8_13	0.7%	0.1%	0.1%	0.0%	0.2%	1.1%	1.9%	0.4%	1.5%
long_term_health_problem	0.8%	0.1%	2.0%	0.0%	1.2%	1.1%	1.8%	0.6%	0.0%
chdisability	2.9%	4.8%	0.1%	0.4%	0.2%	5.7%	1.9%	8.7%	1.4%
subjective_health_13	1.3%	0.1%	0.6%	1.2%	1.5%	1.2%	2.9%	0.6%	2.0%
hsleep	0.8%	0.0%	0.3%	0.0%	0.7%	1.2%	2.4%	0.4%	1.7%
hcare	0.9%	0.1%	0.3%	0.6%	0.7%	1.2%	2.4%	0.6%	1.7%
hchore	0.9%	0.1%	0.3%	0.2%	0.7%	1.2%	2.4%	0.8%	1.7%
htask	1.0%	0.3%	0.3%	0.6%	0.7%	1.2%	2.4%	0.8%	1.7%
hwork	1.0%	0.4%	0.3%	0.3%	0.7%	1.2%	2.4%	0.6%	1.7%
hschool	0.9%	0.0%	0.3%	0.2%	0.7%	1.2%	2.4%	0.7%	1.7%
hstudy	0.9%	0.1%	0.3%	0.1%	0.7%	1.2%	2.4%	0.7%	1.7%
hplay	0.9%	0.0%	0.3%	0.0%	0.7%	1.2%	2.4%	0.6%	1.7%
chldwork_during_school	0.8%	0.0%	0.1%	0.0%	1.2%	1.1%	2.1%	0.7%	1.5%
missed_school	8.3%	5.5%	33.8%	10.6%	4.2%	2.3%	2.9%	3.4%	3.5%
math_score_13	3.4%	6.2%	0.3%	4.9%	5.9%	1.6%	2.6%	1.3%	4.7%
math_score_improvement	5.1%	8.6%	0.4%	7.3%	6.7%	3.4%	3.6%	4.3%	6.2%
read_score_13	53.3%	**100%**	23.7%	**100%**	0.2%	**100%**	1.4%	**100%**	1.4%
ppvt_score_13	8.4%	15.9%	20.4%	5.4%	0.8%	10.7%	9.0%	2.3%	2.5%
ppvt_score_improvement	15.2%	18.0%	31.0%	11.1%	11.4%	19.2%	19.2%	5.1%	7.0%
noncog_friend	1.1%	0.0%	0.4%	0.0%	1.6%	1.2%	2.2%	0.9%	2.3%
noncog_hardtalk	3.9%	5.6%	0.6%	10.9%	1.7%	2.7%	2.5%	5.7%	1.8%
noncog_incgame	3.5%	5.5%	0.4%	10.6%	1.6%	2.7%	2.2%	3.5%	1.8%
noncog_lead	8.5%	1.1%	1.4%	0.3%	1.7%	52.9%	3.7%	2.3%	4.1%
noncog_helpchld	3.7%	5.8%	0.6%	11.1%	1.6%	2.7%	2.2%	3.8%	1.7%
noncog_trust	1.1%	0.3%	0.7%	0.1%	1.7%	1.4%	2.0%	0.8%	1.7%
noncog_selfefficiacy	1.2%	0.0%	0.6%	0.1%	1.8%	1.4%	2.1%	1.2%	2.1%
noncog_selfesteem	1.0%	0.0%	0.6%	0.0%	1.7%	1.4%	2.1%	0.7%	1.7%
expected_grade	4.6%	5.5%	0.7%	10.6%	1.5%	3.4%	3.4%	3.8%	7.8%
dreamjob_sector	2.2%	0.6%	0.7%	0.6%	2.0%	3.4%	3.9%	2.5%	4.3%
vocational_dreamjob_dummy	7.3%	2.4%	1.8%	3.3%	4.6%	12.8%	6.5%	19.6%	7.4%
academic_dreamjob_dummy	7.3%	2.4%	1.8%	3.3%	4.6%	12.8%	6.5%	19.6%	7.4%
dadage_atbirth	12.4%	24.9%	15.8%	8.1%	2.4%	23.3%	16.9%	4.1%	3.6%

momage_atbirth	2.8%	9.3%	2.2%	2.9%	0.9%	4.4%	0.4%	2.0%	0.3%
careage_atbirth	0.1%	0.1%	0.1%	0.0%	0.0%	0.2%	0.1%	0.0%	0.0%
dadpassed	12.5%	23.8%	15.8%	4.5%	5.3%	22.9%	17.9%	3.9%	6.0%
mompassed	3.3%	8.2%	3.3%	1.4%	1.7%	4.3%	3.1%	1.9%	2.7%
primarycaregiver	1.7%	0.1%	0.1%	0.1%	0.2%	5.0%	6.0%	0.6%	1.5%
parent_sick	0.4%	0.0%	0.1%	0.0%	0.2%	0.0%	1.4%	0.4%	1.3%
hhsize	0.6%	0.0%	0.1%	0.0%	0.2%	0.9%	1.7%	0.4%	1.3%
male012	0.6%	0.0%	0.1%	0.0%	0.2%	0.9%	1.7%	0.4%	1.3%
female012	0.6%	0.0%	0.1%	0.0%	0.2%	0.9%	1.7%	0.4%	1.3%
bornbef	0.4%	0.3%	0.0%	0.0%	0.0%	1.1%	1.4%	0.4%	0.3%
bornaft	1.2%	0.4%	1.4%	0.0%	0.7%	1.1%	3.8%	0.4%	1.5%
total_children_household	1.3%	0.4%	1.4%	0.0%	0.7%	1.1%	4.8%	0.4%	1.7%
household_primary_job	25.5%	0.7%	0.4%	0.6%	0.1%	**100%**	**100%**	1.1%	1.1%
ownhouse	13.0%	0.0%	0.1%	0.0%	0.2%	0.9%	1.8%	**100%**	1.3%
hq	50.4%	**100%**	**100%**	0.0%	0.2%	0.9%	1.7%	**100%**	**100%**
sv	50.4%	**100%**	**100%**	0.0%	0.2%	0.9%	1.7%	**100%**	**100%**
cd	50.4%	**100%**	**100%**	0.0%	0.2%	1.1%	1.7%	**100%**	**100%**
debt	0.6%	0.1%	0.3%	0.0%	0.3%	0.9%	1.3%	0.6%	1.4%
dadcantread	54.3%	13.5%	**100%**	8.9%	**100%**	6.2%	**100%**	6.2%	**100%**
momcantread	51.9%	5.9%	**100%**	3.7%	**100%**	1.8%	**100%**	3.8%	**100%**
carecantread	50.6%	0.1%	**100%**	0.4%	**100%**	0.9%	**100%**	3.0%	**100%**
mom_edu_attended_formaleducation	4.5%	9.0%	5.2%	2.8%	3.4%	5.3%	4.6%	2.4%	3.5%
mom_edu_beyond_primaryeducation	4.5%	9.0%	5.2%	2.8%	3.4%	5.3%	4.6%	2.4%	3.5%
mom_edu_attended_postsecondary	4.5%	9.0%	5.2%	2.8%	3.4%	5.3%	4.6%	2.4%	3.5%
mom_edu_attended_vocational	4.5%	9.0%	5.2%	2.8%	3.4%	5.3%	4.6%	2.4%	3.5%
dad_edu_attended_formaleducation	14.8%	24.6%	18.3%	8.7%	7.4%	26.3%	21.4%	5.1%	6.5%
dad_edu_beyond_primaryeducation	14.8%	24.6%	18.3%	8.7%	7.4%	26.3%	21.4%	5.1%	6.5%
dad_edu_attended_postsecondary	14.8%	24.6%	18.3%	8.7%	7.4%	26.3%	21.4%	5.1%	6.5%
dad_edu_attended_vocational	14.8%	24.6%	18.3%	8.7%	7.4%	26.3%	21.4%	5.1%	6.5%
care_edu_attended_formaleducation	0.7%	0.1%	0.6%	0.1%	0.5%	0.9%	1.7%	0.6%	1.5%
care_edu_beyond_primaryeducation	0.7%	0.1%	0.6%	0.1%	0.5%	0.9%	1.7%	0.6%	1.5%
care_edu_attended_postsecondary	0.7%	0.1%	0.6%	0.1%	0.5%	0.9%	1.7%	0.6%	1.5%
care_edu_attended_vocational	0.7%	0.1%	0.6%	0.1%	0.5%	0.9%	1.7%	0.6%	1.5%
expected_age_married	3.5%	8.6%	0.1%	4.8%	0.9%	1.6%	4.1%	4.2%	3.6%
expected_age_earning	3.0%	4.4%	0.4%	7.4%	2.3%	1.2%	2.8%	2.4%	3.4%
expected_age_leaving_school	5.5%	4.4%	1.0%	16.7%	2.0%	1.8%	3.1%	10.4%	5.0%
realistic_expectations_parents	6.7%	4.8%	4.3%	6.3%	4.0%	2.1%	6.1%	7.2%	18.5%
parents_dreamjob_sector	2.9%	0.8%	0.3%	1.3%	1.4%	1.8%	2.0%	0.6%	15.4%

parents_vocational_dreamjob_dummy	9.2%	1.5%	1.3%	6.5%	7.9%	10.8%	12.6%	12.8%	19.9%
parents_academic_dreamjob_dummy	9.2%	1.5%	1.3%	6.5%	7.9%	10.8%	12.6%	12.8%	19.9%
formal_education_useful	39.4%	65.8%	67.6%	63.9%	67.3%	12.8%	13.2%	9.7%	15.2%
education_during_financial_hardship	1.1%	0.7%	0.1%	0.4%	0.9%	1.2%	1.8%	2.1%	1.7%
quality_primary_school	1.7%	1.8%	1.1%	2.8%	2.0%	1.4%	1.8%	0.8%	1.6%
typesite_w1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
popsize	10.0%	8.6%	5.4%	0.2%	0.7%	26.6%	21.8%	9.5%	7.1%
timecap	27.9%	35.9%	35.7%	18.1%	13.3%	58.1%	45.6%	9.5%	7.1%
agriculture_jobs	11.5%	8.6%	5.4%	6.3%	6.5%	26.6%	21.8%	9.5%	7.1%
factory_jobs	18.2%	8.6%	5.4%	16.4%	21.6%	26.6%	21.8%	23.7%	21.7%
craft_jobs	18.9%	8.6%	5.4%	18.9%	24.0%	26.6%	21.8%	23.7%	21.7%
public_secondary_available	10.1%	8.6%	5.4%	0.2%	0.7%	26.8%	22.1%	9.5%	7.1%
private_secondary_available	11.0%	8.6%	5.4%	0.2%	0.7%	26.6%	21.8%	14.7%	10.3%
lower_vocational_available	12.8%	8.6%	6.7%	5.8%	4.6%	26.6%	21.8%	16.1%	12.2%
public_higher_vocational_available	11.2%	8.6%	5.4%	0.8%	1.2%	26.6%	21.8%	14.7%	10.3%
private_higher_vocational_available	31.0%	8.6%	5.4%	3.1%	2.5%	**100%**	**100%**	16.1%	12.2%

Appendix C: Descriptive Statistics on all Covariates

Table 12: Descriptive Statistics on all Covariates

	Vocational Secondary		General Secondary		Dropped Out Post Primary	
	OC	YC	OC	YC	OC	YC
Outcomes:						
Hours per week worked	32.92	25.13	34.83	29.2	39.53	28.77
Hourly income (us $)	0.41	0.33	0.63	0.44	0.24	0.16
Any IGA (binary)	0.60	0.44	0.68	0.51	0.69	0.44
Formal IGA (binary)	0.43	0.26	0.53	0.34	0.39	0.26
Self-employed (binary)	0.09	0.21	0.10	0.15	0.15	0.22
Not employed in farming (binary)	0.48	0.35	0.60	0.44	0.46	0.35
Attended higher education	0.54	0.23	0.82	0.43	0.00	0.00
Survey Characteristics						
Age at measurement outcome variables	22.08	18.8	23.48	19.05	23.97	19.54
Work experience in years when measuring outcomes	6.71	3.48	5.17	1.42	2.83	0.92
Individual Characteristics						
Gender: male	0.54	0.55	0.49	0.49	0.53	0.53
Child health: relative weight at 8 years	-1.7	-1.57	-1.39	-0.96	-1.99	-1.62
Relative height at 8 years	-1.5	-1.28	-1.36	-1.24	-1.64	-1.46
Relative BMI at 8 years	-1.09	-1.11	-0.78	-0.28	-1.24	-1
Serious illness between 8 and 13: Yes (binary)	0.12	0.26	0.11	0.17	0.15	0.2
not known	0.01	0	0	0.01	0	0.01
Long term health problem: Yes (binary)	0.06	0.08	0.09	0.1	0.09	0.1
not known	0.01	0.02	0	0.01	0	0
Disablity: Yes (binary)	0.03	0	0.02	0.01	0.02	0.02
not known	0.01	0	0.02	0.01	0.01	0.01
Self-reported health (scale 1-9)	4.45	5.06	4.99	5.84	4.08	5.31
Time use in hours (age 13): sleep	8.95	9.24	8.89	9.27	9.03	9.26
Taking care of family	0.52	0.24	0.51	0.59	0.45	0.53
Doing chores at home	1.51	1.2	1.44	1.17	1.51	1.39
Performing tasks	0.67	0.35	0.48	0.42	0.91	0.8
Working	0.06	0	0.05	0.03	0.11	0.03
At school	5.75	7.29	5.49	6.34	5.3	6.19
Studying	2.22	1.96	2.82	2.05	1.97	1.77
Playing	3.59	3.6	4	3.96	4.16	3.97
Cognitive scores: Math test at 13	508.58	473.13	541.76	511.45	471.82	455.25
Math test improvement 8 to 13	-18.42	126.52	4.15	130.4	-18.31	121.55
Reading score at 13	NA	0.1	NA	-0.03	NA	-0.42
Language score at 13	2.65	2.55	2.91	2.73	2.37	2.36
Language test improvement 8 to 13	0.46	1.21	0.28	1.34	0.28	1.23
Non-cognitive skills: Number of friends	7.89	6.42	10.23	7.09	8.26	5.81

Hard to talk to others: always	0.11	0.09	0.09	0.11	0.08	0.09
never	0.67	0.68	0.75	0.6	0.7	0.65
sometimes	0.2	0.22	0.15	0.28	0.18	0.25
not known	0.02	0.01	0.01	0.01	0.05	0.01
Including friends in games: always	0.69	0.67	0.82	0.62	0.77	0.69
never	0.02	0.05	0.02	0.05	0.02	0.02
sometimes	0.28	0.27	0.16	0.32	0.18	0.27
not known	0.01	0.01	0.01	0.01	0.04	0.01
Taking the lead: always	0.15	0.25	0.18	0.27	0.15	0.19
never	0.39	0.4	0.36	0.45	0.46	0.46
sometimes	0.39	0.34	0.31	0.26	0.35	0.32
not known	0.07	0.02	0.15	0.02	0.03	0.03
Helping other children at school: always	0.38	0.38	0.48	0.44	0.41	0.43
never	0.05	0.14	0.06	0.07	0.06	0.07
sometimes	0.55	0.47	0.45	0.48	0.49	0.49
not known	0.02	0.01	0.01	0.01	0.04	0.01
Trust in others (1-100 scale)	77.84	67.23	71.99	64.46	78.54	68.82
Self-efficiacy (1-100 scale)	75.86	64.04	79.45	64.43	76.11	62.87
Self-esteem (1-100 scale)	79.14	64.63	85.19	68.36	76.82	63.52
Child working during school: yes	0.1	0.3	0.11	0.62	0.11	0.44
not known	0.01	0.01	0	0.01	0	0.01
Missed more than one week of school: yes	0.13	0.13	0.08	0.08	0.13	0.08
not known	0.01	0.02	0.01	0.01	0.04	0.11
Expectations: self-expected grade: (below) primary	0.01	0.02	0.01	0.04	0.01	0.03
lower-secondary	0.05	0.04	0.04	0.02	0.14	0.09
technical/vocational college	0.06	0.02	0.07	0.08	0.06	0.03
university/college	0.72	0.8	0.83	0.76	0.56	0.62
upper-secondary	0.14	0.1	0.05	0.07	0.19	0.19
not known	0.02	0.02	0.01	0.03	0.04	0.03
Sector of dreamjob: Education and Research	0.47	0.28	0.58	0.24	0.42	0.3
Healthcare	0.23	0.36	0.1	0.27	0.16	0.24
other	0.07	0.02	0.11	0.05	0.11	0.05
Public Administration and Services	0.09	0.11	0.04	0.15	0.08	0.14
Services and Management	0.04	0.04	0.04	0.04	0.02	0.03
Skilled Trades and Manual Labor	0.09	0.17	0.1	0.23	0.2	0.21
not known	0.02	0.02	0.01	0.03	0.01	0.02
Dreamjob is vocational: Yes	0.11	0.15	0.12	0.23	0.2	0.21
not known	0.06	0.03	0.1	0.05	0.09	0.05
Dreamjob requires academic study_Yes	0.76	0.65	0.71	0.54	0.62	0.58
not known	0.06	0.03	0.1	0.05	0.09	0.05
Household Characteristics						
Dad age at birth	17.07	15.21	15.23	15.34	16.76	16.66
Mom age at birth	15.47	14.57	15.89	16.22	16.18	16.5
Careage at birth	18.13	16.81	18.43	18.43	18.95	18.8
Dad passed: Yes	0.02	0.01	0.02	0.01	0.02	0.02
not known	0.15	0.13	0.1	0.12	0.11	0.1
Mom passed: Yes	0.01	0	0.01	0	0.01	0.01

not known	0.06	0.01	0.02	0.02	0.03	0.03
Primarycaregiver: nonrelatives	0	0	0	0	0	0
parent	0.94	0.88	0.96	0.93	0.95	0.92
relatives	0.04	0.11	0.02	0.03	0.05	0.06
sibling	0.01	0	0	0	0.01	0.01
not known	0.01	0.01	0.01	0.03	0	0.02
Household size at age 13	5.61	5.02	5.16	4.94	5.67	5.27
Number of boys aged 0-12 in household	1.62	1.49	1.48	1.47	1.61	1.53
Number of girls aged 0-12 in household	1.59	1.38	1.43	1.48	1.6	1.57
Parents sick: Yes	0.11	0.07	0.08	0.07	0.12	0.08
not known	0	0	0	0.01	0	0.01
Children born before	2.83	2.13	2.34	2.32	2.89	2.75
Children born after	2.16	1.83	1.85	1.97	2.21	2.19
Total children household	4.01	2.97	3.19	3.3	4.1	3.98
Household primary job: agriculture	0.34	0.25	0.27	0.18	0.48	0.41
casual labor	0.13	0.21	0.08	0.09	0.16	0.15
child care	0	0	0	0	0	0
construction and repairs	0.03	0.03	0.04	0.02	0.03	0.04
crafts and manufacturing	0.05	0.08	0.05	0.03	0.03	0.04
food/local drink preparation	0.04	0.02	0.02	0.01	0.04	0.02
other	0.09	0.07	0.1	0.05	0.1	0.08
public sector	0.07	0.04	0.01	0.01	0.03	0.02
services	0.14	0.16	0.16	0.08	0.1	0.11
not known	0.11	0.13	0.26	0.54	0.04	0.13
Dad can't read: yes	0.24	NA	0.12	NA	0.3	NA
not known	0.09	NA	0.07	NA	0.09	NA
Mom can't read: yes	0.43	NA	0.24	NA	0.52	NA
not known	0.06	NA	0.03	NA	0.04	NA
Caretaker can't read: yes	0.47	NA	0.26	NA	0.56	NA
not known	0.01	NA	0.01	NA	0.01	NA
Mom attended formal education: Yes	0.53	0.65	0.74	0.8	0.47	0.56
not known	0.06	0.01	0.03	0.03	0.04	0.05
Mom attended education beyond primary: Yes	0.22	0.25	0.35	0.36	0.11	0.14
not known	0.06	0.01	0.03	0.03	0.04	0.05
Mom attended post-secondary education: Yes	0.03	0.02	0.06	0.06	0.01	0.01
not known	0.06	0.01	0.03	0.03	0.04	0.05
Mom attended vocational education: Yes	0.02	0.02	0.04	0.04	0	0.01
not known	0.06	0.01	0.03	0.03	0.04	0.05
Dad attended formal education: Yes	0.58	0.67	0.77	0.79	0.58	0.65
not known	0.16	0.15	0.12	0.13	0.13	0.13
Dad attended education beyond primary: Yes	0.28	0.35	0.43	0.38	0.18	0.19
not known	0.16	0.15	0.12	0.13	0.13	0.13
Dad attended post-secondary education: Yes	0.06	0.07	0.08	0.08	0.01	0.02
not known	0.16	0.15	0.12	0.13	0.13	0.13
Dad attended vocational education: Yes	0.04	0.04	0.05	0.05	0	0.01
not known	0.16	0.15	0.12	0.13	0.13	0.13
Care attended formal education: Yes	0.55	0.63	0.76	0.82	0.48	0.6
not known	0.01	0	0	0.01	0	0.01

Care attended education beyond primary: Yes	0.23	0.24	0.36	0.36	0.1	0.15
not known	0.01	0	0	0.01	0	0.01
Care attended post-secondary education: Yes	0.03	0.04	0.06	0.06	0	0.01
not known	0.01	0	0	0.01	0	0.01
Care attended vocational education: Yes	0.02	0.03	0.04	0.05	0	0.01
not known	0.01	0	0	0.01	0	0.01
Formal education useful: no. it is not useful	0.09	0.11	0.23	0.46	0.03	0.11
yes but it is not essential	0.02	0.03	0.04	0.03	0.04	0.05
yes. it is essential	0.42	0.44	0.48	0.31	0.42	0.37
not known	0.46	0.41	0.24	0.2	0.51	0.46
During financial hardship: let child stay in school	0.95	0.94	0.96	0.94	0.93	0.91
child leave school	0.04	0.05	0.03	0.04	0.06	0.08
not known	0.01	0.01	0.01	0.02	0.01	0.01
The primary school is of high quality: agree	0.36	0.63	0.38	0.56	0.29	0.57
disagree	0.04	0.08	0.08	0.1	0.04	0.05
more or less	0.01	0.15	0.04	0.21	0	0.15
strongly agree	0.55	0.11	0.48	0.1	0.63	0.2
strongly disagree	0.02	0	0.01	0.02	0.02	0.02
Not known	0.02	0.02	0.01	0.01	0.01	0.01
Expected age child marries: 18-21	0.08	0.09	0.09	0.09	0.19	0.16
22-26	0.46	0.4	0.45	0.41	0.45	0.45
27-30	0.31	0.4	0.33	0.41	0.23	0.31
31+	0.05	0.08	0.04	0.06	0.04	0.05
Before 18	0	0.02	0	0	0.01	0.01
no expectation	0.04	0	0.04	0	0.03	0
not known	0.06	0.02	0.04	0.03	0.05	0.02
Expected age child earns thier own income: 18-19	0.1	0.08	0.08	0.12	0.13	0.14
20-22	0.32	0.21	0.3	0.29	0.35	0.31
23-26	0.37	0.52	0.41	0.43	0.27	0.36
27+	0.05	0.14	0.05	0.07	0.05	0.1
Before 18	0.05	0.02	0.05	0.06	0.1	0.07
no expectation	0.06	0	0.07	0	0.05	0
not known	0.05	0.02	0.04	0.02	0.04	0.02
Expected age child leaves school 18-19	0.11	0.1	0.06	0.09	0.18	0.18
20-21	0.25	0.27	0.17	0.17	0.19	0.22
22-23	0.19	0.21	0.23	0.33	0.13	0.22
24-25	0.14	0.25	0.2	0.27	0.11	0.19
26+	0.01	0.07	0.04	0.05	0.02	0.05
Before 18	0.09	0.06	0.09	0.07	0.16	0.11
no expectation	0.1	0	0.14	0	0.11	0
not known	0.11	0.03	0.08	0.02	0.1	0.02
Child is likely to achieve their educational goal: Yes	0.91	0.9	0.9	0.87	0.84	0.83
not known	0.05	0.06	0.04	0.08	0.06	0.09
Sector dreamjob according to parents: Education and Research	0.3	0.28	0.33	0.23	0.26	0.24
Healthcare	0.35	0.26	0.24	0.21	0.22	0.2
other	0.04	0.07	0.1	0.1	0.15	0.07
Public Administration and Services	0.11	0.13	0.09	0.1	0.16	0.21
Services and Management	0.04	0.04	0.08	0.11	0.04	0.06

Skilled Trades and Manual Labor	0.16	0.18	0.15	0.21	0.17	0.16
not known	0.01	0.03	0.01	0.04	0.01	0.06
Dreamjob by parents is vocational: Yes	0.16	0.2	0.16	0.21	0.17	0.17
not known	0.04	0.09	0.09	0.11	0.09	0.11
Dreamjob by parents requires academic study: Yes	0.74	0.63	0.66	0.5	0.63	0.62
not known	0.04	0.09	0.09	0.11	0.09	0.11
Family owns their house: yes	0.66	0.79	0.51	0.8	0.52	0.83
not known	0.14	0	0.35	0.01	0.37	0.01
Food security: we always eat enough, what we want	0.03	0.32	0.1	0.39	0.01	0.23
we eat enough, not always what we like	0.05	0.49	0.13	0.49	0.01	0.56
we frequently do not eat enough	0	0.02	0.01	0.02	0	0.03
we sometimes do not eat enough	0.01	0.17	0.02	0.09	0.01	0.17
not known	0.9	0	0.74	0.01	0.97	0.01
Wealth index (0-1)	0.45	0.54	0.55	0.6	0.39	0.5
Housing quality (0-1)	0.52	0.61	0.53	0.51	0.49	0.54
Access to services (0-1)	0.62	0.66	0.73	0.79	0.57	0.66
Consumer durables owned (0-1)	0.26	0.36	0.36	0.44	0.19	0.34
Household in debt: Yes	0.48	0.33	0.47	0.37	0.56	0.48
not known	0.01	0.01	0	0.01	0	0.01
Community Characteristics						
Population	137.78	124.22	139.99	119.41	145.58	141.05
Time to provincial capital (hours)	13.91	14.69	11.23	10.65	12.63	12.55
Urban locality	0.38	0.37	0.42	0.44	0.21	0.2
Agriculture: most important	0.64	0.67	0.61	0.67	0.7	0.72
not important	0.14	0.18	0.19	0.11	0.09	0.07
somewhat important	0.09	0.04	0.05	0.06	0.12	0.14
not known	0.13	0.1	0.14	0.16	0.08	0.06
Factory jobs: not important	0.67	0.58	0.67	0.73	0.72	0.7
somewhat important	0.15	0.22	0.12	0.06	0.1	0.13
not known	0.18	0.2	0.21	0.21	0.18	0.17
Craft jobs: not important	0.55	0.66	0.51	0.52	0.54	0.56
somewhat important	0.26	0.13	0.27	0.26	0.27	0.27
Not known	0.19	0.21	0.22	0.22	0.19	0.17
Public secondary available: no. and not in a nearby locality	0	0	0	0	0.04	0.04
no. but there is one in a nearby locality	0.4	0.47	0.23	0.22	0.35	0.34
yes	0.5	0.44	0.65	0.63	0.54	0.57
not known	0.1	0.1	0.12	0.14	0.07	0.05
Private secondary available: no. and not in a nearby locality	0.33	0.16	0.3	0.35	0.47	0.47
no. but there is one in a nearby locality	0.4	0.59	0.41	0.36	0.39	0.41
yes	0.15	0.15	0.16	0.14	0.06	0.05
not known	0.11	0.1	0.13	0.15	0.09	0.07
Lower vocational available: no. and not in a nearby locality	0.37	0.22	0.31	0.32	0.49	0.45
no. but there is one in a nearby locality	0.4	0.56	0.42	0.41	0.36	0.42
yes	0.09	0.08	0.11	0.1	0.04	0.04
not known	0.13	0.13	0.16	0.17	0.11	0.08

Public higher vocational available: no. and not in a nearby locality	0.16	0.1	0.24	0.24	0.34	0.3
no. but there is one in a nearby locality	0.58	0.67	0.5	0.48	0.53	0.57
yes	0.14	0.13	0.12	0.13	0.05	0.06
not known	0.11	0.1	0.14	0.15	0.09	0.07
Private higher vocational available: no. and not in a nearby locality	0.29	0.21	0.28	0.21	0.41	0.35
no. but there is one in a nearby locality	0.46	0.49	0.32	0.19	0.43	0.4
yes	0.04	0.07	0.07	0.03	0.05	0.08
not known	0.21	0.23	0.33	0.57	0.11	0.17

Notes: a) all descriptives are on non-imputed data, b) Vocational secondary is defined as once having been enrolled in non-tertiary TVET, even if earlier enrolled in general secondary, dropped-out is defined as having been enrolled in the final grade of primary, but never have been enrolled in the final grade of upper-secondary. c) For categorical variables values are proportions, for numerical variables values are the non-standardized. d) not known encompasses missing data, refusals to answer and “I do not know answers”

Appendix D: Balance in Sub Samples

D.1. Vocational Secondary vs. General Secondary

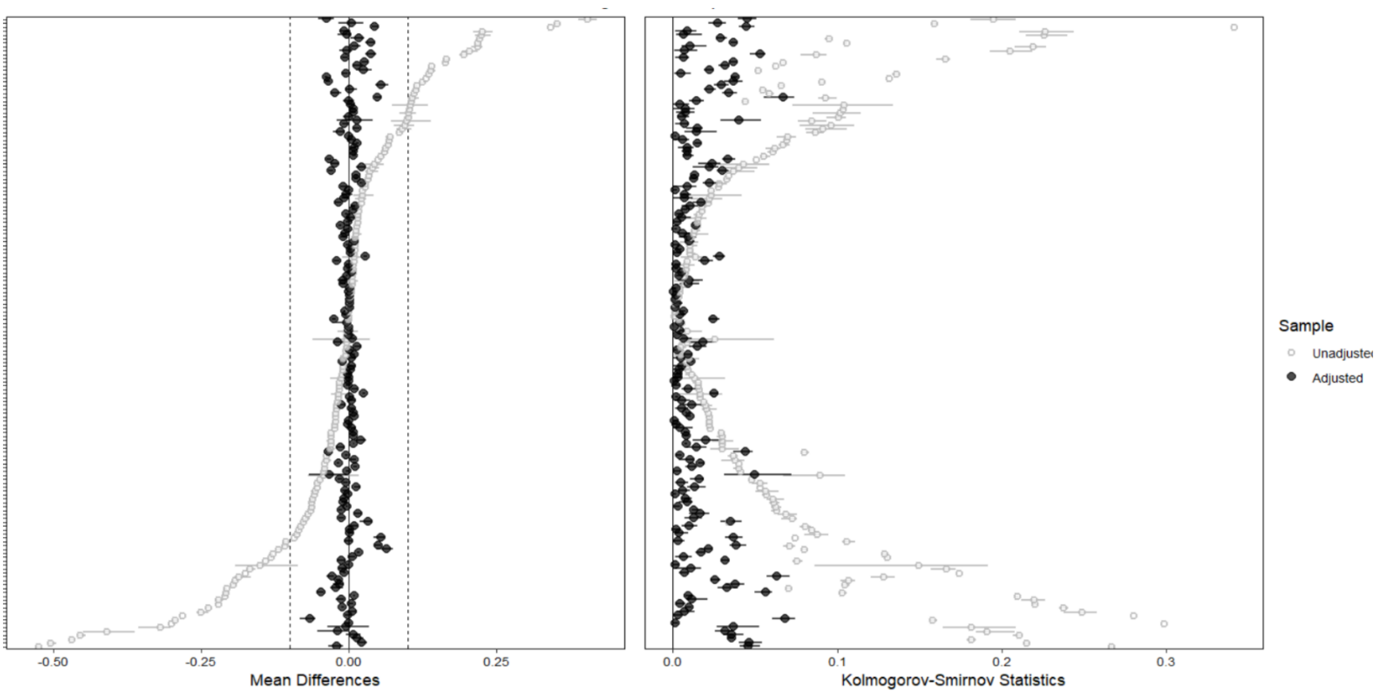


Figure 7: Vocational vs. General, Covariates Balance after Propensity Weighting

D.2. Vocational Secondary vs. Dropouts

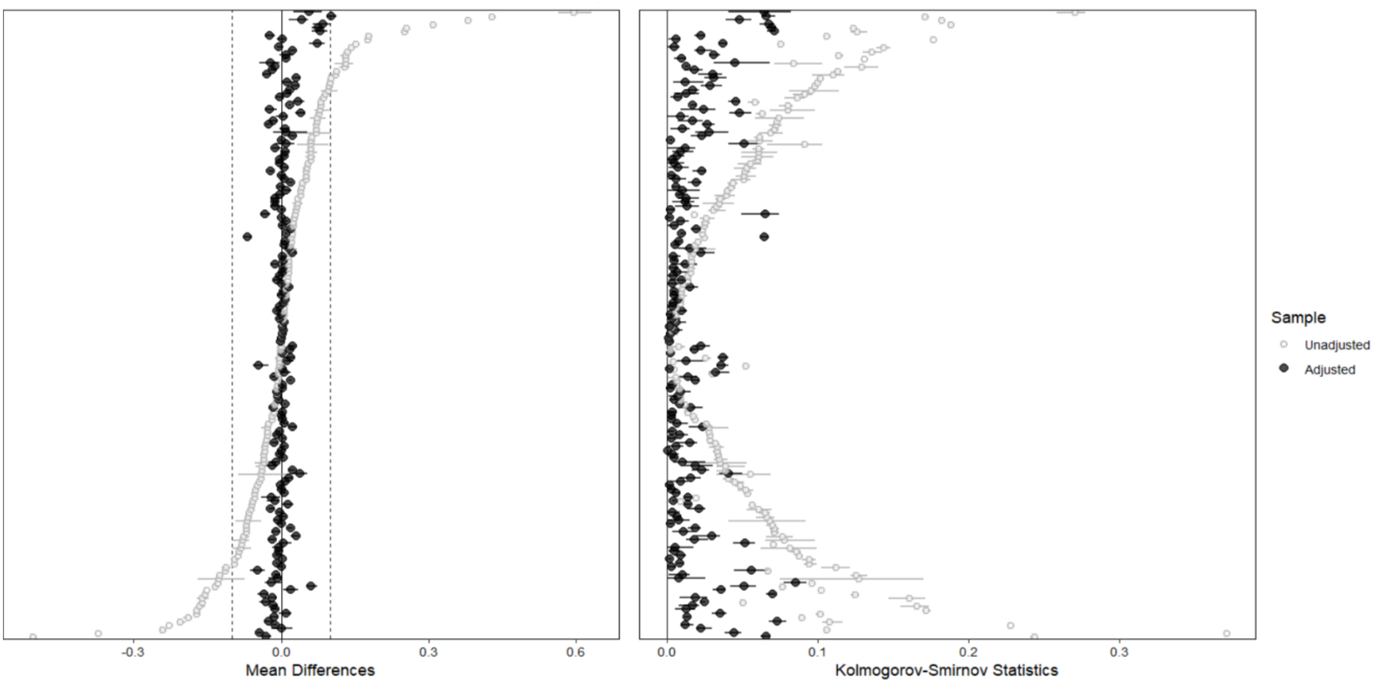
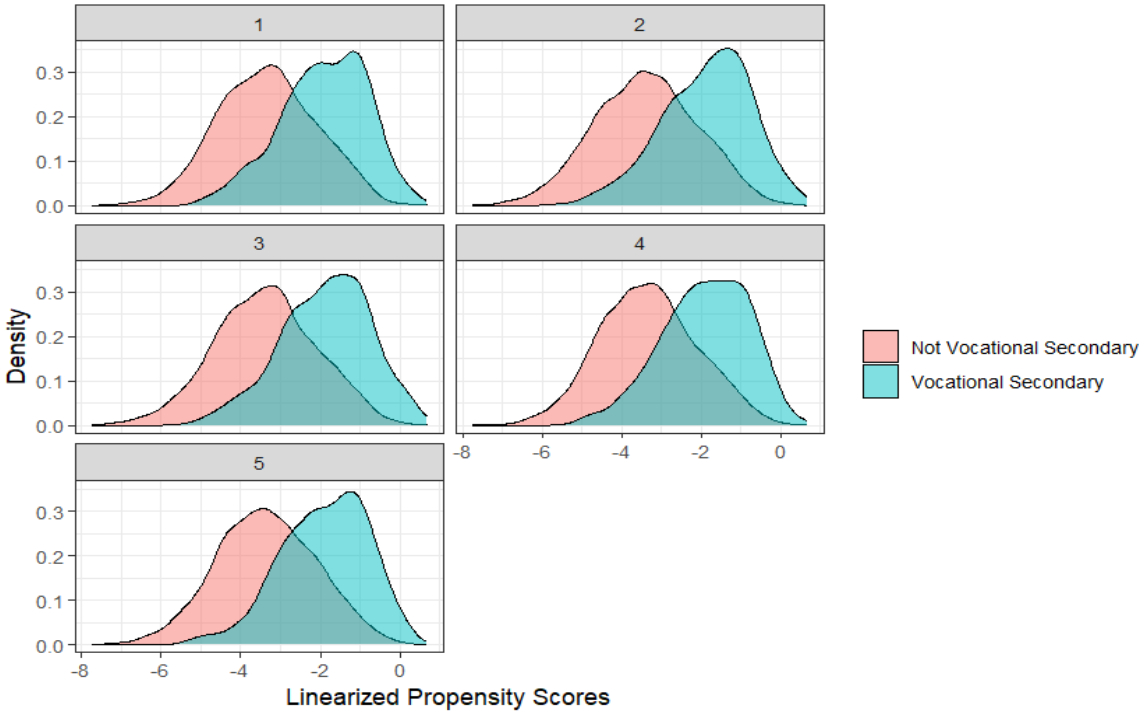


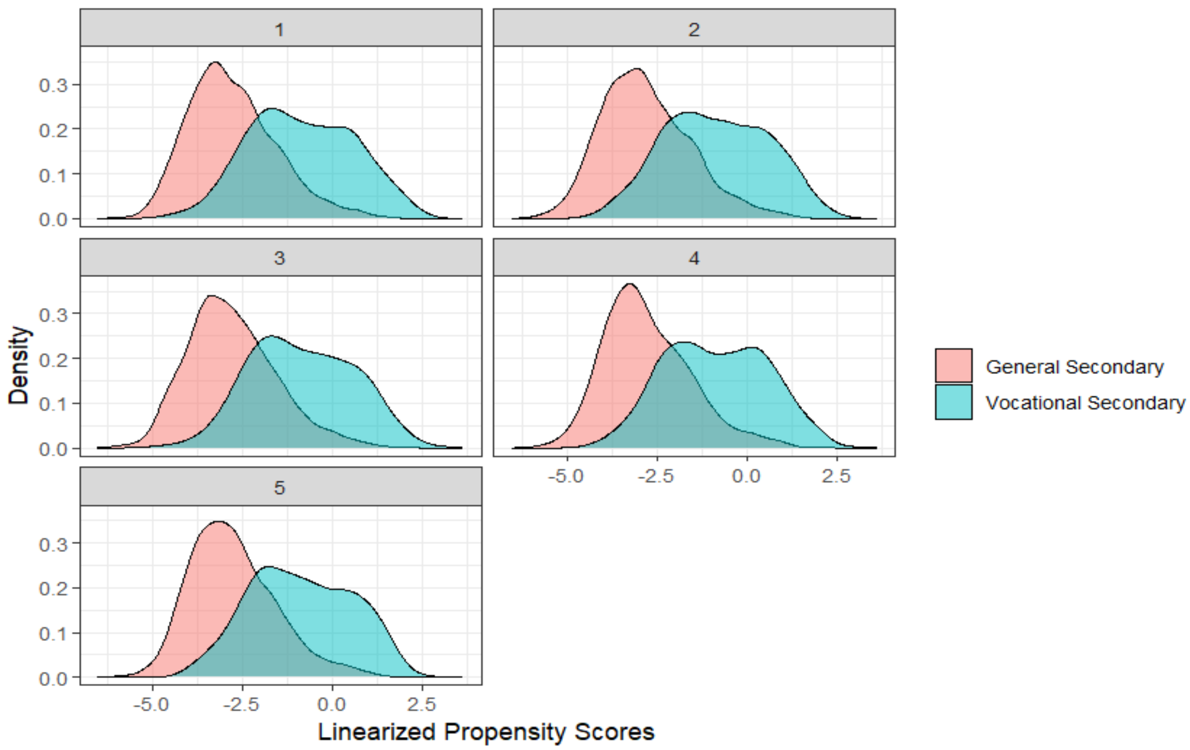
Figure 8: Vocational vs. Dropouts, Covariate Balance after Propensity Weighting

Appendix E: Area of Common Support

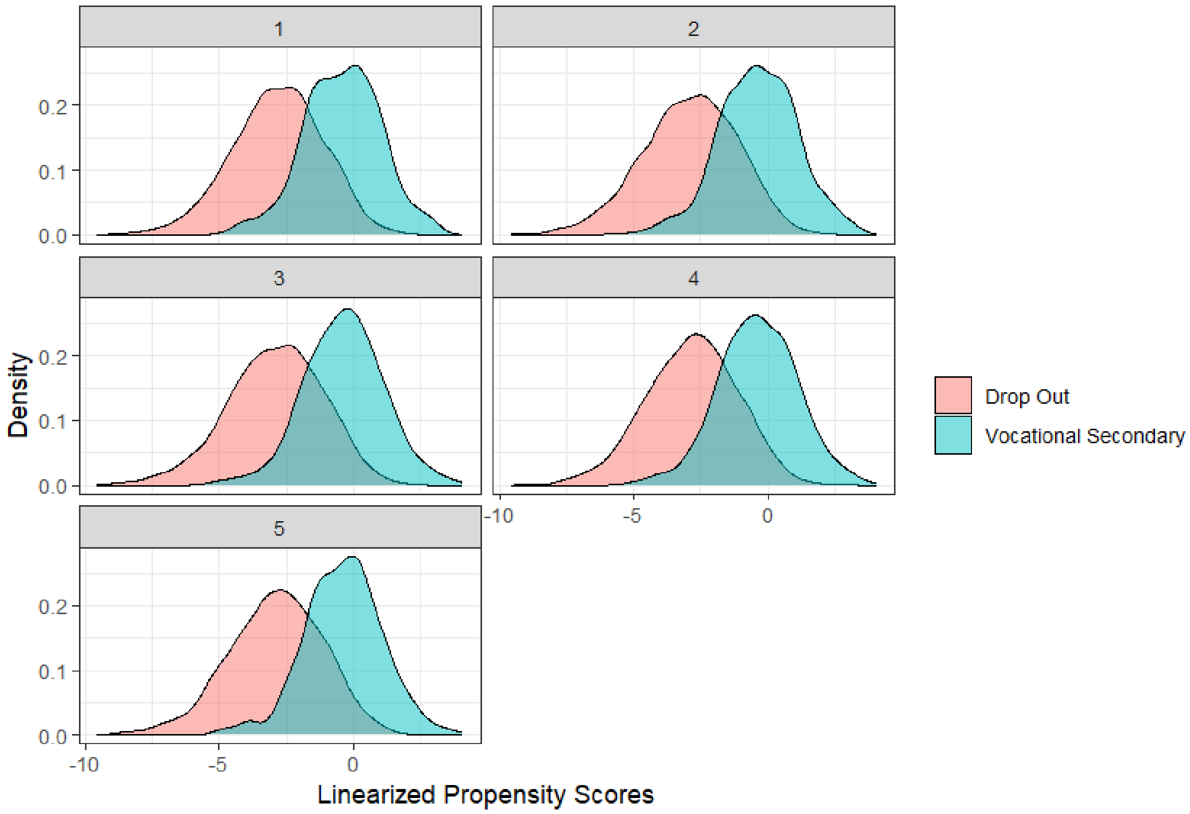
E.1. Vocational Secondary vs. Everybody Else



E.2. Vocational Secondary vs. General Secondary



E.3. Vocational Secondary vs. Dropouts

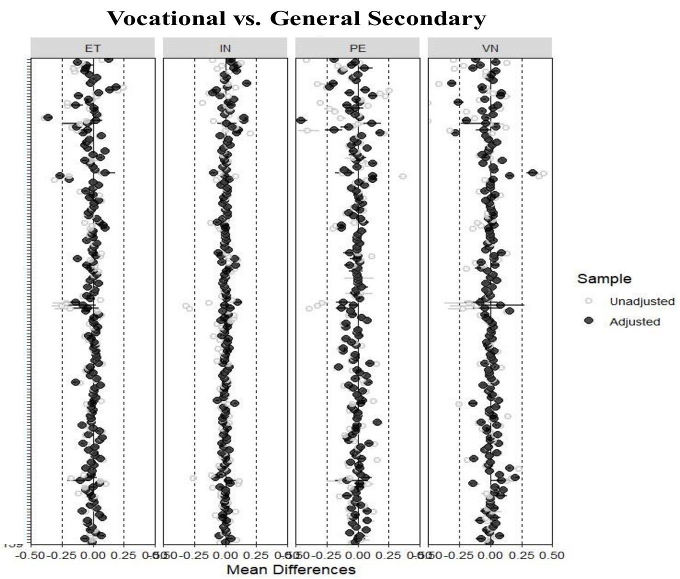


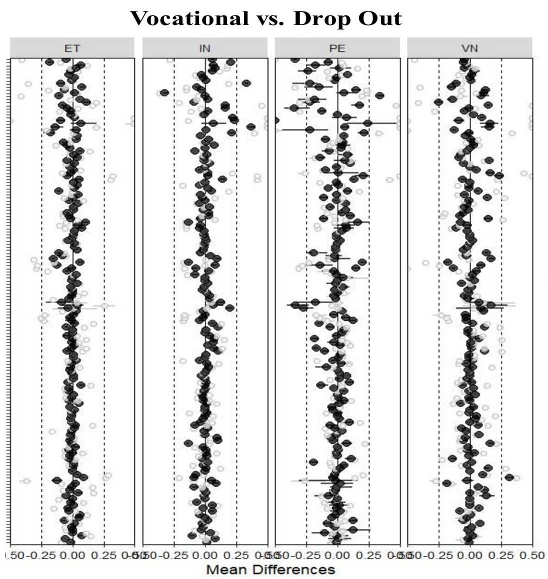
Appendix F: Balance Across Sub-Groups

Table 13: Overview of Covariate Balance across Moderators

Well balanced (all SMD within 0.25 for all subgroups)	Acceptably balanced (few outliers for some subgroups with SMD > .25)	Unbalanced (Many outliers across all subgroups with SMD > .25)
Gender	Country (only Peru)	Propensity Scores
Handcrafts jobs	Math test score	Time since Graduation
	Language test scores	
	Leadership	
	Self-efficacy	
	Self-esteem	
	Vocational dreamjob	
	Hours of studying	
	Population size	
	Agricultural jobs	
	Factory jobs	

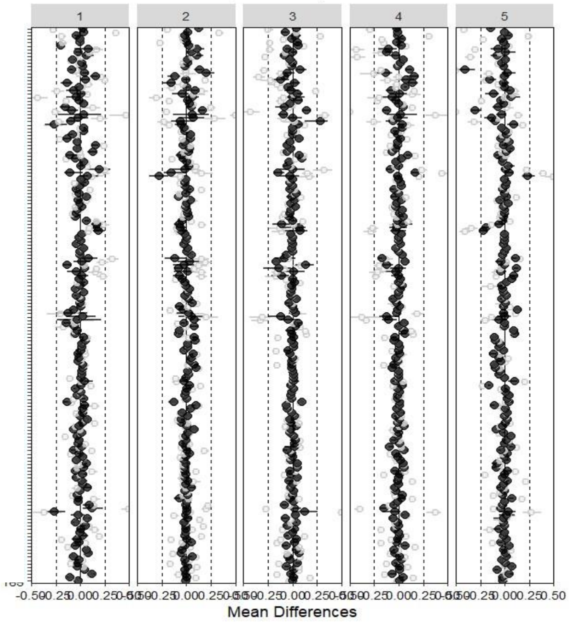
Country Heterogeneity



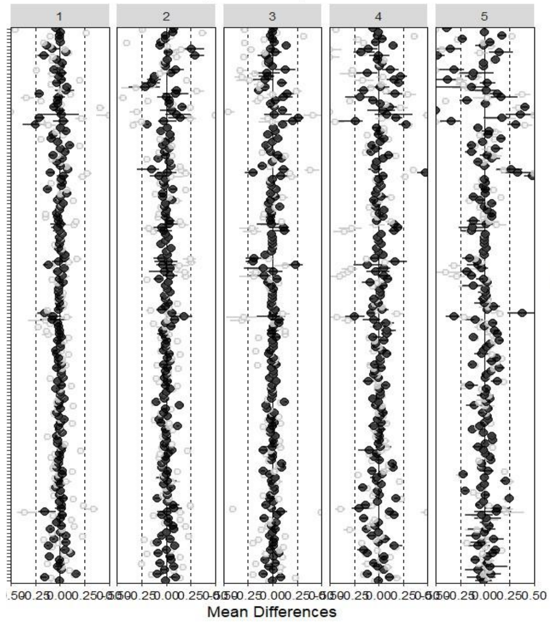


Propensity Score General Secondary

Vocational vs. General Secondary

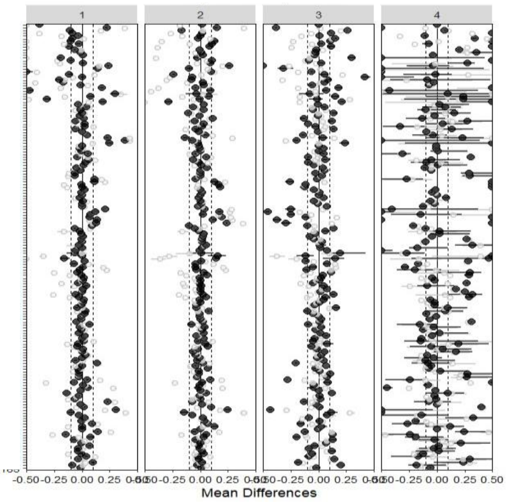


Vocational vs. Drop Out

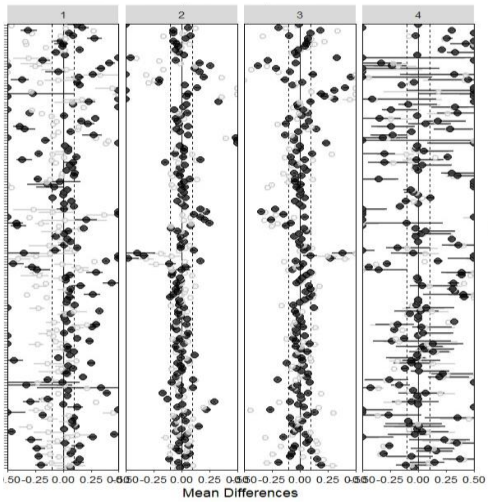


Time since Graduation

Vocational vs. General Secondary



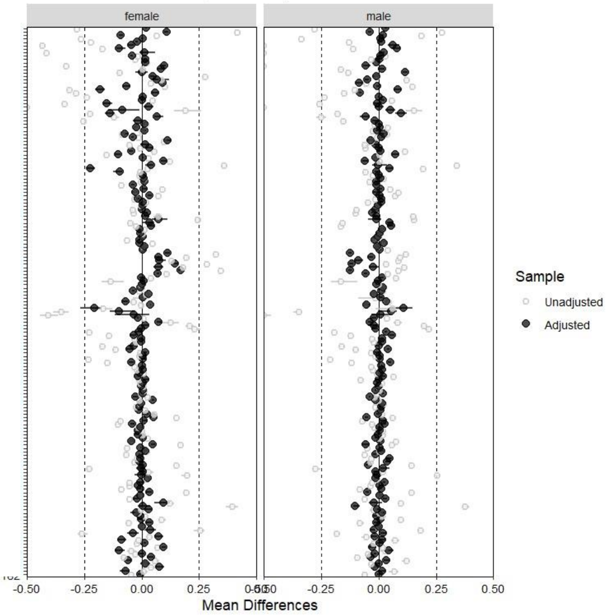
Vocational vs. Drop Out



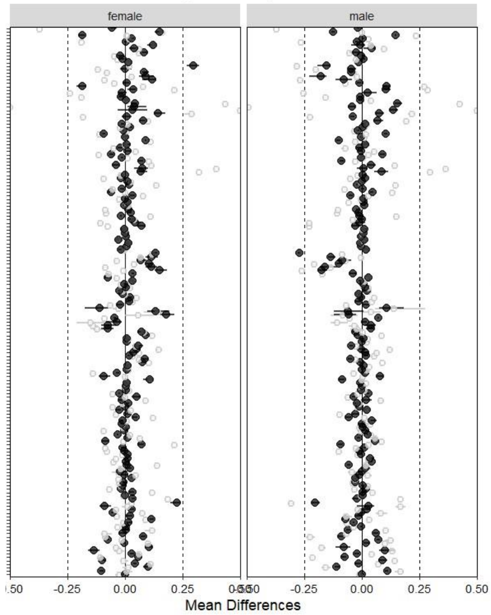
Personal Characteristics

Gender

Vocational vs. General Secondary

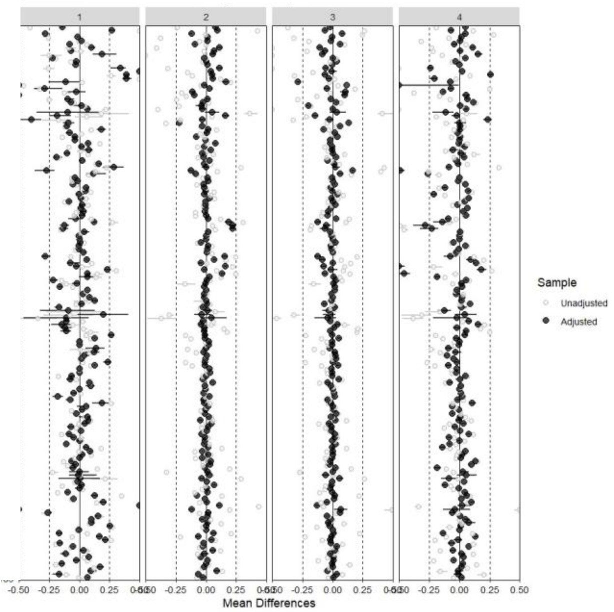


Vocational vs. Drop Out

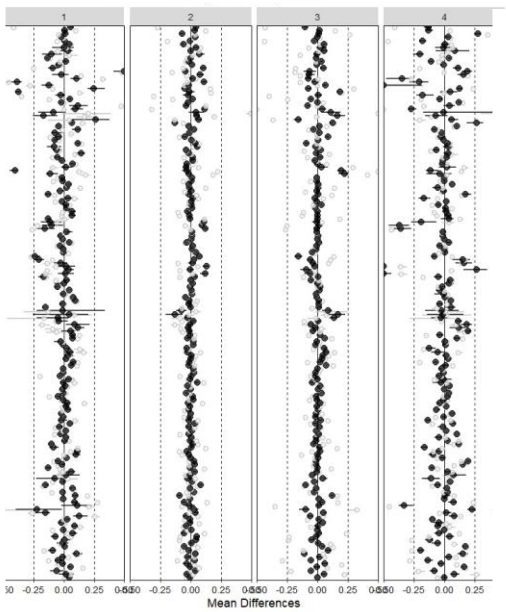


Math Test Score

Vocational vs. General Secondary

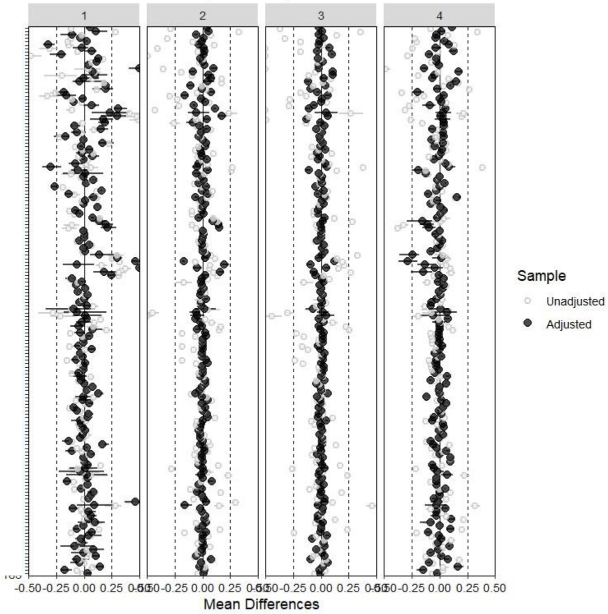


Vocational vs. Drop Out

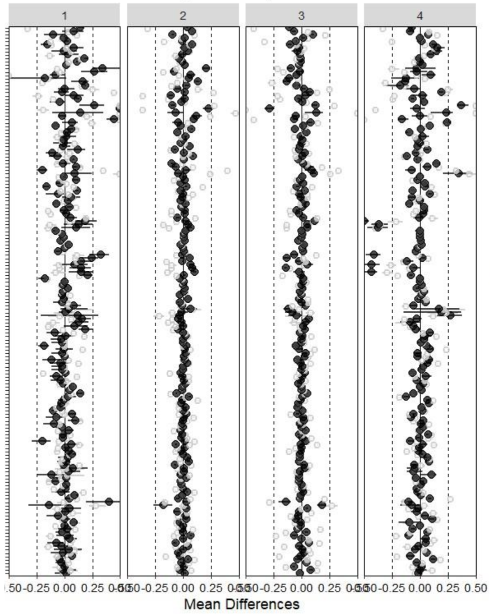


Language Test Score

Vocational vs. General Secondary

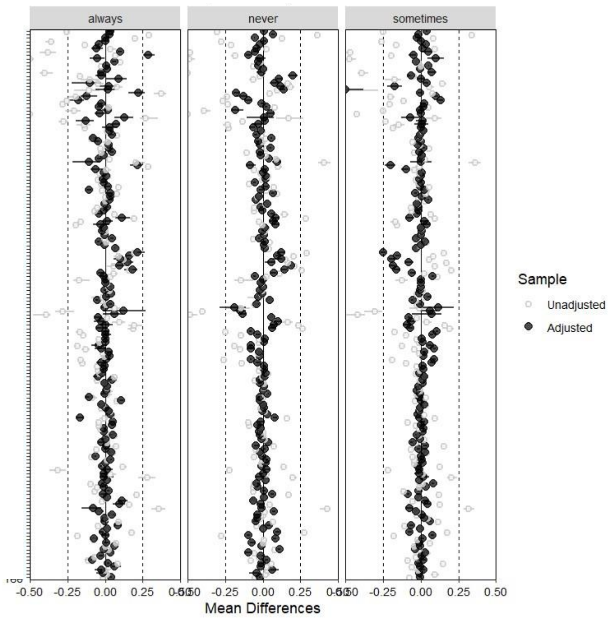


Vocational vs. Drop Out

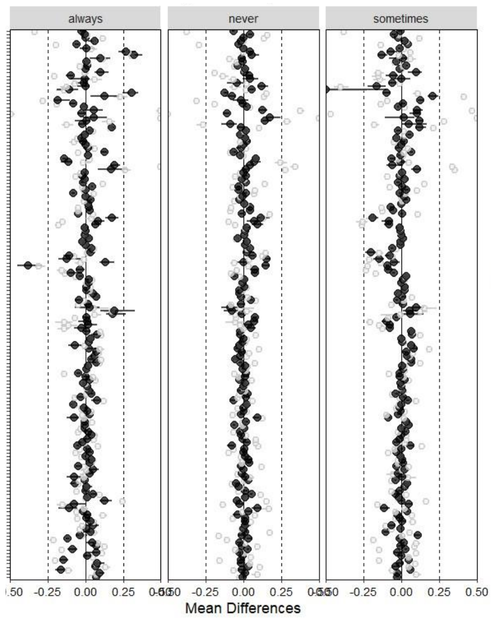


Leadership

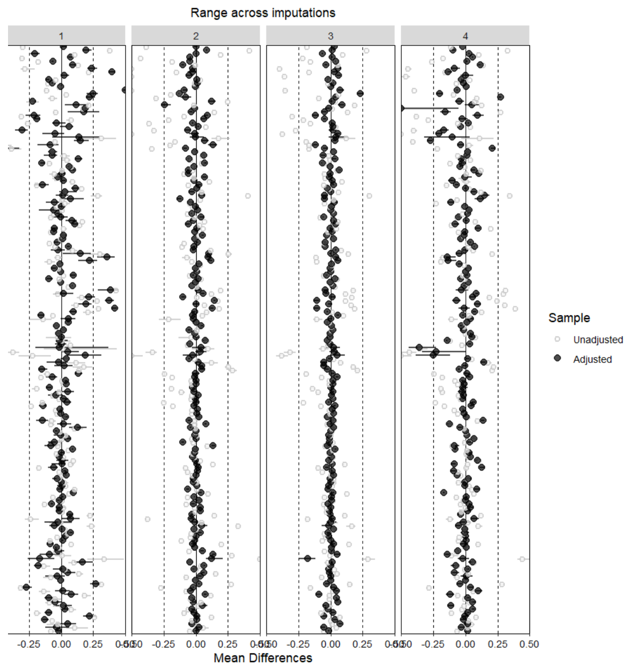
Vocational vs. General Secondary

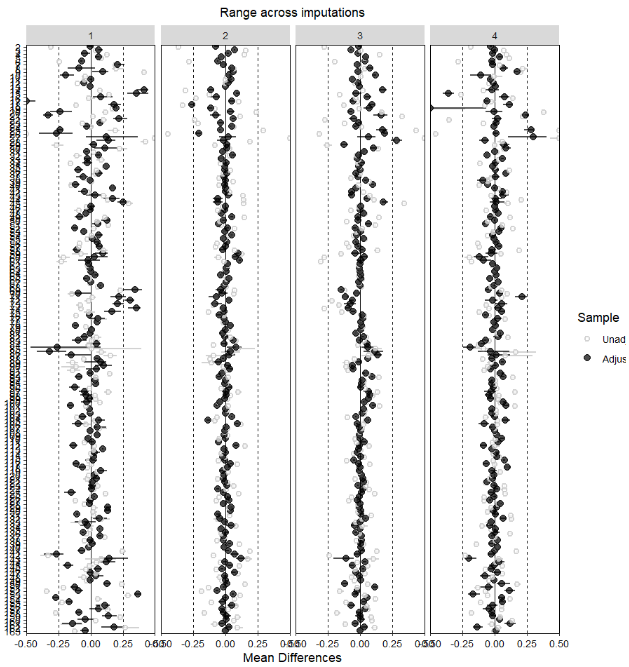


Vocational vs. Drop Out



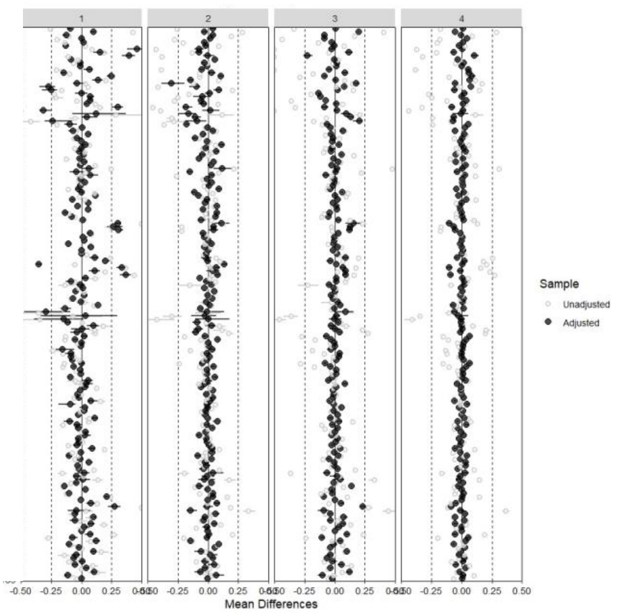
Self-Efficacy



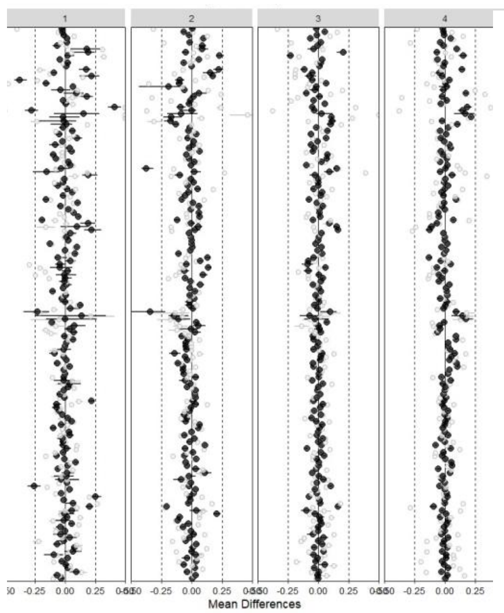


Self Esteem

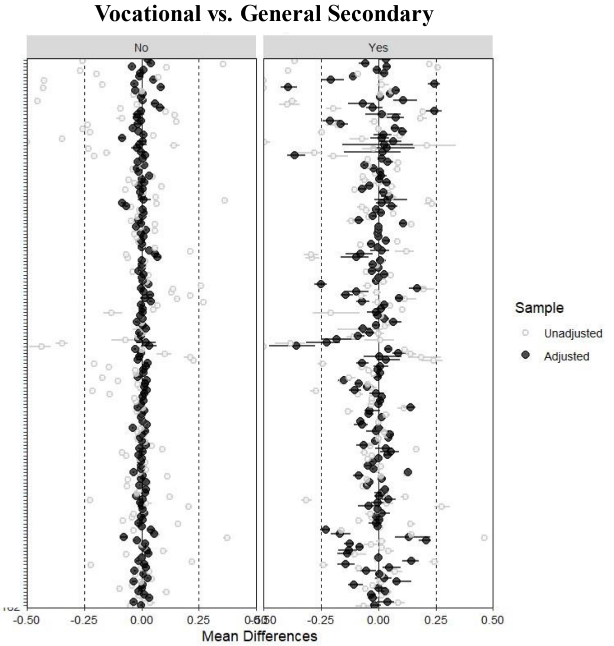
Vocational vs. General Secondary

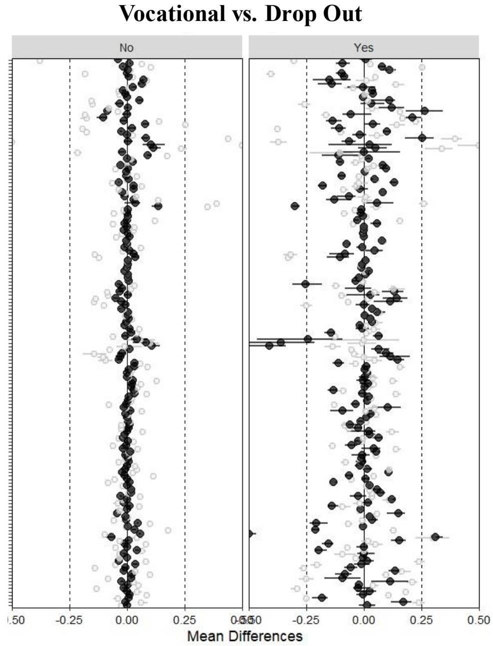


Vocational vs. Drop Out

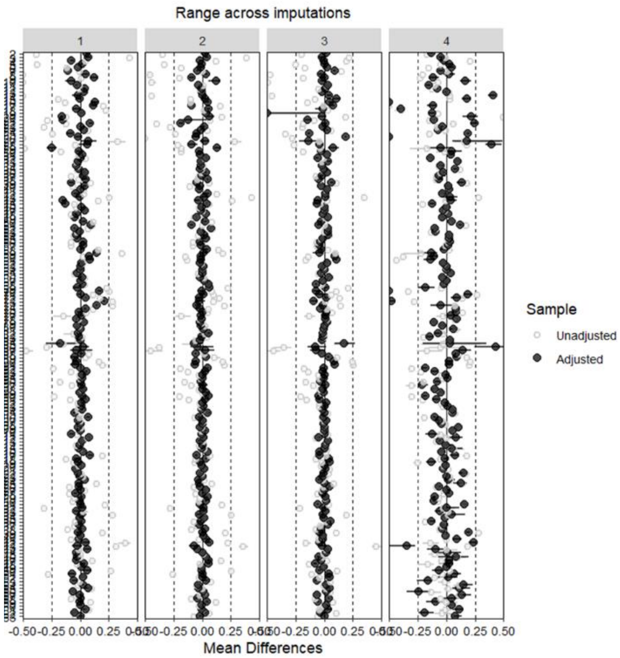


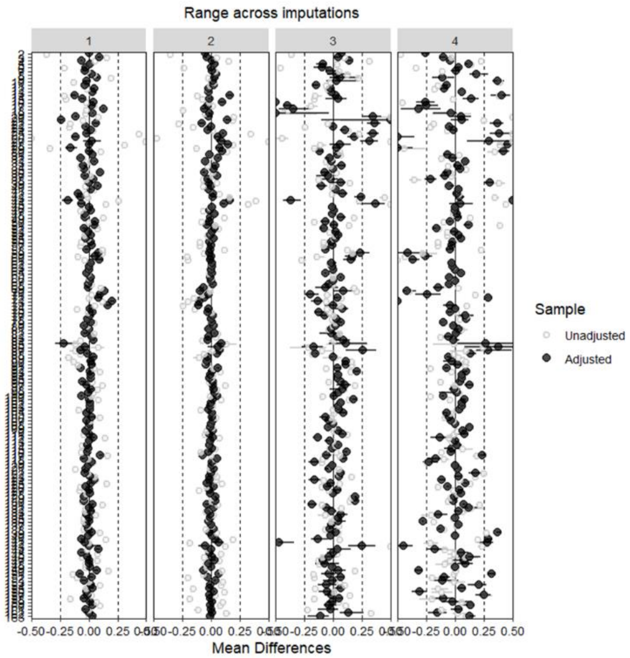
Having a Vocational Dreamjob





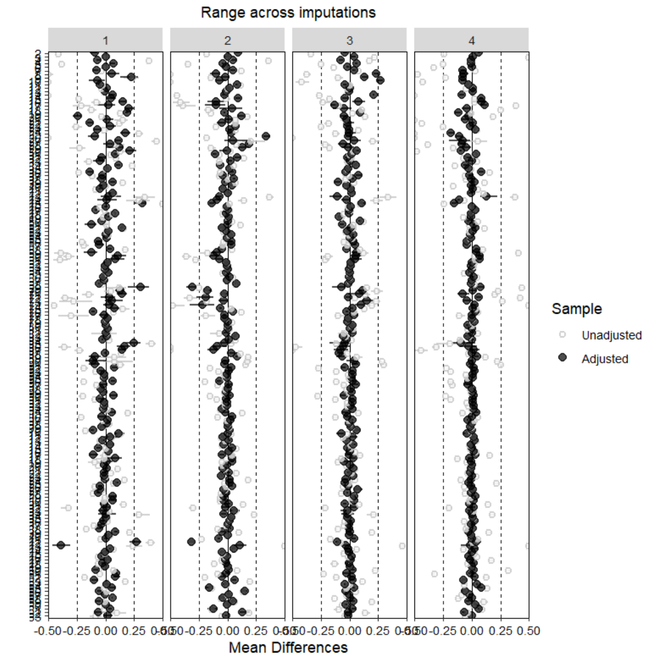
Hours of Studying

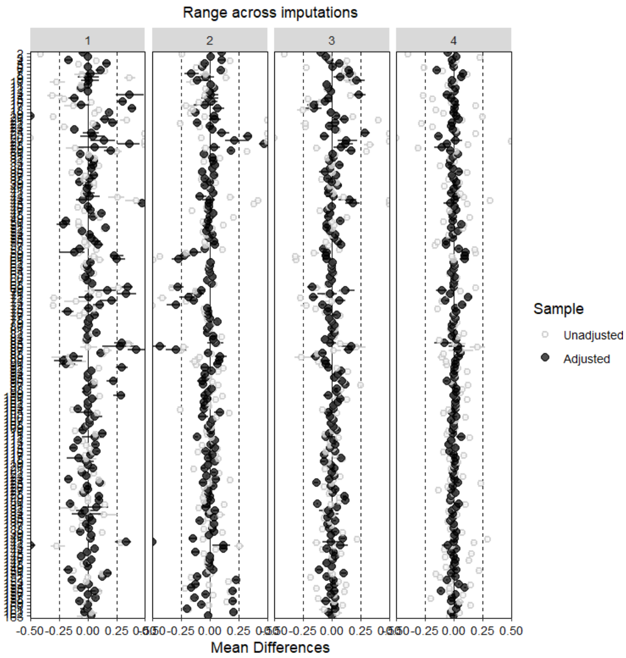




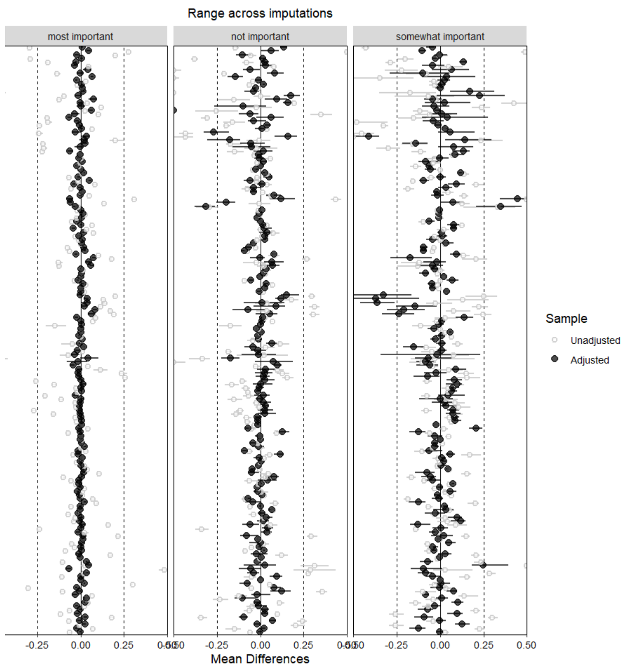
Community Characteristics

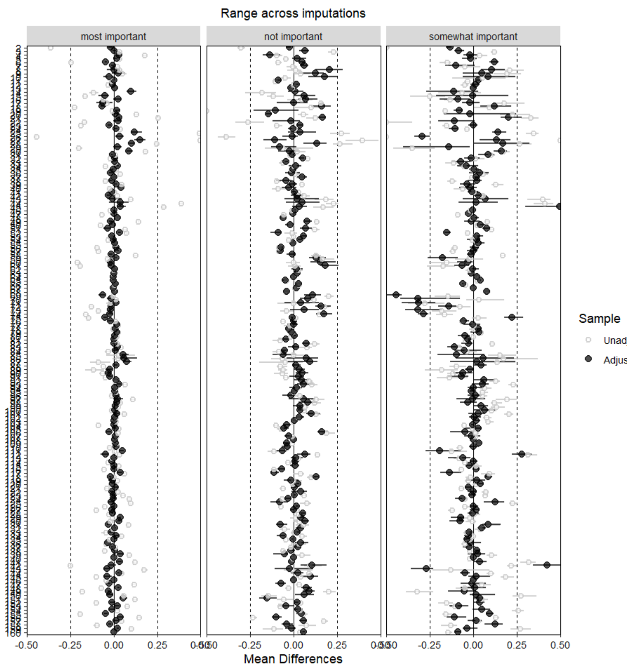
Population Size





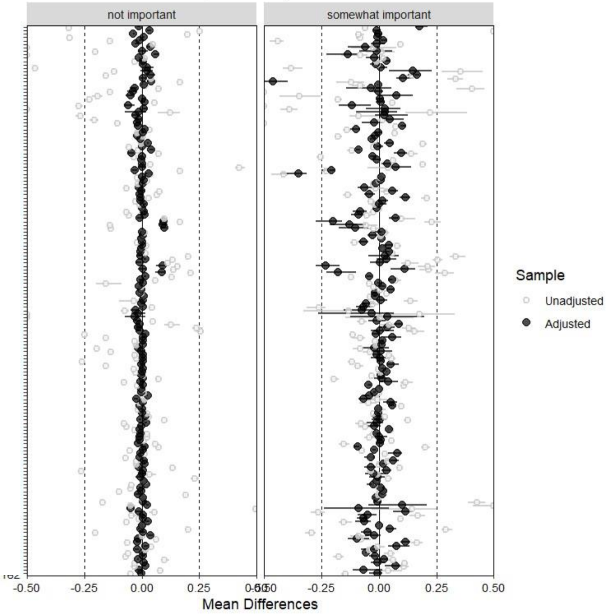
Agricultural Jobs



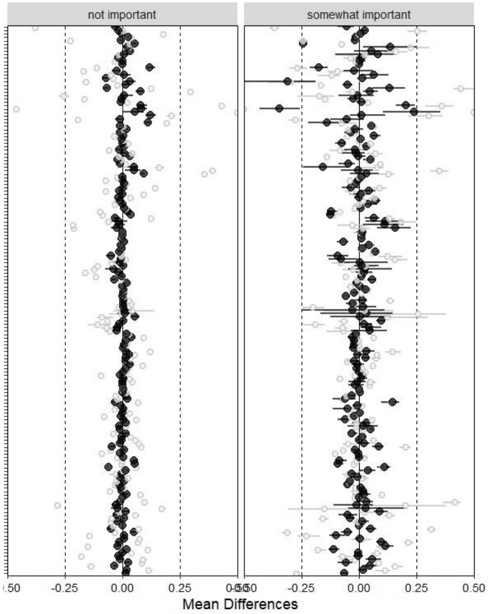


Factory jobs

Vocational vs. General Secondary

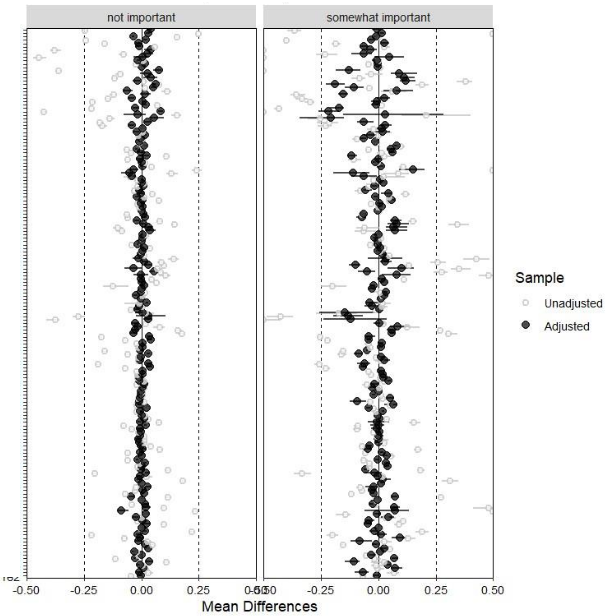


Vocational vs. Drop Out

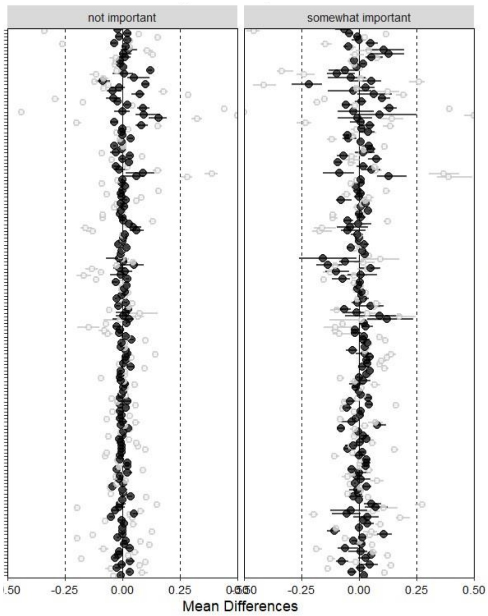


Handcrafts and small manufacturing jobs

Vocational vs. General Secondary



Vocational vs. Drop Out



Appendix G: Robustness Checks with GBM-Propensity Scores

The main tables in the paper are reproduced below using propensity scores calculated with a Generalized Boosted Model. GBM propensity scores also balance the data successfully, but slightly less well and are thus not preferred. However, I show below all main findings are robust to changing the computational method of the propensity scores.

Table 14: ATT estimates with GBM Propensity Scores

ATT	Hours per Week Worked	Hourly Income	Any IGA	Formal Work	Self Employed	Non Farming IGA	Attending Higher Education
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Panel A: Full Sample							
Treatment	0.06 (0.05)	-0.001 (0.05)	0.033 (0.025)	0.033 (0.026)	-0.035 (0.017)**	0.031 (0.025)	-0.043 (0.022)*
Adjusted R^2	0.197	0.236	0.191	0.146	0.109	0.165	0.397
Mean outcome	0	0	0.554	0.365	0.164	0.443	0.313
Effective Sample Size		Non Vocational: N = 1843			Vocational: N = 467		
Panel B: Vocational vs. General Secondary							
Treatment	0.052 (0.053)	-0.018 (0.052)	0.006 (0.029)	-0.006 (0.029)	-0.027 (0.019)	0.006 (0.029)	-0.26 (0.024)***
Adjusted R^2	0.198	0.232	0.181	0.139	0.144	0.18	0.361
Mean outcome	-0.120	0.137	0.582	0.426	0.126	0.506	0.587
Effective Sample Size		Non Vocational: N = 607			Vocational: N = 467		
Panel C: Vocational vs. Drop Out							
Treatment	0.087 (0.071)	0.119 (0.059)**	0.087 (0.033)***	0.087 (0.034)**	-0.057 (0.026)**	0.064 (0.035)*	0.294 (0.028)***
Adjusted R^2	0.253	0.246	0.249	0.191	0.094	0.205	0.452
Mean outcome	0.015	-0.193	0.531	0.324	0.183	0.402	0.084
Effective Sample Size		Non Vocational: N = 531			Vocational: N = 467		
Cohort FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Country FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Country*Year FE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MI	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Notes: a)*** p < .01, ** p < .05, * p < .1. b) Average marginal effects are computed using MarginalEffects package in R, c) all estimates use the full set of individual, family and sector covariates, and include cohort, country and country * year fixed effects c) all estimates are pooled from OLS regressions of five multiple imputed datasets using the Mice Package, and standard errors are robust of type HC1, d) all coefficients estimate the average effect on the treated.

Robustness: All results above are within 0.04 sd. or 1 p.p. and are statistically as significant as the regression results with GLM-propensity scores, where the impact of TVE is higher than estimated by GLM-propensity scores, with effects for hourly wage, any IGA and farming now significant.

Table 15: Heterogeneity by Country with GBM-Propensity Scores

ATT	Hours per Week Worked	Hourly Income	Any IGA	Formal Work	Self Employed	Non Farming IGA	Attending Higher Education
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Panel A: Vocational vs. General Secondary							
Ethiopia	-0.02 (0.107)	-0.091 (0.078)	-0.041 (0.057)	-0.021 (0.057)	-0.039 (0.04)	-0.032 (0.057)	-0.439 (0.045)***
India	0.164 (0.077)**	0.114 (0.086)	0.063 (0.041)	0.023 (0.041)	-0.019 (0.029)	0.073 (0.041)*	-0.135 (0.036)***
Peru	0.055 (0.125)	-0.233 (0.194)	-0.032 (0.062)	-0.071 (0.071)	0.025 (0.044)	-0.036 (0.067)	-0.261 (0.061)***
Vietnam	-0.113 (0.125)	-0.027 (0.048)	-0.014 (0.06)	0.003 (0.07)	-0.073 (0.029)**	-0.061 (0.065)	-0.17 (0.059)***
Mean outcome	-0.120	0.137	0.582	0.426	0.126	0.506	0.587
Panel B: Vocational vs. Drop Out after Primary							
Ethiopia	0.096 (0.121)	0.086 (0.079)	0.137 (0.055)**	0.144 (0.052)***	-0.095 (0.046)**	0.12 (0.057)**	0.092 (0.042)**
India	0.059 (0.109)	0.183 (0.104)*	0.059 (0.055)	0.055 (0.054)	-0.033 (0.037)	0.026 (0.055)	0.478 (0.043)***
Peru	0.426 (0.159)***	0.262 (0.244)	0.112 (0.083)	0.063 (0.089)	0.046 (0.062)	0.08 (0.088)	0.259 (0.066)***
Vietnam	-0.069 (0.129)	-0.022 (0.057)	0.022 (0.06)	0.041 (0.073)	-0.085 (0.035)**	0.012 (0.071)	0.376 (0.063)***
Mean outcome	0.015	-0.193	0.531	0.324	0.183	0.402	0.084

Notes: a)*** p < .01, ** p < .05, * p < .1. b) Average marginal effects are computed using MarginalEffects package in R, c) all estimates use the full set of individual, family and sector covariates (minus the multicollinear ones), and include cohort, country and country * year fixed effects c) all estimates are pooled from OLS regressions of five multiple imputed datasets using the Mice Package, and standard errors are robust of type HC1, d) all coefficients estimate the average effect on the treated.

Robustness: The main significant effects are robust to using GBM-propensity scores. The results are quantitively very similar for India, Vietnam, Ethiopia, and Peru, being within 0.04 sd. or 2 p.p.

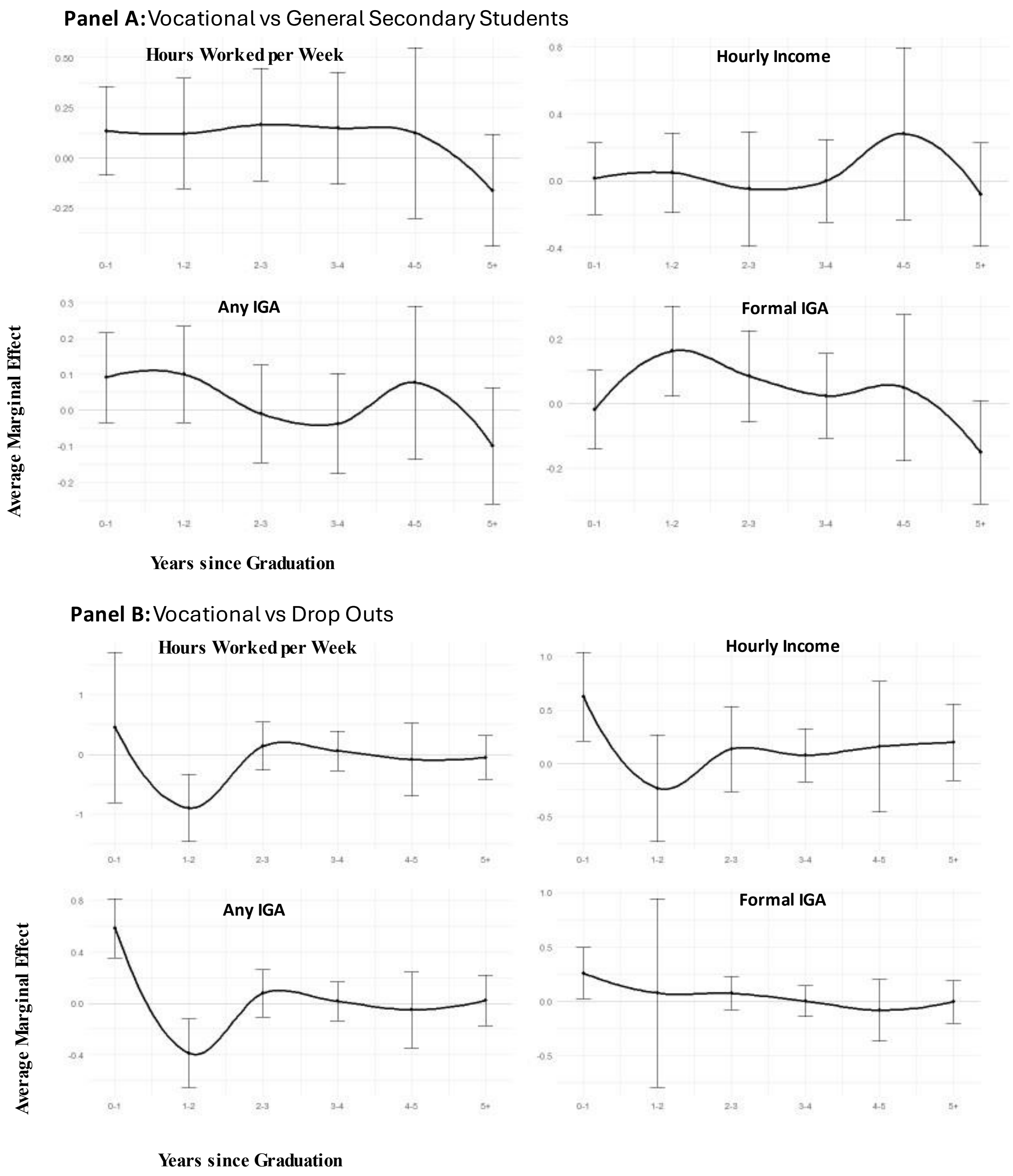


Figure 9: ATT over time with GBM-Propensity Scores

Robustness: The trend is equal to the GLM-specification. Estimates are also almost equal, except that the 0-1 group generally has slightly larger average marginal effect using GBM propensity scores.

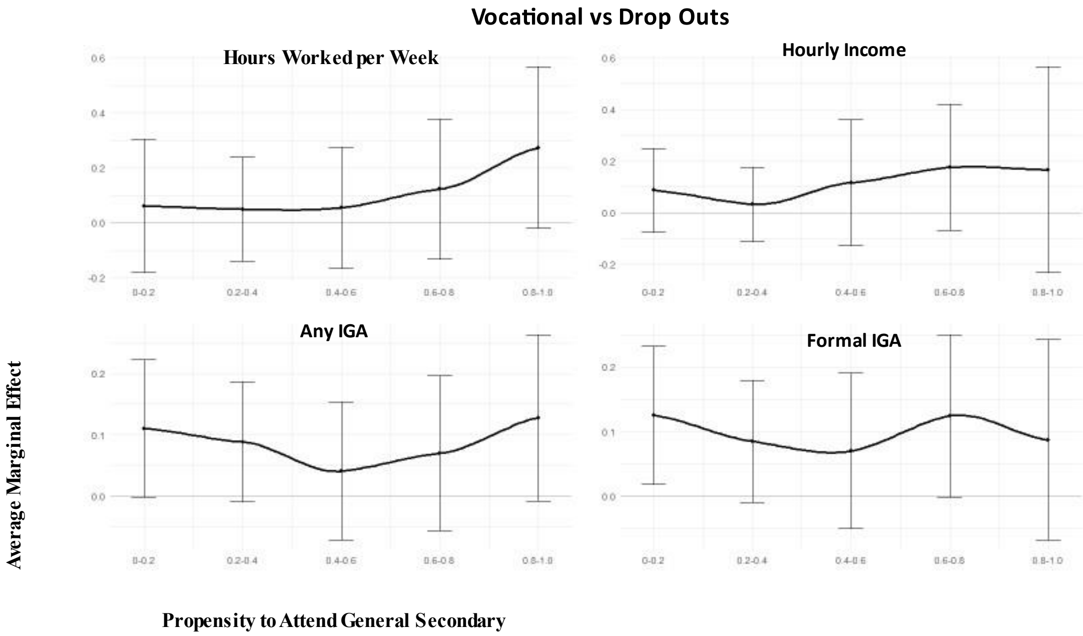


Figure 10: Heterogeneous Effects by Propensity to Attend General Secondary using GBM-Propensity Scores

Robustness: Using the GBM-specification, there is slightly more evidence for larger impacts among higher propensity scores for specifically hours worked per week. Hourly income also has a slight upward climbing slope, but this effect is small, and insignificant. For formal IGA and any IGA there is still no evidence that propensity to attend general education moderates in any way the impact of vocational education.

Appendix H: Table for Heterogeneous Effects by Country

Table 16: Heterogeneous Effects by Country with GLM-Propensity Scores

ATT	Hours per Week Worked	Hourly Income	Any IGA	Formal Work	Self Employed	Non Farming IGA	Attending Higher Education
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Panel A: Vocational vs. General Secondary							
Ethiopia	-0.02	-0.091	-0.041	-0.021	-0.039	-0.032	-0.439
	(0.107)	(0.078)	(0.057)	(0.057)	(0.04)	(0.057)	(0.045)***
India	0.164	0.114	0.063	0.023	-0.019	0.073	-0.135
	(0.077)**	(0.086)	(0.041)	(0.041)	(0.029)	(0.041)*	(0.036)***
Peru	0.055	-0.233	-0.032	-0.071	0.025	-0.036	-0.261
	(0.125)	(0.194)	(0.062)	(0.071)	(0.044)	(0.067)	(0.061)***
Vietnam	-0.113	-0.027	-0.014	0.003	-0.073	-0.061	-0.17
	(0.125)	(0.048)	(0.06)	(0.07)	(0.029)**	(0.065)	(0.059)***
Mean outcome	-0.120	0.137	0.582	0.426	0.126	0.506	0.587
Panel B: Vocational vs. Drop Out after Primary							
Ethiopia	0.096	0.086	0.137	0.144	-0.095	0.12	0.092
	(0.121)	(0.079)	(0.055)**	(0.052)***	(0.046)**	(0.057)**	(0.042)**
India	0.059	0.183	0.059	0.055	-0.033	0.026	0.478
	(0.109)	(0.104)*	(0.055)	(0.054)	(0.037)	(0.055)	(0.043)***
Peru	0.426	0.262	0.112	0.063	0.046	0.08	0.259
	(0.159)***	(0.244)	(0.083)	(0.089)	(0.062)	(0.088)	(0.066)***
Vietnam	-0.069	-0.022	0.022	0.041	-0.085	0.012	0.376
	(0.129)	(0.057)	(0.06)	(0.073)	(0.035)**	(0.071)	(0.063)***
Mean outcome	-0.015	-0.193	0.513	0.324	0.183	0.402	0.084

Notes: a)*** p < .01, ** p < .05, * p < .1. b) Average marginal effects are computed using MarginalEffects package in R, c) all estimates use the full set of individual, family and sector covariates (minus the multicollinear ones), and include cohort, country and country * year fixed effects c) all estimates are pooled from OLS regressions of five multiple imputed datasets using the Mice Package, and standard errors are robust of type HC1, d) all coefficients estimate the average effect on the treated.

Supplementary Materials I. Access to Raw Data, Processed Data and Coding

https://drive.google.com/drive/folders/1e4VQ8vnc2gUZ40tufYFvxDstW13kRCmS?usp=sharing

The coding files are very long, and thus it might not be ideal to directly copy them from the supplementary materials. Instead, I included above a link to a Google Drive with all the necessary material to replicate the analysis. After downloading the full map, and setting this map as the working directory, downloading the necessary packages, the code should work. It is critical to not change the names of the raw data files.

- In the Google Drive there are several sub-maps. Unproc_data includes all the raw Young Lives data from waves 1-6 and the constructed data files made by Young lives, but also the corresponding data dictionaries. The raw files are identical to those downloaded directly from Young Lives Survey, but have gotten a new prefix for easier data handling. Unproc_data also includes historical forex rates for the four countries to US$.
- In the map proc_data are the processed data-files which were used for the data analysis in the thesis. These are produces by the R-markdown file "data preparation", and can directly be loaded into the analysis without having to reload the data preparation. There are 3 main datasets, final_data_mi_dep1 that corresponds to multiple imputed dataset for the full sample. Final_data_mi_dep2 and dep3 are subsets of these for vocational vs general and vocational vs drop-out respectively.
- There are also two R-markdown files with code. Data preparation handles all the processing from raw into final data, including the multiple imputation. Multiple imputed analysis handles all the data analysis mentioned in the thesis. Further instructions on how to use these files are included in the Markdown.
- A specific codebook on the processed final_data_dep1/dep2/dep3 is included below in Supplementary Materials II.

II: Codebook with Descriptions of Outcomes, Treatment and Covariates

II.I. Outcomes:

Name	Unit	Source	Computation
Hours Worked Per Week	Standardised	Raw data	In Ethiopia, India and Peru: The number of days per week worked in the last 90 days * the number of hours worked per day on average in the last 90 days In Vietnam: The number of days worked per month / the number of hours worked per day / 4.345 assuming no weeks of holidays in the measured time-period. This is different because the framing of the question was different in Vietnam. Hours worked per week is set to zero if Any IGA is equal to zero.
Hourly Income	Standardised	Raw data	Survey respondents filled in their earned income (including informal/in-kind) for their preferred time-period in the local currency. These were multiplied and then divided to hourly wage, assuming 4.345 working weeks a month on average. Thus hourly income assumes full-time work. Using income per week did not change the results. Before standardization, wages were exchanged from local currency to US$ using the exchange rate on the date of interview according to Yahoo Finance. Any values labelled other, not known or refused to answer are labelled as NA and later imputed, unless the person answered they were unemployed, then salary was imputed with zero.
Any IGA	Binary	Raw data	1 = if worked at least one hour during the 7 days before 1= if not worked during the last 7 days, but did have a job 0 otherwise, except if participant was not present then NA.
Formal IGA	Binary	Raw data	1 if type of activity is one of "Regular Salaried Employment", "Salaried Farmer", "Salaried worker", "Wage Employment (Agriculture)", "Salaried Worker", "Annual Farm Servant", "Working for wage in non-agricultural activities (e.g. in mine/workshop/factory/construction/making food or drink" “Waged worked”)

			0 if type of activity is not part of that list. Only the first activity is considered: defined as the activity with the highest income
Self-employed	Binary	Raw data	1 if string “Self-Employed”, “Independent”, “own farm”, “selling goods” or “making” was in type of activity 0 otherwise Only the first activity is considered: defined as the activity with the highest income
Non-Farming IGA	Binary	For call 6: Constructed dataset by Young Lives For wave 5: Derived from raw data	Wave 6: see Young Lives Documentation Wave 5: if “farm”, “agriculture”, “food-crops”, “non-food, including horticulture, sericulture and floriculture” or “livestock” is detected in the type of activity, then non-farming IGA is 0. Otherwise it is 1 if employed, and 0 if unemployed. “other” is coded as NA.
Attended higher education	Binary	Raw data	1 if respondent attended (technical) college, lower-level vocational tertiary degree, a teacher’s education, pedological institute or university for at least one year, 0 otherwise.

II.2. Outcomes:

Raw data is derived from childlevel (wave 3), educationhistoryindex (wave 4 and wave 5) and arch (wave 6)

Treatment definitions:

- Attended vocational secondary: students has been enrolled for at least one year in an institute labelled vocational secondary.
- Attended general secondary: students has never been enrolled in vocational secondary, and was enrolled in the final year of upper-secondary education.
- Drop-out: student was enrolled in the final year of primary school, has never been enrolled in vocational secondary, and has never been enrolled in the final year of upper-secondary.
- Attended higher education: student has been enrolled for at least a year at a college or university level

Overview of which labels were included in which school category:

Type:	**Names in dataset**
Ethiopia:	
Vocational Secondary	TVET $1^{st}/2^{nd}/3^{th}/4^{th}$ Year
Non-Formal	religious education, kindergarden, creche day-care
Primary	Grade 1-8
General Lower Secondary	Grade 9/10
General Upper Secondary	Grade 11/12
Higher education	College: all cycles of primary teaching certificates and pre-school teaching certificates University: Secondary teaching, undergraduates and masters (probably also includes technical diploma's since those aren't mentioned separately)
India:	
Vocational Secondary	Vocational
Non-Formal	religious education, adult literacy
Primary	Grade 1-8
General Lower Secondary	Grade 9/10
General Upper Secondary	Grade 11/12
Higher education	College: technical colleges University: undergraduates and masters
Vietnam:	
Vocational Secondary	Vocational secondary schools
Non-Formal	religious education, adult literacy, non-formal continued education, short-term vocational, "any pre-primary" "any pre-primary grade" (pre)-kindergarten
Primary	Grade 1-5
General Lower Secondary	Grade 6-9
General Upper Secondary	Grade 10-12

Higher education	College: vocational college, post-secondary technological institute, Professional Secondary (vocational college is tertiary education, since all students already have an upper secondary diploma before enrolling. University: undergraduates and masters
Peru:	
Vocational Secondary	Cent. Tecnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO
Non-Formal	religious education, kindergarden, creche day-care
Primary	Grade 1-6
General Lower Secondary	first cycle, grade 7/8/9 (by own definition, lower-secondary is not a seperate entity in Peru)
General Upper Secondary	second cycle, grade 10/11
Higher education	College: technical or pedagogical institute (technical is vocational college, pedagological is training for education), No Univ. Completa regular college University: University Sup. (Includes Officials School) and masters

II.3. Baseline Characteristics

When possible data was collected from wave 2 in OC, and wave 4 in YC. However, for both cohorts I also used wave 3 for some unique questions. Individual characteristics derive from surveys directly filled in by the surveyed child (except childhood health), household characteristics are derived from surveys filled in by the head of the household, and community characteristics are aggregated responses from “experts” in the community (aggregation is done before data was published by Young Lives). The last column indicates if the variable is also included in the outcome models, or only in the propensity score weighting. Ethnic groups and religion are excluded to keep a sufficient sample size of treated in each group, the others are excluded to prevent multicollinearity. For variables computed by Young Lives, I refer to Briones (2018).

Variable	name in dataframe	Computation	Included in outcome model
Individual Characteristics			
General Demographics			

Gender	chsex	Based on a single survey question	Yes
Ethnic group	cheth	Self-reported, based on a single survey question	No
Religion	chrel	Self-reported, based on a single survey question	No
Health			
Child weight	Zweight_8	Weight for age z-score at age 8, if flagged for being a likely data error according to WHO, then NA.	No
Child height	Zheight_8	Height for age z-score at age 8, if flagged for being a likely data error according to WHO, then NA.	Yes
Child BMI	Zbmi_8	BMI for age z-score at age 8, if flagged for being a likely data error according to WHO, then NA.	Yes
Serious illness	Chillness_8_13	Was child seriously ill during the last three years (for OC wave 2, for YC wave 3 and 4).	Yes
Chronic illness	long_term_health_problem	Wave 2 for OC, wave 5 for YC (Binary), self-reported Did child suffer from a long-term health problem?	Yes
Disability	chdisability	Only available for wave 4, does child suffer from a permanent disability (binary)? Disablity cannot be affected by the treatment, and can thus still be used as a covariate, also for the OC.	Yes
Self-reported health	subjective_health_13	Wave 2 for OC, wave 4 for YC. Self-reported score on how healthy child generally felt. Scale is 1 (low) to 9 (highest)	Yes
Time Use			
Child labour	chldwork_during_school	Did child work while still going to primary/secondary school: work defined as any activity that generates monetary income (Binary). W2 OC, W4 YC	Yes
Missed school	missed_school	Missed school for more than one week during the last year (Binary). W2 OC, W4 YC	Yes
Hours spent …	Hsleep	Measured during wave 2 for the OC, and wave 4 for the YC. Open questions asking the children how many hours a day the spend on this activity. Originally measured in hours, afterwards standardised.	Yes
…	Hcare		Yes
…	Hchore		Yes
…	Htask		Yes
…	Hwork		Yes
…	Hschool		Yes
…	Hstudy		Yes

…	Hplays		Yes
Cognitive Skills			
Math scores	math_score_13	Item response theory scores at age 13 for a) mathematics, b) reading and c) Peabody picture vocabulary tests. The tests were conducted by Young Lives personnel, and total scores were also computed by Young Lives. Afterwards, I standardised the scores to have mean 0.	Yes
Reading scores	read_score_13		Yes
Vocabulary scores	ppvt_score_13		Yes
Vocabulary learned	ppvt_score_improvement	Improvement between ages 8 and 13 on score of Peabody picture vocabulary test and math test.	Yes
Math learned	math_score_improvement		Yes
Non-Cognitive Skills			
Friends	Noncog_Friend	Number of friends spoken during the last 7-days (standardised) (W3 OC, W4 YC)	Yes
Extrovertness	Noncog_hardtalk	Ordinal categorical: Do you find it hard to talk to others in your class? (W3 OC, W4 YC)	Yes
Sociability	noncog_incgame	Ordinal categorical: Do friends include you in their games? (W3 OC, W4 YC)	Yes
Leadership	Noncog_lead	Ordinal categorical: Do friends perceive you as a leader? (W3 OC, W4 YC)	Yes
Helping	Noncog_helpchld	Ordinal categorical: Do you help other children with problems at school? (W3 OC, W4 YC)	Yes
Trust	Noncog_trust	Index of 3 questions: (W3 OC, W4 YC) I. Most people in my neighbourhood can be trusted II. I believe the government does what is right for people like me III. I feel safe when I go out of the house on my own The index is the average of the responses available and is standardised.	Yes
Self-efficacy	Noncog_self_efficacy	Index of 5 questions: (W3 OC, W4 YC) I. If I try hard I can improve my situation in life II. people in my family make all the decisions about how I spend my time III. I like to make plans for my future studies and work IV. If I study hard I will be rewarded with a better job in the future V. I have no choice about the work I do The index is the average of the responses available and is standardised.	Yes
Self-esteem	Noncog_self_esteem	Index of 5 questions based on the Pride scale (W3 OC, W4 YC) I. I am ashamed of my clothes II. I am ashamed of my shoes	Yes

		III. I am often embarrassed because I do not have the right supplies for school IV. I am worried that I don't have the correct uniform V. I am embarrassed by the work I have to do The index is the average of the responses available and is standardised. Some questions were phrased negatively for the OC, and positively for YC.	
Child's expectations and aspirations			
Expected highest grade of education	expected_grade	Education grade you would like to complete are 20yrs old (W3 OC, W4 YC) Answers grouped in below primary, lower secondary, upper secondary, technical/vocational college, college/university, following the grouping set out in I.2. There was no option to choose for vocational secondary. There was also no distinction in most countries between college and university.	Yes
Preferred sector of working	dreamjob_sector	Job you think you will be doing when you are 25 (W3 OC, W4 YC). Answers grouped in sectors: I. Healthcare II. Education and research III. Services and management IV. Public administration and public services V. Skilled trades and manual labor VI. other	No
Preferred sector requires most likely vocational education	vocational_dreamjob_dummy	Is your dreamjob vocational in nature? (Binary, self-constructed, W3 OC, W4 YC). 1 if in "construction worker", "cook", "driver", "engineer", "fireman/woman", "fisherman", "fisherman/woman", "labourer", "mason", "mechanic", "tailor", "taxi driver", "domestic worker", "farmer", "painter/decorator", "traditional occupation", "market trader/shop assistant", "painter", "trader/businessman/woman", "trader", "market trader", NA if dreamjob is not known, 0 otherwise	Yes
Preferred sector requires most likely academic training	academic_dreamjob_dummy	Dummy whether job requires academic study (Binary, self-constructed, W3 OC, W4 YC). 1 if in "doctor", "dentist", "nurse", "vet", "veterinary", "teacher", "lecturer", "scientist", "university student", "university student/other form of further education", "accountant", "lawyer", "management", "manager/management", "computer operator", "civil servant", "politician", "president of the country", "president/leader of country"", NA if dreamjob is not known, 0 otherwise	Yes
<u>Household Characteristics</u>			
Household's demographics			

Dad's age	dadage_atbirth	Age in years at birth of respondent Calculated by current age – age child	Yes
Mom's age	momage_atbirth		Yes
Caretaker's age	careage_atbirth		No
Dad passed away	dadpassed	Dad/mom passed away before starting secondary (Binary, W2 OC, W4 YC)	Yes
Mom passed away	mompassed		Yes
Family relationship to primary caregiver	primarycaregiver	The relationship between respondent and primary-caregiver (categorical, W2 OC, W4 YC). Either parent, relatives, sibling or nonrelatives.	Yes
One of parents seriously ill	parent_sick	One or both parents fell seriously ill since last survey (binary, W2 OC, W4 YC)	Yes
Household's size			
Household size	hhsize	All numeric, and directly derived from surveys in W2 (OC) and W4 (YC). Note that there might be differences between household size and number of children born, since it is likely some live with their extended families.	Yes
Numbers of boys aged between 0-12 in household	male012		No
Numbers of girls aged between 0-12 in household	female012		No
Numbers of children born before respondent	bornbef		Yes
Numbers of children born after respondent	bornaft		Yes
Number of children in the household	total_children_household		No
Education of parents/caretakers			
… Can't Read	dadcantread	… can't read according to Young Lives Surveyer (3-level categorical variable), W2 (OC), W4 (YC).	Yes
	momcantread		Yes

	carecantread		No
… attended any formal education	mom_edu_attended_formaleducation	… attended formal education, defined as primary or more (binary, W2 OC, W4 YC) "religious education" and "Adult literacy" are not considered formal education	Yes
	dad_edu_attended_formaleducation		Yes
	care_edu_attended_formaleducation		No
… attended education higher than primary	mom_edu_beyond_primaryeducation	… attended an educational degree higher then primary, defined as being enrolled in grade 8+ or studying at vocational, college or university (binary, W2 OC, W4 YC).	Yes
	dad_edu_beyond_primaryeducation		Yes
	care_edu_beyond_primaryeducation		No
… attended higher education	mom_edu_attended_postsecondary	… attended an educational degree higher then secondary, defined as being enrolled in one of Post-secondary, vocational", "Technical, pedagogical, CETPRO (complete)", "Technical, pedagogical, CETPRO (incomplete)", "Vocational, technical college", "Masters, doctorate", "University", "University (complete)", "University (incomplete)" (Binary, W2 OC, W4 YC)	Yes
	dad_edu_attended_postsecondary		Yes
	care_edu_attended_postsecondary		No
… attended vocational secondary/tertiary	mom_edu_attended_vocational	… attended any of "Post-secondary, vocational", "Technical, pedagogical, CETPRO (complete)", "Technical, pedagogical, CETPRO (incomplete)", "Vocational, technical college"), data specifically for vocational secondaries is lacking	Yes
	dad_edu_attended_vocational		Yes
	care_edu_attended_vocational		No
Household's perception of education			
Perceived quality of primary school	quality_primary_school	Nearest primary school provides a good quality education for children (5-level categorical variable, W3 (OC) W4 (YC))	Yes
Usefullness of formal education	formal_education_useful	Do you think formal education has been useful in your life (5-level categorical variable, W3 (OC) W4 (YC))	Yes
Should child stay in school during financial hardship	education_during_financial_hardship	12yr old son/daughter at school - family needs money - what should family do? 3 levels (stay in school, leave or not known) If child is a girl, then the case of a daughter is used. If child is a boy, then the son's case is used	Yes
Household's expectation			
Expected age of marriage	expected_age_married	At what age should child get married (Categorical variable, no expectations and then age groups, W3 YC, W4 OC)	Yes
Expected age to earn self-sustaining income	expected_age_earning	At what age should child earn money to support (Categorical variable, no expectations and then age groups, W3 YC, W4 OC)	Yes

Expected age of leaving school	expected_age_leaving_school	At what age should child leave full-time education (Categorical variable, no expectations and then age groups, W3 YC, W4 OC)	No
Do parents expect their childs to meet their expectations	realistic_expectations_parents	Do you think child will complete desired level of education (Yes, no, not known) (W3 YC, W4 OC)	Yes
Sector they hope their child will work later	parents_dreamjob_sector	What job would you most like child to do in the future (W3 OC, W4 YC). Answers grouped in sectors: I. Healthcare II. Education and research III. Services and management IV. Public administration and public services V. Skilled trades and manual labor VI. other	No
Is that sector vocational	parents_vocational_dreamjob_dummy	Dummy is job is vocational in nature: (Binary, self-constructed, W3 OC, W4 YC). 1 if one of "construction worker", "cook", "driver", "engineer", "fireman/woman", "fisherman", "fisherman/woman", "labourer", "mason", "mechanic", "tailor", "taxi driver", "domestic worker", "farmer", "painter/decorator", "traditional occupation", "market trader/shop assistant", "painter", "trader/businessman/woman", "trader", "market trader", "ingeniero(a)", "cocinero(a)", "trabajador (a) de construcci", "pintor(a) / decorador(a)", "mec", "chofer", "chofer de taxi (taxista)", "pescador(a)", "trabajadora dom", "sastre", "agricultor(a)", "factory worker", "cient", "l", "trader/ businessman/woman", "vendedor en mercado / ayudante en tienda", zero otherwise unless dreamjob is unknown, then it is NA.	Yes
Is that sector academic	parents_academic_dreamjob_dummy	Dummy whether job requires academic study (Binary, self-constructed, W3 OC, W4 YC). 1 if one of "doctor", "dentist", "nurse", "vet", "veterinary", "teacher", "lecturer", "scientist", "university student", "university student/other form of further education", "accountant", "lawyer", "management", "manager/management", "computer operator", "civil servant", "politician", "president of the country", "president/leader of country", "profesor universitario", "profesor(a)", "estudiante universitario / otra educaci", "abogado(a)", "contador", "operador(a) de computadora", "religious leader/priest/sheikh", "veterinario(a)", "religious leader/priest/shaik", "presidente del pa", "president of country", "piloto"	Yes
Household's economic situation			
Sector of primary job	household_primary_job	Most important money-making activity of the household in last 12 months Grouped into sectors	No

		I. agriculture II. crafts and manufacturing III. services IV. construction and repairs V. causal labor The type of jobs named from the dreamjob questions, thus the grouping is also different.	
Do they own their house	ownhouse	Do you own the house you're living at? (Binary, W2 OC, W4 YC)	Yes
Index of housing quality	hq	An index by Young Lives measuring the quality of the house, both in structure, accessibility and luxuries available (numeric, standardised, W2 OC, W4 YC)	Yes
Index of access to services	sv	An index by Young Lives measuring access to public services, both whether they are close by and affordable (numeric, standardised, W2 OC, W4 YC)	Yes
Index of possession of consumer durables	cd	An index by Young Lives measuring the type of consumer durables owned by the household (numeric, standardised, W2 OC, W4 YC)	Yes
Is household in debt	debt	Do you have any serious debts (Binary, W2 OC, W4 YC).	Yes
Household economic shock			
Household felt victim to crime	shock_crime	Is one if one of the 8 indicators for falling victim to crime equals one, zero otherwise (binary, W2 OC, W4 YC)	Yes
Head of household lost their job	shock_household_job_loss	Is one if either a) shock-loss of job b) source of income/ or c) family enterprise is one, zero otherwise (W2 OC, W4 YC).	Yes
Household felt victim to natural disaster	shock_natural_disaster	Is one if one of the 13 indicators for natural disaster equals one, zero otherwise (W2 OC, W4 YC)	Yes
Household house damaged	shock_house_collapse	Is one if one of the 3 indicators for damage to house equals one, zero otherwise (W2 OC, W4 YC)	Yes
Community Characteristics (Only available for W2)			
Type of Community			
Urban vs. rural	typesite_w1	Rural vs. Urban dummy (sW1 OC/YC)	Yes

Population size of town	popsize	approximately, how many people (including children) live in this locality? (standardised, W2 OC/YC)	Yes
Time to reach district capital	timecap	Time to district capital by public transport in minutes (standardised, W2 OC/YC).	Yes
Type of jobs available			
Importance of agriculture	agriculture_jobs	Local land used for agriculture? Local land used for industry? Local land used for handicraft/small scale manufacturing? All are codes as three factors: not important, somewhat important, most important (W2 OC/YC)	Yes
Importance of factories	factory_jobs		Yes
Importance of craft jobs	craft_jobs		Yes
Type of education available			
Education available?	public_secondary_available	All are coded as three factors: - No, and not in a nearby locality - No, but there is in a nearby locality - Yes (W2 OC/YC)	Yes
	private_secondary_available		No
	lower_vocational_available		Yes
	public_higher_vocational_available		Yes
	private_higher_vocational_available		Yes

III: Code for Data Preparation

~~~~
---
title: "Part 1. Data Preperation"
author: "Xavier Friesen"
date: "`r Sys.Date()`"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```


#0. Manual:
Chapter 1-3 collect treatment, covariates and outcome data from the raw data.
Chapter 4 merges these together into one single dataframe, with missing data, for analysis
Chapter 5 provide descriptive statistics on that dataframe
Chapter 6 computes 5 imputed dataframes that will be used for the final analysis.

Each chapter works seperately, since it loads the data it needs from the enviroinment. Importantly,
each chapter should be run from top-to-bottom and only once to guarentee correct results.

#1. Covariates
The goal is to build a wide dataframe with one entry per child with all the available covariates.

The covariates are split out accross many different sub-dataframes and waves, and also have to be
cleaned and processed as much as possible.

The code below does not yet remove data errors. This is done later during the merging

## 1.1. Standard Functions
```{r}
read.stata <- function(file_path, country, rem.number = T) {
   library(haven)
   library(dplyr)
   library(stringr)

   # Read the data file
   data <- NULL
   data <- read_dta(file_path)

   # Process the data
   data <- data %>%
     rename_with(~tolower(.), everything()) %>%
     mutate(across(where(is.labelled), as_factor)) %>%
     mutate(across(where(is.factor), as.character)) %>%
     mutate(across(where(is.character), ~trimws(.x, which = "left"))) %>%
     mutate(across(where(is.factor), as.factor)) #remove starting spaces
~~~~

```
    if ("childcode" %in% names(data) && !is.null(country)) {
      data <- data %>% mutate(childcode = as.factor(childcode))
      country_code <- toupper(substr(country, 1, 2))
      data <- data %>%
        mutate(childcode = paste0(country_code, childcode))
    }

    if (rem.number == T) {
      # Remove numbers from column names
      names(data) <- gsub(pattern = "[0-9]", replacement = "", names(data))
    }

    return(data)
 }

adjust_childcode <- function(df, country) {
    temp_name <- paste0(toupper(country), "0") # Temporary name to check
    df %>% mutate(childcode = if_else(str_sub(childcode, 1, 3) == temp_name,
                                  paste0(toupper(substr(country, 1, 2)),
                                      str_sub(childcode, 4)), childcode))
}
```

1.2 Collecting raw data
We start with information from the constructed files -> which is preferred, since most of the coding is already done then.

```{r}
library(dplyr)
ind <- read.stata("unproc_data/constructed_data/Constructed Wave 1-5/stata/stata13/in_constructed.dta", country = "in", rem.number = F) %>%
  rename(childcode = childid) %>%
  adjust_childcode(country = "in") %>%
  mutate(across(where(is.numeric), as.character)) #for merging

pe <- read.stata("unproc_data/constructed_data/Constructed Wave 1-5/stata/stata13/pe_constructed.dta", country = "pe", rem.number = F)  %>%
  rename(childcode = childid,
        commid = placeid) %>%
  adjust_childcode(country = "pe") %>%
  mutate(across(where(is.numeric), as.character)) #for merging

vn <- read.stata("unproc_data/constructed_data/Constructed Wave 1-5/stata/stata13/vn_constructed.dta", country = "vn", rem.number = F) %>%
  rename(childcode = childid) %>%
  adjust_childcode(country = "vn") %>% rename(wi = wi_new) %>%
  mutate(across(where(is.numeric), as.character)) #for merging

et <- read.stata("unproc_data/constructed_data/Constructed Wave 1-5/stata/stata13/et_constructed.dta", country = "et", rem.number = F)%>%
  rename(childcode = childid) %>%
  adjust_childcode(country = "et") %>% rename(wi = wi_new) %>%
```

```
  mutate(across(where(is.numeric), as.character)) #for merging

merged_con <- bind_rows(ind, pe, vn, et) %>% filter(childcode != "childid") %>%
mutate(countrycode = substr(childcode, 1, 2)) %>% mutate(commid = if_else(commid == "", NA,
commid)) %>% mutate(
    across(
      starts_with("sh"),
      ~ case_when(
        . == "yes" ~ "1",
        . == "no" ~ "0",
        TRUE ~ .
      )
    )
  ) #chancing labelling of shock variables to 1 = Yes and 0 = no
remove(ind, pe, vn, et)
```

We will also need information from wave 2 and wave 4 questionaires

```{r}
childlevel <- c("childid", "dtopi", "bornbef", "bornaft", "primocc", "debt", "spyr11", "spyr12",
"spyr13", "spyr14", "famson", "famdtr", "csv1", "scuseful", "cfuturjb", "cambitn", "gradlike",
"expgrade", "expmar", "expedu", "expearn")

#Household questions about child
et_oc <- read.stata("unproc_data/raw_data/oc/w2_oc_et_childlevel12yro.dta", country = "et",
rem.number = F) %>% dplyr::select(any_of(childlevel)) %>% rename(childcode = childid, dint =
dtopi) %>% adjust_childcode(country = "et")

in_oc <- read.stata("unproc_data/raw_data/oc/w2_oc_in_childlevel12yro.dta", country = "in",
rem.number = F) %>% dplyr::select(any_of(childlevel)) %>% rename(childcode = childid, dint =
dtopi)  %>% adjust_childcode(country = "in")

vn_oc <- read.stata("unproc_data/raw_data/oc/w2_oc_vn_childlevel12yro.dta", country = "vn",
rem.number = F) %>% dplyr::select(any_of(childlevel)) %>% rename(childcode = childid, dint =
dtopi)  %>% adjust_childcode(country = "vn")

pe_oc <- read.stata("unproc_data/raw_data/oc/w2_oc_pe_childlevel12yro.dta", country = "pe",
rem.number = F) %>% dplyr::select(any_of(childlevel)) %>% rename(childcode = childid, dint =
dtopi)  %>% adjust_childcode(country = "pe")

merged_oc <- bind_rows(et_oc, in_oc, vn_oc, pe_oc)
remove(et_oc, in_oc, vn_oc, pe_oc, childlevel)

#Questions to child directly
child <- c("childid", "cdint", "missch", "chldwork", "friend", "lead", "incgame", "hardtalk",
"helpchld", "chfuture", "cgrdlike")

et_oc_ch <- read.stata("unproc_data/raw_data/oc/w2_oc_et_childquest12yr.dta", country = "et",
rem.number = F) %>% dplyr::select(any_of(child), starts_with("PS"), starts_with("AG"),
```

```
starts_with("TR")) %>% rename(childcode = childid, dint = cdint) %>% dplyr::select(-c(ps1, ps3, ps5, ps9, tr1)) %>% adjust_childcode(country = "et") #not available for YC

in_oc_ch <- read.stata("unproc_data/raw_data/oc/w2_oc_in_childquest12yr.dta", country = "in", rem.number = F) %>% dplyr::select(any_of(child), starts_with("PS"), starts_with("AG"), starts_with("TR")) %>% rename(childcode = childid, dint = cdint)%>% dplyr::select(-c(ps1, ps3, ps5, ps9, tr1)) %>% adjust_childcode(country = "in") #not available for YC

vn_oc_ch <- read.stata("unproc_data/raw_data/oc/w2_oc_vn_childquest12yr.dta", country = "vn", rem.number = F) %>% dplyr::select(any_of(child), starts_with("PS"), starts_with("AG"), starts_with("TR")) %>% rename(childcode = childid, dint = cdint) %>% dplyr::select(-c(ps1, ps3, ps5, ps9, tr1))  %>% adjust_childcode(country = "vn") #not available for YC

pe_oc_ch <- read.stata("unproc_data/raw_data/oc/w2_oc_pe_childquest12yr.dta", country = "pe", rem.number = F) %>% dplyr::select(any_of(child), starts_with("PS"), starts_with("AG"), starts_with("TR")) %>% rename(childcode = childid, dint = cdint) %>% dplyr::select(-c(ps1, ps3, ps5, ps9, tr1))  %>% adjust_childcode(country = "pe") #not available for YC

merged_oc_ch <- bind_rows(et_oc_ch, in_oc_ch, vn_oc_ch, pe_oc_ch)
remove(et_oc_ch, in_oc_ch, vn_oc_ch, pe_oc_ch, child)

## Unfortunately not all these data points are available in Wave 4 for YC, I mostly have to use wave 3, and very rarely wave 2 and 4. This means the children were generally younger when asked about the non-cognitive skills. The coding is also slightly different, and some non-cognitive tests are positively rather than negatively framed.
vars <- c("childid", "debtr3", "obtnlnr3", "spyrr311", "spyrr312", "spyrr313", "spyrr314", "r3csv1", "cfutjbr3", "grdlker3", "expgrdr3")

in_yc_w3 <- read.stata("unproc_data/raw_data/yc/w3_yc_in_householdleve.dta", country = "in", rem.number = F) %>% dplyr::select(any_of(vars)) %>% dplyr::select(-obtnlnr3) %>% rename(childcode = childid) %>% adjust_childcode(country = "in")

vn_yc_w3 <- read.stata("unproc_data/raw_data/yc/w3_yc_vn_householdleve.dta", country = "vn", rem.number = F) %>% dplyr::select(any_of(vars))  %>% rename(childcode = childid) %>% adjust_childcode(country = "vn")

pe_yc_w3 <- read.stata("unproc_data/raw_data/yc/w3_yc_pe_householdleve.dta", country = "pe", rem.number = F) %>% dplyr::select(any_of(vars)) %>% rename(debtr3 = obtnlnr3, school_fees = spyrr311, other_school_payments = spyrr312) %>% dplyr::select(-spyrr313)  %>% rename(childcode = childid)  %>% adjust_childcode(country = "pe") #different name for this variable in peru dataset, also be carefull spyrr is not seperated per gender

et_yc_w3 <- read.stata("unproc_data/raw_data/yc/w3_yc_et_householdleve.dta", country = "et", rem.number = F) %>% dplyr::select(any_of(vars))  %>% rename(childcode = childid) %>% adjust_childcode(country = "et")
merged_yc_w3 <- bind_rows(et_yc_w3, in_yc_w3, vn_yc_w3, pe_yc_w3) %>%
  rename(
      spyr11 = spyrr311,
      spyr12 = spyrr312,
      spyr13 = spyrr313,
      spyr14 = spyrr314,
```

```
      csv1 = r3csv1,
      cfuturjb = cfutjbr3,
      gradlike = grdlker3,
      expgrade = expgrdr3)
remove(et_yc_w3, in_yc_w3, vn_yc_w3, pe_yc_w3, vars)

vars <- c("childid", "bornbef", "primocc", "famson", "famdtr", "scuseful")
#primary occupation not available for Peru, these 5 variables are only asked in wave 2

in_yc_w2 <- read.stata("unproc_data/raw_data/yc/w2_yc_in_childlevel5yrol.dta", country = "in",
rem.number = F) %>% dplyr::select(any_of(vars))  %>% rename(childcode = childid) %>%
adjust_childcode(country = "in")
vn_yc_w2 <- read.stata("unproc_data/raw_data/yc/w2_yc_vn_childlevel5yrol.dta", country = "vn",
rem.number = F) %>% dplyr::select(any_of(vars))  %>% rename(childcode = childid) %>%
adjust_childcode(country = "vn")
pe_yc_w2 <- read.stata("unproc_data/raw_data/yc/w2_yc_pe_childlevel5yrol.dta", country = "pe",
rem.number = F) %>% dplyr::select(any_of(vars))   %>% rename(childcode = childid) %>%
adjust_childcode(country = "pe")
et_yc_w2 <- read.stata("unproc_data/raw_data/yc/w2_yc_et_childlevel5yrol.dta", country = "et",
rem.number = F) %>% dplyr::select(any_of(vars))  %>% rename(childcode = childid) %>%
adjust_childcode(country = "et")

merged_yc_w2 <- bind_rows(et_yc_w2, in_yc_w2, vn_yc_w2, pe_yc_w2)
remove(et_yc_w2, in_yc_w2, vn_yc_w2, pe_yc_w2, vars)

vars <- c("childcode", "cambtnr4", "getmarr4", "lveedcr4", "ernmnyr4")
in_yc_w4 <- read.stata("unproc_data/raw_data/yc/w4_yc_in_youngerhouseh.dta", country = "in",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% adjust_childcode(country = "in")
vn_yc_w4 <- read.stata("unproc_data/raw_data/yc/w4_yc_vn_youngerhous.dta", country = "vn",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% adjust_childcode(country = "vn")
pe_yc_w4 <- read.stata("unproc_data/raw_data/yc/w4_yc_pe_youngerhouse.dta", country = "pe",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% adjust_childcode(country = "pe")
et_yc_w4 <- read.stata("unproc_data/raw_data/yc/w4_yc_et_youngerhouseh.dta", country = "et",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% adjust_childcode(country = "et")
merged_yc_w4 <- bind_rows(et_yc_w4, in_yc_w4, vn_yc_w4, pe_yc_w4) %>% rename(
  cambitn = cambtnr4,
  expmar = getmarr4,
  expedu = lveedcr4,
  expearn = ernmnyr4
)

remove(et_yc_w4, in_yc_w4, vn_yc_w4, pe_yc_w4, vars)

#lastly childs born after
vars <- c("childcode", "chlbrnr4")
in_yc_w4_fam <- read.stata("unproc_data/raw_data/yc/w4_yc_in_householdroste.dta", country = "in",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% group_by(childcode) %>%  summarise(
    chlbrnr4 = if(all(is.na(chlbrnr4), na.rm = FALSE)) {
      NA_real_  # Return NA if all values are NA within the group
    } else {
      max(chlbrnr4, na.rm = TRUE)  # Compute the max, ignoring NAs
```

```
  },
  .groups = "drop") %>% adjust_childcode(country = "in") #for some reason dataframe returns many
rows per childcode with NA for chlbrnr4
vn_yc_w4_fam <- read.stata("unproc_data/raw_data/yc/w4_yc_vn_householdros.dta", country = "vn",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% group_by(childcode) %>%  summarise(
  chlbrnr4 = if(all(is.na(chlbrnr4), na.rm = FALSE)) {
    NA_real_  # Return NA if all values are NA within the group
  } else {
    max(chlbrnr4, na.rm = TRUE)  # Compute the max, ignoring NAs
  },
  .groups = "drop")  %>% adjust_childcode(country = "vn")
pe_yc_w4_fam <- read.stata("unproc_data/raw_data/yc/w4_yc_pe_householdrost.dta", country = "pe",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% group_by(childcode) %>%  summarise(
  chlbrnr4 = if(all(is.na(chlbrnr4), na.rm = FALSE)) {
    NA_real_  # Return NA if all values are NA within the group
  } else {
    max(chlbrnr4, na.rm = TRUE)  # Compute the max, ignoring NAs
  },
  .groups = "drop")  %>% adjust_childcode(country = "pe")
et_yc_w4_fam <- read.stata("unproc_data/raw_data/yc/w4_yc_et_householdroste.dta", country = "et",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% group_by(childcode) %>% summarise(
  chlbrnr4 = if(all(is.na(chlbrnr4), na.rm = FALSE)) {
    NA_real_  # Return NA if all values are NA within the group
  } else {
    max(chlbrnr4, na.rm = TRUE)  # Compute the max, ignoring NAs
  },
  .groups = "drop") %>% adjust_childcode(country = "et")
merged_yc_w4_fam <- bind_rows(et_yc_w4_fam, in_yc_w4_fam, vn_yc_w4_fam, pe_yc_w4_fam)
%>% rename(bornaft = chlbrnr4)

remove(et_yc_w4_fam, in_yc_w4_fam, vn_yc_w4_fam, pe_yc_w4_fam, vars)

#and now let's make one dataframe
merged_yc <-  merged_yc_w2 %>%
 full_join(merged_yc_w3, by = "childcode") %>%
 full_join(merged_yc_w4, by = "childcode") %>%
 full_join(merged_yc_w4_fam, by = "childcode") %>% rename(debt = debtr3)
remove(merged_yc_w2, merged_yc_w3, merged_yc_w4, merged_yc_w4_fam)

#and at the child level -> different datasets first wave 3
vars <- c("childid", "mssdscr3", "evrdayr3", "misschr3", "chwrkr3", "nmfrndr3", "leaderr3",
"incgmer3", "hrdtlkr3", "hlpchlr3", "ftrwrkr3", "ctrustr3", "cgovrgr3", "csfeowr3", "ctryhdr3",
"cpldecr3", "cftrwrr3", "cbrjobr3", "cnochcr3", "cashclr3", "cashshr3", "cembbkr3", "cwrunir3",
"cashwkr3")

in_yc_ch_w3 <- read.stata("unproc_data/raw_data/yc/w3_yc_in_childleve.dta", country = "in",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% rename(childcode = childid) %>%
adjust_childcode(country = "in")
vn_yc_ch_w3 <- read.stata("unproc_data/raw_data/yc/w3_yc_vn_childleve.dta", country = "vn",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% rename(childcode = childid, mssdscr3 =
misschr3) %>% adjust_childcode(country = "vn") #cgovrgr3 is missing
```

```
pe_yc_ch_w3 <- read.stata("unproc_data/raw_data/yc/w3_yc_pe_childleve.dta", country = "pe",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% rename(childcode = childid, mssdscr3 =
misschr3) %>% adjust_childcode(country = "pe")
et_yc_ch_w3 <- read.stata("unproc_data/raw_data/yc/w3_yc_et_childleve.dta", country = "et",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% rename(childcode = childid, mssdscr3 =
evrdayr3) %>% adjust_childcode(country = "et") %>% dplyr::select(-misschr3)

merged_yc_ch_w3 <- bind_rows(et_yc_ch_w3, in_yc_ch_w3, vn_yc_ch_w3, pe_yc_ch_w3) %>%
rename(
  missch = mssdscr3,
  chldwrk = chwrkr3,
  friend = nmfrndr3,
  lead = leaderr3,
  incgame = incgmer3,
  hardtalk = hrdtlkr3,
  helpchld = hlpchlr3,
  chfuture = ftrwrkr3,
  tr2 = ctrustr3,
  tr3 = cgovrgr3,
  tr4 = csfeowr3,
  ag1 = ctryhdr3,
  ag2 = cpldecr3,
  ag3 = cftrwrr3,
  ag4 = cbrjobr3,
  ag5 = cnochcr3,
  ps2 = cashclr3,
  ps7 = cashshr3,
  ps4 = cembbkr3,
  ps8 = cwrunir3,
  ps6 = cashwkr3
)

remove(et_yc_ch_w3, in_yc_ch_w3, vn_yc_ch_w3, pe_yc_ch_w3, vars)

#one more from wave 4
vars <- c("childcode", "cldstdr4")
in_yc_ch_w4 <- read.stata("unproc_data/raw_data/yc/w4_yc_in_youngerch.dta", country = "in",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% adjust_childcode(country = "in")
vn_yc_ch_w4 <- read.stata("unproc_data/raw_data/yc/w4_yc_vn_younger.dta", country = "vn",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% adjust_childcode(country = "vn") #cgovrgr3
is missing
pe_yc_ch_w4 <- read.stata("unproc_data/raw_data/yc/w4_yc_pe_youngerch.dta", country = "pe",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% adjust_childcode(country = "pe")
et_yc_ch_w4 <- read.stata("unproc_data/raw_data/yc/w4_yc_et_youngerch.dta", country = "et",
rem.number = F) %>% dplyr::select(any_of(vars)) %>% adjust_childcode(country = "et")

merged_yc_ch_w4 <- bind_rows(et_yc_ch_w4, in_yc_ch_w4, vn_yc_ch_w4, pe_yc_ch_w4)
remove(et_yc_ch_w4, in_yc_ch_w4, vn_yc_ch_w4, pe_yc_ch_w4, vars)

merged_yc_ch <-  merged_yc_ch_w3 %>%
  full_join(merged_yc_ch_w4, by = "childcode") %>% rename(chldwork = chldwrk)
```

```
remove(merged_yc_ch_w3, merged_yc_ch_w4)

#Removing NA's
#the above has some complicated NA-codes, which should be changed into NA
library(dplyr)
library(dplyr)

merged_oc <- merged_oc %>%
  # Convert character columns to lowercase except 'childcode'
  mutate(across(
    where(is.character) & !starts_with("childcode"),
    ~ tolower(.)
  )) %>%
  # Replace specific values with NA in character columns except 'childcode'
  mutate(across(
    where(is.character) & !starts_with("childcode"),
    ~ if_else(. %in% c("missing", "nk", "n/a", "0", "refused to answer"), NA_character_, .)
  )) %>% #different na placeholders
  # Change negative values in 'spyr' columns to NA
  mutate(across(starts_with("spyr"), ~ if_else(. < 0, NA_real_, .))) %>%
  # Change 77 values in 'born' columns to NA
  mutate(across(starts_with("born"), ~ if_else(. ==77, NA_real_, .)))


merged_yc <- merged_yc %>%
  # Convert character columns to lowercase except 'childcode'
  mutate(across(
    where(is.character) & !starts_with("childcode"),
    ~ tolower(.)
  )) %>%
  # Replace specific values with NA in character columns except 'childcode'
  mutate(across(
    where(is.character) & !starts_with("childcode"),
    ~ if_else(. %in% c("missing", "nk", "n/a", "0", "79", "refused to answer"), NA_character_, .)
#different na placeholders
  )) %>%
  # Change negative values in numeric columns to NA
  mutate(across(where(is.numeric), ~ if_else(. < 0, NA_real_, .)))

merged_oc_ch <- merged_oc_ch %>%
  # Convert character columns to lowercase except 'childcode'
  mutate(across(
    where(is.character) & !starts_with("childcode"),
    ~ tolower(.)
  )) %>% dplyr::select(-c(agestop, transsch)) %>% #accidentally loaded variables
  # Replace specific values with NA in character columns except 'childcode'
  mutate(across(
    where(is.character) & !starts_with("childcode"),
    ~ if_else(. %in% c("missing", "nk", "n/a", "0", "refused to answer"), NA_character_, .)
  )) %>% #different NA'values
  # Change negative values in friend columns to NA
```

~~~
  mutate(friend = if_else(friend < 0, NA_real_, friend))


merged_yc_ch <- merged_yc_ch %>% mutate(childcode = toupper(childcode)) %>%
  # Convert character columns to lowercase except 'childcode'
  mutate(across(
    where(is.character) & !starts_with("childcode"),
    ~ tolower(.)
  )) %>% rename(cgrdlike = cldstdr4) %>%
  # Replace specific values with NA in character columns except 'childcode'
  mutate(across(
    where(is.character) & !starts_with("childcode"),
    ~ if_else(. %in% c("missing", "nk", "n/a", "0", "refused to answer"), NA_character_, .)
  )) %>% #different NA placeholders
  # Change negative values in friend columns to NA
  mutate(friend = case_when(
    as.numeric(friend) < 0 ~ NA,
    friend == 88 ~ NA, #other error
    friend == 99 ~ NA, #other error
    TRUE ~ friend #otherwise stay the same
    ),
    friend = as.numeric(friend))  %>%
  group_by(childcode) %>%
  filter(!(n() > 1 & if_all(everything(), is.na) & !is.na(childcode))) %>%
  ungroup()


```


##1.3 Individual Characteristics
child demographics
```{r}
library(dplyr)
library(lubridate)  # For handling dates and times

#Start with the basic dataframe which remains constant over time.By design nobody was not attending
round 1, so we obtain this information from wave 1, constructed data.

#childcode is id, countrycode signals country, yc is cohort, panel12345 shows if data available for
wave 1-5
#chsex = gender, chethnic = ethnic group, chldrel = religion
#commid, region and typesite characterize the type of place the child is living in during w1
#Year of birth is calculated by substracting age in months at wave 1 from date of interview
covariates <- merged_con %>%
  filter(round == 1) %>%
  mutate(
    dint = as.Date(dint),  # Convert 'dint' to a Date object
    agemon = as.numeric(agemon),  # Ensure 'agemon' is numeric
    date_of_birth = dint %m-% months(agemon),  # Subtract age in months from date to get date of
birth
    year_of_birth = year(date_of_birth)  # Extract the year from the date of birth
~~~

~~~
  ) %>%
  dplyr::select(childcode, countrycode, yc, panel12345, chsex, year_of_birth, chethnic, chldrel,
commid, region, typesite) %>% rename(chrel = chldrel, commid_w1 = commid, region_w1 = region,
typesite_w1 = typesite)


#Deceased == 1 if deceased during wave 1 to wave 5, 0 otherwise
deceased <- merged_con %>%
  group_by(childcode) %>%
  # Rename 'deceased' column to 'deceased2' right at the data input step to avoid confusion
  rename(deceased2 = deceased) %>%
  # Create a new 'deceased' column: 1 if any 'deceased2' is 'yes' in the group, 0 otherwise
  mutate(deceased = as.integer("yes" %in% deceased2)) %>%
  # Filter for specific round after determining deceased status
  filter(round == 1) %>%
  # Make sure to get unique entries
  distinct(childcode, deceased) %>%
  ungroup()
remove(deceased)

# Moved between wave 1 and wave 3  #pre-secondary
# 0 if commid = the same, 1 if differs
moved <- merged_con %>%
  # Filter to keep only records from round 1 or round 3
  filter(round == 1 | round == 3) %>%
  group_by(childcode) %>%
  # Modify the summarisation logic to account for NA values
  summarise(
    moved = if (anyNA(commid)) {
      NA_integer_  # Return NA if any commid is NA
    } else {
      ifelse(n_distinct(commid) > 1, 1, 0)  # Check if there's more than one unique commid
    },
    .groups = "drop"  # Drop grouping for final data frame
  )
covariates <- covariates %>% left_join(moved, by = "childcode")
remove(moved)
```
~~~

child health

~~~
```{r}
#Early childhood health around age 8 (wave 1 for OC, wave 3 for YC)
### 3 indicators:  weight-for-age z-score, height-for-age z-score, BMI-for-age z-score
### fwfa, fhfa and fbfa are flags, indicating very likely errors based on WHO-criteria, so if those are
one we replace the score with NA
library(dplyr)

# Define the  variables
vars <- c("zwfa", "zhfa", "zbfa", "fwfa", "fhfa", "fbfa")

# Filter and process data for the Older cohort
~~~

```
older_cohort <- merged_con %>%
  filter(yc == "Older cohort", round == 1) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 1, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Filter and process data for the Younger cohort
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 3) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 3, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Combine and process data for both cohorts
combined_data <- bind_rows(older_cohort, younger_cohort) %>% mutate(
  zweight_8 = case_when(
    fwfa == 1 ~ NA,
    TRUE ~ zwfa),
  zheight_8 = case_when(
    fhfa == 1 ~ NA,
    TRUE ~ zhfa),
  zbmi_8 = case_when(
    fbfa == 1 ~ NA,
    TRUE ~ zbfa),
) %>% dplyr::select(childcode, zweight_8, zheight_8, zbmi_8)

# Join the combined data back to the original dataset
covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(combined_data, younger_cohort, older_cohort, vars)

#Early adolescent health betwen 8 and 13
### Occurence of serious health illness dummy (available w2 and w4 and w5)
### Long-term health problem dummy (w2 oc and w5 yc)
### Permanent disablity dummy  (available w4 only, but it's not going to be affected by treatment, so can also be used as pre-observed characterisc) (Requires Note in-text!!)
### Child's subjective wellbeing (w2 oc, w4 yc)

# Serious health illness dummy
# oc 1 if chilness == 1 in round 2, yc 1 if chilness == 1 in round 4 or 5
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 4) %>%
  group_by(childcode) %>%
  summarize(
    chillness_8_13 = if (all(is.na(chillness))) NA_integer_
            else if (any(chillness == "yes", na.rm = TRUE)) 1
            else 0,
    .groups = 'drop'
  )

older_cohort <- merged_con %>%
  filter(yc == "Older cohort", round == 2) %>%
  group_by(childcode) %>%
  summarize(
```

```
    chillness_8_13 = if (all(is.na(chillness))) NA_integer_
              else if (any(chillness == "yes", na.rm = TRUE)) 1
              else 0,
    .groups = 'drop'
  )

# Combine results and create a lookup table
chillness <- bind_rows(younger_cohort, older_cohort)
covariates <- covariates %>% left_join(chillness, by = "childcode")
remove(chillness, younger_cohort, older_cohort)

# long term health problem
# oc 1 if chhprob == 1 in round 2, yc 1 if chhprob == 1 in round 5
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 5) %>%
  group_by(childcode) %>%
  summarize(
    chhprob = if (all(is.na(chhprob))) NA_integer_
    else if (any(chhprob == "yes", na.rm = TRUE)) 1
    else 0,
    .groups = 'drop'
  )

older_cohort <- merged_con %>%
  filter(yc == "Older cohort", round == 2) %>%
  group_by(childcode) %>%
  summarize(
    chhprob = if (all(is.na(chhprob))) NA_integer_
    else if (any(chhprob == "yes", na.rm = TRUE)) 1
    else 0,
    .groups = 'drop'
  )

# Combine results and create a lookup table
chhprob <- bind_rows(younger_cohort, older_cohort) %>% rename(long_term_health_problem =
chhprob)
covariates <- covariates %>% left_join(chhprob, by = "childcode")
remove(chhprob, younger_cohort, older_cohort)

#disablities
disablity <- merged_con %>% filter(round == 4) %>% dplyr::select(childcode, chdisability,
chdisscale) %>% rename(chdis_scale = chdisscale) %>% mutate(chdisability = as.factor(chdisability))
covariates <- covariates %>% left_join(disablity, by = "childcode")
remove(disablity)

# longer term health problem
# oc is cladder in round 2, yc is cladder in round 5
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 4)

older_cohort <- merged_con %>%
```

```
  filter(yc == "Older cohort", round == 2)

# Combine results and create a lookup table
cladder <- bind_rows(younger_cohort, older_cohort) %>% dplyr::select(childcode, cladder) %>%
rename(subjective_health_13 = cladder)
covariates <- covariates %>% left_join(cladder, by = "childcode")

remove(cladder, younger_cohort, older_cohort)

```

time-use
We will use wave 2 for the older cohort and wave 4 for the younger cohort to assure similar ages when time usage is measured
```{r}
library(dplyr)
# Define the  variables
vars <- c("hsleep", "hcare", "hchore", "htask", "hwork", "hschool", "hstudy", "hplay")

# Filter and process data for the Older cohort
older_cohort <- merged_con %>%
  filter(yc == "Older cohort", round == 2) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 2, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Filter and process data for the Younger cohort
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 4) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 4, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Combine the processed data for both cohorts
combined_data <- bind_rows(older_cohort, younger_cohort)

# Join the combined data back to the original dataset
covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(combined_data, younger_cohort, older_cohort, vars)
```

cognitive skills
This requires different dataframes

```{r}
library(tidyr)
adjust_childcode_allcountries <- function(df) {
    # Generate patterns to match based on the provided countries list
    countries <- c("IN", "PE", "VN", "ET")
    patterns <- paste0("^", toupper(countries), "0")

    df %>% mutate(
      childcode = if_else(
```

```
          str_detect(childcode, paste(patterns, collapse = "|")), # Check if starts with any of "IN0",
"PE0", etc.
          paste0(
            str_sub(childcode, 1, 2),
            str_sub(childcode, 4) #removes third character
          ),
          childcode
        )
    )
}

math <- read.stata("unproc_data/constructed_data/Constructed Wave 1-
5/stata/stata13/all_countries_math_irt_scores.dta", rem.number = F) %>%
  rename(childcode = childid) %>%
  adjust_childcode_allcountries() %>% dplyr::select(childcode, round, yc, math) %>%
rename(math_score = math)

reading <- read.stata("unproc_data/constructed_data/Constructed Wave 1-
5/stata/stata13/all_countries_reading_irt_scores.dta", rem.number = F) %>%
  rename(childcode = childid) %>%
  adjust_childcode_allcountries() %>% dplyr::select(childcode, round, yc, read_score)

#not avaible for OC (only w4), YC w4

ppvt <- read.stata("unproc_data/constructed_data/Constructed Wave 1-
5/stata/stata13/all_countries_ppvt_irt_scores.dta", rem.number = F) %>%
  rename(childcode = childid) %>%
  adjust_childcode_allcountries() %>% dplyr::select(childcode, round, yc, ppvt_score)

#available OC w2 and w3, YC w2, w3, 4, 5

#math
#1st indicator Score in Wave 2 for OC, wave 4 for YC
#2nd indicator Improvement in between Wave 2 and 3 for OC and Wave 3 and 4 for YC (latter being
less ideal, but there is no data for w1, and w5 cannot be used since it's post secondary enrollment)


# Filter and organize data for the older cohort for Wave 2 and 3
older_cohort <- math %>%
  filter(yc == "Older" & (round == 2 | round == 3)) %>% pivot_wider(
    names_from = round,
    values_from = math_score,
    names_prefix = "score_round_"
  ) %>% mutate(
    math_score_13 = score_round_2,
    math_score_improvement = score_round_3 - score_round_2
  ) %>% dplyr::select(childcode, math_score_13, math_score_improvement)

# Filter and organize data for the younger cohort for Wave 3 and 4
younger_cohort <- math %>%
  filter(yc == "Younger" & (round == 3 | round == 4))  %>% pivot_wider(
```

```
    names_from = round,
    values_from = math_score,
    names_prefix = "score_round_"
  ) %>% mutate(
    math_score_13 = score_round_4,
    math_score_improvement = score_round_4 - score_round_3
  ) %>% dplyr::select(childcode, math_score_13, math_score_improvement)

combined_data <- bind_rows(older_cohort, younger_cohort)

# Join the combined data back to the original dataset
covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(younger_cohort, older_cohort, combined_data)

## Reading
# Filter and organize data for the younger cohort for Wave 3 and 4
younger_cohort <- reading %>% filter(round == 4 & yc == "Younger") %>% rename(
    read_score_13 = read_score
  ) %>% dplyr::select(childcode, read_score_13)

# Join the combined data back to the original dataset
covariates <- covariates %>% left_join(younger_cohort, by = "childcode")
remove(younger_cohort)

## PPVT
#1st indicator Score in Wave 2 for OC, wave 4 for YC
#2nd indicator Improvement in between Wave 2 and 3 for OC and Wave 3 and 4 for YC (latter being less ideal, but there is no data for w1, and w5 cannot be used since it's post secondary enrollment)


# Filter and organize data for the older cohort for Wave 2 and 3
older_cohort <- ppvt %>%
  filter(yc == "Older" & (round == 2 | round == 3)) %>% pivot_wider(
    names_from = round,
    values_from = ppvt_score,
    names_prefix = "score_round_"
  ) %>% mutate(
    ppvt_score_13 = score_round_2,
    ppvt_score_improvement = score_round_3 - score_round_2
  ) %>% dplyr::select(childcode, ppvt_score_13, ppvt_score_improvement)

# Filter and organize data for the younger cohort for Wave 3 and 4
younger_cohort <- ppvt %>%
  filter(yc == "Younger" & (round == 3 | round == 4))  %>% pivot_wider(
    names_from = round,
    values_from = ppvt_score,
    names_prefix = "score_round_"
  ) %>% mutate(
    ppvt_score_13 = score_round_4,
    ppvt_score_improvement = score_round_4 - score_round_3
  ) %>% dplyr::select(childcode, ppvt_score_13, ppvt_score_improvement)
```

~~~
combined_data <- bind_rows(older_cohort, younger_cohort)

# Join the combined data back to the original dataset
covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(younger_cohort, older_cohort, combined_data)
remove(reading, ppvt, math)

#assessed level of writing and reading by research assistant (W2 OC, W4 YC)
vars <- c("levlwrit", "levlread", "literate")

# Filter and process data for the Older cohort
older_cohort <- merged_con %>%
  filter(yc == "Older cohort", round == 2) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 2, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Filter and process data for the Younger cohort
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 4) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 4, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Combine the processed data for both cohorts
combined_data <- bind_rows(older_cohort, younger_cohort)

# Join the combined data back to the original dataset
covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(combined_data, younger_cohort, older_cohort, vars)

```
~~~

Non-cognitive skills and Personality
Personality some single-questions

~~~
```{r}
#Number of friends spoken during the last 7-days -> indicator for extravertness
#Numeric: no manipulation required
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, friend), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, friend), by = "childcode") %>%
  mutate(noncog_friend = coalesce(friend.x, friend.y)) %>%
  dplyr::select(-friend.x, -friend.y)

#Talking with others: Do you find it hard to talk to others in your class?
#no manipulation required
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, hardtalk), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, hardtalk), by = "childcode") %>%
  mutate(noncog_hardtalk = coalesce(hardtalk.x, hardtalk.y)) %>%
  dplyr::select(-hardtalk.x, -hardtalk.y)
~~~

```
#incgame: do friends include you in your games
#no data manipulationo necessary
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, incgame), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, incgame), by = "childcode") %>%
  mutate(noncog_incgame = coalesce(incgame.x, incgame.y)) %>%
  dplyr::select(-incgame.x, -incgame.y)

#leadership skills: do friends perceive you as a leader
#oc is 3 factors, in yc yes should be always, and no should be never
merged_yc_ch <- merged_yc_ch %>% mutate(
  lead = case_when(
    lead == "no" ~ "never",
    lead == "yes" ~ "always",
    TRUE ~ lead
  ))

#merge
covariates <-  covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, lead), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, lead), by = "childcode") %>%
  mutate(noncog_lead = coalesce(lead.x, lead.y)) %>%
  dplyr::select(-lead.x, -lead.y)

#HELPCHLD
#no data mnaipulation required
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, helpchld), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, helpchld), by = "childcode") %>%
  mutate(noncog_helpchld = coalesce(helpchld.x, helpchld.y)) %>%
  dplyr::select(-helpchld.x, -helpchld.y)

```

Non cognitive skills
several indexes -> but be careful several questions were positively phrased for yc and negatively for oc
indexes are re-scaled to 0-100 with 100 being high.
They are calculated by average, not counting NAs

```{r}
replace_categorical_with_numerical <- function(x) {
  x <- as.character(x)  # Convert all inputs to character
  case_when(
    x == "strongly agree" ~ 4,
    x == "agree" ~ 3,
    x == "more or less" ~ 2,
    x == "disagree" ~ 1,
    x == "strongly disagree" ~ 0,
    x %in% c("0", "1", "2", "3", "4") ~ as.numeric(x),
    TRUE ~ NA_real_
```

```
  ) }

replace_categorical_with_numerical_negatively <- function(x) {
  x <- as.character(x)  # Convert all inputs to character
  case_when(
    x == "strongly agree" ~ 0,
    x == "agree" ~ 1,
    x == "more or less" ~ 2,
    x == "disagree" ~ 3,
    x == "strongly disagree" ~ 4,
    x %in% c("0", "1", "2", "3", "4") ~ as.numeric(x),
    TRUE ~ NA_real_
  ) #for negatively phrased
} #function to change categories into numerics to calculate index for negatively phrased questions

#Trust
merged_oc_ch <- merged_oc_ch %>%
  mutate(across(starts_with("tr"), ~ if_else(. == "no", "strongly disagree", .)),
         across(starts_with("tr"), ~ if_else(. == "yes", "strongly agree", .)),
         across(starts_with("tr"), ~ if_else(. == "3", "more or less", .))) #correcting different codes

merged_oc_ch <- merged_oc_ch %>%
  mutate(across(c(tr2, tr3, tr4), replace_categorical_with_numerical)) %>%
  rowwise() %>%
  mutate(
    noncog_trust = ifelse(all(is.na(c(tr2, tr3, tr4))), NA_real_, mean(c(tr2, tr3, tr4), na.rm = TRUE)),
    noncog_trust = noncog_trust/4*100) %>% #standardizing
  ungroup()

merged_yc_ch <- merged_yc_ch %>%
  mutate(across(c(tr2, tr3, tr4), replace_categorical_with_numerical)) %>%
  rowwise() %>%
  mutate(
    noncog_trust = ifelse(all(is.na(c(tr2, tr3, tr4))), NA_real_, mean(c(tr2, tr3, tr4), na.rm = TRUE)),
    noncog_trust = noncog_trust/4*100) %>% #standardizing
  ungroup()

#merge
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, noncog_trust), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, noncog_trust), by = "childcode") %>%
  mutate(noncog_trust= coalesce(noncog_trust.x, noncog_trust.y)) %>%
  dplyr::select(-noncog_trust.x, -noncog_trust.y)

#self efficicacy, question 2 and 5 negatively phrased
merged_oc_ch <- merged_oc_ch %>%
  mutate(across(starts_with("ag"), ~ if_else(. == "no", "strongly disagree", .)),
         across(starts_with("ag"), ~ if_else(. == "yes", "strongly agree", .))) #correcting different codes

merged_oc_ch <- merged_oc_ch %>%
  mutate(across(c(ag1, ag3, ag4), replace_categorical_with_numerical),
```

```
      across(c(ag2, ag5), replace_categorical_with_numerical_negatively)) %>%
  rowwise() %>%
  mutate(
    noncog_selfefficiacy = ifelse(all(is.na(c(ag1, ag2, ag3, ag4, ag5))), NA_real_, mean(c(ag1, ag2,
ag3, ag4, ag5), na.rm = TRUE)),
    noncog_selfefficiacy = noncog_selfefficiacy/4*100) %>% #standardizing
  ungroup()

merged_yc_ch <- merged_yc_ch %>%
  mutate(across(c(ag1, ag3, ag4), replace_categorical_with_numerical),
      across(c(ag2, ag5), replace_categorical_with_numerical_negatively)) %>%
  rowwise() %>%
  mutate(
    noncog_selfefficiacy = ifelse(all(is.na(c(ag1, ag2, ag3, ag4, ag5))), NA_real_, mean(c(ag1, ag2,
ag3, ag4, ag5), na.rm = TRUE)),
    noncog_selfefficiacy = noncog_selfefficiacy/4*100) %>% #standardizing
  ungroup()

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, noncog_selfefficiacy), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, noncog_selfefficiacy), by = "childcode") %>%
  mutate(noncog_selfefficiacy= coalesce(noncog_selfefficiacy.x, noncog_selfefficiacy.y)) %>%
  dplyr::select(-noncog_selfefficiacy.x, -noncog_selfefficiacy.y)

#self-esteem
# for OC all negatively framed
# for YC only 4 negatively framed
merged_oc_ch <- merged_oc_ch %>%
  mutate(across(starts_with("ps"), ~ if_else(. == "no", "strongly disagree", .)),
      across(starts_with("ps"), ~ if_else(. == "yes", "strongly agree", .)),
      across(starts_with("ps"), ~ if_else(. == "5", "strongly agree", .))) #correcting different codes

merged_oc_ch <- merged_oc_ch %>%
  mutate(across(c(ps2, ps4, ps6, ps7, ps8), replace_categorical_with_numerical_negatively)) %>%
  rowwise() %>%
  mutate(
    noncog_selfesteem = ifelse(all(is.na(c(ps2, ps4, ps6, ps7, ps8))), NA_real_, mean(c(ps2, ps4, ps6,
ps7, ps8), na.rm = TRUE)),
    noncog_selfesteem = noncog_selfesteem/4*100) %>% #standardizing
  ungroup()

merged_yc_ch <- merged_yc_ch %>%
  mutate(across(c(ps2, ps6, ps7, ps8), replace_categorical_with_numerical),
      across(c(ps4), replace_categorical_with_numerical_negatively)) %>%
  rowwise() %>%
  mutate(
    noncog_selfesteem = ifelse(all(is.na(c(ps2, ps4, ps6, ps7, ps8))), NA_real_, mean(c(ps2, ps4, ps6,
ps7, ps8), na.rm = TRUE)),
    noncog_selfesteem = noncog_selfesteem/4*100) %>% #standardizing
  ungroup()
```

````
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, noncog_selfesteem), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, noncog_selfesteem), by = "childcode") %>%
  mutate(noncog_selfesteem= coalesce(noncog_selfesteem.x, noncog_selfesteem.y)) %>%
  dplyr::select(-noncog_selfesteem.x, -noncog_selfesteem.y)
```

early employment and education
```{r}
#Did child work while still going to school: work defined as any activity that generates monetary
income
#No manipulation necessary

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, chldwork), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, chldwork), by = "childcode") %>%
  mutate(chldwork_during_school = coalesce(chldwork.x, chldwork.y)) %>%
  dplyr::select(-chldwork.x, -chldwork.y)

##Missed school for more than one week during the last year

#No manipulation required
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, missch), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, missch), by = "childcode") %>%
  mutate(missed_school = coalesce(missch.x, missch.y)) %>%
  dplyr::select(-missch.x, -missch.y)

#expected grade
#We summarize the potential answers in categories, sometimes also based on countries if educational
systems differ

#unfortunately, the labelling of merged_yc was done much worse than merged_oc
merged_yc_ch <- merged_yc_ch %>% mutate(
  countrycode = substr(childcode, 1, 2),
  expected_grade = case_when(
    is.na(cgrdlike) ~ NA,
    cgrdlike %in% c("other", "other  (specify)", "other (specify)", "otro (especificar)", "ns") ~
NA_character_,
    cgrdlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, 9, "none", "grade 1", "grade 2", "grade 3", "grade 4", "grade
5", "grade 6", "grade 7", "grade 8", "grade 9", "adult literacy", "religious education", "ninguno", "nqc",
"rade 7", "programa de alfabetizaci") & countrycode %in% c("ET", "IN") ~ "(below) primary",
    cgrdlike %in% c(10, 11, "grade 10", "grade 11") & countrycode %in% c("ET", "IN") ~ "lower-
secondary",
    cgrdlike %in% c(12, "grade 12", "centro t") & countrycode %in% c("ET", "IN") ~ "upper-
secondary",
    cgrdlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, "none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5",
"grade 6", "grade 7", "grade 8", "adult literacy", "religious education", "ninguno", "nqc", "rade 7",
"programa de alfabetizaci") & countrycode == "VN" ~ "(below) primary",
    cgrdlike %in% c(9, 10, 11, "grade 9", "grade 10", "grade 11") & countrycode == "VN" ~ "lower-
secondary",
````

```
    cgrdlike %in% c(12, "grade 12", "centro t") & countrycode == "VN" ~ "upper-secondary",

    cgrdlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, "none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5",
"grade 6", "grade 7", "grade 8", "adult literacy", "religious education", "ninguno", "nqc", "rade 7",
"programa de alfabetizaci") & countrycode == "PE" ~ "(below) primary",
    cgrdlike %in% c(9, 10, 11,"grade 9", "grade 10", "grade 11") & countrycode == "PE" ~ "lower-
secondary",
    cgrdlike %in% c(12, "grade 12", "centro t") & countrycode == "PE" ~ "upper-secondary",

    cgrdlike %in% c("post-secondary"," post-secondary, vocational", "sup. no univ. completa  (t",
"vocational")~ "technical/vocational college",
    cgrdlike %in% c("university", "univ. completa  (incluye escuela de oficiales)", "postgrado (maestr",
"post-graduate (e.g. masters, phd)", "degree (graduate)", "degree(graduate)") ~ "university/college",
    TRUE ~ NA
  )
)

merged_oc_ch <- merged_oc_ch %>% mutate(
  countrycode = substr(childcode, 1, 2),
  expected_grade = case_when(
    is.na(cgrdlike) ~ NA,
    cgrdlike == "other" ~ NA_character_,
    cgrdlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, 9, "none") & countrycode %in% c("ET", "IN") ~ "(below)
primary",
    cgrdlike %in% c(10, 11) & countrycode %in% c("ET", "IN") ~ "lower-secondary",
    cgrdlike %in% c(12) & countrycode %in% c("ET", "IN") ~ "upper-secondary",
    cgrdlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, "none") & countrycode == "VN" ~ "(below) primary",
    cgrdlike %in% c(9, 10, 11) & countrycode == "VN" ~ "lower-secondary",
    cgrdlike %in% c(12) & countrycode == "VN" ~ "upper-secondary",

    cgrdlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, "none") & countrycode == "PE" ~ "(below) primary",
    cgrdlike %in% c(9, 10, 11) & countrycode == "PE" ~ "lower-secondary",
    cgrdlike %in% c(12) & countrycode == "PE" ~ "upper-secondary",

    cgrdlike %in% c("complete technical college","incomplete technical college", "post-secondary,
vocational")~ "technical/vocational college",
    cgrdlike %in% c("incomplete university", "complete university", "university/college") ~
"university/college",
    TRUE ~ NA
  )
)

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, expected_grade), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, expected_grade), by = "childcode") %>%
  mutate(expected_grade = coalesce(expected_grade.x, expected_grade.y)) %>%
  dplyr::select(-expected_grade.x, -expected_grade.y)

```

Childhood job expectation

```{r}
merged_yc_ch <- merged_yc_ch %>% mutate(chfuture = case_when(
  chfuture %in% c("43", "44") ~ "other", #unlabelled
  TRUE ~ chfuture
))

merged_oc_ch <- merged_oc_ch %>% mutate(chfuture = case_when(
  chfuture %in% c("43", "44") ~ "other", #unlabelled
  TRUE ~ chfuture
))

merged_yc_ch <- merged_yc_ch %>%
   mutate(dreamjob_sector = case_when(
    is.na(chfuture) == T ~ NA_character_,
    chfuture %in% c("doctor", "dentist", "nurse", "vet", "veterinary") ~ "Healthcare",
    chfuture %in% c("teacher", "lecturer", "scientist", "student/university student", "university
student/other form of further education") ~ "Education and Research",
    chfuture %in% c("administrative assistant/secretary",  "district collector", "religious
leader/priest/sheikh", "accountant", "lawyer", "management", "manager/management", "computer
operator", "trader/businessman/woman") ~ "Services and Management",
    chfuture %in% c("civil servant", "politician", "president of the country", "president/leader of
country", "soldier", "policeman/woman", "conductor") ~ "Public Administration and Services",
    chfuture %in% c("construction worker", "cook", "driver", "engineer", "fireman/woman",
"fisherman", "fisherman/woman", "labourer", "mason", "mechanic", "tailor", "taxi driver", "domestic
worker", "farmer", "painter/decorator", "traditional occupation", "pilot", "market trader/shop
assistant") ~ "Skilled Trades and Manual Labor",
    chfuture %in% c("artist", "actor/actress", "singer", "sportsman/woman", "full-time
parent/housewife", "fulltime parent/housewife", "other") ~ "other",
    TRUE ~ "not found"
  )) %>% mutate(vocational_dreamjob_dummy = case_when(
       is.na(chfuture) == T ~ NA,
       chfuture %in% c("construction worker", "cook", "driver", "engineer", "fireman/woman",
"fisherman", "fisherman/woman", "labourer",
 "mason", "mechanic", "tailor", "taxi driver", "domestic worker",
"farmer", "painter/decorator", "traditional occupation",
"market trader/shop assistant", "painter", "trader/businessman/woman",
 "trader", "market trader") ~ "Yes",
       chfuture == "other" ~ NA,
       TRUE ~ "No"),
       academic_dreamjob_dummy = case_when(
       is.na(chfuture) == T ~ NA,
       chfuture %in% c("doctor", "dentist", "nurse", "vet", "veterinary",   "teacher", "lecturer",
"scientist", "university student",
  "university student/other form of further education", "accountant", "lawyer", "management",
"manager/management", "computer operator", "civil servant", "politician", "president of the country",
"president/leader of country") ~ "Yes",
       chfuture == "other" ~ NA,
       TRUE ~ "No"),
)

merged_oc_ch <- merged_oc_ch %>%
```

```
  mutate(dreamjob_sector = case_when(
   is.na(chfuture) == T ~ NA_character_,
   chfuture %in% c("doctor", "dentist", "nurse", "vet", "veterinary") ~ "Healthcare",
   chfuture %in% c("teacher", "lecturer", "scientist", "student/university student", "university
student/other form of further education", "university student") ~ "Education and Research",
   chfuture %in% c("administrative assistant/secretary",  "district collector", "religious
leader/priest/sheikh", "accountant", "lawyer", "management", "manager/management", "computer
operator", "administrative assistent/secretary", "trader/businessman/woman", "trader", "religious
leader", "market trader") ~ "Services and Management",
   chfuture %in% c("civil servant", "politician", "president of the country", "president/leader of
country", "soldier", "policeman/woman", "conductor", "policeman", "fireman") ~ "Public
Administration and Services",
   chfuture %in% c("construction worker", "cook", "driver", "engineer", "fireman/woman",
"fisherman", "fisherman/woman", "labourer", "mason", "mechanic", "tailor", "taxi driver", "domestic
worker", "farmer", "painter/decorator", "traditional occupation", "pilot", "market trader/shop
assistant", "painter") ~ "Skilled Trades and Manual Labor",
   chfuture %in% c("artist", "actor/actress", "singer", "sportsman/woman", "full-time
parent/housewife", "fulltime parent/housewife", "other", "sportsman", "actor") ~ "other",
   TRUE ~ "not found"
  )) %>% mutate(vocational_dreamjob_dummy = case_when(
      is.na(chfuture) == T ~ NA,
      chfuture %in% c("construction worker", "cook", "driver", "engineer", "fireman/woman",
"fisherman", "fisherman/woman", "labourer",
 "mason", "mechanic", "tailor", "taxi driver", "domestic worker",
"farmer", "painter/decorator", "traditional occupation",
"market trader/shop assistant", "painter", "trader/businessman/woman",
 "trader", "market trader") ~ "Yes",
      chfuture == "other" ~ NA,
      TRUE ~ "No"),
      academic_dreamjob_dummy = case_when(
      is.na(chfuture) == T ~ NA,
      chfuture %in% c("doctor", "dentist", "nurse", "vet", "veterinary",   "teacher", "lecturer",
"scientist", "university student",
  "university student/other form of further education", "accountant", "lawyer", "management",
"manager/management", "computer operator", "civil servant", "politician", "president of the country",
"president/leader of country") ~ "Yes",
      chfuture == "other" ~ NA,
      TRUE ~ "No"),
)

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, dreamjob_sector), by = "childcode") %>%
  left_join(dplyr::select(merged_yc_ch, childcode, dreamjob_sector), by = "childcode") %>%
  mutate(dreamjob_sector = coalesce(dreamjob_sector.x, dreamjob_sector.y)) %>%
  dplyr::select(-dreamjob_sector.x, -dreamjob_sector.y)

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, vocational_dreamjob_dummy), by = "childcode")
%>%
  left_join(dplyr::select(merged_yc_ch, childcode, vocational_dreamjob_dummy), by = "childcode")
%>%
```

```
  mutate(vocational_dreamjob_dummy = coalesce(vocational_dreamjob_dummy.x,
vocational_dreamjob_dummy.y)) %>%
  dplyr::select(-vocational_dreamjob_dummy.x, -vocational_dreamjob_dummy.y)

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc_ch, childcode, academic_dreamjob_dummy), by = "childcode")
%>%
  left_join(dplyr::select(merged_yc_ch, childcode, academic_dreamjob_dummy), by = "childcode")
%>%
  mutate(academic_dreamjob_dummy = coalesce(academic_dreamjob_dummy.x,
academic_dreamjob_dummy.y)) %>%
  dplyr::select(-academic_dreamjob_dummy.x, -academic_dreamjob_dummy.y)


```

1.4 Household Characteristics
Household basic characteristics

```{r}
#We include mother, father and caregivers age when child was born
#Mum or dad passed away before secondary (wave 2 oc, wave 4yc)
#living with mom or dad (1), or another member of household (0), wave 2 (oc), wave 4 (yc)
#And an indicator for whether parents were sick during wave 2(oc) or wave 4 (yc)
#Household size in wave 2 (oc) and wave 4 (yc) and number of children aged 0-12

ages <- merged_con %>% filter(round == 1) %>% mutate(across(c(dadage, momage, careage,
agemon),  as.numeric)) %>% mutate(
    dadage_atbirth = round(dadage - (agemon/12), 0),
    momage_atbirth = round(momage - (agemon/12), 0),
    careage_atbirth = round(careage - (agemon/12), 0),
) %>% dplyr::select(childcode, dadage_atbirth, momage_atbirth, careage_atbirth)
covariates <- covariates %>% left_join(ages, by = "childcode")

vars <- c("dadlive", "momlive", "carerel", "hhsize", "male05", "male612", "female05", "female612",
"shfam4", "shfam5")

# Filter and process data for the Older cohort
older_cohort <- merged_con %>%
  filter(yc == "Older cohort", round == 2) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 2, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Filter and process data for the Younger cohort
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 4) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 4, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

combined_data <- bind_rows(older_cohort, younger_cohort) %>% mutate(
  dadpassed = case_when(
```

```
    is.na(dadlive) == T ~ NA,
    dadlive == "Has died" ~ 1,
    TRUE ~ 0),
  mompassed = case_when(
    is.na(momlive) == T ~ NA,
    momlive == "Has died" ~ 1,
    TRUE ~ 0),
  primarycaregiver = case_when(
    is.na(carerel) == T ~ NA,
    carerel %in% c("Biological parent", "Non-biological parent") ~ "parent",
    carerel %in% c("Father-in-law/mother-in-law", "Grandparent", "grandparents", "Other-relative",
"Uncle/aunt") ~ "relatives",
    carerel == "Sibling" ~ "sibling",
    carerel == "Other-nonrelative" ~ "nonrelatives",
    TRUE ~ NA),
  parents_primarycaregiver = case_when(
    is.na(primarycaregiver) == T ~ NA,
    primarycaregiver == "parent" ~ 1,
    TRUE ~ 0),
  hhsize = as.numeric(hhsize),
  male012 = as.numeric(as.numeric(male05) + as.numeric(male612)),
  female012 = as.numeric(as.numeric(female05) + as.numeric(female612)),
  parent_sick = case_when(
    is.na(shfam4) == T & is.na(shfam5) == T ~ NA,
    shfam4 == 1 ~ 1,
    shfam4 == 1 ~ 1,
    TRUE ~ 0
  )) %>% dplyr::select(childcode, dadpassed, mompassed, primarycaregiver, parents_primarycaregiver,
hhsize, male012, female012, parent_sick)

covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(older_cohort, younger_cohort, combined_data, vars, ages)


#children born before and after Young lives child
#No manipulation required
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, bornbef), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, bornbef), by = "childcode") %>%
  mutate(bornbef = coalesce(bornbef.x, bornbef.y)) %>%
  dplyr::select(-bornbef.x, -bornbef.y)

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, bornaft), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, bornaft), by = "childcode") %>%
  mutate(bornaft = coalesce(bornaft.x, bornaft.y)) %>%
  dplyr::select(-bornaft.x, -bornaft.y)

#total children, children born before + after + 1 (young lives survey)
covariates <- covariates %>%
  mutate(total_children_household = bornbef + bornaft + 1)
```

````
```

Household Primary Occupation
```{r}
#first grouping the different occupations
group_jobs_into_sectors <- function(job) {
  case_when(
    is.na(job == T) ~ NA_character_,
    job %in% c("farming/agriculture", "forestry/logging", "fishing") ~ "agriculture",
    job %in% c("factory work", "handicrafts", "pottery", "weaving", "tailor/sewing") ~ "crafts and manufacturing",
    job %in% c("beauty salon", "barber shop", "entertainment services",
               "nursing/medicinal services", "teaching", "security guard", "transportation", "food/local drink preperation", "trading") ~ "services",
    job %in% c("construction", "blacksmith", "mechanic services", "plumbing services") ~ "construction and repairs",
    job %in% c("civil servant", "army") ~ "public sector",
    job %in% c("casual/intermittent labour", "domestic work", "collecting firewood/water to sell", "child work") ~ "casual labor",
    TRUE ~ job
  )
}

#transform variables
merged_oc <- merged_oc %>% mutate(job = group_jobs_into_sectors(primocc))
merged_yc <- merged_yc %>% mutate(job = group_jobs_into_sectors(primocc))

#merge
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, job), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, job), by = "childcode") %>%
  mutate(household_primary_job = coalesce(job.x, job.y)) %>%
  dplyr::select(-job.x, -job.y)

```

Parents and caregivers level of education
```{r}
#once again time of measurement is wave 2 for OC, wave 4 for YC
vars <- c("momedu", "dadedu", "caredu", "dadcantread", "momcantread", "carecantread")

# Filter and process data for the Older cohort
older_cohort <- merged_con %>%
  filter(yc == "Older cohort", round == 2) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 2, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Filter and process data for the Younger cohort
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 4) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 4, ., NA_real_), .names = "{.col}")) %>%
````

```
  dplyr::select(childcode, all_of(vars))

#combine and transforming into better working categories
#one in buckets, dummies for no formal education, completed primary school, completed secondary
school, and participated in post-secondary education.
combined_data <- bind_rows(older_cohort, younger_cohort) %>% mutate(
  mom_edu = case_when(
    is.na(momedu) == T ~ NA,
    momedu == "Other" ~ NA,
    momedu %in% c("None", "religious education", "Adult literacy") ~ "No Formal Education",
    momedu %in% c("Grade 1", "Grade 2", "Grade 3", "Grade 4") ~ "Grade 1-4",
    momedu %in% c("Grade 5", "Grade 6", "Grade 7") ~ "Grade 5-7",
    momedu %in% c("Grade 8", "Grade 9", "Grade 10") ~ "Grade 8-10",
    momedu %in% c("Grade 11", "Grade 12") ~"Grade 11-12",
    momedu %in% c("Post-secondary, vocational", "Technical, pedagogical, CETPRO (complete)",
"Technical, pedagogical, CETPRO (incomplete)", "Vocational, technical college") ~ "Vocational post-
secondary education",
    momedu %in% c("Masters, doctorate", "University", "University (complete)", "University
(incomplete)") ~ "(in)complete university",
    TRUE ~ momedu
  ), dad_edu = case_when(
  is.na(dadedu) == T ~ NA,
  dadedu == "Other" ~ NA,
  dadedu %in% c("None", "religious education", "Adult literacy") ~ "No Formal Education",
  dadedu %in% c("Grade 1", "Grade 2", "Grade 3", "Grade 4") ~ "Grade 1-4",
  dadedu %in% c("Grade 5", "Grade 6", "Grade 7") ~ "Grade 5-7",
  dadedu %in% c("Grade 8", "Grade 9", "Grade 10") ~ "Grade 8-10",
  dadedu %in% c("Grade 11", "Grade 12") ~"Grade 11-12",
  dadedu %in% c("Post-secondary, vocational", "Technical, pedagogical, CETPRO (complete)",
"Technical, pedagogical, CETPRO (incomplete)", "Vocational, technical college") ~ "Vocational post-
secondary education",
  dadedu %in% c("Masters, doctorate", "University", "University (complete)", "University
(incomplete)") ~ "(in)complete university",
  TRUE ~ dadedu
), care_edu = case_when(
  is.na(caredu) == T ~ NA,
  caredu == "Other" ~ NA,
  caredu %in% c("None", "religious education", "Adult literacy") ~ "No Formal Education",
  caredu %in% c("Grade 1", "Grade 2", "Grade 3", "Grade 4") ~ "Grade 1-4",
  caredu %in% c("Grade 5", "Grade 6", "Grade 7") ~ "Grade 5-7",
  caredu %in% c("Grade 8", "Grade 9", "Grade 10") ~ "Grade 8-10",
  caredu %in% c("Grade 11", "Grade 12") ~"Grade 11-12",
  caredu %in% c("Post-secondary, vocational", "Technical, pedagogical, CETPRO (complete)",
"Technical, pedagogical, CETPRO (incomplete)", "Vocational, technical college") ~ "Vocational post-
secondary education",
  caredu %in% c("Masters, doctorate", "University", "University (complete)", "University
(incomplete)") ~ "(in)complete university",
  TRUE ~ caredu
), mom_edu_attended_formaleducation = case_when(
  is.na(mom_edu) == T ~ NA,
  mom_edu == "No Formal Education" ~ 0,
```

```
  TRUE ~ 1),
  mom_edu_beyond_primaryeducation = case_when(
  is.na(mom_edu) == T ~ NA,
  mom_edu %in% c("No Formal Education", "Grade 1-4", "Grade 5-7") ~ 0,
  TRUE ~ 1),
  mom_edu_attended_postsecondary = case_when(
  is.na(mom_edu) == T ~ NA,
  mom_edu %in% c("Vocational post-secondary education", "(in)complete university") ~ 1,
  TRUE ~ 0),
  mom_edu_attended_vocational = case_when(
  is.na(mom_edu) == T ~ NA,
  mom_edu %in% c("Vocational post-secondary education") ~ 1,  #could be that parent attended
vocational, and then continued studying, but that is not detectable in the data
  TRUE ~ 0),
dad_edu_attended_formaleducation = case_when(
  is.na(dad_edu) == T ~ NA,
  dad_edu == "No Formal Education" ~ 0,
  TRUE ~ 1),
dad_edu_beyond_primaryeducation = case_when(
  is.na(dad_edu) == T ~ NA,
  dad_edu %in% c("No Formal Education", "Grade 1-4", "Grade 5-7") ~ 0,
  TRUE ~ 1),
dad_edu_attended_postsecondary = case_when(
  is.na(dad_edu) == T ~ NA,
  dad_edu %in% c("Vocational post-secondary education", "(in)complete university") ~ 1,
  TRUE ~ 0),
dad_edu_attended_vocational = case_when(
  is.na(dad_edu) == T ~ NA,
  dad_edu %in% c("Vocational post-secondary education") ~ 1,  #could be that parent attended
vocational, and then continued studying, but that is not detectable in the data
  TRUE ~ 0), care_edu_attended_formaleducation = case_when(
  is.na(care_edu) == T ~ NA,
  care_edu == "No Formal Education" ~ 0,
  TRUE ~ 1),
care_edu_beyond_primaryeducation = case_when(
  is.na(care_edu) == T ~ NA,
  care_edu %in% c("No Formal Education", "Grade 1-4", "Grade 5-7") ~ 0,
  TRUE ~ 1),
care_edu_attended_postsecondary = case_when(
  is.na(care_edu) == T ~ NA,
  care_edu %in% c("Vocational post-secondary education", "(in)complete university") ~ 1,
  TRUE ~ 0),
care_edu_attended_vocational = case_when(
  is.na(care_edu) == T ~ NA,
  care_edu %in% c("Vocational post-secondary education") ~ 1,  #could be that parent attended
vocational, and then continued studying, but that is not detectable in the data
  TRUE ~ 0))

covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(older_cohort, younger_cohort, combined_data, vars)
```

```

Educational expectactions and Willingness to Pay
```{r}
#Parents perceptino of usefullness of formal education in their own life
#correcting typos
merged_oc <- merged_oc %>% mutate(scuseful = case_when(
  scuseful == "no it has not been useful" ~ "no, it is not useful",
  scuseful == "yes it is essential" ~ "yes, it is essential",
  TRUE ~ scuseful
))

merged_yc <- merged_yc %>% mutate(scuseful = case_when(
  scuseful == "no it has not been useful" ~ "no, it is not useful",
  scuseful == "yes it is essential" ~ "yes, it is essential",
  TRUE ~ scuseful
))

#merge and name
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, scuseful), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, scuseful), by = "childcode") %>%
  mutate(formal_education_useful = coalesce(scuseful.x, scuseful.y)) %>%
  dplyr::select(-scuseful.x, -scuseful.y)

#12yr old son/daugther at school - family needs money - what should family do
#Covariates dummy takes the value for the gender of the surveyed child
temp <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, famson), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, famson), by = "childcode") %>%
  mutate(famson = coalesce(famson.x, famson.y)) %>%
  dplyr::select(-famson.x, -famson.y)

temp <- temp %>%
  left_join(dplyr::select(merged_oc, childcode, famdtr), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, famdtr), by = "childcode") %>%
  mutate(famdtr = coalesce(famdtr.x, famdtr.y)) %>%
  dplyr::select(-famdtr.x, -famdtr.y)

#Generate variable with correct value
temp <- temp %>% mutate(across(starts_with("fam"), ~ if_else(. == "other", NA_character_, .)))
%>% #remove others
mutate(education_during_financial_hardship = case_when(
  chsex == "male" ~ famson,
  chsex == "female" ~ famdtr,
  is.na(chsex) == T ~ NA_character_,
  TRUE ~ NA_character_
))

#merge

```
covariates <- covariates %>%
  left_join(dplyr::select(temp, childcode, education_during_financial_hardship), by = "childcode")
#Perceived quality of nearest primary school
#some double coding in OC dataset
merged_oc <- merged_oc %>% mutate(csv1 = case_when(
  csv1 == "more or less (some doubt)" ~ "more or less",
  csv1 == "no (don't agree)" ~ "disagree",
  csv1 == "yes (agree)" ~ "agree",
  TRUE ~ csv1
))

#now merge
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, csv1), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, csv1), by = "childcode") %>%
  mutate(quality_primary_school = coalesce(csv1.x, csv1.y)) %>%
  dplyr::select(-csv1.x, -csv1.y)

#At what age should child get married?
#making categories -> for easier interpretation
merged_oc <- merged_oc %>%
  mutate(expected_age_married = case_when(
    is.na(expmar) ~ NA_character_,
    expmar == "no expectation" ~ "no expectation",
    as.numeric(expmar) < 18 ~ "Before 18",
    as.numeric(expmar) >= 18 & as.numeric(expmar) <= 21 ~ "18-21",
    as.numeric(expmar) >= 22 & as.numeric(expmar) <= 26 ~ "22-26",
    as.numeric(expmar) >= 27 & as.numeric(expmar) <= 30 ~ "27-30",
    as.numeric(expmar) >= 31 ~ "31+",
    TRUE ~ NA_character_
  ))

merged_yc <- merged_yc %>%
  mutate(expected_age_married = case_when(
    is.na(expmar) ~ NA_character_,
    expmar == 77 | expmar == 88 ~ NA_character_, #NA codes
    expmar == "0" ~ "no expectation", #for YC 0 is no expectation
    as.numeric(expmar) < 18 ~ "Before 18",
    as.numeric(expmar) >= 18 & as.numeric(expmar) <= 21 ~ "18-21",
    as.numeric(expmar) >= 22 & as.numeric(expmar) <= 26 ~ "22-26",
    as.numeric(expmar) >= 27 & as.numeric(expmar) <= 30 ~ "27-30",
    as.numeric(expmar) >= 31 ~ "31+",
    TRUE ~ NA_character_
  ))

#merge
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, expected_age_married), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, expected_age_married), by = "childcode") %>%
  mutate(expected_age_married = coalesce(expected_age_married.x, expected_age_married.y)) %>%
  dplyr::select(-expected_age_married.x, -expected_age_married.y)
```

```
#expected age earnings
#making categories -> for easier interpretation
merged_oc <- merged_oc %>%
  mutate(expected_age_earning = case_when(
    is.na(expearn) ~ NA_character_,
    expearn == "no expectation" ~ "no expectation",
    as.numeric(expearn) < 18 ~ "Before 18",
    as.numeric(expearn) >= 18 & as.numeric(expearn) <= 19 ~ "18-19",
    as.numeric(expearn) >= 20 & as.numeric(expearn) <= 22 ~ "20-22",
    as.numeric(expearn) >= 23 & as.numeric(expearn) <= 26 ~ "23-26",
    as.numeric(expearn) >= 27 ~ "27+",
    TRUE ~ NA_character_
  ))

merged_yc <- merged_yc %>%
  mutate(expected_age_earning = case_when(
    is.na(expearn) ~ NA_character_,
    expearn == 77 | expearn == 88 ~ NA_character_, #NA codes
    expearn == "0" ~ "no expectation", #for YC 0 is no expectation
    as.numeric(expearn) < 18 ~ "Before 18",
    as.numeric(expearn) >= 18 & as.numeric(expearn) <= 19 ~ "18-19",
    as.numeric(expearn) >= 20 & as.numeric(expearn) <= 22 ~ "20-22",
    as.numeric(expearn) >= 23 & as.numeric(expearn) <= 26 ~ "23-26",
    as.numeric(expearn) >= 27 ~ "27+",
    TRUE ~ NA_character_
  ))

#merge
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, expected_age_earning), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, expected_age_earning), by = "childcode") %>%
  mutate(expected_age_earning = coalesce(expected_age_earning.x, expected_age_earning.y)) %>%
  dplyr::select(-expected_age_earning.x, -expected_age_earning.y)

## expected age leaving school
#making categories -> for easier interpretation
merged_oc <- merged_oc %>%
  mutate(expected_age_leaving_school = case_when(
    is.na(expedu) ~ NA_character_,
    expedu == "no expectation" ~ "no expectation",
    as.numeric(expedu) < 18 ~ "Before 18",
    as.numeric(expedu) >= 18 & as.numeric(expedu) <= 19 ~ "18-19",
    as.numeric(expedu) >= 20 & as.numeric(expedu) <= 21 ~ "20-21",
    as.numeric(expedu) >= 22 & as.numeric(expedu) <= 23 ~ "22-23",
    as.numeric(expedu) >= 24 & as.numeric(expedu) <= 25 ~ "24-25",
    as.numeric(expedu) >= 26 ~ "26+",
    TRUE ~ NA_character_
  ))

merged_yc <- merged_yc %>%
```

```
  mutate(expected_age_leaving_school = case_when(
    is.na(expedu) ~ NA_character_,
    expedu == 77 | expedu == 88 ~ NA_character_, #NA codes
    expedu == "0" ~ "no expectation", #for YC 0 is no expectation
    as.numeric(expedu) < 18 ~ "Before 18",
    as.numeric(expedu) >= 18 & as.numeric(expedu) <= 19 ~ "18-19",
    as.numeric(expedu) >= 20 & as.numeric(expedu) <= 21 ~ "20-21",
    as.numeric(expedu) >= 22 & as.numeric(expedu) <= 23 ~ "22-23",
    as.numeric(expedu) >= 24 & as.numeric(expedu) <= 25 ~ "24-25",
    as.numeric(expedu) >= 26 ~ "26+",
    TRUE ~ NA_character_
  ))

#merge
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, expected_age_leaving_school), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, expected_age_leaving_school), by = "childcode") %>%
  mutate(expected_age_leaving_school = coalesce(expected_age_leaving_school.x,
expected_age_leaving_school.y)) %>%
  dplyr::select(-expected_age_leaving_school.x, -expected_age_leaving_school.y)
```

Expected grades and jobs

```{r}
#parents desired grades for their children
merged_yc <- merged_yc %>% mutate(
  countrycode = substr(childcode, 1, 2),
  expected_grade_by_parents = case_when(
    is.na(gradlike) ~ NA_character_,
    gradlike %in% c("other") ~ NA_character_,
    gradlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, 9, "none", "grade 1", "grade 2", "grade 3", "grade 4", "grade
5", "grade 6", "grade 7", "grade 8", "grade 9", "adult literacy", "religious education") & countrycode
%in% c("ET", "IN") ~ "(below) primary",
    gradlike %in% c(10, 11, "grade 10", "grade 11") & countrycode %in% c("ET", "IN") ~ "lower-
secondary",
    gradlike %in% c(12, "grade 12") & countrycode %in% c("ET", "IN") ~ "upper-secondary",
    gradlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, "none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5",
"grade 6", "grade 7", "grade 8", "adult literacy", "religious education") & countrycode == "VN" ~
"(below) primary",
    gradlike %in% c(9, 10, 11, "grade 9", "grade 10", "grade 11") & countrycode == "VN" ~ "lower-
secondary",
    gradlike %in% c(12, "grade 12") & countrycode == "VN" ~ "upper-secondary",

    gradlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, "none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5",
"grade 6", "grade 7", "grade 8", "adult literacy", "religious education") & countrycode == "PE" ~
"(below) primary",
    gradlike %in% c(9, 10, 11,"grade 9", "grade 10", "grade 11") & countrycode == "PE" ~ "lower-
secondary",
    gradlike %in% c(12, "grade 12") & countrycode == "PE" ~ "upper-secondary",
```

gradlike %in% c("complete technical college", "complete technical or pedagogical institute", "incomplete technical college", "post-secondary, vocational", "post-secondary/vocational") ~ "technical/vocational college",
    gradlike %in% c("complete university", "masters or doctorate at university", "masters/higher education", "university", "university degree") ~ "university/college",
    TRUE ~ NA
)
)

merged_oc <- merged_oc %>% mutate(
  countrycode = substr(childcode, 1, 2),
  expected_grade_by_parents = case_when(
    is.na(gradlike) ~ NA,
    gradlike == "other" ~ NA_character_,
    gradlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, 9, "none", "adult literacy", "religious education") & countrycode %in% c("ET", "IN") ~ "(below) primary",
    gradlike %in% c(10, 11) & countrycode %in% c("ET", "IN") ~ "lower-secondary",
    gradlike %in% c(12) & countrycode %in% c("ET", "IN") ~ "upper-secondary",

    gradlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, "none", "adult literacy", "religious education") & countrycode == "VN" ~ "(below) primary",
    gradlike %in% c(9, 10, 11) & countrycode == "VN" ~ "lower-secondary",
    gradlike %in% c(12) & countrycode == "VN" ~ "upper-secondary",

    gradlike %in% c(1, 2, 3, 4, 5, 6, 7, 8, "none", "adult literacy", "religious education") & countrycode == "PE" ~ "(below) primary",
    gradlike %in% c(9, 10, 11) & countrycode == "PE" ~ "lower-secondary",
    gradlike %in% c(12) & countrycode == "PE" ~ "upper-secondary",

    gradlike %in% c("complete technical college","incomplete technical college", "post-secondary, vocational") ~ "technical/vocational college",
    gradlike %in% c("incomplete university", "complete university", "university") ~ "university/college",
    TRUE ~ NA
)
)

#Now we have dummy for: Do you think child will complete desired level of educatio?
#no manipulation required
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, expgrade), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, expgrade), by = "childcode") %>%
  mutate(realistic_expectations_parents = coalesce(expgrade.x, expgrade.y)) %>%
  dplyr::select(-expgrade.x, -expgrade.y)

```

household job expectation
```{r}
merged_yc <- merged_yc %>% mutate(cambitn = case_when(

```
  cambitn %in% c("12", "40", "43", "44") ~ "other", #unlabelled
  TRUE ~ cambitn
))

merged_oc <- merged_oc %>% mutate(cambitn = case_when(
  cambitn %in% c("12", "40", "43", "44") ~ "other", #unlabelled
  TRUE ~ cambitn
))

# Update merged_yc
merged_yc <- merged_yc %>%
  mutate(parents_dreamjob_sector = case_when(
    is.na(cambitn) ~ NA_character_,
    cambitn %in% c("doctor", "dentist", "nurse", "vet", "veterinary", "enfermera(o)", "dentista",
"veterinario(a)") ~ "Healthcare",
    cambitn %in% c("teacher", "lecturer", "scientist", "student/university student", "university
student/other form of further education", "profesor universitario", "profesor(a)", "estudiante
universitario / otra educaci") ~ "Education and Research",
    cambitn %in% c("administrative assistant/secretary", "district collector", "religious
leader/priest/sheikh", "accountant", "lawyer", "management", "manager/management", "computer
operator", "administrative assistent/secretary", "trader/businessman/woman", "trader", "religious
leader", "market trader", "comerciante / negociante", "trader/ businessman/woman", "administrador",
"abogado(a)", "contador", "operador(a) de computadora", "vendedor en mercado / ayudante en tienda",
"religious leader/priest/shaik", "asistente administrativo / secretaria") ~ "Services and Management",
    cambitn %in% c("civil servant", "politician", "president of the country", "president/leader of
country", "soldier", "policeman/woman", "conductor", "policeman", "fireman", "military
man/woman", "soldado/ffaa", "presidente del pa", "president of country", "polic", "bombero(a)") ~
"Public Administration and Services",
    cambitn %in% c("construction worker", "cook", "driver", "engineer", "fireman/woman",
"fisherman", "fisherman/woman", "labourer", "mason", "mechanic", "tailor", "taxi driver", "domestic
worker", "farmer", "painter/decorator", "traditional occupation", "pilot", "piloto", "market trader/shop
assistant", "painter", "ingeniero(a)", "cocinero(a)", "trabajador (a) de construcci", "pintor(a) /
decorador(a)", "mec", "chofer", "chofer de taxi (taxista)", "pescador(a)", "trabajadora dom", "sastre",
"agricultor(a)", "factory worker", "l", "cient") ~ "Skilled Trades and Manual Labor",
    cambitn %in% c("artist", "actor/actress", "singer", "sportsman/woman", "sportman/woman", "full-
time parent/housewife", "fulltime parent/housewife", "sportsman", "actor", "artista", "actor/actriz",
"deportista", "cantante", "periodista", "padre / madre a tiempo completo / ama de casa", "other",
"other, specify", "otro (especificar)", "ns", "nqc", "not known", "na", "pe") ~ "Other",
    TRUE ~ "not found"
  ),
  parents_vocational_dreamjob_dummy = case_when(
    is.na(cambitn) ~ NA_character_,
    cambitn %in% c("construction worker", "cook", "driver", "engineer", "fireman/woman",
"fisherman", "fisherman/woman", "labourer", "mason", "mechanic", "tailor", "taxi driver", "domestic
worker", "farmer", "painter/decorator", "traditional occupation", "market trader/shop assistant",
"painter", "trader/businessman/woman", "trader", "market trader", "ingeniero(a)", "cocinero(a)",
"trabajador (a) de construcci", "pintor(a) / decorador(a)", "mec", "chofer", "chofer de taxi (taxista)",
"pescador(a)", "trabajadora dom", "sastre", "agricultor(a)", "factory worker", "cient", "l", "trader/
businessman/woman", "vendedor en mercado / ayudante en tienda") ~ "Yes",
```

```
    cambitn %in% c("other", "other, specify", "otro (especificar)", "ns", "nqc", "not known", "na") ~
NA_character_,
    TRUE ~ "No"
  ),
  parents_academic_dreamjob_dummy = case_when(
    is.na(cambitn) ~ NA_character_,
    cambitn %in% c("doctor", "dentist", "nurse", "vet", "veterinary", "teacher", "lecturer", "scientist",
"university student", "university student/other form of further education", "accountant", "lawyer",
"management", "manager/management", "computer operator", "civil servant", "politician", "president
of the country", "president/leader of country", "profesor universitario", "profesor(a)", "estudiante
universitario / otra educaci", "abogado(a)", "contador", "operador(a) de computadora", "religious
leader/priest/sheikh", "veterinario(a)", "religious leader/priest/shaik", "presidente del pa", "president of
country", "piloto") ~ "Yes",
    cambitn %in% c("other", "other, specify", "otro (especificar)", "ns", "nqc", "not known", "na") ~
NA_character_,
    TRUE ~ "No"
  ))

# Update merged_oc
merged_oc <- merged_oc %>%
  mutate(parents_dreamjob_sector = case_when(
    is.na(cambitn) ~ NA_character_,
    cambitn %in% c("doctor", "dentist", "nurse", "vet", "veterinary", "enfermera(o)", "dentista",
"veterinario(a)") ~ "Healthcare",
    cambitn %in% c("teacher", "lecturer", "scientist", "student/university student", "university
student/other form of further education", "profesor universitario", "profesor(a)", "estudiante
universitario / otra educaci", "university student") ~ "Education and Research",
    cambitn %in% c("administrative assistant/secretary", "district collector", "religious
leader/priest/sheikh", "accountant", "lawyer", "management", "manager/management", "computer
operator", "administrative assistent/secretary", "trader/businessman/woman", "trader", "religious
leader", "market trader", "comerciante / negociante", "trader/ businessman/woman", "administrador",
"abogado(a)", "contador", "operador(a) de computadora", "vendedor en mercado / ayudante en tienda",
"religious leader/priest/shaik", "asistente administrativo / secretaria") ~ "Services and Management",
    cambitn %in% c("civil servant", "politician", "president of the country", "president/leader of
country", "soldier", "policeman/woman", "conductor", "policeman", "fireman", "military
man/woman", "soldado/ffaa", "presidente del pa", "president of country", "polic", "bombero(a)") ~
"Public Administration and Services",
    cambitn %in% c("construction worker", "cook", "driver", "engineer", "fireman/woman",
"fisherman", "fisherman/woman", "labourer", "mason", "mechanic", "tailor", "taxi driver", "domestic
worker", "farmer", "painter/decorator", "traditional occupation", "pilot", "piloto", "market trader/shop
assistant", "painter", "ingeniero(a)", "cocinero(a)", "trabajador (a) de construcci", "pintor(a) /
decorador(a)", "mec", "chofer", "chofer de taxi (taxista)", "pescador(a)", "trabajadora dom", "sastre",
"agricultor(a)", "factory worker", "l", "cient") ~ "Skilled Trades and Manual Labor",
    cambitn %in% c("artist", "actor/actress", "singer", "sportsman/woman", "sportman/woman", "full-
time parent/housewife", "fulltime parent/housewife", "sportsman", "actor", "artista", "actor/actriz",
"deportista", "cantante", "periodista", "padre / madre a tiempo completo / ama de casa", "other",
"other, specify", "otro (especificar)", "ns", "nqc", "not known", "na", "pe") ~ "Other",
    TRUE ~ "not found"
  ),
  parents_vocational_dreamjob_dummy = case_when(
    is.na(cambitn) ~ NA_character_,
```

````
    cambitn %in% c("construction worker", "cook", "driver", "engineer", "fireman/woman",
"fisherman", "fisherman/woman", "labourer", "mason", "mechanic", "tailor", "taxi driver", "domestic
worker", "farmer", "painter/decorator", "traditional occupation", "market trader/shop assistant",
"painter", "trader/businessman/woman", "trader", "market trader", "ingeniero(a)", "cocinero(a)",
"trabajador (a) de construcci", "pintor(a) / decorador(a)", "mec", "chofer", "chofer de taxi (taxista)",
"pescador(a)", "trabajadora dom", "sastre", "agricultor(a)", "factory worker", "cient", "l", "trader/
businessman/woman", "vendedor en mercado / ayudante en tienda") ~ "Yes",
    cambitn %in% c("other", "other, specify", "otro (especificar)", "ns", "nqc", "not known", "na") ~
NA_character_,
    TRUE ~ "No"
  ),
  parents_academic_dreamjob_dummy = case_when(
    is.na(cambitn) ~ NA_character_,
    cambitn %in% c("doctor", "dentist", "nurse", "vet", "veterinary", "teacher", "lecturer", "scientist",
"university student", "university student/other form of further education", "accountant", "lawyer",
"management", "manager/management", "computer operator", "civil servant", "politician", "president
of the country", "president/leader of country", "profesor universitario", "profesor(a)", "estudiante
universitario / otra educaci", "abogado(a)", "contador", "operador(a) de computadora", "religious
leader/priest/sheikh", "veterinario(a)", "religious leader/priest/shaik", "presidente del pa", "president of
country", "piloto", "university student") ~ "Yes",
    cambitn %in% c("other", "other, specify", "otro (especificar)", "ns", "nqc", "not known", "na") ~
NA_character_,
    TRUE ~ "No"
  ))

#merging
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, parents_dreamjob_sector), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, parents_dreamjob_sector), by = "childcode") %>%
  mutate(parents_dreamjob_sector = coalesce(parents_dreamjob_sector.x, parents_dreamjob_sector.y))
%>%
  dplyr::select(-parents_dreamjob_sector.x, -parents_dreamjob_sector.y)

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, parents_vocational_dreamjob_dummy), by =
"childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, parents_vocational_dreamjob_dummy), by =
"childcode") %>%
  mutate(parents_vocational_dreamjob_dummy = coalesce(parents_vocational_dreamjob_dummy.x,
parents_vocational_dreamjob_dummy.y)) %>%
  dplyr::select(-parents_vocational_dreamjob_dummy.x, -parents_vocational_dreamjob_dummy.y)

covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, parents_academic_dreamjob_dummy), by =
"childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, parents_academic_dreamjob_dummy), by =
"childcode") %>%
  mutate(parents_academic_dreamjob_dummy = coalesce(parents_academic_dreamjob_dummy.x,
parents_academic_dreamjob_dummy.y)) %>%
  dplyr::select(-parents_academic_dreamjob_dummy.x, -parents_academic_dreamjob_dummy.y)
```
````
````

Household economic situation

```{r}
#Indicators
#household economic situation at W2 (oc) and w4 (yc),
#Indicators; Owning house, food security during last 12 months, wealth index, housing quality index, access to services index, consumer durables index (food security has quite some NA's)

#once again time of measurement is wave 2 for OC, wave 4 for YC
vars <- c("ownhouse", "foodsec", "wi", "hq", "sv", "cd")

# Filter and process data for the Older cohort
older_cohort <- merged_con %>%
  filter(yc == "Older cohort", round == 2) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 2, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))

# Filter and process data for the Younger cohort
younger_cohort <- merged_con %>%
  filter(yc == "Younger cohort", round == 4) %>%
  mutate(across(all_of(vars), ~ ifelse(round == 4, ., NA_real_), .names = "{.col}")) %>%
  dplyr::select(childcode, all_of(vars))
combined_data <- bind_rows(older_cohort, younger_cohort)
covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(older_cohort, younger_cohort, combined_data, vars)

#Does your household have significant debt? Dummy
#No manipulation required
covariates <- covariates %>%
  left_join(dplyr::select(merged_oc, childcode, debt), by = "childcode") %>%
  left_join(dplyr::select(merged_yc, childcode, debt), by = "childcode") %>%
  mutate(debt = coalesce(debt.x, debt.y)) %>%
  dplyr::select(-debt.x, -debt.y)

```

Ocurrence of negative economic, regulatory and natural shocks
during wave 2 for OC
during wave 4 for YC

```{r}
library(dplyr)
library(tidyr)
older_cohort <- merged_con %>% filter(round == 2 & yc == "Older cohort") %>%
dplyr::select(childcode, starts_with("sh"))
younger_cohort <- merged_con %>% filter(round == 4 & yc == "Younger cohort") %>%
dplyr::select(childcode, starts_with("sh"))

#shock crime, (1) if one of shcrime1 to 8 is equal to one, zero otherwise
#for OC there is more detail available, for YC there is just one dummy
```

~~~
#natural disaster, (1) if one of shenv1 to 13 is equal to one, zero otherwise

#house collapse, (1) if one of shhouse1 to 3 is equal to one, zero otherwise
#for OC there is more detail available, for YC there is just one dummy

#household job loss, (1) shock-loss of job/source of income/family enterprise 0 otherwise

combined_data <- bind_rows(older_cohort, younger_cohort) %>%
mutate(
# Crime shock: 1 if any of shcrime1 to shcrime8 is 1, 0 if all are 0, NA if all are NA
   shock_crime = ifelse(
     rowSums(dplyr::select(., starts_with("shcrime")) == 1, na.rm = TRUE) > 0, 1,
     ifelse(
       rowSums(!is.na(dplyr::select(., starts_with("shcrime"))), na.rm = TRUE) == 0, NA_integer_,
       0
     )
   ),

# Household job loss shock: directly use shecon5 if available
   shock_household_job_loss = case_when(
     is.na(shecon5) ~ NA_integer_,
     shecon5 == 1 ~ 1,
     shecon5 == 0 ~ 0
   ),

# Natural disaster shock: 1 if any of shenv1 to shenv13 is 1, 0 if all are 0, NA if all are NA
   shock_natural_disaster = ifelse(
     rowSums(dplyr::select(., starts_with("shenv")) == 1, na.rm = TRUE) > 0, 1,
     ifelse(
       rowSums(!is.na(dplyr::select(., starts_with("shenv"))), na.rm = TRUE) == 0, NA_integer_,
       0
     )
   ),

# House collapse shock: 1 if any of shhouse1 to shhouse3 is 1, 0 if all are 0, NA if all are NA
   shock_house_collapse = ifelse(
     rowSums(dplyr::select(., starts_with("shhouse")) == 1, na.rm = TRUE) > 0, 1,
     ifelse(
       rowSums(!is.na(dplyr::select(., starts_with("shhouse"))), na.rm = TRUE) == 0, NA_integer_,
       0 ))) %>% dplyr::select(childcode, shock_crime, shock_household_job_loss,
shock_natural_disaster, shock_house_collapse)

covariates <- covariates %>% left_join(combined_data, by = "childcode")
remove(older_cohort, younger_cohort, combined_data)

```
~~~

## 1.5. Community level
Collecting data

~~~
```{r}
~~~

```
comm_in <- read.stata("unproc_data/raw_data/oc/in_r2_comm.dta", country = "in", rem.number = F)
%>% dplyr::select(placeid, popsize, timecap, agric, indust, handicr, lndmale, cnstmale, factmale,
nurspub, secrtry, pubsec, prvtscnd, ceos, posttech, privtech, pbsectim, psttchtm, prvtchtm, ceotim)
%>% rename(
  privtech_time = psttchtm,
  govtech_time = prvtchtm,
  ceostim = ceotim
) %>%
  mutate_all(as.character)


comm_et <- read.stata("unproc_data/raw_data/oc/et_r2_comm.dta", country = "et", rem.number = F)
%>% dplyr::select(placeid, popsize, timecap, agric, indust, handicr, lndmale, cnstmale, factmale,
nurspub, secrtry, pubsec, prvtscnd, ceos, gvpsttch, prpsttch, pbsectim, gvptchtm, prptchtm, ceostim)
%>% rename(
  posttech = gvpsttch,
  privtech = prpsttch,
  privtech_time = prptchtm,
  govtech_time = gvptchtm
) %>%
  mutate_all(as.character)

comm_pe <- read.stata("unproc_data/raw_data/oc/pe_r2_comm.dta", country = "pe", rem.number = F)
%>% dplyr::select(placeid, popsize, timecap, agric, indust, handicr, lndmale, cnstmale, factmale,
nurspub, secrtry, pubsec, prvtscnd, ceos, posttech, pbsectim, psttchtm, ceostim) %>% rename(
  govtech_time = psttchtm #no private tech institutions
) %>%
  mutate_all(as.character)

comm_vn <- read.stata("unproc_data/raw_data/oc/vn_r2_comm.dta", country = "vn", rem.number =
F) %>% dplyr::select(commid, popsize, timecap, agric, indust, handicr, lndmale, cnstmale, factmale,
nurspub, secrtry, pubsec, prvtscnd, ceos, gvpsttch, pvpsttch, pbsectim, govtchtm, prvtsctm, ceostim)
%>% rename(
  posttech = gvpsttch,
  privtech = pvpsttch,
  privtech_time = prvtsctm,
  govtech_time = govtchtm,
  placeid = commid
) %>%
  mutate_all(as.character)

columns_to_numeric <- c("popsize", "timecap", "lndmale", "cnstmale", "factmale",
                        "nurspub", "secrtry", "pbsectim", "govtech_time",
                        "privtech_time", "ceostim")

#correcting NA's and 88 values
comm <- bind_rows(comm_et, comm_in, comm_pe, comm_vn) %>%
  mutate_all(~ replace(., . %in% c("na", "NA", "nk", "NK", "they are not paid for that", "they do not
work in this job", "missing", "88"), NA)) %>%
  mutate(across(all_of(columns_to_numeric), as.numeric)) %>%
mutate(across(all_of(columns_to_numeric), ~ replace(., . < 0, NA))) %>%
```

````
  rename(commid_w1 = placeid)

remove(comm_et, comm_in, comm_pe, comm_vn, columns_to_numeric)

```

demographics + available jobs
```{r}
#popsize + timetodistrictcapital can just be merged
temp <- comm %>% dplyr::select(commid_w1, popsize, timecap)
covariates <- covariates %>% left_join(temp, by = "commid_w1")
remove(temp)

#jobs available based on land used for that purpose
#0 = not used,
#1 = 2nd/3th most important use
#2 = most important

comm <- comm %>% mutate(agriculture_jobs = case_when(
  is.na(agric) == T ~ NA,
  agric == "most important use" ~ "most important",
  agric == "The most important use" ~ "most important",
  agric %in% c("2nd most imp use", "3rd most imp use", "The second most important use", "this use
does exist but not imp") ~ "somewhat important",
  agric %in% c("No land used for this", "no local land is used for this purpose", "0") ~ "not important",
  TRUE ~ "not found"
),
factory_jobs = case_when(
  is.na(indust) == T ~ NA,
  indust == "most important use" ~ "most important",
  indust == "The most important use" ~ "most important",
  indust %in% c("2nd most imp use", "3rd most imp use", "The second most important use", "this use
does exist but not imp", "The third most important use") ~ "somewhat important",
  indust %in% c("No land used for this", "no local land is used for this purpose", "0") ~ "not
important",
  TRUE ~ "not found"
), craft_jobs = case_when(
  is.na(handicr) == T ~ NA,
  handicr == "most important use" ~ "most important",
  handicr == "The most important use" ~ "most important",
  handicr %in% c("2nd most imp use", "3rd most imp use", "The second most important use", "this use
does exist but not imp", "The third most important use") ~ "somewhat important",
  handicr %in% c("No land used for this", "no local land is used for this purpose", "0") ~ "not
important",
  TRUE ~ "not found"
))

#merge
temp <- comm %>% dplyr::select(commid_w1, agriculture_jobs, factory_jobs, craft_jobs)
covariates <- covariates %>% left_join(temp, by = "commid_w1")
remove(temp)
````

```

education available
```{r}
# For all:
# No, and not in a nearby locality
# No, but there is in a nearby locality
# Yes

#public secondary
comm <- comm %>% mutate(
  public_secondary_available = case_when(
    is.na(pubsec) == T ~ NA,
    pubsec %in% c("no and there is no such facility in a nearby locality") ~ "no, and not in a nearby locality",
    pubsec %in% c("no, but there is in a nearby locality", "No, but there is one in a nearby locality") ~ "no, but there is one in a nearby locality",
    pubsec %in% c("yes", "Yes") ~ "yes",
    TRUE ~ "not found"
  ),

  private_secondary_available = case_when(
  is.na(prvtscnd) == T ~ NA,
  prvtscnd %in% c("99") ~ NA,
  prvtscnd %in% c("no and there is no such facility in a nearby locality", "No, and there is no such facility in a nearby locality") ~ "no, and not in a nearby locality",
  prvtscnd %in% c("no, but there is in a nearby locality", "No, but there is one in a nearby locality") ~ "no, but there is one in a nearby locality",
  prvtscnd %in% c("yes", "Yes") ~ "yes",
  TRUE ~ "not found"
),

lower_vocational_available = case_when(
  is.na(ceos) == T ~ NA,
  ceos %in% c("99", "77") ~ NA,
  ceos %in% c("no and there is no such facility in a nearby locality", "No, and there is no such facility in a nearby locality") ~ "no, and not in a nearby locality",
  ceos %in% c("no, but there is in a nearby locality", "No, but there is one in a nearby locality") ~ "no, but there is one in a nearby locality",
  ceos %in% c("yes", "Yes") ~ "yes",
  TRUE ~ "not found"
),
public_higher_vocational_available = case_when(
  is.na(posttech) == T ~ NA,
  posttech %in% c("99", "77") ~ NA,
  posttech %in% c("no and there is no such facility in a nearby locality", "No, and there is no such facility in a nearby locality") ~ "no, and not in a nearby locality",
  posttech %in% c("no, but there is in a nearby locality", "No, but there is one in a nearby locality") ~ "no, but there is one in a nearby locality",
  posttech %in% c("yes", "Yes") ~ "yes",

```
  TRUE ~ "not found"
),
private_higher_vocational_available = case_when(
  is.na(privtech) == T ~ NA,
  privtech %in% c("99", "77") ~ NA,
  privtech %in% c("no and there is no such facility in a nearby locality", "No, and there is no such facility in a nearby locality") ~ "no, and not in a nearby locality",
  privtech %in% c("no, but there is in a nearby locality", "No, but there is one in a nearby locality") ~ "no, but there is one in a nearby locality",
  privtech %in% c("yes", "Yes") ~ "yes",
  TRUE ~ "not found"
))

#merge
temp <- comm %>% dplyr::select(commid_w1, public_secondary_available, private_secondary_available, lower_vocational_available, public_higher_vocational_available, private_higher_vocational_available)
covariates <- covariates %>% left_join(temp, by = "commid_w1")
remove(temp)

```

## 1.6. Save data
```{r}
write.csv(covariates, "proc_data/covariates.csv", row.names = FALSE)
```

#2. Educational History
Goal:
Input: unprocessed data from different waves
2.1 to 2.4 create a dataframe per country with per student their type of education for every year.
2.5. creates dummies based on that long dataframe, and transforms this into a wide dataframe that is the final output

##2.1. Custom functions
libaries
```{r}
library(dplyr)
library(ggplot2)
library(tidyr)
library(stringr)
```

Function 1:
Reads stata data file
makes variable names lowercase
adds country code to childcode
and if desired removes numbers from the variables.
changes numeric factors into labels

```{r}
read.stata <- function(file_path, country, rem.number = T) {
    library(haven)
    library(dplyr)
    library(stringr)

    # Read the data file
    data <- NULL
    data <- read_dta(file_path)

    # Process the data
    data <- data %>%
      rename_with(~tolower(.), everything()) %>%
      mutate(across(where(is.labelled), as_factor)) %>%
      mutate(across(where(is.factor), as.character)) %>%
      mutate(across(where(is.character), ~trimws(.x, which = "left"))) %>%
      mutate(across(where(is.factor), as.factor)) #remove starting spaces

    if ("childcode" %in% names(data) && !is.null(country)) {
        data <- data %>% mutate(childcode = as.factor(childcode))
        country_code <- toupper(substr(country, 1, 2))
        data <- data %>%
          mutate(childcode = paste0(country_code, childcode))
      }

   if (rem.number == T) {
      # Remove numbers from column names
      names(data) <- gsub(pattern = "[0-9]", replacement = "", names(data))
      }

    return(data)
  }
```

Function 2: calls and prepare the necessary datasets
```{r}
datasets <- function(cohort = "oc", country = "in") {
  cohort <- tolower(cohort)
  country <- tolower(country)

  # Function to read Stata file and preprocess data
  read.stata <- function(file_path, country, rem.number = T) {
    library(haven)
    library(dplyr)
    library(stringr)

    # Read the data file
    data <- NULL
    data <- read_dta(file_path)

    # Process the data
```

```
  data <- data %>%
    rename_with(~tolower(.), everything()) %>%
    mutate(across(where(is.labelled), as_factor)) %>%
    mutate(across(where(is.factor), as.character)) %>%
    mutate(across(where(is.character), ~trimws(.x, which = "left"))) %>%
    mutate(across(where(is.factor), as.factor)) #remove starting spaces

  if ("childcode" %in% names(data) && !is.null(country)) {
      data <- data %>% mutate(childcode = as.factor(childcode))
      country_code <- toupper(substr(country, 1, 2))
      data <- data %>%
        mutate(childcode = paste0(country_code, childcode))
    }

 if (rem.number == T) {
    # Remove numbers from column names
    names(data) <- gsub(pattern = "[0-9]", replacement = "", names(data))
    }

  return(data)
 }

 # Construct the file path dynamically
 path_wave3 <- sprintf("unproc_data/raw_data/%s/w3_%s_%s_childleve.dta", cohort, cohort,
country)
 path_wave4 <- sprintf("unproc_data/raw_data/%s/w4_%s_%s_educationhistoryindexchil.dta",
cohort, cohort, country)
 path_wave5 <- sprintf("unproc_data/raw_data/%s/w5_%s_%s_educationhistoryindexchild.dta",
cohort, cohort, country)
 path_call2 <- sprintf("unproc_data/raw_data/%s/w6_%s_%s_c2_arch.dta", cohort, cohort, country)
 path_call3 <- sprintf("unproc_data/raw_data/%s/w6_%s_%s_c3_arch.dta", cohort, cohort, country)
 path_call5 <- sprintf("unproc_data/raw_data/%s/w6_%s_%s_c5_arch.dta", cohort, cohort, country)

 # Call the read.stata function, passing 'country' explicitly for each necessary dataset
 wave3 <- read.stata(path_wave3, country, rem.number = F) %>% rename(childcode = childid)
 wave4 <- read.stata(path_wave4, country)
 wave5 <- read.stata(path_wave5, country)
 call2 <- read.stata(path_call2, country) %>% dplyr::select(c(childcode, dint, curgrdcov, curschcov,
cureducov, tmeschcov, lstschcov, lstgrdcov))
 call3 <- read.stata(path_call3, country) %>% dplyr::select(c(childcode, dint, curgrdcov, curschcov,
cureducov))
 call5 <- read.stata(path_call5, country) %>% dplyr::select(c(childcode, dintcov, curgrdcov, curshcov,
cureducov, tmeschcov, lstshcov, lstgrdcov)) %>% rename(dint = dintcov, curschcov = curshcov,
lstschcov = lstshcov)

 temp_name <- paste0(toupper(country), "0") #in Wave 3 some IDs are wrongly numbered
 wave3 <- wave3 %>%
 mutate(childcode = if_else(str_sub(childcode, 1, 3) == temp_name,
                 paste0(toupper(substr(country, 1, 2)), str_sub(childcode, 4)),
                 childcode))
```

  datasets <- list(wave3, wave4, wave5, call2, call3, call5)
  names(datasets) <- paste(country, cohort, c("wave3", "wave4", "wave5", "call2", "call3", "call5"), sep
= "_")
  Sys.setlocale("LC_ALL", "Dutch")
  return(datasets)
}

```

Function 3:
For wave 6 there are no pre-made educational histories
Gets most recent education based on call 2, 3 and 5 in wave 6.
Questions about current education were asked during phone calls 2, 3 and 5.
Questions about past education were asked during phone calls 2 and 5, but not year-by-year
  Call 5 was more elaborate, and is thus used unless there was no response.

```{r}
w6.recentedu <- function(call2, call3, call5, country) {
#if not enrolled, current education should be equal to none
#then cureducov == "No" or "Never attended", if NK, NA or refused to answer then it should be not
known
call2 <- call2 %>% mutate(curgrdcov = as.character(curgrdcov)) %>%
  mutate(curgrdcov = if_else(cureducov %in% c("No", "Never attended"), "None", curgrdcov)) %>%
  mutate(curgrdcov = if_else(cureducov %in% c("NK", "Refused to answer"), "Not Known",
curgrdcov)) %>% mutate(curgrdcov = if_else(is.na(cureducov) == T, "Not Known", curgrdcov)) %>%
rename(edgrader = curgrdcov, tyscr = curschcov)

call3 <- call3 %>% mutate(curgrdcov = as.character(curgrdcov)) %>%
  mutate(curgrdcov = if_else(cureducov %in% c("No", "Never attended"), "None", curgrdcov)) %>%
  mutate(curgrdcov = if_else(cureducov %in% c("NK", "Refused to answer"), "Not Known",
curgrdcov)) %>% mutate(curgrdcov = if_else(is.na(cureducov) == T, "Not Known", curgrdcov)) %>%
rename(edgrader = curgrdcov, tyscr = curschcov)


call5 <- call5 %>% mutate(curgrdcov = as.character(curgrdcov)) %>%
  mutate(curgrdcov = if_else(cureducov %in% c("No", "Never attended"), "None", curgrdcov)) %>%
  mutate(curgrdcov = if_else(cureducov %in% c("NK", "Refused to answer"), "Not Known",
curgrdcov)) %>% mutate(curgrdcov = if_else(is.na(cureducov) == T, "Not Known", curgrdcov))%>%
rename(edgrader = curgrdcov, tyscr = curschcov)

#converting dates to correct format to calculate schoolyears. Vietnam and India starts a new semester
in summer, India, Peru and Vietnam start the new semester at new year.

country <- tolower(country)
if (country %in% c("vn","in")) {
  Sys.setlocale("LC_ALL", "English")
  call2 <- call2 %>%
    mutate(
      dint = as.POSIXct(dint, format = "%d/%m/%Y %H:%M:%S"),  # Direct conversion using base R
      year = case_when(

```
      between(dint, as.POSIXct("2019-06-30"), as.POSIXct("2020-10-16")) ~ "2019-2020", #call 2
refers to schoolyear 2019-2020, despite being conducted in october
      between(dint, as.POSIXct("2020-10-15"), as.POSIXct("2021-06-30")) ~ "2020-2021",
      between(dint, as.POSIXct("2021-06-30"), as.POSIXct("2022-07-01")) ~ "2021-2022",
      TRUE ~ NA_character_
)
)

 call3 <- call3 %>%
  mutate(
   dint = as.POSIXct(dint, format = "%d/%m/%Y %H:%M:%S"), # Direct conversion using base R
   year = case_when(
      between(dint, as.POSIXct("2019-06-30"), as.POSIXct("2020-10-16")) ~ "2019-2020", #call 2
refers to schoolyear 2019-2020, despite being conducted in october
      between(dint, as.POSIXct("2020-10-15"), as.POSIXct("2021-06-30")) ~ "2020-2021",
      between(dint, as.POSIXct("2021-06-30"), as.POSIXct("2022-07-01")) ~ "2021-2022",
      TRUE ~ NA_character_
)
)

 call5 <- call5 %>%
  mutate(
   dint = as.POSIXct(dint, format = "%d-%b-%y"), # Direct conversion using base R
   year = case_when(
      between(dint, as.POSIXct("2019-06-30"), as.POSIXct("2020-10-16")) ~ "2019-2020", #call 2
refers to schoolyear 2019-2020, despite being conducted in october
      between(dint, as.POSIXct("2020-10-15"), as.POSIXct("2021-06-30")) ~ "2020-2021",
      between(dint, as.POSIXct("2021-06-30"), as.POSIXct("2022-07-01")) ~ "2021-2022",
      TRUE ~ NA_character_
)
)
}

if (country =="et") { #in ethopia school years are from january to december
 Sys.setlocale("LC_ALL", "English")
 call2 <- call2 %>%
  mutate(
   dint = as.POSIXct(dint, format = "%d/%m/%Y %H:%M:%S"), # Direct conversion using base R
   year = case_when(
      between(dint, as.POSIXct("2019-01-01"), as.POSIXct("2019-12-01")) ~ "2019",
      between(dint, as.POSIXct("2019-12-01"), as.POSIXct("2020-12-01")) ~ "2020",
      between(dint, as.POSIXct("2020-12-01"), as.POSIXct("2021-12-01")) ~ "2021",
      between(dint, as.POSIXct("2021-12-01"), as.POSIXct("2022-12-31")) ~ "2022",
      TRUE ~ NA_character_
)
)

 call3 <- call3 %>%
  mutate(
   dint = as.POSIXct(dint, format = "%d/%m/%Y %H:%M:%S"), # Direct conversion using base R
   year = case_when(
```

```
      between(dint, as.POSIXct("2019-01-01"), as.POSIXct("2019-12-01")) ~ "2019",
      between(dint, as.POSIXct("2019-12-01"), as.POSIXct("2020-12-01")) ~ "2020",
      between(dint, as.POSIXct("2020-12-01"), as.POSIXct("2021-12-01")) ~ "2021",
      between(dint, as.POSIXct("2021-12-01"), as.POSIXct("2022-12-31")) ~ "2022",
      TRUE ~ NA_character_
    )
  )

 call5 <- call5 %>%
  mutate(
    dint = as.POSIXct(dint, format = "%d-%b-%y"),  # Direct conversion using base R
    year = case_when(
      between(dint, as.POSIXct("2019-01-01"), as.POSIXct("2019-12-01")) ~ "2019",
      between(dint, as.POSIXct("2019-12-01"), as.POSIXct("2020-12-01")) ~ "2020",
      between(dint, as.POSIXct("2020-12-01"), as.POSIXct("2021-12-01")) ~ "2021",
      between(dint, as.POSIXct("2021-12-01"), as.POSIXct("2022-12-31")) ~ "2022",
      TRUE ~ NA_character_
    )
  )
}


if (country == "pe") { #in peru school years are from january to december, but language is spanish,
meaning dic for wave 5 has to be translated to Dec
 Sys.setlocale("LC_ALL", "English")
 call2 <- call2 %>%
  mutate(
    dint = as.POSIXct(dint, format = "%d/%m/%Y %H:%M:%S"),  # Direct conversion using base R
    year = case_when(
      between(dint, as.POSIXct("2019-01-01"), as.POSIXct("2019-12-01")) ~ "2019",
      between(dint, as.POSIXct("2019-12-01"), as.POSIXct("2020-12-01")) ~ "2020",
      between(dint, as.POSIXct("2020-12-01"), as.POSIXct("2021-12-01")) ~ "2021",
      between(dint, as.POSIXct("2021-12-01"), as.POSIXct("2022-12-31")) ~ "2022",
      TRUE ~ NA_character_
    )
  )

 call3 <- call3 %>%
  mutate(
    dint = as.POSIXct(dint, format = "%d/%m/%Y %H:%M:%S"),  # Direct conversion using base R
    year = case_when(
      between(dint, as.POSIXct("2019-01-01"), as.POSIXct("2019-12-01")) ~ "2019",
      between(dint, as.POSIXct("2019-12-01"), as.POSIXct("2020-12-01")) ~ "2020",
      between(dint, as.POSIXct("2020-12-01"), as.POSIXct("2021-12-01")) ~ "2021",
      between(dint, as.POSIXct("2021-12-01"), as.POSIXct("2022-12-31")) ~ "2022",
      TRUE ~ NA_character_
    )
  )

 call5 <- call5 %>%
  mutate(dint = str_replace_all(dint, fixed("-dic-"), "-Dec-"),
```

```
        dint = str_replace_all(dint, fixed("-ene-"), "-Jan-")) %>%
    mutate(
      dint = as.POSIXct(dint, format = "%d-%b-%y"),  # Direct conversion using base R
      year = case_when(
        between(dint, as.POSIXct("2019-01-01"), as.POSIXct("2019-12-01")) ~ "2019",
        between(dint, as.POSIXct("2019-12-01"), as.POSIXct("2020-12-01")) ~ "2020",
        between(dint, as.POSIXct("2020-12-01"), as.POSIXct("2021-12-01")) ~ "2021",
        between(dint, as.POSIXct("2021-12-01"), as.POSIXct("2022-12-31")) ~ "2022",
        TRUE ~ NA_character_
      )
    )
}
call2 <- call2 %>% dplyr::select(c(childcode, edgrader, tyscr, year, dint))
call3 <- call3 %>% dplyr::select(c(childcode, edgrader, tyscr, year, dint))
call5 <- call5 %>% dplyr::select(c(childcode, edgrader, tyscr, year, dint))

Sys.setlocale("LC_ALL", "Dutch")
output <- merge.waves(call2, call3, call5)
return(output)
}
```

Function 4:
Gets education obtained between 2017 and 2019 based on call 2 and 5 in wave 6.

```{r}
w6.edu20172020 <- function(call2, call5, wave15edu, country) {
  library(dplyr)
  merged_df <- NULL
  # Step 1: Merge data from call2 into call5 for missing replacement purposes
  merged_df <- call5 %>%
    left_join(call2, by = "childcode", suffix = c(".5", ".2"))

  # Define the columns for which replacements need to be made if there are NA's in call5
  replace_columns <- c("lstgrdcov", "lstschcov", "tmeschcov")

# Loop through the columns and replace NA values in call5 with values from call2 where available
  for (col in replace_columns) {
    merged_df <- merged_df %>%
      mutate("{col}.5" := coalesce(!!sym(paste0(col, ".5")), !!sym(paste0(col, ".2"))))
  }

  # dplyr::select only the original call5 columns (now potentially updated with call2 values where there
were NA's)
  merged_df <- merged_df %>%
      dplyr::select(childcode, dint = dint.5, tmeschcov = tmeschcov.5, lstschcov = lstschcov.5, lstgrdcov
= lstgrdcov.5)

  # Step 2 -> transform dataset into similar format
  # For school-year, we leave years intact
  # Before 2016 -> means Wave 5 covers this, so these can be removed
  # the same for 2016 -> wave 5 covers 2015-2016
```

```
  # for india and vietnam also 2016-2017 is covered in wave 6
  # 2017 onwards
  # Month + year -> code as year

  # Before 2019 -> (only for those imputed with call 2) data requires special attention
  # or NK -> also requires special attention.

#for India and Vietnam schoolyear lasts from summer to summer, thus if 2019 was the last time
attending education, schoolyear should be 2018 -2019.
if (country %in% c("in", "vn")) {
  Sys.setlocale("LC_ALL", "English")
  merged_df <- merged_df[merged_df$tmeschcov != "Before 2016", ]
  merged_df <- merged_df[merged_df$tmeschcov != "2016", ]
  merged_df <- merged_df[merged_df$tmeschcov != "2017", ]
  merged_df <- merged_df[!is.na(merged_df$childcode), ]
  merged_df <- merged_df %>%
    mutate(tmeschcov = case_when(
      str_detect(tmeschcov, "2017") ~ "2016-2017",
      str_detect(tmeschcov, "2018") ~ "2017-2018",
      str_detect(tmeschcov, "2019") ~ "2018-2019",
      str_detect(tmeschcov, "2020") ~ "2019-2020",
      str_detect(tmeschcov, "2021") ~ "2020-2021",
      str_detect(tmeschcov, "2022") ~ "2021-2022",
      TRUE ~ tmeschcov  # Keep original if no specific year is mentioned
    ))
}

if (country %in% c("pe", "et")) {
  Sys.setlocale("LC_ALL", "English")
  merged_df <- merged_df[merged_df$tmeschcov != "Before 2016", ]
  merged_df <- merged_df[merged_df$tmeschcov != "2016", ]
  merged_df <- merged_df[!is.na(merged_df$childcode), ]
  merged_df <- merged_df %>%
    mutate(tmeschcov = case_when(
      str_detect(tmeschcov, "2022") ~ "2022",
      str_detect(tmeschcov, "2021") ~ "2021",
      str_detect(tmeschcov, "2020") ~ "2020",
      TRUE ~ tmeschcov  # Keep original if no specific year is mentioned
    ))
}

  #For before 2019 and NK we check with the last wave 5 data to see if the new education is actually
new.

  #obtain highest educational status
  temp <- merged_df[merged_df$tmeschcov == "Before 2019" | merged_df$tmeschcov == "NK", ]
  library(dplyr)

  #Function to check matches within the same childcode
    check_matches_within_childcode <- function(temp_df, edu_df) {
      # Initialize an empty vector to store childcodes with matches
```

```
    matching_childcodes <- vector()

    # Iterate over each unique childcode in temp_df
    unique_childcodes <- unique(temp_df$childcode)

    for (code in unique_childcodes) {
      # Extract the lstgrdcov value(s) for this childcode in temp_df
      temp_values <- temp_df %>%
        filter(childcode == code) %>%
        pull(lstgrdcov)

      # Extract the edgrader values for this childcode in edu_df
      edu_values <- edu_df %>%
        filter(childcode == code) %>%
        pull(edgrader)

      # Check if any temp_values are in edu_values
      if (any(temp_values %in% edu_values)) {
        matching_childcodes <- c(matching_childcodes, code)
      }
    }

    return(matching_childcodes)
   }
  matching_childcodes <- check_matches_within_childcode(temp, wave15edu)
  new <- temp %>%
  filter(!childcode %in% matching_childcodes)

#remove values with before 2019 and NK, and add new to only include the new education. For those
we assume schoolyear is 2017-2018 for India and Vietnam. 2017 for peru and ethiopia

  country <- tolower(country)
  if (country %in% c("et", "pe")) {
  new <- new %>% mutate(tmeschcov = "2017")
  merged_df <- merged_df[merged_df$tmeschcov != "Before 2019", ]
  merged_df <- merged_df[merged_df$tmeschcov != "NK", ]
  merged_df <- bind_rows(merged_df, new)
  }

  if (country %in% c("vn", "in")) {
  new <- new %>% mutate(tmeschcov = "2017-2018")
  merged_df <- merged_df[merged_df$tmeschcov != "Before 2019", ]
  merged_df <- merged_df[merged_df$tmeschcov != "NK", ]
  merged_df <- bind_rows(merged_df, new)
  }

  #change output into final style
  final_df <- merged_df %>% dplyr::select(childcode, tmeschcov, lstschcov, lstgrdcov) %>%
rename(year = tmeschcov, edgrader = lstgrdcov, tyscr = lstschcov)

  return(final_df)
```

```
}
```

Function 5: To combine different waves
-> it combines the rows unlimited different dataframes
-> Removes unncessary variables
-> confirms factors to characters to avoid level mismatch
-> and checks for duplicating values

```{r}
merge.waves <- function(data1, data2, ..., col.remove = c("STATER", "DSSCR", "SCHNMER")) {
  # Convert factors to characters to avoid level mismatch issues
  data1[] <- lapply(data1, function(x) if(is.factor(x)) as.character(x) else x)
  data2[] <- lapply(data2, function(x) if(is.factor(x)) as.character(x) else x)

  # Combine all datasets
  all_data <- list(data1, data2, ...)
  all_data <- lapply(all_data, function(data) {
    if(is.factor(data)) as.character(data) else data
  })

  # Optionally handle additional datasets
  output <- bind_rows(all_data)
  names(output) <- tolower(names(output))

  # Remove duplicates
  key_cols <- names(output)[1:3]
  output <- output %>%
    mutate(na_count = rowSums(is.na(across(everything())))) %>%
    group_by(across(all_of(key_cols))) %>%
    mutate(min_na_count = min(na_count, na.rm = TRUE)) %>%
    filter(na_count == min_na_count) %>%
    dplyr::select(-na_count, -min_na_count) %>%
    ungroup()


  # Directly subset to remove specified columns if they exist in the data
  output <- output[, !names(output) %in% col.remove, drop = FALSE]
  return(output)
}

```

##2.2. Older Cohort
India
Everything is available
```{r}
ind <- datasets(country = "in", cohort = "oc")
list2env(ind, envir = .GlobalEnv)

edu_1999 <- in_oc_wave3 %>% pivot_longer(
```

```
    cols = starts_with("grder39"),
    names_to = "year",
    values_to = "edgrader") %>%  mutate(year = as.numeric(gsub("grder39", "", year)) + 1990) %>%
  left_join(# type of school 1999 AND before
      in_oc_wave3 %>%
        pivot_longer(
          cols = starts_with("tyscr39"),  # dplyr::select columns that start with 'tyscr
          values_to = "tyscr",        # New column for the tyscr score
          names_to = "year"           # New column for the year
        ) %>%
        mutate(year = as.numeric(gsub("tyscr39", "", year)) + 1990),
      by = c("childcode", "year"))  %>%
  mutate(year = paste(year, year+1, sep = "-")) %>% #indian schoolyears start at half-year
  mutate(year = as.factor(year)) %>%
  mutate(edgrader = ifelse(edgrader == "N/A", "Not known", edgrader)) %>%
  mutate(edgrader = ifelse(is.na(edgrader) == T, "Not known", edgrader)) %>%
  dplyr::select(year, edgrader, tyscr, childcode)

edu_2000 <- in_oc_wave3 %>%
  pivot_longer(
    cols = starts_with("grder30"),  # dplyr::select columns that start with 'grder30'
    names_to = "year",             # New column for the year
    values_to = "edgrader"         # New column for the educational grade
  ) %>%

  mutate(year = as.numeric(gsub("grder30", "", year)) + 2000) %>%
  left_join(# type of school 2000 AND AFTER
      in_oc_wave3 %>%
        pivot_longer(
          cols = starts_with("tyscr30"),  # dplyr::select columns that start with 'tyscr
          values_to = "tyscr",        # New column for the tyscr score
          names_to = "year"           # New column for the year
        ) %>%
        mutate(year = as.numeric(gsub("tyscr30", "", year)) + 2000), # Adjust year
      by = c("childcode", "year"))  %>%
  dplyr::select(year, edgrader, tyscr, childcode) %>%
  mutate(year = paste(year, year+1, sep = "-")) %>% #indian schoolyears start at half-year
  mutate(year = as.factor(year)) %>%
  mutate(edgrader = ifelse(edgrader == "N/A", "Not known", edgrader)) %>%
  mutate(edgrader = ifelse(is.na(edgrader) == T, "Not known", edgrader))

edu_wave4 <- in_oc_wave4 %>% dplyr::select(childcode, edchstr, edgrader, tyscr) %>% rename(year
= edchstr) %>% mutate(edgrader = if_else(edgrader == "NK", "Not known", edgrader)) %>%
mutate(edgrader = if_else(is.na(edgrader)==T, "Not known", edgrader))

edu_wave5 <- in_oc_wave5 %>% dplyr::select(childcode, edchstr, edgrader, tyscr) %>% rename(year
= edchstr) %>% mutate(edgrader = if_else(edgrader == "NK", "Not known", edgrader)) %>%
mutate(edgrader = if_else(is.na(edgrader)==T, "Not known", edgrader))

temp_wave15 <- merge.waves(edu_1999, edu_2000, edu_wave4, edu_wave5)
edu_recent <- w6.recentedu(in_oc_call2, in_oc_call3, in_oc_call5, country = "in")
```

```
edu_2017 <- w6.edu20172020(in_oc_call2, in_oc_call5, temp_wave15, country = "in")

edu_in_oc <- merge.waves(edu_1999, edu_2000, edu_wave4, edu_wave5, edu_2017, edu_recent)
temp <- names(ind)
remove(temp)
remove(edu_1999, edu_2000, edu_wave4, edu_wave5, edu_2017, edu_recent, temp_wave15)

```

Ethiopia
Lots of years missing. Yearly data available from 2009 onwards.

Years for wave 4 and 5 are off by seven years (!!!). This is corrected
Older cohort is not aged 18/19 in 2006

```{r}
et <- datasets(country = "et", cohort = "oc")
list2env(et, envir = .GlobalEnv)

edu_wave4 <- et_oc_wave4 %>% dplyr::select(childcode, edchstr, grder, tyscr) %>% rename(year = edchstr, edgrader = grder) %>% mutate(year = as.character(year)) %>% mutate(year = as.numeric(year) + 7)  %>% mutate(year = as.character(year)) %>% #adding seven years
mutate(edgrader = if_else(edgrader == "NK", "Not known", edgrader)) %>% mutate(edgrader = if_else(is.na(edgrader)==T, "Not known", edgrader))

edu_wave5 <- et_oc_wave5 %>% dplyr::select(childcode, edchstr, edgrader, tyscr) %>% rename(year = edchstr) %>% mutate(year = as.character(year)) %>% mutate(year = as.numeric(year) + 7) %>% mutate(year = as.character(year))

temp_wave15 <- merge.waves(edu_wave4, edu_wave5)
edu_recent <- w6.recentedu(et_oc_call2, et_oc_call3, et_oc_call5, country = "et")
edu_2017 <- w6.edu20172020(et_oc_call2, et_oc_call5, temp_wave15, country = "et")

edu_et_oc <- merge.waves(edu_wave4, edu_wave5, edu_2017, edu_recent)
remove(edu_wave4, edu_wave5, edu_2017, edu_recent, temp_wave15)
```

vietnam
Yearly data available from 2000 onwards.
No problems further.

```{r}
vn <- datasets(country = "vn", cohort = "oc")
list2env(vn, envir = .GlobalEnv)

edu_wave4 <- vn_oc_wave4 %>% dplyr::select(childcode, edchstr, edgrader, tyscr) %>% rename(year = edchstr) %>% mutate(edgrader = if_else(edgrader == "NK", "Not known", edgrader)) %>% mutate(edgrader = if_else(is.na(edgrader)==T, "Not known", edgrader))

edu_wave5 <- vn_oc_wave5 %>% dplyr::select(childcode, edchstr, edgrader, tyscr) %>% rename(year = edchstr)
```

temp_wave15 <- merge.waves(edu_wave4, edu_wave5)
edu_recent <- w6.recentedu(vn_oc_call2, vn_oc_call3, vn_oc_call5, country = "vn")
edu_2017 <- w6.edu20172020(vn_oc_call2, vn_oc_call5, temp_wave15, country = "vn")

edu_vn_oc <- merge.waves(edu_wave4, edu_wave5, edu_2017, edu_recent)
remove(edu_wave4, edu_wave5, edu_2017, edu_recent, temp_wave15)
```

Peru
Data missing, available from 2009 onwards '
Also partly in spanish

```{r}
peru <- datasets(country = "pe", cohort = "oc")
list2env(peru, envir = .GlobalEnv)

#For peru, the coding for non-attending students is not yet done.
edu_wave4 <- pe_oc_wave4 %>% dplyr::select(childcode, edchstr, grder, tyscr, atdschr) %>%
rename(year = edchstr, edgrader = grder, atscr = atdschr) %>%
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader))

#for Wave 5 non-attending students have been coded "none", so NA is unknown
edu_wave5 <- pe_oc_wave5 %>% dplyr::select(childcode, edchstr, grder, tyscr) %>% rename(year =
edchstr, edgrader = grder) %>% mutate(edgrader = if_else(is.na(edgrader)==T, "Not known",
edgrader)) %>% mutate(edgrader = if_else(edgrader == "NK", "Not known", edgrader))

temp_wave15 <- merge.waves(edu_wave4, edu_wave5)
edu_recent <- w6.recentedu(pe_oc_call2, pe_oc_call3, pe_oc_call5, country = "pe")
edu_2017 <- w6.edu20172020(pe_oc_call2, pe_oc_call5, temp_wave15, country = "pe")

edu_pe_oc <- merge.waves(edu_wave4, edu_wave5, edu_2017, edu_recent)
remove(edu_wave4, edu_wave5, edu_2017, edu_recent, temp_wave15)
```

##2.3. Younger Cohort
India
2010 - 2017 yearly
then we have yearly for 2020 and 2021 and highest grade achieved in between.
However, call 5 from wave 6 has no date of interview (data error), so we put that data on 1-11-2021 for all, similar to the dates in other countries and for the older cohort

```{r}
ind_yc <- datasets(country = "in", cohort = "yc")
list2env(ind_yc, envir = .GlobalEnv)

in_yc_call5 <- in_yc_call5 %>% mutate(dint = "01-Nov-21")

#Mutate functions are to replace current grade with None if not in school, or unknown if the variable is unknown.

```
edu_wave4 <- in_yc_wave4 %>% dplyr::select(childcode, edchstr, grder, tyscr, atscr) %>%
rename(year = edchstr, edgrader = grder) %>%
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
mutate(edgrader = if_else(atscr %in% c("N/A", "NA", "NK", "Refused to answer"), "Not known",
edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader)) %>% mutate(edgrader =
if_else(is.na(edgrader) == T, if_else(atscr %in% c("Yes", "S"), "In school, unknown grade", edgrader),
edgrader)) %>% dplyr::select(-atscr)

edu_wave5 <- in_yc_wave5 %>% dplyr::select(childcode, edchstr, edgrader, tyscr, atdschr) %>%
rename(year = edchstr, atscr = atdschr) %>%
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
mutate(edgrader = if_else(atscr %in% c("N/A", "NA", "NK", "Refused to answer"), "Not known",
edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader)) %>%
mutate(edgrader = if_else(is.na(edgrader) == T, if_else(atscr %in% c("Yes", "S"), "In school, unknown
grade", edgrader), edgrader)) %>% dplyr::select(-atscr)

temp_wave15 <- merge.waves(edu_wave4, edu_wave5)
edu_recent <- w6.recentedu(in_yc_call2, in_yc_call3, in_yc_call5, country = "in")
edu_2017 <- w6.edu20172020(in_yc_call2, in_yc_call5, temp_wave15, country = "in")

edu_in_yc <- merge.waves(edu_wave4, edu_wave5, edu_2017, edu_recent)
remove(edu_wave4, edu_wave5, edu_2017, edu_recent, temp_wave15)
```

ethiopia
2002-2016 yearly
then when available.

Years for wave 4 and 5 are off by seven years (!!!). This is corrected

```{r}
et_yc <- datasets(country = "et", cohort = "yc")
list2env(et_yc, envir = .GlobalEnv)

edu_wave4 <- et_yc_wave4 %>% dplyr::select(childcode, edchstr, grder, tyscr, atscr) %>%
rename(year = edchstr, edgrader = grder) %>% mutate(year = as.character(year)) %>% mutate(year =
as.numeric(year) + 7)  %>% mutate(year = as.character(year))  %>%
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
mutate(edgrader = if_else(atscr %in% c("N/A", "NA", "NK", "Refused to answer"), "Not known",
edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader)) %>% mutate(edgrader =
if_else(is.na(edgrader) == T, if_else(atscr %in% c("Yes", "S"), "In school, unknown grade", edgrader),
edgrader)) %>% dplyr::select(-atscr)

edu_wave5 <- et_yc_wave5 %>% dplyr::select(childcode, edchstr, edgrader, tyscr, atdschr) %>%
rename(year = edchstr, atscr = atdschr) %>% mutate(year = as.character(year)) %>% mutate(year =
as.numeric(year) + 7)  %>% mutate(year = as.character(year)) %>%
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
```

```
mutate(edgrader = if_else(atscr %in% c("N/A", "NA", "NK", "Refused to answer"), "Not known",
edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader)) %>%
mutate(edgrader = if_else(is.na(edgrader) == T, if_else(atscr %in% c("Yes", "S"), "In school, unknown
grade", edgrader), edgrader)) %>% dplyr::select(-atscr)

temp_wave15 <- merge.waves(edu_wave4, edu_wave5)
edu_recent <- w6.recentedu(et_yc_call2, et_yc_call3, et_yc_call5, country = "et") %>%
mutate(edgrader = if_else(is.na(edgrader)==T, "Not known", edgrader))
edu_2017 <- w6.edu20172020(et_yc_call2, et_yc_call5, temp_wave15, country = "et") %>%
mutate(edgrader = if_else(is.na(edgrader)==T, "Not known", edgrader))

edu_et_yc <- merge.waves(edu_wave4, edu_wave5, edu_2017, edu_recent)
remove(edu_wave4, edu_wave5, edu_2017, edu_recent, temp_wave15)

```

peru
2009-2016 yearly
then when available.

Some of the data is not translated

```{r}
pe_yc <- datasets(country = "pe", cohort = "yc")
list2env(pe_yc, envir = .GlobalEnv)

edu_wave4 <- pe_yc_wave4 %>% dplyr::select(childcode, edchstr, grder, tyscr, atscr) %>%
rename(year = edchstr, edgrader = grder)  %>%
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
mutate(edgrader = if_else(atscr %in% c("N/A", "NA", "NK", "Refused to answer"), "Not known",
edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader)) %>% mutate(edgrader =
if_else(is.na(edgrader) == T, if_else(atscr %in% c("Yes", "S"), "In school, unknown grade", edgrader),
edgrader)) %>% dplyr::select(-atscr)

edu_wave5 <- pe_yc_wave5 %>%dplyr::select(childcode, edchstr, grder, tyscr, atscr) %>%
rename(year = edchstr, edgrader = grder) %>%
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
mutate(edgrader = if_else(atscr %in% c("N/A", "NA", "NK", "Refused to answer"), "Not known",
edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader)) %>%
mutate(edgrader = if_else(is.na(edgrader) == T, if_else(atscr %in% c("Yes", "S"), "In school, unknown
grade", edgrader), edgrader)) %>% dplyr::select(-atscr)

temp_wave15 <- merge.waves(edu_wave4, edu_wave5)
edu_recent <- w6.recentedu(pe_yc_call2, pe_yc_call3, pe_yc_call5, country = "pe") %>%
mutate(edgrader = if_else(is.na(edgrader)==T, "Not known", edgrader))
edu_2017 <- w6.edu20172020(pe_yc_call2, pe_yc_call5, temp_wave15, country = "pe") %>%
mutate(edgrader = if_else(is.na(edgrader)==T, "Not known", edgrader))
```

edu_pe_yc <- merge.waves(edu_wave4, edu_wave5, edu_2017, edu_recent)
remove(edu_wave4, edu_wave5, edu_2017, edu_recent, temp_wave15)
```

vietnam
2005-2017 regularly
then similar to before
```{r}
vn_yc <- datasets(country = "vn", cohort = "yc")
list2env(vn_yc, envir = .GlobalEnv)

edu_wave4 <- vn_yc_wave4 %>% dplyr::select(childcode, edchstr, grder, tyscr, atscr) %>%
rename(year = edchstr, edgrader = grder) %>%
  mutate(year = str_replace(year, "-([0-9]{2})$", function(x) paste0("-20", substring(x, 2)))) %>%
#ensures year is comparable to other countries
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
mutate(edgrader = if_else(atscr %in% c("N/A", "NA", "NK", "Refused to answer"), "Not known",
edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader)) %>% mutate(edgrader =
if_else(is.na(edgrader) == T, if_else(atscr %in% c("Yes", "S"), "In school, unknown grade", edgrader),
edgrader)) %>% dplyr::select(-atscr)

edu_wave5 <- vn_yc_wave5 %>% dplyr::select(childcode, edchstr, grder, tyscr, atscr) %>%
rename(year = edchstr, edgrader = grder)  %>%
  mutate(year = str_replace(year, "-([0-9]{2})$", function(x) paste0("-20", substring(x, 2)))) %>%
mutate(edgrader = if_else(atscr == "No", "None", edgrader)) %>%
mutate(edgrader = if_else(atscr %in% c("N/A", "NA", "NK", "Refused to answer"), "Not known",
edgrader)) %>%
mutate(edgrader = if_else(is.na(atscr) == T, "Not known", edgrader)) %>%
mutate(edgrader = if_else(is.na(edgrader) == T, if_else(atscr %in% c("Yes", "S"), "In school, unknown
grade", edgrader), edgrader)) %>% dplyr::select(-atscr)

temp_wave15 <- merge.waves(edu_wave4, edu_wave5)
edu_recent <- w6.recentedu(vn_yc_call2, vn_yc_call3, vn_yc_call5, country = "vn")
edu_2017 <- w6.edu20172020(vn_yc_call2, vn_yc_call5, temp_wave15, country = "vn")

edu_vn_yc <- merge.waves(edu_wave4, edu_wave5, edu_2017, edu_recent)
remove(edu_wave4, edu_wave5, edu_2017, edu_recent, temp_wave15)
```

##2.4. Saving Datasets
```{r}
write.csv(edu_in_oc, "proc_data/edu_in_oc.csv", row.names = FALSE)
write.csv(edu_in_yc, "proc_data/edu_in_yc.csv", row.names = FALSE)

write.csv(edu_et_oc, "proc_data/edu_et_oc.csv", row.names = FALSE)
write.csv(edu_et_yc, "proc_data/edu_et_yc.csv", row.names = FALSE)

write.csv(edu_pe_oc, "proc_data/edu_pe_oc.csv", row.names = FALSE)
write.csv(edu_pe_yc, "proc_data/edu_pe_yc.csv", row.names = FALSE)

```
write.csv(edu_vn_oc, "proc_data/edu_vn_oc.csv", row.names = FALSE)
write.csv(edu_vn_yc, "proc_data/edu_vn_yc.csv", row.names = FALSE)
```

2.5. Transforming long dataset into useful dummies
The next step will create educational dummies, since the educational system differs slightly between countries, we will do this seperately per country.

Ethiopia
non-formal: religious education, kindergarden, creche day-care
Vocational: TVET 1st, 2nd, 3th, 4th year of TVET
    ## The exact years seem a bit random
    ## Most people enroll in TVET
Primary: grade 1-8
Lower secondary: first cycle, grade 9/10
Upper secondary: second cycle, grade 11/12
College: all cycles of primary teaching certificates and pre-school teaaching certificates
University: Secondary teaching, undergraduates and masters
  (probably also includes technical diploma's since those aren't mentioned seperately)

```{r}
edu_et <- bind_rows(read_csv("proc_data/edu_et_oc.csv"), read_csv("proc_data/edu_et_yc.csv"))
%>% mutate(year = as.character(year))

library(dplyr)
library(stringr)
### Cleaning Ethiopia
# Function to clean and precisely categorize education history
adapt_edu_et <- function(x) {
  case_when(
    x == "second cycle of primary teaching certificate grade 8" ~ "second cycle of primary teaching certificate year 2",
    x == "first cycle of primary teaching certificate grade 1" ~ "first cycle of primary teaching certificate year 1",
    x == "secondary education, teacher (diploma)" ~ "secondary education teacher diploma",
    x == "second cycle of primary teaching certificate grade 7" ~ "second cycle of primary teaching certificate year 1",
    x == "first cycle of primary teaching certificate grade 2" ~ "first cycle of primary teaching certificate year 2",
    x == "grade 11 (secondary second cycle preparatory programme)" ~ "grade 11",
    x == "grade 12 (secondary second cycle preparatory programme)" ~ "grade 12",
    x == "first cycle of primary teaching certificate  (grade 1-4)/1st year" ~ "first cycle of primary teaching certificate year 1",
    x == "first cycle of primary teaching certificate  (grade 1-4)/2nd year" ~ "first cycle of primary teaching certificate year 2",
    x == "first cycle of primary teaching certificate (grade 1-4)/1st" ~ "first cycle of primary teaching certificate year 1",
    x == "first cycle of primary teaching certificate (grade 1-4)/2nd" ~ "first cycle of primary teaching certificate year 2",
```

```
    x == "second cycle of primary teaching certificate (grades 5-8)/1st year" ~ "second cycle of primary
teaching certificate year 1",
    x == "second cycle of primary teaching certificate (grades 5-8)/2nd year" ~ "second cycle of
primary teaching certificate year 2",
    x == "second cycle of primary teaching certificate (grades 5-8)/1s" ~ "second cycle of primary
teaching certificate year 1",
    x == "preschool teacher certificate (6 months to one year)" ~ "preschool teacher certificate",
    x == "grade 10 (secondary first cycle)" ~ "grade 10",
    x == "grade 9 (secondary first cycle)" ~ "grade 9",
    x == "kindergarten (kg)" ~ "(pre)-kindergarten",
    x == "pre-kg/nursery" ~ "(pre)-kindergarten",
    x == "creche/day-care" ~ "(pre)-kindergarten",
    x == "00 = none" ~ "none",
    x == "not known" ~ NA_character_,
    x == "not applicable" ~ NA_character_,
    x == "secondary education, teacher (bachelor's degree holder and above)/1st or second or 3rd year"
~ "secondary education teacher diploma",
    x == "secondary education, teacher (diploma holder)/1st or 2nd year" ~ "secondary education
teacher diploma",
     x == "secondary education, teacher (diploma holder)/1st or 2nd/ year" ~ "secondary education
teacher diploma",
    x == "secondary education, teacher (diploma holder)/1st or 2nd/ ye" ~ "secondary education teacher
diploma",
    x == "secondary education, teacher (bachelor's degree holder and above)/1st or 2nd or 3rd year" ~
"secondary education teacher diploma",
    x == "secondary education, teacher (bachelor's degree holder and a" ~ "secondary education teacher
diploma",
    x == "secondary education, teacher (bachelor's degree holder and above)/1st year" ~ "secondary
education teacher diploma year 1",
    x == "secondary education, teacher (bachelor's degree holder and above)/2nd year" ~ "secondary
education teacher diploma year 2",
    x == "secondary education, teacher (bachelor's degree holder and above)/3rd year" ~ "secondary
education teacher diploma year 3",
    x == "secondary education, teacher (diploma holder)/1st year" ~ "secondary education teacher
diploma year 1",
    x == "secondary education, teacher (diploma holder)/2nd year" ~ "secondary education teacher
diploma year 2",
    x %in% c("other", "other, specify") ~ "other",
    TRUE ~ x  # Default to return original if no match
  )
}

clean_education_category <- function(x) {
  x <- tolower(x) %>% stringr::str_trim()  # Convert to lower case and trim white spaces

  # Standardize kindergarten entries
  x <- str_replace_all(x, "kindergarden", "kindergarten")

  # Correct and unify grade entries, removing leading zeros and extra descriptions
  x <- str_replace_all(x, "\\bgrade 0?(\\d+)\\b", "grade \\1")
  x <- str_replace_all(x, "\\bgrade (\\d+) \\(primary\\)", "grade \\1")
```

```
  x <- str_replace_all(x, "\\bprimary grade (\\d+)\\b", "grade \\1")

  # Handle TVET entries to keep the year information
  x <- str_replace_all(x, "tvet/\\d+.*year.*", function(m) {
    year <- str_extract(m, "\\d+")
    paste0("TVET year ", year)
  })

  # Normalize secondary grade entries
  x <- str_replace_all(x, "secondary first cycle grade (\\d+)", "grade \\1")
  x <- str_replace_all(x, "secondary second cycle preparatory programme grade (\\d+)", "grade \\1")

  # Specifically handle undergraduate degree entries
  # Extract and keep the year, stripping away unnecessary descriptions
  x <- str_replace_all(x, "undergraduate degree.*?(\\d+)\\w*\\syear.*", "undergraduate year \\1")
  x <- str_replace_all(x, "undergraduate year (\\d+)\\s.*", "undergraduate year \\1")

  x
}

edu_et$edgrader_new = edu_et$edgrader
edu_et$edgrader_new = clean_education_category(edu_et$edgrader_new)
edu_et$edgrader_new = adapt_edu_et(edu_et$edgrader_new)
table(edu_et$edgrader_new)

#create a dataframe to be used for merging
edu_et_dummies <- edu_et %>% group_by(childcode) %>% slice(n()) %>% ungroup() %>%
dplyr::select(childcode)

#Now we will use that to create dummies for whether they completed different levels
## Be careful year 2021 could mean they are still studying

#step 1: obtaining last formal grade and last year of study (highest grade is too difficult)
#Getting last known grade, and last known year of studying
edu_et_cleaned <- edu_et %>%
  filter(!is.na(edgrader_new) & !edgrader_new %in% c("none", "religious education", "other")) %>%
  arrange(childcode, year)

#if everything is equal to none, then we need a seperate flag indicator
temp <- edu_et %>%
  group_by(childcode) %>%
  mutate(all_na = if_else(all(is.na(edgrader_new) | edgrader_new == "none"), 1, 0)) %>%
  slice(n()) %>%
  ungroup() %>% dplyr::select(childcode, all_na)

last_grade <- edu_et_cleaned %>%
  group_by(childcode) %>%
  slice(n()) %>%
  ungroup() %>% rename(lastgrade = edgrader_new, lastyearstudy = year) %>% dplyr::select(
    childcode, lastgrade, lastyearstudy)
  #be careful 2021 could likely mean they are still studying
```

```
#combining with dummies
edu_et_dummies <- edu_et_dummies %>% full_join(last_grade, by = "childcode") %>%
full_join(temp, by = "childcode") %>% mutate(
    lastgrade = ifelse(all_na == 1, "none", lastgrade),
    lastyearstudy = ifelse(all_na == 1, "never", lastyearstudy)
)

remove(temp, last_grade, edu_et_cleaned)

#step 2: creating dummies whether they completed pre-defined educational levels for primary and
secondary, this method is not possible for higher education due to too many YC still not having
achieved that level of education

#We do not have consistent dummies on whether they have succesfully completed the grade, thus
enrolling in the final grade is considered "completion
edu_et_dummies <- edu_et_dummies %>% mutate(
  attended_formal_education = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade == "none" ~ 0,
    TRUE ~ 1
  ),
  completed_primary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade
7") ~ 0,
    TRUE ~ 1
  ),
   completed_general_lower_secondary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade
7", "grade 8", "grade 9") ~ 0,
    TRUE ~ 1
  ),
  completed_general_upper_secondary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade
7", "grade 8", "grade 9", "grade 10", "grade 11") ~ 0,
    TRUE ~ 1
  )
)

#step 3: Obtaining higher education enrollment
temp <- edu_et %>%
  group_by(childcode) %>%
  mutate(
  attended_vocational_secondary = case_when(
    all(is.na(edgrader_new)) == T ~ NA,
    any(grepl("TVET", edgrader_new)) == T ~ 1,
    TRUE ~ 0),
  attended_kindergarten = case_when(
```

```
    all(is.na(edgrader_new)) == T ~ NA,
    any(grepl("(pre)-kindergarten", edgrader_new)) == T ~ 1,
    TRUE ~ 0),
  attended_college = case_when(
    all(is.na(edgrader_new)) == T ~ NA,
    any(grepl("primary teaching certificate", edgrader_new)) == T ~ 1,
    any(grepl("preschool teacher certificate", edgrader_new)) == T ~ 1,
    TRUE ~ 0),
  attended_university = case_when(
    all(is.na(edgrader_new)) == T ~ NA,
    any(grepl("undergraduate", edgrader_new)) == T ~ 1,
    any(grepl("secondary education teacher diploma", edgrader_new)) == T ~ 1,
    any(grepl("masters or doctoral at university", edgrader_new)) == T ~ 1,
    TRUE ~ 0),
  attended_graduate = case_when(
    all(is.na(edgrader_new)) == T ~ NA,
    any(grepl("masters or doctoral at university", edgrader_new)) == T ~ 1,
    TRUE ~ 0),
  attended_higher_education = case_when(
    all(is.na(edgrader_new)) == T ~ NA,
    attended_college == 1 ~ 1,
    attended_university == 1 ~ 1,
    attended_graduate == 1 ~ 1,
    TRUE ~ 0)) %>%
  ungroup()
```

#step 4 obtaining years of graduation and enrollment when possible, and creating other helpful dummies

#Years are an approximation -> if between 2016 and 2019 the data does not show the precise year, so it could be the same, or the exact grade could be missing, which is why we add a few extra just in case.

```
temp <- temp %>%
  mutate(year = as.numeric(year)) %>%  # Convert 'year' to numeric to ensure calculations are correct
  arrange(childcode, year) %>%
  mutate(
    completed_primary = grepl("grade 8|grade 9|grade 10|grade 11|grade 12|undergraduate|masters or doctoral at university|primary teaching certificate|preschool teacher certificate", edgrader_new, ignore.case = TRUE),
    completed_lower_sec = grepl("grade 10|grade 11|grade 12|undergraduate|masters or doctoral at university|primary teaching certificate|preschool teacher certificate", edgrader_new, ignore.case = TRUE),
    completed_upper_sec = grepl("grade 12|undergraduate|masters or doctoral at university|primary teaching certificate|preschool teacher certificate", edgrader_new, ignore.case = TRUE),
    enrolled_college = grepl("primary teaching certificate|preschool teacher certificate", edgrader_new, ignore.case = TRUE),
    enrolled_university = grepl("undergraduate|secondary education teacher diploma|masters or doctoral at university", edgrader_new, ignore.case = TRUE),
    attended_voc_sec_thisyear = grepl("TVET", edgrader_new, ignore.case = TRUE)
  ) %>%
  group_by(childcode) %>%
```

```
  mutate(
    year_graduated_primary = if(any(completed_primary, na.rm = TRUE))
       min(year[completed_primary], na.rm = TRUE) else NA_real_,
    year_graduated_lower_sec = if(any(completed_lower_sec, na.rm = TRUE))
      min(year[completed_lower_sec], na.rm = TRUE) else NA_real_,
    year_graduated_upper_sec = if(any(completed_upper_sec, na.rm = TRUE))
      min(year[completed_upper_sec], na.rm = TRUE) else NA_real_,
    year_graduated_college = if(any(enrolled_college, na.rm = TRUE))
      max(year[enrolled_college], na.rm = TRUE) else NA_real_,
    year_graduated_university = if(any(enrolled_university, na.rm = TRUE))
      max(year[enrolled_university], na.rm = TRUE) else NA_real_,
    first_tvet_year = if(any(attended_voc_sec_thisyear, na.rm = TRUE))
      min(year[attended_voc_sec_thisyear], na.rm = TRUE) else NA_real_,
    last_tvet_year = if(any(attended_voc_sec_thisyear, na.rm = TRUE))
      max(year[attended_voc_sec_thisyear], na.rm = TRUE) else NA_real_,
    graduated_gen_primary_before_tvet = if_else(
      !is.na(year_graduated_primary) & !is.na(first_tvet_year) &
        year_graduated_primary < first_tvet_year, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_)),
    graduated_gen_lower_secondary_before_tvet = if_else(
      !is.na(year_graduated_lower_sec) & !is.na(first_tvet_year) &
        year_graduated_lower_sec < first_tvet_year, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_)),
    graduated_gen_upper_secondary_before_tvet = if_else(
      !is.na(year_graduated_upper_sec) & !is.na(first_tvet_year) &
        year_graduated_upper_sec < first_tvet_year, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_))) %>%
  ungroup()

#dplyr::select the relevant variables and compress into one row
temp <- temp %>% dplyr::select(childcode, attended_vocational_secondary, attended_kindergarten,
attended_college, attended_university, attended_graduate, attended_higher_education,
year_graduated_primary, year_graduated_lower_sec, year_graduated_upper_sec,
year_graduated_college, year_graduated_university, first_tvet_year, last_tvet_year,
graduated_gen_primary_before_tvet, graduated_gen_lower_secondary_before_tvet,
graduated_gen_upper_secondary_before_tvet)

temp <- temp %>% group_by(childcode) %>% slice(n())

edu_et_dummies <- edu_et_dummies %>% full_join(temp, by = "childcode") %>%
  mutate(continued_after_upper_secondary = case_when(
    is.na(attended_vocational_secondary) == T | is.na(attended_higher_education == T) ~ NA,
    attended_vocational_secondary == 1 & attended_higher_education == 1 ~ 1,
    completed_general_upper_secondary == 1 & attended_higher_education == 1 ~ 1,
    attended_vocational_secondary == 1 & attended_higher_education == 0 ~ 1,
    completed_general_upper_secondary == 1 & attended_higher_education == 0 ~ 1,
    TRUE ~ NA
  ))

remove(temp)
```

India
non-formal: religious education, adult literacy
Vocational: labelled only "vocational" is a bit unclear what the exact indian name of the school is. Students can enroll with only lower secondary diploma -> so an alternative to upper-secondary.
Primary: grade 1-8
Lower secondary: grade 9/10
Upper secondary: grade 11/12
College: technical colleges
University: undergraduates and masters

Importantly there are a lot less years missing here, so data between 2016 and 2020 is more reliable

```{r}
edu_in <- bind_rows(read_csv("proc_data/edu_in_oc.csv"), read_csv("proc_data/edu_in_yc.csv"))

adapt_edu_in <- function(x) {
  case_when(
    x %in% c("Not known", "Not Known", "NK", "Refused to answer", "In school, unknown grade") ~ NA_character_,
    x %in% c("1", "Grade 1", "Grade 01") ~ "grade 1",
    x %in% c("2", "Grade 2", "Grade 02") ~ "grade 2",
    x %in% c("3", "Grade 3", "Grade 03") ~ "grade 3",
    x %in% c("4", "Grade 4", "Grade 04") ~ "grade 4",
    x %in% c("5", "Grade 5", "Grade 05", "Primary (Class 5)") ~ "grade 5",
    x %in% c("6", "Grade 6", "Grade 06") ~ "grade 6",
    x %in% c("7", "Grade 7", "Grade 07") ~ "grade 7",
    x %in% c("8", "Grade 8", "Grade 08") ~ "grade 8",
    x %in% c("9", "Grade 9", "Grade 09") ~ "grade 9",
    x %in% c("10", "Grade 10", "Matriculation certificate (Class 10)") ~ "grade 10",
    x %in% c("11", "Grade 11") ~ "grade 11",
    x %in% c("12", "Grade 12", "Senior Secondary school certificate /  Intermediate certific", "13") ~ "grade 12",
    x == "University degree (graduate)" ~ "undergraduate",
    x == "None" ~ "none",
    x == "Vocational" ~ "vocational",
    x == "Post-secondary / technological institute" ~ "technical college",
    x == "Univeristy degree (postgraduate)" ~ "masters or doctoral at university",
    x == "Post-secondary technological institute" ~ "technical college",
    x == "Religious education" ~ "religious education",
    x == "Adult literacy" ~ "adult literacy",
    x == "Post-graduate" ~ "masters or doctoral at university",
    x == "Other, specify" ~ "other",
    x == "Diploma in technical education" ~ "technical college",
    x == "Post-graduate university degree (completed)" ~ "masters or doctoral at university",
    x == "Post-graduate university degree (second year)" ~ "masters or doctoral at university",
    x == "University degree (third year under graduate)" ~ "undergraduate",
    x == "University degree (second year under graduate)" ~ "undergraduate",
    x == "Post-graduate university degree (first year)" ~ "masters or doctoral at university",
    x == "University degree (first year under graduate)" ~ "masters or doctoral at university",
```

```
    x == "No" ~ "none",
    x == "Other" ~ "other",
    x == "University degree (completed)" ~ "undergraduate",
    TRUE ~ x  # Default case to return the input if no conditions are matched
  )
}

edu_in$edgrader_new = edu_in$edgrader
edu_in$edgrader_new = adapt_edu_in(edu_in$edgrader_new)
table(edu_in$edgrader_new)

#create a dataframe to be used for merging
edu_in_dummies <- edu_in %>% group_by(childcode) %>% slice(n()) %>% ungroup() %>%
dplyr::select(childcode)

#Now we will use that to create dummies for whether they completed different levels
## Be careful year 2021 could mean they are still studying

#step 1: obtaining last formal grade and last year of study (highest grade is too difficult)
#Getting last known grade, and last known year of studying
edu_in_cleaned <- edu_in %>%
  filter(!is.na(edgrader_new) & !edgrader_new %in% c("none", "religious education", "other", "adult
literacy")) %>%
  arrange(childcode, year)

#if everything is equal to none, then we need a seperate flag indicator
temp <- edu_in %>%
  group_by(childcode) %>%
  mutate(all_na = if_else(all(is.na(edgrader_new) | edgrader_new == "none"), 1, 0)) %>%
  slice(n()) %>%
  ungroup() %>% dplyr::select(childcode, all_na)

last_grade <- edu_in_cleaned %>%
  group_by(childcode) %>%
  slice(n()) %>%
  ungroup() %>% rename(lastgrade = edgrader_new, lastyearstudy = year) %>% dplyr::select(
    childcode, lastgrade, lastyearstudy)
#be careful 2021 could likely mean they are still studying

#combining with dummies
edu_in_dummies <- edu_in_dummies %>% full_join(last_grade, by = "childcode") %>%
full_join(temp, by = "childcode") %>% mutate(
  lastgrade = ifelse(all_na == 1, "none", lastgrade),
  lastyearstudy = ifelse(all_na == 1, "never", lastyearstudy)
)

remove(temp, last_grade, edu_in_cleaned)

#step 2: creating dummies whether they completed pre-defined educational levels for primary and
secondary, this method is not possible for higher education due to too many YC still not having
achieved that level of education
```

```
#We do not have consistent dummies on whether they have succesfully completed the grade, thus
enrolling in the final grade is considered "completion
edu_in_dummies <- edu_in_dummies %>% mutate(
  attended_formal_education = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade == "none" ~ 0,
    TRUE ~ 1
  ),
  completed_primary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade
7") ~ 0,
    TRUE ~ 1
  ),
  completed_general_lower_secondary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade
7", "grade 8", "grade 9") ~ 0,
    TRUE ~ 1
  ),
  completed_general_upper_secondary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade
7", "grade 8", "grade 9", "grade 10", "grade 11") ~ 0,
    TRUE ~ 1
  )
)

#step 3: Obtaining higher education enrollment
temp <- edu_in %>%
  group_by(childcode) %>%
  mutate(
    attended_vocational_secondary = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("vocational", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_kindergarten = NA, #no information for india on kindergarten,
    attended_college = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("technical college", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_university = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("undergraduate", edgrader_new)) == T ~ 1,
      any(grepl("masters or doctoral at university", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_graduate = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("masters or doctoral at university", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
```

```
    attended_higher_education = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      attended_college == 1 ~ 1,
      attended_university == 1 ~ 1,
      attended_graduate == 1 ~ 1,
      TRUE ~ 0)) %>%
  ungroup()

#step 4 obtaining years of graduation and enrollment when possible, and creating other helpful
dummies

#Years are an approximation -> if between 2016 and 2019 the data does not show the precise year, so
it could be the same, or the exact grade could be missing, which is why we add a few extra just in
case.
temp <- temp %>%
  mutate(year_con = substr(year, 1, 4)) %>%
  mutate(year_con = as.numeric(year_con)) %>%  # Convert 'year_con' to numeric to ensure
calculations are correct
  arrange(childcode, year_con) %>%
  mutate(
    completed_primary = grepl("grade 8|grade 9|grade 10|grade 11|grade 12|undergraduate|masters or
doctoral at university|technical college", edgrader_new, ignore.case = TRUE),
    completed_lower_sec = grepl("grade 10|grade 11|grade 12|undergraduate|masters or doctoral at
university|technical college", edgrader_new, ignore.case = TRUE),
    completed_upper_sec = grepl("grade 12|undergraduate|masters or doctoral at university|technical
college", edgrader_new, ignore.case = TRUE),
    enrolled_college = grepl("technical college", edgrader_new, ignore.case = TRUE),
    enrolled_university = grepl("undergraduate|masters or doctoral at university", edgrader_new,
ignore.case = TRUE),
    attended_voc_sec_thisyear_con = grepl("vocational", edgrader_new, ignore.case = TRUE)
  ) %>%
  group_by(childcode) %>%
  mutate(
    year_con_graduated_primary = if(any(completed_primary, na.rm = TRUE))
      min(year_con[completed_primary], na.rm = TRUE) else NA_real_,
    year_con_graduated_lower_sec = if(any(completed_lower_sec, na.rm = TRUE))
      min(year_con[completed_lower_sec], na.rm = TRUE) else NA_real_,
    year_con_graduated_upper_sec = if(any(completed_upper_sec, na.rm = TRUE))
      min(year_con[completed_upper_sec], na.rm = TRUE) else NA_real_,
    year_con_graduated_college = if(any(enrolled_college, na.rm = TRUE))
      max(year_con[enrolled_college], na.rm = TRUE) else NA_real_,
    year_con_graduated_university = if(any(enrolled_university, na.rm = TRUE))
      max(year_con[enrolled_university], na.rm = TRUE) else NA_real_,
    first_tvet_year_con = if(any(attended_voc_sec_thisyear_con, na.rm = TRUE))
      min(year_con[attended_voc_sec_thisyear_con], na.rm = TRUE) else NA_real_,
    last_tvet_year_con = if(any(attended_voc_sec_thisyear_con, na.rm = TRUE))
      max(year_con[attended_voc_sec_thisyear_con], na.rm = TRUE) else NA_real_,
    graduated_gen_primary_before_tvet = if_else(
      !is.na(year_con_graduated_primary) & !is.na(first_tvet_year_con) &
        year_con_graduated_primary < first_tvet_year_con, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_)),
```

```
    graduated_gen_lower_secondary_before_tvet = if_else(
      !is.na(year_con_graduated_lower_sec) & !is.na(first_tvet_year_con) &
        year_con_graduated_lower_sec < first_tvet_year_con, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_)),
    graduated_gen_upper_secondary_before_tvet = if_else(
      !is.na(year_con_graduated_upper_sec) & !is.na(first_tvet_year_con) &
        year_con_graduated_upper_sec < first_tvet_year_con, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_))) %>%
    mutate( #convert back to halfyear years
    year_graduated_primary = if_else(
      !is.na(year_con_graduated_primary),
      paste(year_con_graduated_primary, year_con_graduated_primary + 1, sep = "-"),
      NA_character_ ),
    year_graduated_lower_sec = if_else(
      !is.na(year_con_graduated_lower_sec),
      paste(year_con_graduated_lower_sec, year_con_graduated_lower_sec + 1, sep = "-"),
      NA_character_),
    year_graduated_upper_sec = if_else(
      !is.na(year_con_graduated_upper_sec),
      paste(year_con_graduated_upper_sec, year_con_graduated_upper_sec + 1, sep = "-"),
      NA_character_),
    year_graduated_college = if_else(
      !is.na(year_con_graduated_college),
      paste(year_con_graduated_college, year_con_graduated_college + 1, sep = "-"),
      NA_character_ ),
     year_graduated_university = if_else(
      !is.na(year_con_graduated_university),
      paste(year_con_graduated_university, year_con_graduated_university + 1, sep = "-"),
      NA_character_ ),
    first_tvet_year = if_else(
      !is.na(first_tvet_year_con),
      paste(first_tvet_year_con, first_tvet_year_con + 1, sep = "-"),
      NA_character_ ),
    last_tvet_year = if_else(
      !is.na(last_tvet_year_con),
      paste(last_tvet_year_con, last_tvet_year_con + 1, sep = "-"),
      NA_character_
    )) %>%
  ungroup()

#dplyr::select the relevant variables and compress into one row
temp <- temp %>% dplyr::select(childcode, attended_vocational_secondary, attended_kindergarten,
attended_college, attended_university, attended_graduate, attended_higher_education,
year_graduated_primary, year_graduated_lower_sec, year_graduated_upper_sec,
year_graduated_college, year_graduated_university, first_tvet_year, last_tvet_year,
graduated_gen_primary_before_tvet, graduated_gen_lower_secondary_before_tvet,
graduated_gen_upper_secondary_before_tvet)

temp <- temp %>% group_by(childcode) %>% slice(n())

edu_in_dummies <- edu_in_dummies %>% full_join(temp, by = "childcode") %>%
```

```
  mutate(continued_after_upper_secondary = case_when(
    is.na(attended_vocational_secondary) == T | is.na(attended_higher_education == T) ~ NA,
    attended_vocational_secondary == 1 & attended_higher_education == 1 ~ 1,
    completed_general_upper_secondary == 1 & attended_higher_education == 1 ~ 1,
    attended_vocational_secondary == 1 & attended_higher_education == 0 ~ 1,
    completed_general_upper_secondary == 1 & attended_higher_education == 0 ~ 1,
    TRUE ~ NA
  ))

remove(temp)
```

Vietnam
non-formal: religious education, adult literacy, non-formal continued education, short-term vocational
pre-primary: "any pre-primary" "any pre-primary grade" (pre)-kindergarten
Vocational secondary: vocational secondary schools
Primary: grade 1-5
Lower secondary: grade 6-9
Upper secondary: grade 10-11-12
College: vocational college, post-secondary technological institute, Professional Secondary
  # also vocational is assumed to be a college (students generally already have an upper-secondary diploma)
University: undergraduates and masters

However there is also a significant numbers of observations coded college/university, for now we consider those university -> but it makes most sense to use post-secondary education rather than specific differences between colleges and universities.

Importantly there are a lot less years missing here, so data between 2016 and 2020 is more reliable

```{r}
edu_vn <- bind_rows(read_csv("proc_data/edu_vn_oc.csv"), read_csv("proc_data/edu_vn_yc.csv"))

adapt_edu_vn <- function(x) {
  case_when(
    x %in% c("00 = None", "None") ~ "none",
    x %in% c("Not known", "NK", "In school, unknown grade", "Not Known") ~ NA,
    x %in% c("Primary (Grade 1)", "Grade 1") ~ "grade 1",
    x %in% c("Primary (Grade 2)", "Grade 2") ~ "grade 2",
    x %in% c("Primary (Grade 3)", "Grade 3") ~ "grade 3",
    x %in% c("Primary (Grade 4)", "Grade 4") ~ "grade 4",
    x %in% c("Primary (Grade 5)", "Grade 5") ~ "grade 5",
    x %in% c("Lower Secondary Education  (Grade 6)", "Grade 6") ~ "grade 6",
    x %in% c("Lower Secondary Education  (Grade 7)", "Grade 7") ~ "grade 7",
    x %in% c("Lower Secondary Education  (Grade 8)", "Grade 8") ~ "grade 8",
    x %in% c("Lower Secondary Education  (Grade 9)", "Grade 9") ~ "grade 9",
    x %in% c("Upper Secondary Education  (Grade 10)", "Grade 10") ~ "grade 10",
    x %in% c("Upper Secondary Education  (Grade 11)", "Grade 11") ~ "grade 11",
    x %in% c("Upper Secondary Education  (Grade 12)", "Grade 12") ~ "grade 12",
    x %in% c("Other,(specify)", "Other, (specify)", "Other, specify", "other") ~ "other",
    x == "Any pre-primary grade" ~ "(pre)-kindergarten",
```

```
    x == "Any pre-primary" ~ "(pre)-kindergarten",
    x == "College education (1st year)" ~ "college year 1",
    x == "University education (under graduate 1st year )" ~ "undergraduate year 1",
    x == "Vocational Secondary School ( 1st  year)" ~ "vocational secondary year 1",
    x == "Vocational Secondary School completion" ~ "vocational secondary year 2",
    x == "Professional Secondary (1st  years)" ~ "college year 1", #professional secondary == junior
college
    x == "Vocational College (1st year)" ~ "vocational college year 1",
    x == "Vocational College (2nd year)" ~ "vocational college year 2",
    x == "In the job, evening/weekend college education" ~ "college",
    x == "Professional Secondary (2nd  years)" ~ "college year 2",
    x == "University education (under graduate 2nd  year)" ~ "undergraduate year 2",
    x == "College education (2nd year)" ~ "college year 2",
    x == "Short term Vocational Training" ~ "short-term vocational",
    x == "Vocational Secondary School ( 2nd  year)" ~ "vocational secondary year 2",
    x == "Centre for continued education (non-formal student)" ~ "non-formal continued education",
    x == "College education completion" ~ "college",
    x == "Professional Secondary (2nd years)" ~ "college year 2",
    x == "University education (undergraduate 1st year)" ~ "undergraduate year 1",
    x == "University education (undergraduate 2nd year)" ~ "undergraduate year 2",
    x == "University education (undergraduate 3rd year)" ~ "undergraduate year 3",
    x == "University education (undergraduate 4th year)" ~ "undergraduate year 4",
    x == "University education completion" ~ "undergraduate",
    x == "Vocational Secondary School (2nd year)" ~ "vocational secondary year 2",
    x == "Vocational Secondary School (1st year)" ~ "vocational secondary year 1",
    x == "Professional Secondary (1st years)" ~ "college year 1",
    x == "Professional Secondary completion" ~ "college",
    x == "Professional Secondary (3rd years)" ~ "college year 3",
    x == "Vocational college completion" ~ "vocational college",
    x == "University education (undergraduate 5th year)" ~ "undergraduate year 5",
    x == "Post-graduate completion" ~ "masters or doctoral at university",
    x == "Post-graduate education" ~ "masters or doctoral at university",
    x == "In the job, evening/weekend undergraduate in university" ~ "undergraduate",
    x == "Degree (graduate)" ~ "masters or doctoral at university",
    x == "Post-graduate degree (e.g. Masters, PhD.)" ~ "masters or doctoral at university",
    x == "Post-graduate (vd: Masters, PhD." ~ "masters or doctoral at university",
    x == "Post-secondary technological institute" ~ "technical college",
    x == "Post-secondary technological institute/Vocational" ~ "technical college",
    x == "Vocational" ~ "vocational college",
    x == "University/College" ~ "uni/col",
    TRUE ~ x) }

edu_vn$edgrader_new = edu_vn$edgrader
edu_vn$edgrader_new = adapt_edu_vn(edu_vn$edgrader_new)
table(edu_vn$edgrader_new)

#create a dataframe to be used for merging
edu_vn_dummies <- edu_vn %>% group_by(childcode) %>% slice(n()) %>% ungroup() %>%
dplyr::select(childcode)

#Now we will use that to create dummies for whether they completed different levels
```

```
## Be careful year 2021 could mean they are still studying

#step 1: obtaining last formal grade and last year of study (highest grade is too difficult)
#Getting last known grade, and last known year of studying
edu_vn_cleaned <- edu_vn %>%
  filter(!is.na(edgrader_new) & !edgrader_new %in% c("none", "religious education", "other", "adult
literacy", "non-formal continued education", "short-term vocational")) %>% #excluding non-formal
education
  arrange(childcode, year)

#if everything is equal to none, then we need a seperate flag indicator
temp <- edu_vn %>%
  group_by(childcode) %>%
  mutate(all_na = if_else(all(is.na(edgrader_new) | edgrader_new == "none"), 1, 0)) %>%
  slice(n()) %>%
  ungroup() %>% dplyr::select(childcode, all_na)

last_grade <- edu_vn_cleaned %>%
  group_by(childcode) %>%
  slice(n()) %>%
  ungroup() %>% rename(lastgrade = edgrader_new, lastyearstudy = year) %>% dplyr::select(
    childcode, lastgrade, lastyearstudy)
#be careful 2021 could likely mean they are still studying

#combining with dummies
edu_vn_dummies <- edu_vn_dummies %>% full_join(last_grade, by = "childcode") %>%
full_join(temp, by = "childcode") %>% mutate(
  lastgrade = ifelse(all_na == 1, "none", lastgrade),
  lastyearstudy = ifelse(all_na == 1, "never", lastyearstudy)
)

remove(temp, last_grade, edu_vn_cleaned)

#step 2: creating dummies whether they completed pre-defined educational levels for primary and
secondary, this method is not possible for higher education due to too many YC still not having
achieved that level of education

#We do not have consistent dummies on whether they have succesfully completed the grade, thus
enrolling in the final grade is considered "completion
edu_vn_dummies <- edu_vn_dummies %>% mutate(
  attended_formal_education = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade == "none" ~ 0,
    TRUE ~ 1
  ),
  completed_primary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4") ~ 0,
    TRUE ~ 1
  ),
  completed_general_lower_secondary = case_when(
```

```
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade 7", "grade 8") ~ 0,
    TRUE ~ 1
  ),
  completed_general_upper_secondary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade 7", "grade 8", "grade 9", "grade 10", "grade 11") ~ 0,
    TRUE ~ 1
  )
)

#step 3: Obtaining higher education enrollment
temp <- edu_vn %>%
  group_by(childcode) %>%
  mutate(
    attended_vocational_secondary = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("vocational secondary", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_kindergarten = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("(pre)-kindergarten", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_college = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("technical college", edgrader_new)) == T ~ 1,
      any(grepl("college", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_university = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("undergraduate", edgrader_new)) == T ~ 1,
      any(grepl("masters or doctoral at university", edgrader_new)) == T ~ 1,
      any(grepl("uni/col", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_graduate = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("masters or doctoral at university", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_higher_education = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      attended_college == 1 ~ 1,
      attended_university == 1 ~ 1,
      attended_graduate == 1 ~ 1,
      TRUE ~ 0)) %>%
  ungroup()

#step 4 obtaining years of graduation and enrollment when possible, and creating other helpful dummies
```

```
#Years are an approximation -> if between 2016 and 2019 the data does not show the precise year, so
it could be the same, or the exact grade could be missing, which is why we add a few extra just in
case.
temp <- temp %>%
  mutate(year_con = substr(year, 1, 4)) %>%
  mutate(year_con = as.numeric(year_con)) %>%  # Convert 'year_con' to numeric to ensure
calculations are correct
  arrange(childcode, year_con) %>%
  mutate(
    completed_primary = grepl("grade 5|grade 6|grade 7|grade 8|grade 9|grade 10|grade 11|grade
12|college|uni/col|undergraduate|masters or doctoral at university", edgrader_new, ignore.case =
TRUE),
    completed_lower_sec = grepl("grade 9|grade 10|grade 11|grade
12|college|uni/col|undergraduate|masters or doctoral at university", edgrader_new, ignore.case =
TRUE),
    completed_upper_sec = grepl("grade 12|college|uni/col|undergraduate|masters or doctoral at
university", edgrader_new, ignore.case = TRUE),
    enrolled_college = grepl("college", edgrader_new, ignore.case = TRUE),
    enrolled_university = grepl("uni/col|undergraduate|masters or doctoral at university", edgrader_new,
ignore.case = TRUE),
    attended_voc_sec_thisyear_con = grepl("vocational secondary", edgrader_new, ignore.case =
TRUE)
  ) %>%
  group_by(childcode) %>%
  mutate(
    year_con_graduated_primary = if(any(completed_primary, na.rm = TRUE))
      min(year_con[completed_primary], na.rm = TRUE) else NA_real_,
    year_con_graduated_lower_sec = if(any(completed_lower_sec, na.rm = TRUE))
      min(year_con[completed_lower_sec], na.rm = TRUE) else NA_real_,
    year_con_graduated_upper_sec = if(any(completed_upper_sec, na.rm = TRUE))
      min(year_con[completed_upper_sec], na.rm = TRUE) else NA_real_,
    year_con_graduated_college = if(any(enrolled_college, na.rm = TRUE))
      max(year_con[enrolled_college], na.rm = TRUE) else NA_real_,
    year_con_graduated_university = if(any(enrolled_university, na.rm = TRUE))
      max(year_con[enrolled_university], na.rm = TRUE) else NA_real_,
    first_tvet_year_con = if(any(attended_voc_sec_thisyear_con, na.rm = TRUE))
      min(year_con[attended_voc_sec_thisyear_con], na.rm = TRUE) else NA_real_,
    last_tvet_year_con = if(any(attended_voc_sec_thisyear_con, na.rm = TRUE))
      max(year_con[attended_voc_sec_thisyear_con], na.rm = TRUE) else NA_real_,
    graduated_gen_primary_before_tvet = if_else(
      !is.na(year_con_graduated_primary) & !is.na(first_tvet_year_con) &
        year_con_graduated_primary < first_tvet_year_con, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_)),
    graduated_gen_lower_secondary_before_tvet = if_else(
      !is.na(year_con_graduated_lower_sec) & !is.na(first_tvet_year_con) &
        year_con_graduated_lower_sec < first_tvet_year_con, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_)),
    graduated_gen_upper_secondary_before_tvet = if_else(
      !is.na(year_con_graduated_upper_sec) & !is.na(first_tvet_year_con) &
        year_con_graduated_upper_sec < first_tvet_year_con, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_))) %>%
```

```
  mutate( #convert back to halfyear years
    year_graduated_primary = if_else(
      !is.na(year_con_graduated_primary),
      paste(year_con_graduated_primary, year_con_graduated_primary + 1, sep = "-"),
      NA_character_ ),
    year_graduated_lower_sec = if_else(
      !is.na(year_con_graduated_lower_sec),
      paste(year_con_graduated_lower_sec, year_con_graduated_lower_sec + 1, sep = "-"),
      NA_character_),
    year_graduated_upper_sec = if_else(
      !is.na(year_con_graduated_upper_sec),
      paste(year_con_graduated_upper_sec, year_con_graduated_upper_sec + 1, sep = "-"),
      NA_character_),
    year_graduated_college = if_else(
      !is.na(year_con_graduated_college),
      paste(year_con_graduated_college, year_con_graduated_college + 1, sep = "-"),
      NA_character_ ),
    year_graduated_university = if_else(
      !is.na(year_con_graduated_university),
      paste(year_con_graduated_university, year_con_graduated_university + 1, sep = "-"),
      NA_character_ ),
    first_tvet_year = if_else(
      !is.na(first_tvet_year_con),
      paste(first_tvet_year_con, first_tvet_year_con + 1, sep = "-"),
      NA_character_ ),
    last_tvet_year = if_else(
      !is.na(last_tvet_year_con),
      paste(last_tvet_year_con, last_tvet_year_con + 1, sep = "-"),
      NA_character_
    )) %>%
  ungroup()

#dplyr::select the relevant variables and compress into one row
temp <- temp %>% dplyr::select(childcode, attended_vocational_secondary, attended_kindergarten,
attended_college, attended_university, attended_graduate, attended_higher_education,
year_graduated_primary, year_graduated_lower_sec, year_graduated_upper_sec,
year_graduated_college, year_graduated_university, first_tvet_year, last_tvet_year,
graduated_gen_primary_before_tvet, graduated_gen_lower_secondary_before_tvet,
graduated_gen_upper_secondary_before_tvet)

temp <- temp %>% group_by(childcode) %>% slice(n())

edu_vn_dummies <- edu_vn_dummies %>% full_join(temp, by = "childcode") %>%
  mutate(continued_after_upper_secondary = case_when(
    is.na(attended_vocational_secondary) == T | is.na(attended_higher_education == T) ~ NA,
    attended_vocational_secondary == 1 & attended_higher_education == 1 ~ 1,
    completed_general_upper_secondary == 1 & attended_higher_education == 1 ~ 1,
    attended_vocational_secondary == 1 & attended_higher_education == 0 ~ 1,
    completed_general_upper_secondary == 1 & attended_higher_education == 0 ~ 1,
    TRUE ~ NA
  ))
```

```
remove(temp)
```

Peru
non-formal: religious education, kindergarden, creche day-care
Vocational secondary: Cent. Tecnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO
Primary: grade 1-6
Lower secondary: first cycle, grade 7/8/9 (by own definition, lower-secondary is not a seperate entity in Peru)
Upper secondary: second cycle, grade 10/11
College: technical or pedagogical institute (technical is vocational college, pedagolological is training for education), No Univ. Completa regular college
University: University Sup. (Includes Officials School) and masters

```{r}
edu_pe <- bind_rows(read_csv("proc_data/edu_pe_oc.csv",  locale = locale(encoding = "UTF-8")), read_csv("proc_data/edu_pe_yc.csv",  locale = locale(encoding = "UTF-8"))) %>% mutate(edgrader = iconv(edgrader, from = "UTF-8", to = "UTF-8", sub = "byte")) %>%
mutate(edgrader = gsub("<e9>", "e", edgrader)) #fixing é
library(dplyr)
library(stringr)

#partly in spanish
adapt_edu_pe <- function(x) {
  case_when(
    x %in% c("Not known", "Not Known", "NS") ~ NA_character_,
    x %in% c("1", "Primary Grade 1", "Grade 1") ~ "grade 1",
    x %in% c("2", "Primary Grade 2", "Grade 2") ~ "grade 2",
    x %in% c("3", "Primary Grade 3", "Grade 3") ~ "grade 3",
    x %in% c("4", "Primary Grade 4", "Grade 4") ~ "grade 4",
    x %in% c("5", "Primary Grade 5", "Grade 5", "5th Grade of Primary") ~ "grade 5",
    x %in% c("6", "Primary Grade 6", "Grade 6", "6th Grade of Primary") ~ "grade 6",
    x %in% c("7", "1st of Secondary", "Grade 7", "Secondary Grade 1") ~ "grade 7",
    x %in% c("8", "2nd of Secondary", "Grade 8", "Secondary Grade 2") ~ "grade 8",
    x %in% c("9", "3th of Secondary", "Grade 9", "3rd of Secondary", "Secondary Grade 3") ~ "grade 9",
    x %in% c("10", "4th of Secondary", "Grade 10", "Secondary Grade 4") ~ "grade 10",
    x %in% c("11", "5th of Secondary", "Grade 11", "Secondary Grade 5") ~ "grade 11",
    x %in% c("None", "Ninguno") ~ "none",
    x == "Sup. No Univ. Incompleta (t" ~ "college",
    x %in% c("Otro (especificar)", "Other (specify)", "Other (Specify)") ~ "other",
    x %in% c("Incomplete Cent. Tecnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO", "Complete Cent. Tecnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO") ~ "vocational secondary",
    x == "Sup. Universitaria Incompleta(incluye Escuela de Oficiales)" ~ "undergraduate",
    x == "Cent. T" ~ "vocational secondary",
    x == "Sup. No Univ. Completa(t" ~ "college",
    x == "Sup. Universitaria Completa (incluye Escuela de Oficiales)" ~ "undergraduate",
    x == "Incomplete technical or pedagogical institute" ~ "technical or pedagogical college",
    x == "Complete technical or pedagogical institute" ~ "technical or pedagogical college",
```

```
    x == "Incomplete university" ~ "undergraduate",
    x == "Incomplete Cent. Técnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO" ~ "vocational
secondary",
    x == "Complete university" ~ "undergraduate",
    x == "Complete Cent. Técnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO" ~ "vocational
secondary",
    x == "Complete university (incl. 'Escuela de Oficiales')" ~ "undergraduate",
    x == "Complete technical or pedagogical institute (incl. Escuela de Sub Oficiales)" ~ "technical or
pedagogical college",
    x == "Incomplete technical or pedagogical institute (incl.Escuela de Sub Officiales)" ~ "technical or
pedagogical college",
    x == "Incomplete university (incl. 'Escuela de Oficiales')" ~ "undergraduate",
    x == "Sup. Universitaria Incompleta(incluye  Escuela de Oficiales)" ~ "undergraduate",
    x == "Complete University Sup. (includes Officers School)" ~ "undergraduate",
    x == "Incomplete University Sup. (Includes Officials School)" ~ "undergraduate",
    x == "Sup. No Univ. (Technical or pedagogical or SubOficial Schoo" ~ "technical or pedagogical
college",
    x == "Masters or doctoral at university" ~ "masters or doctoral at university",
    x == "Cent. Tecnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO Complete" ~ "vocational
secondary",
    x == "Sup . No Univ. (Technical or pedagogical or SubOfficial Sch" ~ "technical or pedagogical
college",
    x == "Productive Technical Center CETPRO / Occupational Education" ~ "vocational secondary",
    x == "Postgraduate (Master's or Doctorate)" ~ "masters or doctoral at university",
    x == "Cent. Tecnico Productivo CETPRO/Cent. Edu. Ocupacional CEO Incomplete" ~ "vocational
secondary",
    TRUE ~ x  # Keeps the original value if no condition is matched
  )
}

edu_pe$edgrader_new = edu_pe$edgrader
edu_pe$edgrader_new = adapt_edu_pe(edu_pe$edgrader_new)
table(edu_pe$edgrader_new)

#create a dataframe to be used for merging
edu_pe_dummies <- edu_pe %>% group_by(childcode) %>% slice(n()) %>% ungroup() %>%
dplyr::select(childcode)

#Now we will use that to create dummies for whether they completed different levels
## Be careful year 2021 could mean they are still studying

#step 1: obtaining last formal grade and last year of study (highest grade is too difficult)
#Getting last known grade, and last known year of studying
edu_pe_cleaned <- edu_pe %>%
  filter(!is.na(edgrader_new) & !edgrader_new %in% c("none", "other")) %>%
  arrange(childcode, year)

#if everything is equal to none, then we need a seperate flag indicator
temp <- edu_pe %>%
  group_by(childcode) %>%
  mutate(all_na = if_else(all(is.na(edgrader_new) | edgrader_new == "none"), 1, 0)) %>%
```

```
  slice(n()) %>%
  ungroup() %>% dplyr::select(childcode, all_na)

last_grade <- edu_pe_cleaned %>%
  group_by(childcode) %>%
  slice(n()) %>%
  ungroup() %>% rename(lastgrade = edgrader_new, lastyearstudy = year) %>% dplyr::select(
    childcode, lastgrade, lastyearstudy)
#be careful 2021 could likely mean they are still studying

#combining with dummies
edu_pe_dummies <- edu_pe_dummies %>% full_join(last_grade, by = "childcode") %>%
full_join(temp, by = "childcode") %>% mutate(
  lastgrade = ifelse(all_na == 1, "none", lastgrade),
  lastyearstudy = ifelse(all_na == 1, "never", lastyearstudy)
)

remove(temp, last_grade, edu_pe_cleaned)

#step 2: creating dummies whether they completed pre-defined educational levels for primary and
secondary, this method is not possible for higher education due to too many YC still not having
achieved that level of education

#We do not have consistent dummies on whether they have succesfully completed the grade, thus
enrolling in the final grade is considered "completion
edu_pe_dummies <- edu_pe_dummies %>% mutate(
  attended_formal_education = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade == "none" ~ 0,
    TRUE ~ 1
  ),
  completed_primary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5") ~ 0,
    TRUE ~ 1
  ),
  completed_general_lower_secondary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade
7", "grade 8") ~ 0,
    TRUE ~ 1
  ),
  completed_general_upper_secondary = case_when(
    is.na(lastgrade) ~ NA,
    lastgrade %in% c("none", "grade 1", "grade 2", "grade 3", "grade 4", "grade 5", "grade 6", "grade
7", "grade 8", "grade 9", "grade 10") ~ 0,
    TRUE ~ 1
  )
)

#step 3: Obtaining higher education enrollment
```

```
temp <- edu_pe %>%
  group_by(childcode) %>%
  mutate(
    attended_vocational_secondary = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("vocational secondary", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_kindergarten = NA, #no info
    attended_college = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("college", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_university = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("undergraduate", edgrader_new)) == T ~ 1,
      any(grepl("masters or doctoral at university", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_graduate = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      any(grepl("masters or doctoral at university", edgrader_new)) == T ~ 1,
      TRUE ~ 0),
    attended_higher_education = case_when(
      all(is.na(edgrader_new)) == T ~ NA,
      attended_college == 1 ~ 1,
      attended_university == 1 ~ 1,
      attended_graduate == 1 ~ 1,
      TRUE ~ 0)) %>%
  ungroup()

#step 4 obtaining years of graduation and enrollment when possible, and creating other helpful
dummies

#Years are an approximation -> if between 2016 and 2019 the data does not show the precise year, so
it could be the same, or the exact grade could be missing, which is why we add a few extra just in
case.
temp <- temp %>%
  mutate(year = as.numeric(year)) %>%  # Convert 'year' to numeric to ensure calculations are correct
  arrange(childcode, year) %>%
  mutate(
    completed_primary = grepl("grade 6|grade 7|grade 8| grade 9|grade 10|grade
11|college|undergraduate|masters or doctoral at university", edgrader_new, ignore.case = TRUE),
    completed_lower_sec = grepl("grade 9|grade 10|grade 11|college|undergraduate|masters or doctoral
at university", edgrader_new, ignore.case = TRUE),
    completed_upper_sec = grepl("grade 11|college|undergraduate|masters or doctoral at university",
edgrader_new, ignore.case = TRUE),
    enrolled_college = grepl("college", edgrader_new, ignore.case = TRUE),
    enrolled_university = grepl("undergraduate|masters or doctoral at university", edgrader_new,
ignore.case = TRUE),
    attended_voc_sec_thisyear = grepl("vocational secondary", edgrader_new, ignore.case = TRUE)
  ) %>%
  group_by(childcode) %>%
```

```
  mutate(
    year_graduated_primary = if(any(completed_primary, na.rm = TRUE))
      min(year[completed_primary], na.rm = TRUE) else NA_real_,
    year_graduated_lower_sec = if(any(completed_lower_sec, na.rm = TRUE))
      min(year[completed_lower_sec], na.rm = TRUE) else NA_real_,
    year_graduated_upper_sec = if(any(completed_upper_sec, na.rm = TRUE))
      min(year[completed_upper_sec], na.rm = TRUE) else NA_real_,
    year_graduated_college = if(any(enrolled_college, na.rm = TRUE))
      max(year[enrolled_college], na.rm = TRUE) else NA_real_,
    year_graduated_university = if(any(enrolled_university, na.rm = TRUE))
      max(year[enrolled_university], na.rm = TRUE) else NA_real_,
    first_tvet_year = if(any(attended_voc_sec_thisyear, na.rm = TRUE))
      min(year[attended_voc_sec_thisyear], na.rm = TRUE) else NA_real_,
    last_tvet_year = if(any(attended_voc_sec_thisyear, na.rm = TRUE))
      max(year[attended_voc_sec_thisyear], na.rm = TRUE) else NA_real_,
    graduated_gen_primary_before_tvet = if_else(
      !is.na(year_graduated_primary) & !is.na(first_tvet_year) &
        year_graduated_primary < first_tvet_year, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_)),
    graduated_gen_lower_secondary_before_tvet = if_else(
      !is.na(year_graduated_lower_sec) & !is.na(first_tvet_year) &
        year_graduated_lower_sec < first_tvet_year, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_)),
    graduated_gen_upper_secondary_before_tvet = if_else(
      !is.na(year_graduated_upper_sec) & !is.na(first_tvet_year) &
        year_graduated_upper_sec < first_tvet_year, 1,
      if_else(attended_vocational_secondary == 1, 0, NA_real_))) %>%
  ungroup()

#dplyr::select the relevant variables and compress into one row
temp <- temp %>% dplyr::select(childcode, attended_vocational_secondary, attended_kindergarten,
attended_college, attended_university, attended_graduate, attended_higher_education,
year_graduated_primary, year_graduated_lower_sec, year_graduated_upper_sec,
year_graduated_college, year_graduated_university, first_tvet_year, last_tvet_year,
graduated_gen_primary_before_tvet, graduated_gen_lower_secondary_before_tvet,
graduated_gen_upper_secondary_before_tvet)

temp <- temp %>% group_by(childcode) %>% slice(n())

edu_pe_dummies <- edu_pe_dummies %>% full_join(temp, by = "childcode") %>%
  mutate(continued_after_upper_secondary = case_when(
    is.na(attended_vocational_secondary) == T | is.na(attended_higher_education == T) ~ NA,
    attended_vocational_secondary == 1 & attended_higher_education == 1 ~ 1,
    completed_general_upper_secondary == 1 & attended_higher_education == 1 ~ 1,
    attended_vocational_secondary == 1 & attended_higher_education == 0 ~ 1,
    completed_general_upper_secondary == 1 & attended_higher_education == 0 ~ 1,
    TRUE ~ NA
  ))

remove(temp)
```

##2.6. Create wide dataframe and save
```{r}
library(dplyr)

edu_et_dummies <- edu_et_dummies %>%
  mutate(across(
    .cols = c(starts_with("year"), ends_with("year")),
    .fns = as.character
  ))
edu_pe_dummies <- edu_pe_dummies %>%
  mutate(across(
    .cols = c(starts_with("year"), ends_with("year")),
    .fns = as.character
  ))

edu_dummies <- bind_rows(edu_et_dummies, edu_in_dummies, edu_pe_dummies,
edu_vn_dummies)

edu_dummies <- edu_dummies %>% dplyr::select(-all_na) %>% mutate(countrycode =
substr(childcode, 1, 2))

#saving datafile
write.csv(edu_dummies, "proc_data/edu_dummies.csv", row.names = FALSE)
```

```{r}
table(edu_dummies$attended_vocational_secondary)
table(edu_dummies$attended_vocational_secondary, edu_dummies$countrycode)
table(edu_dummies$graduated_gen_primary_before_tvet, edu_dummies$countrycode)
table(edu_dummies$graduated_gen_lower_secondary_before_tvet, edu_dummies$countrycode)
table(edu_dummies$graduated_gen_upper_secondary_before_tvet, edu_dummies$countrycode)

```

```{r}
remove(edu_et, edu_in, edu_vn, edu_pe, edu_et_dummies, edu_vn_dummies, edu_in_dummies,
edu_pe_dummies)
```

#3. Outcome Data
Below collects the outcome data per wave available,
it does not check for data errors.
Output is a long dataframe.

3.1. Custom Functions for Wave 6 outcome data
Does not yet exchange income into us$
Does not yet account for irrealistic income/working times
This code generates seperate outputs for different country and cohorts
```{r}
wave6outcome <- function(country, cohort) {
```

```
library(stringr); library(dplyr); library(haven)
datasets2 <- function(country, cohort) {
  cohort <- tolower(cohort)
  country <- tolower(country)

  # Function to read Stata file and preprocess data
  read.stata <- function(file_path, country, rem.number = T) {
    library(haven)
    library(dplyr)
    library(stringr)

    # Read the data file
    data <- NULL
    data <- read_dta(file_path)

    # Process the data
    data <- data %>%
      rename_with(~tolower(.), everything()) %>%
      mutate(across(where(is.labelled), as_factor)) %>%
      mutate(across(where(is.factor), as.character)) %>%
      mutate(across(where(is.character), ~trimws(.x, which = "left"))) %>%
      mutate(across(where(is.factor), as.factor)) #remove starting spaces

    if ("childcode" %in% names(data) && !is.null(country)) {
      data <- data %>% mutate(childcode = as.factor(childcode))
      country_code <- toupper(substr(country, 1, 2))
      data <- data %>%
        mutate(childcode = paste0(country_code, childcode))
    }

    if (rem.number == T) {
      # Remove numbers from column names
      names(data) <- gsub(pattern = "[0-9]", replacement = "", names(data))
    }

    return(data)
  }

  # Construct the file path dynamically
  path_call1 <- sprintf("unproc_data/raw_data/%s/w6_%s_%s_c1_arch.dta", cohort, cohort, country)
  path_call2 <- sprintf("unproc_data/raw_data/%s/w6_%s_%s_c2_arch.dta", cohort, cohort, country)
  path_call3 <- sprintf("unproc_data/raw_data/%s/w6_%s_%s_c3_arch.dta", cohort, cohort, country)
  path_call4 <- sprintf("unproc_data/raw_data/%s/w6_%s_%s_c4_arch.dta", cohort, cohort, country)
  path_call5 <- sprintf("unproc_data/raw_data/%s/w6_%s_%s_c5_arch.dta", cohort, cohort, country)
  path_constr6 <- sprintf("unproc_data/constructed_data/Constructed Wave
6/stata/stata13/%s_constructed_call.dta", country)

  # Call the read.stata function, passing 'country' explicitly for each necessary dataset
  call1 <- read.stata(path_call1, country, rem.number = F)
  call2 <- read.stata(path_call2, country, rem.number = F)
  call3 <- read.stata(path_call3, country, rem.number = F)
```

```
    call4 <- read.stata(path_call3, country, rem.number = F)
    call5 <- read.stata(path_call5, country, rem.number = F)
    constructed_call <- read.stata(path_constr6, country, rem.number = F) %>% rename(childcode =
childid) %>% dplyr::select(childcode, region, yc, call, incall, dint, wi, enrol, enrol2020, work_week,
work_call23, work_mar21, no_work_job, work_bf_cov, econ_sector_bf, econ_sector,
econ_sector_mar21, type_act_bf, type_act, type_act_mar21, typemp_bf, typemp, typemp_mar21,
agri_bf, agri, agri_mar21)

    if (cohort == "yc") {
      constructed_call <- constructed_call %>% subset(yc == "Younger cohort")
    }

    if (cohort == "oc") {
      constructed_call <- constructed_call %>% subset(yc == "Older Cohort")
    }

    #some childcodes are wrongly numbered (having a 0 in after country code
    adjust_childcode <- function(df, country) {
      temp_name <- paste0(toupper(country), "0") # Temporary name to check
      df %>% mutate(childcode = if_else(str_sub(childcode, 1, 3) == temp_name,
                                  paste0(toupper(substr(country, 1, 2)),
                                         str_sub(childcode, 4)), childcode))
    }

    datasets <- list(call1, call2, call3, call4, call5, constructed_call)
    adj_datasets <- lapply(datasets, adjust_childcode, country = country)

    names(adj_datasets) <- paste(country, cohort, c("call1", "call2", "call3", "call4", "call5",
"constructed_call"), sep = "_")
    Sys.setlocale("LC_ALL", "Dutch")
    return(adj_datasets)
  }
data <- datasets2(country = country, cohort = cohort)
list2env(data, envir = .GlobalEnv)

call1_name <- paste(country, "_", cohort, "_call2", sep = "")
call2_name <- paste(country, "_", cohort, "_call2", sep = "")
call3_name <- paste(country, "_", cohort, "_call2", sep = "")
call4_name <- paste(country, "_", cohort, "_call2", sep = "")
call5_name <- paste(country, "_", cohort, "_call5", sep = "")
constructed_call_name <- paste(country, "_", cohort, "_constructed_call", sep = "")

call2 <- get(call2_name)
call5 <- get(call5_name)
constructed_call <- get(constructed_call_name)


############
#Wealth, Salary and QUality of Job
############
# WI
```

```
# Wealth index is already pre-computed, and is only computed for call 2, so the rest should become
NA
constructed_call <- constructed_call %>% mutate(wi = ifelse(call != "Call 2", NA, wi))

if (country %in% c("et", "in", "pe")) {
  #For salary and hours of work we first btain the necessary variables from call 2, 3 and 5
  temp <- call2 %>% dplyr::select(childcode, frmpaycov2, prdcvrcov2, pcedaycov2, erncshcov2,
ernkndcov2, wksmthcov2, dyswkcov2, hrsdaycov2) %>% rename_with(~str_remove(.x, "2$")) %>%
    mutate(ernkndcov = str_remove_all(ernkndcov, "[^\\d-]"),
           erncshcov = str_remove_all(erncshcov, "[^\\d-]"),
           pcedaycov = str_remove_all(pcedaycov, "[^\\d-]")) %>%
    mutate(pcedaycov = if_else(str_detect(pcedaycov, "^-") == T, NA, pcedaycov)) %>%
    mutate(ernkndcov = case_when(
      str_detect(ernkndcov, "^-") ~ NA_character_, #negative numbers were used as placeholders for
NA
      ernkndcov == "" ~ NA, # Employed
      ernkndcov == "00" ~ "0",  # Employed
      ernkndcov == "000" ~ "0",
      TRUE ~ ernkndcov)) %>%
    rename(wrkwkscov = wksmthcov, wrkdyscov = dyswkcov, wrkhrscov = hrsdaycov) %>%
    mutate(erncshcov = as.numeric(erncshcov),
           ernkndcov = as.numeric(ernkndcov),
           pcedaycov = as.numeric(pcedaycov),
           wrkwkscov = as.numeric(wrkwkscov),
           wrkdyscov = as.numeric(wrkdyscov),
           wrkhrscov = as.numeric(wrkhrscov),
           call = "Call 2")

  #different namesfor wave 5
  temp2 <- call5 %>% dplyr::select(childcode, frmpymtncov5, prdpymntcov5, pcspdaycov5,
ntearncshcov5, ntearnkndcov5, wrkwkscov5, wrkdyscov5, wrkhrscov5) %>%
    rename_with(~str_remove(.x, "5$")) %>%
    rename(frmpaycov = frmpymtncov, prdcvrcov = prdpymntcov,
           pcedaycov = pcspdaycov, erncshcov = ntearncshcov,
           ernkndcov = ntearnkndcov)  %>%
    mutate(ernkndcov = str_remove_all(ernkndcov, "[^\\d-]"),
           erncshcov = str_remove_all(erncshcov, "[^\\d-]"),
           pcedaycov = str_remove_all(pcedaycov, "[^\\d-]")) %>% #removing all non-numbers
    mutate(pcedaycov = if_else(str_detect(pcedaycov, "^-") == T, NA, pcedaycov)) %>%
    mutate(ernkndcov = case_when(
      str_detect(ernkndcov, "^-") ~ NA_character_,
      ernkndcov == "" ~ NA, # Employed
      ernkndcov == "00" ~ "0",  # Employed
      ernkndcov == "000" ~ "0",
      TRUE ~ ernkndcov)) %>%  #handling other errors
    mutate(erncshcov = as.numeric(erncshcov),
           ernkndcov = as.numeric(ernkndcov),
           pcedaycov = as.numeric(pcedaycov),
           wrkwkscov = as.numeric(wrkwkscov),
           wrkdyscov = as.numeric(wrkdyscov),
           wrkhrscov = as.numeric(wrkhrscov),
```

```
        call = "Call 5")

  temp <-  bind_rows(temp, temp2)

  # Changing negative values into NA (these were used as errors in coding, but the exact numbers
differ per country)
  temp <- temp %>% mutate(wrkwkscov = ifelse(str_detect(wrkwkscov, "^-") == T, NA, wrkwkscov),
                    wrkdyscov = ifelse(str_detect(wrkdyscov, "^-") == T, NA, wrkdyscov),
                    wrkhrscov = ifelse(str_detect(wrkhrscov, "^-") == T, NA, wrkhrscov))

  # Hours worked per month and week
  temp <- temp %>% mutate(month_hoursworked = wrkwkscov*wrkdyscov*wrkhrscov,
                    week_hoursworked = wrkdyscov*wrkhrscov)

  ################################################################################
  # We calculate monthly income, including informal work and in-kind payment
  # We calculate both weekly income, and hourly equivalent income.
  ################################################################################
  temp <- temp %>% mutate(total_ear = case_when(
    is.na(ernkndcov) & is.na(erncshcov) ~ NA_real_,  # Return NA if both are NA
    TRUE ~ coalesce(ernkndcov, 0) + coalesce(erncshcov, 0)))

  # Now we have to calculate their weekly income.
  # However, applicants themselves could choose their time-period to determine their income. This
thus has to be converted
  # To converse between different timeframes, we assume fulltime work (8 hours a day, 5 days a week,
4.345 weeks a month)
  temp <- temp %>%
    mutate(
      full_time_weekly_income = case_when(
        frmpaycov == "None" ~ 0,  # Return 0 as numeric
        frmpaycov %in% c("Other, specify", "NK", "Refused to Answer", "Other", "Doesn't know",
"Refused to answer", "Debt relief") ~ NA_real_, # Use NA_real_ for numeric NA
        prdcvrcov %in% c("Other, specify", "NK", "Other", "Doesn't know", "Refused to answer", "Debt
relief") ~ NA_real_, # Use NA_real_ for numeric NA
         prdcvrcov == "Per hour" ~ total_ear * 8 * 5,
        prdcvrcov == "Per day" ~ total_ear * 5,
        prdcvrcov == "Per week" ~ total_ear,
        prdcvrcov == "Per month" ~ total_ear / 4.345,
        prdcvrcov == "Per year" ~ total_ear / 4.345 / 12,
        prdcvrcov == "Per piece" ~ total_ear * pcedaycov * 5,
        prdcvrcov == "Fortnightly" ~ total_ear / 2,
        prdcvrcov == "Fortnightly 15" ~ total_ear / 2, #type error in India OC survey
        prdcvrcov == "Biweekly" ~ total_ear / 2,
        TRUE ~ NA_real_  # Handle any other unspecified cases
      )
    )

  #to compare fulltime and parttime, we also calculate earnings per hour
  temp <- temp %>%
    mutate(
```

```
    hourly_income = case_when(
      full_time_weekly_income == 0 ~ 0,
      week_hoursworked == 0 & full_time_weekly_income != 0 ~ NA_real_, #then unknown
      full_time_weekly_income != 0 ~ full_time_weekly_income/coalesce(week_hoursworked, 40),
      TRUE ~ NA_real_
    ),
    real_weekly_income = hourly_income* week_hoursworked)
}

if (country %in% c("vn")) { #unfortunately, vietnam survey uses a different survey, with worked days
per month rather than per week
  #For salary and hours of work we first btain the necessary variables from call 2, 3 and 5
  temp <- call2 %>% dplyr::select(childcode, frmpaycov2, prdcvrcov2, pcedaycov2, erncshcov2,
ernkndcov2, dyswkcov2, hrsdaycov2) %>% rename_with(~str_remove(.x, "2$")) %>%
    mutate(ernkndcov = str_remove_all(ernkndcov, "[^\\d-]"),
           erncshcov = str_remove_all(erncshcov, "[^\\d-]"),
           pcedaycov = str_remove_all(pcedaycov, "[^\\d-]")) %>%
    mutate(pcedaycov = if_else(str_detect(pcedaycov, "^-") == T, NA, pcedaycov))%>%
    mutate(erncshcov = if_else(str_detect(erncshcov, "^-") == T, NA, erncshcov))%>%
    mutate(ernkndcov = case_when(
      str_detect(ernkndcov, "^-") ~ NA_character_, #negative numbers were used as placeholders for
NA
      ernkndcov == "" ~ NA, # Employed
      ernkndcov == "00" ~ "0",  # Employed
      ernkndcov == "000" ~ "0",
      TRUE ~ ernkndcov)) %>%
    rename(wrkdyscov = dyswkcov, wrkhrscov = hrsdaycov) %>%
    mutate(erncshcov = as.numeric(erncshcov),
           ernkndcov = as.numeric(ernkndcov),
           pcedaycov = as.numeric(pcedaycov),
           wrkdyscov = as.numeric(wrkdyscov),
           wrkhrscov = as.numeric(wrkhrscov),
           call = "Call 2")

  #different namesfor wave 5
  temp2 <- call5 %>% dplyr::select(childcode, frmpymtncov5, prdpymntcov5, pcspdaycov5,
ntearncshcov5, ntearnkndcov5, wrkdyscov5, wrkhrscov5) %>%
    rename_with(~str_remove(.x, "5$")) %>%
    rename(frmpaycov = frmpymtncov, prdcvrcov = prdpymntcov,
           pcedaycov = pcspdaycov, erncshcov = ntearncshcov,
           ernkndcov = ntearnkndcov)  %>%
    mutate(pcedaycov = if_else(str_detect(pcedaycov, "^-") == T, NA, pcedaycov))%>%
    mutate(erncshcov = if_else(str_detect(erncshcov, "^-") == T, NA, erncshcov))%>%
    mutate(wrkhrscov = if_else(wrkhrscov == "NK", NA, wrkhrscov),
           wrkdyscov = if_else(wrkdyscov == "NK", NA, wrkdyscov)) %>%
    mutate(ernkndcov = str_remove_all(ernkndcov, "[^\\d-]"),
           erncshcov = str_remove_all(erncshcov, "[^\\d-]"),
           pcedaycov = str_remove_all(pcedaycov, "[^\\d-]")) %>% #removing all non-numbers
    mutate(ernkndcov = case_when(
      str_detect(ernkndcov, "^-") ~ NA_character_,
      ernkndcov == "" ~ NA, # Employed
```

```
    ernkndcov == "00" ~ "0",  # Employed
    ernkndcov == "000" ~ "0",
    TRUE ~ ernkndcov)) %>%  #handling other errors
  mutate(erncshcov = as.numeric(erncshcov),
       ernkndcov = as.numeric(ernkndcov),
       pcedaycov = as.numeric(pcedaycov),
       wrkdyscov = as.numeric(wrkdyscov),
       wrkhrscov = as.numeric(wrkhrscov),
       call = "Call 5")

 temp <-  bind_rows(temp, temp2)

 # Changing missing values into NA
 temp <- temp %>% mutate(wrkdyscov = ifelse(str_detect(wrkdyscov, "^-") == T, NA, wrkdyscov),
                 wrkhrscov = ifelse(str_detect(wrkhrscov, "^-") == T, NA, wrkhrscov))

 # Hours worked per month and week assuming 4.345 weeks a month on average
 temp <- temp %>% mutate(month_hoursworked = wrkdyscov*wrkhrscov,
                 week_hoursworked = round(wrkdyscov*wrkhrscov/4.345, 1),
                 week_daysworked = wrkdyscov / 4.345)

 ################################################################################
 # We calculate monthly income, including informal work and in-kind payment
 # We calculate both weekly income, and hourly equivalent income.
 ################################################################################
 temp <- temp %>% mutate(total_ear = case_when(
  is.na(ernkndcov) & is.na(erncshcov) ~ NA_real_,  # Return NA if both are NA
  TRUE ~ coalesce(ernkndcov, 0) + coalesce(erncshcov, 0)))

 # Now we have to calculate their weekly income.
 # However, applicants themselves could choose their time-period to determine their income. This
thus has to be converted
 # To converse between different timeframes, we assume fulltime work (8 hours a day, 5 days a week,
4.345 weeks a month)
 temp <- temp %>%
  mutate(
    full_time_weekly_income = case_when(
     frmpaycov == "None" ~ 0,  # Return 0 as numeric
     frmpaycov %in% c("Other, specify", "NK", "Refused to Answer", "Other", "Doesn't know",
"Refused to answer", "Debt relief") ~ NA_real_, # Use NA_real_ for numeric NA
     prdcvrcov %in% c("Other, specify", "NK", "Other", "Doesn't know", "Refused to answer", "Debt
relief") ~ NA_real_, # Use NA_real_ for numeric NA
      prdcvrcov == "Per hour" ~ total_ear * 8 * 5,
     prdcvrcov == "Per day" ~ total_ear * 5,
     prdcvrcov == "Per week" ~ total_ear,
     prdcvrcov == "Per month" ~ total_ear / 4.345,
     prdcvrcov == "Per year" ~ total_ear / 4.345 / 12,
     prdcvrcov == "Per piece" ~ total_ear * pcedaycov * 5,
     prdcvrcov == "Fortnightly" ~ total_ear / 2,
     prdcvrcov == "Forthnightly" ~ total_ear / 2, #type error in Vietnam YC survey
     prdcvrcov == "Fortnightly 15" ~ total_ear / 2, #type error in India OC survey
```

```
      prdcvrcov == "Biweekly" ~ total_ear / 2,
      TRUE ~ NA_real_  # Handle any other unspecified cases
    )
  )

 #to compare fulltime and parttime, we also calculate earnings per hour
 temp <- temp %>%
  mutate(
   hourly_income = case_when(
    full_time_weekly_income == 0 ~ 0,
    week_hoursworked == 0 & full_time_weekly_income != 0 ~ NA_real_, #then unknown
    full_time_weekly_income != 0 ~ full_time_weekly_income/coalesce(week_hoursworked, 40),
    TRUE ~ NA_real_
   ),
   real_weekly_income = hourly_income* week_hoursworked)
}

#and join dataframe
merge <- temp %>% dplyr::select(childcode, call, week_hoursworked, month_hoursworked,
full_time_weekly_income, hourly_income, real_weekly_income)

constructed_call <- constructed_call %>%
 left_join(merge, by = c("childcode", "call"))

remove(temp2, merge)

############
#Employment Status
############
#NEET
# Binary Variable 0 if a) worked at least one hour during the 7 days before the call,
# or b) did not work during the 7 days before, but did have a job
# or c) was attending education or training during the same year and 1 otherwise
# we assume that if people do not report being enrolled, or having a job but not going to work that they
are NEET.

constructed_call <- constructed_call %>% mutate(year_int = substr(dint, 1, 4)) %>% #Create new
time variable to characterize the year of interview
 subset(call %in% c("Call 2", "Call 3", "Call 5")) %>%
 mutate(neet = case_when(
  work_week == "Yes" ~ 0,  # Employed
  no_work_job == "Yes" ~ 0,  # Employed
  enrol == "Enrolled" ~ 0,  # Enrolled in school
  incall == "Participant is not present in call" ~ NA,
  TRUE ~ 1  # Not employed or not enrolled in the respective years
 ))

#any_iga
# Binary variable: 1) if a) worked at least one hour during the 7 days before the call which generates
some revenue
```

```
# or b) did not work during the 7 days before, but did have a job (thus does not include continued
education)
constructed_call  <- constructed_call %>% mutate(any_iga = case_when(
  hourly_income > 0 ~ 1, #paid work
  no_work_job == "Yes" ~ 1,  # Employed
  incall == "Participant is not present in call" ~ NA,
  TRUE ~ 0
))

#any_formal job
# Binary variable: 1) if "Regular Salaried Employment" or similar
#                 Wage Employment (Unsalaried/ irregular; Non-agriculture) is not considered formal
work
#                0) otherwise (including unemployed, self-e mployment or dependent worker (working
without contract))
#                 NA if type of activity is not known, but the person is working and not studying)
constructed_call  <- constructed_call %>% mutate(formal_iga = case_when(
  type_act %in% c("Regular Salaried Employment", "Salaried Farmer", "Salaried worker", "Wage
Employment (Agriculture)", "Salaried Worker", "Annual Farm Servant") ~ 1, #Formal
  is.na(type_act) == T & any_iga == 1 & enrol == "Not enrolled" ~ NA, #unknown type of work
  incall == "Participant is not present in call" ~ NA,
  TRUE ~ 0 #Non formal or unemployed
))


#self-employment
#Binary variable 1) if Self Employed is in type of activity
#              0) otherwise
#              99) if type of activity is not known, but the person is working and not studying

constructed_call  <- constructed_call %>% mutate(self_emp = case_when(
  str_detect(type_act, "Self Employed") ~ 1, #Self Employed
  str_detect(type_act, "Self-Employed") ~ 1, #Self Employed
  str_detect(type_act, "Independent") ~ 1, #Self Employed Artisan or trader
  str_detect(type_act, "own farm") ~ 1, #Self Employed
  is.na(type_act) == T & any_iga == 1 & enrol == "Not enrolled" ~ NA,
  incall == "Participant is not present in call" ~ NA,
  TRUE ~ 0 #Not Self Employed
))

#Non-farming IGA
#Binary Variable 1) if any_iga = 1 and agri = "No"
#              0) otherwise
#              99) if sector of employment is unknown, but the person is working and not studying


constructed_call  <- constructed_call %>% mutate(nonfarm_iga = case_when(
  any_iga == 1 & agri == "No" ~ 1, #Employed but not in agri
  any_iga == 1 & agri == "Yes" ~ 0, #Employed in Agri
  any_iga == 0 ~ 0, #unemployed
  TRUE ~ NA_real_
```

))

#month_hoursworked, week_hoursworked and income should be 0 if NA and there is no IGA
constructed_call <- constructed_call %>%
 mutate(
  month_hoursworked = case_when(
   call %in% c("Call 2", "Call 5") & is.na(month_hoursworked) & any_iga == 0 ~ 0,
   call %in% c("Call 2", "Call 5") & is.na(month_hoursworked) ~ NA_real_,
   TRUE ~ month_hoursworked
  ),
  week_hoursworked = case_when(
   call %in% c("Call 2", "Call 5") & is.na(week_hoursworked) & any_iga == 0 ~ 0,
   call %in% c("Call 2", "Call 5") & is.na(week_hoursworked) ~ NA_real_,
   TRUE ~ week_hoursworked
  ),
  full_time_weekly_income = case_when(
   call %in% c("Call 2", "Call 5") & is.na(full_time_weekly_income) & any_iga == 0 ~ 0,
   call %in% c("Call 2", "Call 5") & is.na(full_time_weekly_income) ~ NA_real_,
   TRUE ~ full_time_weekly_income
  ),
  hourly_income = case_when(
   call %in% c("Call 2", "Call 5") & is.na(hourly_income) & any_iga == 0 ~ 0,
   call %in% c("Call 2", "Call 5") & is.na(hourly_income) ~ NA_real_,
   TRUE ~ hourly_income
  ),
  real_weekly_income = case_when(
   is.na(real_weekly_income) & any_iga == 0 ~ 0,
   is.na(real_weekly_income) ~ NA_real_,
   TRUE ~ real_weekly_income
)
)

output <- constructed_call %>% dplyr::select(childcode, call, incall, dint, enrol, wi, week_hoursworked, month_hoursworked, full_time_weekly_income, hourly_income, real_weekly_income, type_act, neet, any_iga, formal_iga, self_emp, nonfarm_iga) %>% rename(inround = incall)

return(output)
remove()
}
```

##3.2. Custom Functions for Wave 5 outcome data
Function for Wave 5 outcome data
Does not yet exchange income into us$
Does not yet account for irrealistic income/working times
This code generates one outputs for all countries and cohorts together

```{r}

```
read.stata <- function(file_path, country, rem.number = T) {
    library(haven)
    library(dplyr)
    library(stringr)

    # Read the data file
    data <- NULL
    data <- read_dta(file_path)

    # Process the data
    data <- data %>%
      rename_with(~tolower(.), everything()) %>%
      mutate(across(where(is.labelled), as_factor)) %>%
      mutate(across(where(is.factor), as.character)) %>%
      mutate(across(where(is.character), ~trimws(.x, which = "left"))) %>%
      mutate(across(where(is.factor), as.factor)) #remove starting spaces

    if ("childcode" %in% names(data) && !is.null(country)) {
      data <- data %>% mutate(childcode = as.factor(childcode))
      country_code <- toupper(substr(country, 1, 2))
      data <- data %>%
        mutate(childcode = paste0(country_code, childcode))
    }

    if (rem.number == T) {
      # Remove numbers from column names
      names(data) <- gsub(pattern = "[0-9]", replacement = "", names(data))
    }

    return(data)
}

adjust_childcode <- function(df, country) {
    temp_name <- paste0(toupper(country), "0") # Temporary name to check
    df %>% mutate(childcode = if_else(str_sub(childcode, 1, 3) == temp_name,
                                  paste0(toupper(substr(country, 1, 2)),
                                         str_sub(childcode, 4)), childcode))
}

wave5_in <- read.stata("unproc_data/raw_data/oc/w5_oc_in_activity.dta", country = "in", rem.number
= F) %>% rename_with(~str_remove(.x, "5$")) %>%
  dplyr::select(childcode, actidr, actr, actdymtr, acthrsr, pymrecr,
       erncshr, hwpaidr, pdpcprr, ernkndr, prfactr) %>%
  rename(type_act = actr, wrkdyscov = actdymtr, wrkhrscov = acthrsr,
       frmpaycov = pymrecr, erncshcov = erncshr, ernkndcov = ernkndr,
       prdcvrcov = hwpaidr, pcedaycov = pdpcprr, work_week = prfactr) %>%
  mutate(actidr = as.character(actidr),
       work_week = as.factor(work_week)) %>%
adjust_childcode(country = "in")
```

```
wave5_pe <- read.stata("unproc_data/raw_data/oc/w5_oc_pe_activity.dta", country = "pe",
rem.number = F) %>% rename_with(~str_remove(.x, "5$")) %>%
  dplyr::select(childcode, actidr, actr, actdaymtr, acthrsr, pymrecr,
         erncshr, hwpaidr, pdpcprr, ernkndr, prfactr) %>%
  rename(type_act = actr, wrkdyscov = actdaymtr, wrkhrscov = acthrsr,
         frmpaycov = pymrecr, erncshcov = erncshr, ernkndcov = ernkndr,
         prdcvrcov = hwpaidr, pcedaycov = pdpcprr, work_week = prfactr)%>%
  mutate(actidr = as.character(actidr),
         work_week = as.factor(work_week)) %>%
adjust_childcode(country = "pe")

wave5_vn <- read.stata("unproc_data/raw_data/oc/w5_oc_vn_activity.dta", country = "vn",
rem.number = F) %>% rename_with(~str_remove(.x, "5$")) %>%
  dplyr::select(childcode, actidr, actr, actwekr, acthrsr, pymrecr,
         erncshr, hwpaidr, pdpcprr, ernkndr, prfactr) %>%
  rename(type_act = actr, wrkdyscov = actwekr, wrkhrscov = acthrsr,
         frmpaycov = pymrecr, erncshcov = erncshr, ernkndcov = ernkndr,
         prdcvrcov = hwpaidr, pcedaycov = pdpcprr, work_week = prfactr) %>%
  mutate(actidr = as.character(actidr),
         work_week = as.factor(work_week)) %>%
adjust_childcode(country = "vn")

wave5_et <- read.stata("unproc_data/raw_data/oc/w5_oc_et_activity.dta", country = "et", rem.number
= F) %>% rename_with(~str_remove(.x, "5$")) %>%
  dplyr::select(childcode, actidr, actr, actdymtr, acthrsr, pymrecr,
         erncshr, hwpaidr, ernkndr, prfactr) %>%
  rename(type_act = actr, wrkdyscov = actdymtr, wrkhrscov = acthrsr,
         frmpaycov = pymrecr, erncshcov = erncshr, ernkndcov = ernkndr,
         prdcvrcov = hwpaidr, work_week = prfactr) %>%
  mutate(actidr = as.character(actidr),
         work_week = as.factor(work_week)) %>%
adjust_childcode(country = "et")


wave5_in_yc <- read.stata("unproc_data/raw_data/yc/w5_yc_in_activity.dta", country = "in",
rem.number = F) %>% rename_with(~str_remove(.x, "5$")) %>%
  dplyr::select(childcode, actidr, actr, actdayr, acthrsr, pymrecr,
         erncshr, hwpaidr, pdpcprr, ernkndr, prfactr) %>%
  rename(type_act = actr, wrkdyscov = actdayr, wrkhrscov = acthrsr,
         frmpaycov = pymrecr, erncshcov = erncshr, ernkndcov = ernkndr,
         prdcvrcov = hwpaidr, pcedaycov = pdpcprr, work_week = prfactr) %>%
  mutate(actidr = as.character(actidr),
         work_week = as.factor(work_week)) %>%
adjust_childcode(country = "in")

wave5_pe_yc <- read.stata("unproc_data/raw_data/yc/w5_yc_pe_activity.dta", country = "pe",
rem.number = F) %>% rename_with(~str_remove(.x, "5$")) %>% rename_with(~str_remove(.x,
"4$")) %>%
  dplyr::select(childcode, actidr, actr, actdayr, acthrsr, pymrecr,
         pdcashr, pdtimer, pdnmpcr, pdkindr) %>%
```

```
  mutate(work_week = if_else(is.na(actr)==F, 1, 0)) %>%
  rename(type_act = actr, wrkdyscov = actdayr, wrkhrscov = acthrsr,
         frmpaycov = pymrecr, erncshcov = pdcashr, ernkndcov = pdkindr,
         prdcvrcov = pdtimer, pcedaycov = pdnmpcr) %>%
  mutate(actidr = as.character(actidr),
         work_week = as.factor(work_week)) %>%
adjust_childcode(country = "pe")

wave5_vn_yc <- read.stata("unproc_data/raw_data/yc/w5_yc_vn_activity.dta", country = "vn",
rem.number = F) %>% rename_with(~str_remove(.x, "5$")) %>%
  dplyr::select(childcode, actidr, actr, actdayr, acthrsr, pymrecr,
         pdcashr, pdtimer, pdnmpcr, pdkindr) %>%
  mutate(work_week = if_else(is.na(actr)==F, 1, 0)) %>%
  rename(type_act = actr, wrkdyscov = actdayr, wrkhrscov = acthrsr,
         frmpaycov = pymrecr, erncshcov = pdcashr, ernkndcov = pdkindr,
         prdcvrcov = pdtimer, pcedaycov = pdnmpcr) %>%
  mutate(actidr = as.character(actidr),
         work_week = as.factor(work_week)) %>%
adjust_childcode(country = "vn")

wave5_et_yc <- read.stata("unproc_data/raw_data/yc/w5_yc_et_activity.dta", country = "et",
rem.number = F) %>% rename_with(~str_remove(.x, "5$")) %>%
  dplyr::select(childcode, actidr, actr, actdayr, acthrsr, pymrecr,
         pdcashr, pdtimer, pdnmpcr, pdkindr) %>%
  mutate(work_week = if_else(is.na(actr)==F, 1, 0)) %>%
  rename(type_act = actr, wrkdyscov = actdayr, wrkhrscov = acthrsr,
         frmpaycov = pymrecr, erncshcov = pdcashr, ernkndcov = pdkindr,
         prdcvrcov = pdtimer, pcedaycov = pdnmpcr) %>%
  mutate(actidr = as.character(actidr),
         work_week = as.factor(work_week))  %>%
adjust_childcode(country = "et")

data <- bind_rows(wave5_et, wave5_et_yc, wave5_in, wave5_in_yc, wave5_pe, wave5_pe_yc,
wave5_vn, wave5_vn_yc)

#cleaning data
data <- data %>%
  mutate(ernkndcov = str_remove_all(ernkndcov, "[^\\d-]"),
         erncshcov = str_remove_all(erncshcov, "[^\\d-]"),
         pcedaycov = str_remove_all(pcedaycov, "[^\\d-]")) %>% #remove letters
  mutate(
    pcedaycov = ifelse(str_detect(pcedaycov, "^-") == T, NA, pcedaycov),
    wrkdyscov = ifelse(str_detect(wrkdyscov, "^-") == T, NA, wrkdyscov),
    wrkhrscov = ifelse(str_detect(wrkhrscov, "^-") == T, NA, wrkhrscov),
    erncshcov = ifelse(str_detect(erncshcov, "^-") == T, NA, erncshcov),
    ernkndcov = case_when(
          str_detect(ernkndcov, "^-") ~ NA_character_, #negative numbers were used as placeholders for
NA
          ernkndcov == "" ~ NA, # Employed
          ernkndcov == "00" ~ "0",  # Employed
          ernkndcov == "000" ~ "0",
```

```
        TRUE ~ ernkndcov),
    type_act = ifelse(type_act == "N/A", NA, type_act),
    type_act = ifelse(type_act == "NA", NA, type_act), #changing NAs into real NAs
    erncshcov = as.numeric(erncshcov),
    ernkndcov = as.numeric(ernkndcov),
    pcedaycov = as.numeric(pcedaycov),
    wrkdyscov = as.numeric(wrkdyscov),
    wrkhrscov = as.numeric(wrkhrscov),
    work_week = case_when(
      work_week == 1 ~ "Yes",
      work_week == 0 ~ "No",
      TRUE ~ work_week),
    actidr = case_when(
      str_detect(actidr, "1") ~ 1,
      str_detect(actidr, "2") ~ 2,
      str_detect(actidr, "3") ~ 3,
      str_detect(actidr, "4") ~ 4,
      str_detect(actidr, "5") ~ 5,
      str_detect(actidr, "6") ~ 6,
      str_detect(actidr, "Primary") ~ 1,
      str_detect(actidr, "Second") ~ 2,
      str_detect(actidr, "Third") ~ 3,
      TRUE ~ NA_real_
    ))

# Compared to Wave 6, in Wave 5 non-paying activities are also considered "working"
# we thus call all activities type_act_alsononiga which includes non-paying (things like domestic
chores, studying, childcare)
# type_act only includes paying activities

data <- data %>% mutate(
      type_act_also_no_iga = type_act,
      type_act = ifelse(frmpaycov != "None", type_act_also_no_iga, NA))

#The following we all calculate per activity then we later synthesize:
# Hours worked per month and week assuming 4.345 weeks a month on average
  data <- data %>% mutate(month_hoursworked = wrkdyscov*wrkhrscov,
                          week_hoursworked = round(wrkdyscov*wrkhrscov/4.345, 1),
                          week_daysworked = wrkdyscov / 4.345)

  # We will also create a formal_work_week variable, which should be equal to "yes" if type_act is
non-NA, this excludes counting non-paying activities as jobs.
  #specifically Domestic chores and childcare or care of elders should not be considered an activity to
enable comparison across waves (this was not an option during Wave 6)
  data <- data %>% mutate(
    work_week = case_when(
      type_act == "Domestic chores" ~ "No",
      type_act == "Childcare or care for others" ~ "No",
      type_act == "Childcare or care of elders" ~ "No",
      type_act == "Non-remunerated household member" ~ "No",
      type_act == "Housewife" ~ "No",
```

```
    is.na(type_act_also_no_iga) == F ~ "Yes",
    TRUE ~ work_week
    ), # making sure work_week is correctly specified
   formal_work_week = case_when(
    is.na(type_act) == T & is.na(type_act_also_no_iga) == F ~ "No",
    is.na(type_act) == T & is.na(type_act_also_no_iga) == T ~ "No",
    TRUE ~ work_week
   )
  )

 # income
 data <- data %>% mutate(total_ear = case_when(
 is.na(ernkndcov) & is.na(erncshcov) ~ NA_real_,  # Return NA if both are NA
 TRUE ~ coalesce(ernkndcov, 0) + coalesce(erncshcov, 0)))

 # Now we have to calculate their weekly income.
 # However, applicants themselves could choose their time-period to determine their income. This
thus has to be converted
 # To converse between different timeframes, we assume fulltime work (8 hours a day, 5 days a week,
4.345 weeks a month)
 data <- data %>%
   mutate(
    full_time_weekly_income = case_when(
     frmpaycov == "None" ~ 0,  # Return 0 as numeric
     frmpaycov %in% c("Other, specify", "Other,specify", "Other, Specify", "NK", "NA", "N/A",
"Refused to Answer", "Other", "Doesn't know", "Refused to answer", "Debt relief") ~ NA_real_, #
Use NA_real_ for numeric NA
     prdcvrcov %in% c("Other, specify", "NK", "Other", "Doesn't know", "Refused to answer", "Debt
relief", "Other (specify)") ~ NA_real_, # Use NA_real_ for numeric NA
     prdcvrcov == "Per hour" ~ total_ear * 8 * 5,
     prdcvrcov == "Per day" ~ total_ear * 5,
     prdcvrcov == "Per week" ~ total_ear,
     prdcvrcov == "Per month" ~ total_ear / 4.345,
     prdcvrcov == "Per year" ~ total_ear / 4.345 / 12,
     prdcvrcov == "Per piece" ~ total_ear * pcedaycov * 5,
     prdcvrcov == "Fortnightly" ~ total_ear / 2,
     TRUE ~ NA_real_  # Handle any other unspecified cases
    )
   )

 #to compare fulltime and parttime, we also calculate earnings per hour
 data <- data %>%
   mutate(
    hourly_income = case_when(
     full_time_weekly_income == 0 ~ 0,
     week_hoursworked == 0 & full_time_weekly_income != 0 ~ NA_real_, #then unknown
     full_time_weekly_income != 0 ~ full_time_weekly_income/coalesce(week_hoursworked, 40),
     TRUE ~ NA_real_
    ),
    real_weekly_income = hourly_income* week_hoursworked)
```

```
#Compared to wave 6 the respondents were not asked for total earnings for their main acitivty, but total earnings per activity. to allow for comparisons across surveys, we only count the respoondents main activity, defined as the activity with their highest real_weekly_income
# Assuming 'data' is your dataset

results <- data %>%
  group_by(childcode) %>%
  arrange(actidr) %>%
  # Check if all real_weekly_income values are NA and calculate max income if not
  mutate(
    all_na_income = all(is.na(real_weekly_income)),
    max_income = ifelse(all_na_income, NA_real_, max(real_weekly_income, na.rm = TRUE))
  ) %>%
  # Filter rows: choose the max income or, if all are NA, the first actidr
  filter(real_weekly_income == max_income | (all_na_income & row_number() == 1)) %>%
  # Resolve ties by dplyr::selecting the minimum actidr
  slice_min(actidr) %>%
  ungroup() %>%
  dplyr::select(-max_income, -all_na_income)  # Clean up by removing the helper columns

#we first have to gather some additional information from the constructed databases

ind <- read.stata("unproc_data/constructed_data/Constructed Wave 1-
5/stata/stata13/in_constructed.dta", country = "in", rem.number = F) %>%
  rename(childcode = childid) %>% dplyr::select(childcode, yc, dint, round, inround, enrol, wi) %>%
  adjust_childcode(country = "in")

pe <- read.stata("unproc_data/constructed_data/Constructed Wave 1-
5/stata/stata13/pe_constructed.dta", country = "pe", rem.number = F)  %>%
  rename(childcode = childid) %>% dplyr::select(childcode, yc, dint, round, inround, enrol, wi) %>%
  adjust_childcode(country = "pe")

vn <- read.stata("unproc_data/constructed_data/Constructed Wave 1-
5/stata/stata13/vn_constructed.dta", country = "vn", rem.number = F) %>%
  rename(childcode = childid) %>% dplyr::select(childcode, yc, dint, round, inround, enrol, wi_new)
%>%
  adjust_childcode(country = "vn") %>% rename(wi = wi_new)

et <- read.stata("unproc_data/constructed_data/Constructed Wave 1-
5/stata/stata13/et_constructed.dta", country = "et", rem.number = F)%>%
  rename(childcode = childid) %>% dplyr::select(childcode, yc, dint, round, inround, enrol, wi_new)
%>%
  adjust_childcode(country = "et") %>% rename(wi = wi_new)

#combine and only include wave 5
con_data <- bind_rows(ind, pe, vn, et) %>% filter(round == 5) %>% filter(childcode != "childid")

#and join dataframe
```

```
merge <- results %>% dplyr::select(childcode, type_act, type_act_also_no_iga, work_week,
formal_work_week, week_hoursworked, month_hoursworked, full_time_weekly_income,
hourly_income, real_weekly_income)

con_data <- con_data %>%
  left_join(merge, by = c("childcode"))

remove(data, merge)


#we first have people without activities, who so far have NAs. But if inround == "Yes" then those
should be equal to 0
con_data <- con_data %>% mutate(
  work_week = case_when(
    is.na(work_week) == T & inround == "yes" ~ "No",
    TRUE ~ work_week
  ),
  formal_work_week = case_when(
    is.na(formal_work_week) == T & inround == "yes" ~ "No",
    TRUE ~ formal_work_week
  ),
  week_hoursworked = case_when(
    is.na(week_hoursworked) == T & inround == "yes" ~ 0,
    TRUE ~ week_hoursworked
  ),
  month_hoursworked = case_when(
    is.na(month_hoursworked) == T & inround == "yes" ~ 0,
    TRUE ~ month_hoursworked
  ),
  full_time_weekly_income = case_when(
    is.na(full_time_weekly_income) == T & inround == "yes" ~ 0,
    TRUE ~ full_time_weekly_income
  ),
  hourly_income = case_when(
    is.na(hourly_income) == T & inround == "yes" ~ 0,
    TRUE ~ hourly_income
  ),
  real_weekly_income = case_when(
    is.na(real_weekly_income) == T & inround == "yes" ~ 0,
    TRUE ~ real_weekly_income
  ))


############
#Employment Status
############
#NEET
# Binary Variable 0 if a) worked at least one hour during the 7 days before the call, or b) did not work
during the 7 days before, but did have a job, we include non-paying jobs like housewife, but exclude
domestic chores specifically
# or c) was attending education or training during the same year and
```

```
#1 otherwise
# we assume that if people do not report being enrolled, or having a job but not going to work that they are NEET.

con_data <- con_data %>% mutate(year_int = substr(dint, 1, 4)) %>%
  mutate(neet = case_when(
    work_week == "Yes" ~ 0,  # Employed
    enrol == "yes" ~ 0,  # Enrolled in school
    inround == "no" ~ NA,
    TRUE ~ 1  # Not working or not enrolled in the respective years
  ))

#any_iga
# Binary variable: 0) if a) worked at least one hour during the 7 days before the call, and generated some income, so unpaid work like family worker etc is excluded.

con_data  <- con_data %>% mutate(any_iga = case_when(
  hourly_income > 0 ~ 1, #paid work
  inround == "no" ~ NA,
  TRUE ~ 0
))

#any_formal job
# Binary variable: 1) if "Regular Salaried Employment" or similar
# Wage Employment (Unsalaried/ irregular; Non-agriculture) is not considered formal work
# 0) otherwise (including unemployed, self-e mployment or dependent worker (working without contract))
# NA if type of activity is not known, but the person is working and not studying)

con_data  <- con_data %>% mutate(formal_iga = case_when(
  type_act %in% c("Regular Salaried Employment", "Wage employment (Agriculture)", "Wage Employment (Agriculture)", "Waged worked", "Working for wage in non-agricultural activities (e.g. in mine/workshop/factory/construction/making food or drink)", "Working for wage in non-agricultural activities, e.g. in mine/workshop/factory/construction/making food or drink", "Annual Farm Servant") ~ 1, #Formal
  is.na(type_act) == T & any_iga == 1 & enrol == "Not enrolled" ~ NA, #unknown type of work
  inround == "no" ~ NA,
  TRUE ~ 0 #Non formal or unemployed
))


#self-employment
#Binary variable 1) if Self Employed is in type of activity
#             0) otherwise
#             99) if type of activity is not known, but the person is working and not studying

con_data  <- con_data %>% mutate(self_emp = case_when(
  str_detect(type_act, "Self Employed") ~ 1, #Self Employed
  str_detect(type_act, "Self-Employed") ~ 1, #Self Employed
  str_detect(type_act, "Self-employed") ~ 1,
  str_detect(type_act, "Self Employed") ~ 1,
```

```
  str_detect(type_act, "elf Employed") ~ 1, #typo
   str_detect(type_act, "Selling goods") ~ 1,
  str_detect(type_act, "Making") ~ 1,
  str_detect(type_act, "Independent") ~ 1, #Self Employed Artisan or trader
  str_detect(type_act, "own farm") ~ 1, #Self Employed
  is.na(type_act) == T & any_iga == 1 & enrol == "Not enrolled" ~ NA,
  inround == "no" ~ NA,
  TRUE ~ 0 #Not Self Employed
))

#Non-farming IGA
#Binary Variable 1) if any_iga = 1 and agri = "No"
#             0) otherwise
#             99) if sector of employment is unknown, but the person is working and not studying


con_data  <- con_data %>% mutate(
  agri = case_when(
    str_detect(type_act, "non-Agriculture") ~ "No",
    str_detect(type_act, "non-agriculture") ~ "No",
    str_detect(type_act, "non- agriculture") ~ "No",
    str_detect(type_act, "not related to agriculture") ~ "No",

    str_detect(type_act, "Farm") ~ "Yes",
    str_detect(type_act, "farm") ~ "Yes",
    str_detect(type_act, "(allied) agriculture") ~ "Yes",
    str_detect(type_act, "agriculture labourer") ~ "Yes",
    str_detect(type_act, "Food crops") ~ "Yes",
    str_detect(type_act, "Non-food, including horticulture, sericulture and floriculture") ~ "Yes",
    str_detect(type_act, "Livestock") ~ "Yes",
    str_detect(type_act, "Agriculture") ~ "Yes",
    str_detect(type_act, "agriculture") ~ "Yes",
    str_detect(type_act, "Other (SPECIFY)") ~ NA, #unclear
    str_detect(type_act, "Other, specify") ~ NA, #unclear
    TRUE ~ "No"),  # and othewise no
  nonfarm_iga = case_when(
  any_iga == 1 & agri == "No" ~ 1, #Employed but not in agri
  any_iga == 1 & agri == "Yes" ~ 0, #Employed in Agri
  any_iga == 0 ~ 0, #unemployed
  TRUE ~ NA_real_)
  )

#month_hoursworked, week_hoursworked and income should be 0 if NA and there is no IGA and it is still NA
con_data <- con_data %>%
  mutate(
    month_hoursworked = case_when(
     is.na(month_hoursworked) & any_iga == 0 ~ 0,
     is.na(month_hoursworked) ~ NA_real_,
    TRUE ~ month_hoursworked
    ),
```

```r
    week_hoursworked = case_when(
     is.na(week_hoursworked) & any_iga == 0 ~ 0,
     is.na(week_hoursworked) ~ NA_real_,
      TRUE ~ week_hoursworked
    ),
    full_time_weekly_income  = case_when(
     is.na(full_time_weekly_income ) & any_iga == 0 ~ 0,
     is.na(full_time_weekly_income ) ~ NA_real_,
      TRUE ~ full_time_weekly_income
    ),
    hourly_income = case_when(
      is.na(hourly_income) & any_iga == 0 ~ 0,
      is.na(hourly_income) ~ NA_real_,
      TRUE ~ hourly_income
    ),
    real_weekly_income = case_when(
      is.na(real_weekly_income) & any_iga == 0 ~ 0,
      is.na(real_weekly_income) ~ NA_real_,
      TRUE ~ real_weekly_income
    )
  )

output_wave5 <- con_data %>% dplyr::select(childcode, inround, dint, enrol, wi, week_hoursworked, month_hoursworked, full_time_weekly_income, hourly_income, real_weekly_income, type_act, neet, any_iga, formal_iga, self_emp, nonfarm_iga) %>% mutate(call = "Wave 5")

remove(con_data, et, ind, pe, vn, results, wave5_et, wave5_et_yc, wave5_in, wave5_in_yc, wave5_pe, wave5_pe_yc, wave5_vn, wave5_vn_yc)
```

3.3. Convert currencies to US$ in data cleaning
Then converts it into us$

```{r}
wave6_et <- wave6outcome(country = "et", cohort = "oc")
wave6_et_yc <- wave6outcome(country = "et", cohort = "yc")
wave6_in <- wave6outcome(country = "in", cohort = "oc")
wave6_in_yc <- wave6outcome(country = "in", cohort = "yc")
wave6_pe <- wave6outcome(country = "pe", cohort = "oc")
wave6_pe_yc <- wave6outcome(country = "pe", cohort = "yc")
wave6_vn <- wave6outcome(country = "vn", cohort = "oc")
wave6_vn_yc <- wave6outcome(country = "vn", cohort = "yc")

wave6 <- bind_rows(wave6_et, wave6_et_yc, wave6_in, wave6_in_yc, wave6_pe, wave6_pe_yc, wave6_vn, wave6_vn_yc)

combined_outcomes <- bind_rows(output_wave5, wave6) %>%
  mutate(inround = case_when(
    inround == "Participant is present in call" ~ "yes",
    inround == "Participant is not present in call" ~ "no",
```

```
  TRUE ~ inround
 ))

remove(wave6_et, wave6_et_yc, wave6_in, wave6_in_yc, wave6_pe, wave6_pe_yc, wave6_vn,
wave6_vn_yc, et_oc_call1, et_oc_call2, et_oc_call3, et_oc_call4, et_oc_call5, et_oc_constructed_call,
in_oc_call1, in_oc_call2, in_oc_call3, in_oc_call4, in_oc_call5, in_oc_constructed_call, pe_oc_call1,
pe_oc_call2, pe_oc_call3, pe_oc_call4, pe_oc_call5, pe_oc_constructed_call, vn_oc_call1,
vn_oc_call2, vn_oc_call3, vn_oc_call4, vn_oc_call5, vn_oc_constructed_call, et_yc_call1,
et_yc_call2, et_yc_call3, et_yc_call4, et_yc_call5, et_yc_constructed_call, in_yc_call1, in_yc_call2,
in_yc_call3, in_yc_call4, in_yc_call5, in_yc_constructed_call, pe_yc_call1, pe_yc_call2, pe_yc_call3,
pe_yc_call4, pe_yc_call5, pe_yc_constructed_call, vn_yc_call1, vn_yc_call2, vn_yc_call3,
vn_yc_call4, vn_yc_call5, vn_yc_constructed_call)
```

Historical data is imported externally from Yahoo Finance, and is publicly available
https://finance.yahoo.com/quote/INRUSD%3DX/history #e.g. for India

```{r}
ind <- read_csv("unproc_data/forex historical data/INRUSD=X.csv") %>% dplyr::select(Date, Open)
%>%
  rename(dint = Date, indrate = Open) %>% mutate(dint = as.Date(dint))
all_dates <- data.frame(dint = seq(min(ind$dint), max(ind$dint), by = "day"))

et <- read_csv("unproc_data/forex historical data/ETBUSD=X.csv") %>% dplyr::select(Date, Open)
%>%
  rename(dint = Date, etrate = Open) %>% mutate(dint = as.Date(dint)) %>%
  right_join(all_dates, by = "dint") %>% mutate(etrate = ifelse(etrate == "null", NA, etrate))%>%
  arrange(dint) %>%
  fill(etrate) #filling in missing values for weekends using last known exchange rate

ind <- ind %>%
  right_join(all_dates, by = "dint") %>% mutate(indrate = ifelse(indrate == "null", NA, indrate))%>%
arrange(dint) %>%
  fill(indrate) #filling in missing values for weekends using last known exchange rate

pe <- read_csv("unproc_data/forex historical data/PENUSD=X.csv") %>% dplyr::select(Date, Open)
%>%
  rename(dint = Date, perate = Open) %>% mutate(dint = as.Date(dint)) %>%
  right_join(all_dates, by = "dint") %>% mutate(perate = ifelse(perate == "null", NA, perate))%>%
  arrange(dint) %>%
  fill(perate) #filling in missing values for weekends using last known exchange rate

vn <- read_csv("unproc_data/forex historical data/VNDUSD=X.csv") %>% dplyr::select(Date, Open)
%>%
  rename(dint = Date, vnrate = Open) %>% mutate(dint = as.Date(dint)) %>%
  right_join(all_dates, by = "dint") %>% mutate(vnrate = ifelse(vnrate == "null", NA, vnrate))%>%
  arrange(dint) %>%
  fill(vnrate) #filling in missing values for weekends using last known exchange rate


combined_outcomes$dint <- as.Date(combined_outcomes$dint)
```

```
combined_outcomes <- combined_outcomes %>%
  left_join(et, by = c("dint" = "dint")) %>%
  left_join(ind, by = c("dint" = "dint")) %>%
  left_join(pe, by = c("dint" = "dint")) %>%
  left_join(vn, by = c("dint" = "dint")) %>%
  mutate(indrate = as.numeric(indrate),
         perate = as.numeric(perate),
         etrate = as.numeric(etrate),
         vnrate = as.numeric(vnrate))

combined_outcomes <- combined_outcomes %>% mutate( #and use the exchange rates to recalculate incomes
  full_time_weekly_income = case_when(
    is.na(full_time_weekly_income) == T ~ NA,
    substr(childcode, 1, 2) == "IN" ~ full_time_weekly_income * indrate,
    substr(childcode, 1, 2) == "PE" ~ full_time_weekly_income * perate,
    substr(childcode, 1, 2) == "VN" ~ full_time_weekly_income * vnrate,
    substr(childcode, 1, 2) == "ET" ~ full_time_weekly_income * etrate,
    TRUE ~ NA
  ),
  hourly_income = case_when(
    is.na(hourly_income) == T ~ NA,
    substr(childcode, 1, 2) == "IN" ~ hourly_income * indrate,
    substr(childcode, 1, 2) == "PE" ~ hourly_income * perate,
    substr(childcode, 1, 2) == "VN" ~ hourly_income * vnrate,
    substr(childcode, 1, 2) == "ET" ~ hourly_income * etrate,
    TRUE ~ NA
  ),
  real_weekly_income = case_when(
    is.na(real_weekly_income) == T ~ NA,
    substr(childcode, 1, 2) == "IN" ~ real_weekly_income * indrate,
    substr(childcode, 1, 2) == "PE" ~ real_weekly_income * perate,
    substr(childcode, 1, 2) == "VN" ~ real_weekly_income * vnrate,
    substr(childcode, 1, 2) == "ET" ~ real_weekly_income * etrate,
    TRUE ~ NA
  )) %>% dplyr::select(-c(etrate, vnrate, indrate, perate))

remove(pe, ind, vn, et, all_dates)
```

##3.4. Save
```{r}
write.csv(combined_outcomes, "proc_data/outcomes_long.csv", row.names = FALSE)
```

#4. Merge into final non-inputed dataframe
This takes the edudummies, covariates and outcomes_long dataframes as inputs, and merges them in an appropiate way into one dataframe which can be used for computation

```{r}
library(dplyr)
library(forcats)
library(cobalt)
library(stargazer)
library(estimatr)
library(MatchIt)
library(WeightIt)
library(gbm)
library(optmatch)
library(sandwich)
library(lmtest)
```

##4.1. Transforming outcomes_long into wide dataframe and truncating
We now take the first year/call available after they have completed TVE or upper-secondary,

- Respondents have to be at least 18, this to get similar samples since nobody finished secondary/vocational before 18, this is to get similar sample sizes

- we then take the first year available after they are no longer inrolled, ignoring call 3 due to the many NAs

Only people that are not yet finished studying are included
If there are NA-values, these are not supplemented by later outcomes.

Later
Then we merge outcome_wide with the covariates and edudummies to get data

```{r}
years_into_dates <- function(data) {
data <- if_else(data == "never", "2000-2001", data)
data <- case_when(
  is.na(data) ~ NA_character_,
  data == "2000-2001" ~ "2001-06-30",
  data == "2001-2002" ~ "2002-06-30",
  data == "2002-2003" ~ "2003-06-30",
  data == "2003-2004" ~ "2004-06-30",
  data == "2004-2005" ~ "2005-06-30",
  data == "2005-2006" ~ "2006-06-30",
  data == "2006-2007" ~ "2007-06-30",
  data == "2007-2008" ~ "2008-06-30",
  data == "2008-2009" ~ "2009-06-30",
  data == "2009-2010" ~ "2010-06-30",
  data == "2010-2011" ~ "2011-06-30",
  data == "2011-2012" ~ "2012-06-30",
  data == "2012-2013" ~ "2013-06-30",
  data == "2013-2014" ~ "2014-06-30",
  data == "2014-2015" ~ "2015-06-30",
  data == "2015-2016" ~ "2016-06-30",
```

```
  data == "2016-2017" ~ "2017-06-30",
  data == "2017-2018" ~ "2018-06-30",
  data == "2018-2019" ~ "2019-06-30",
  data == "2019-2020" ~ "2020-06-30",
  data == "2020-2021" ~ "2021-06-30",
  data == "2021-2022" ~ "2022-06-30",
  data == "2006" ~ "2006-12-31",
  data == "2007" ~ "2006-12-31",
  data == "2008" ~ "2006-12-31",
  data == "2009" ~ "2009-12-31",
  data == "2010" ~ "2010-12-31",
  data == "2011" ~ "2011-12-31",
  data == "2012" ~ "2012-12-31",
  data == "2013" ~ "2013-12-31",
  data == "2014" ~ "2014-12-31",
  data == "2015" ~ "2015-12-31",
  data == "2016" ~ "2016-12-31",
  data == "2017" ~ "2017-12-31",
  data == "2018" ~ "2018-12-31",
  data == "2019" ~ "2019-12-31",
  data == "2020" ~ "2020-12-31",
  data == "2021" ~ "2021-12-31",
  TRUE ~ "Not found"
)
}

outcomes_long <- read.csv("~/1. UU/Thesis/Coding/proc_data/outcomes_long.csv")

library(dplyr)
# Map the call values to numeric for sorting and lagging
call_mapping <- c("Wave 5" = 1, "Call 2" = 2, "Call 3" = 3, "Call 5" = 5)

# Add a numeric call column based on the mapping
outcomes_long <- outcomes_long %>%
  mutate(call_numeric = call_mapping[call])
remove(call_mapping)

#preparing outcomes_long
outcomes_long <- outcomes_long %>% mutate(enrol = case_when(
  enrol == "Enrolled" ~ "yes",
  enrol == "Not enrolled" ~ "no",
  TRUE ~ enrol
)) %>%
  arrange(childcode, call_numeric) %>%
  group_by(childcode) %>%
  mutate(
    wi = if_else(call == "Call 3" & is.na(wi), lag(wi, n = 1, default = NA), wi),
    wi = if_else(call == "Call 5" & is.na(wi), lag(wi, n = 3, default = NA), wi) #lagging wi index since
it was only measured once during wave 6
  ) %>%
  ungroup() %>%
```

```
  dplyr::select(-call_numeric)

#acquiring edu-dummies to merge
edudummies <- read.csv("~/1. UU/Thesis/Coding/proc_data/edu_dummies.csv")

edudummiestomerge <- edudummies %>% dplyr::select(childcode, last_tvet_year,
year_graduated_upper_sec, attended_vocational_secondary, completed_general_upper_secondary)

outcomes_long <- outcomes_long %>% left_join(edudummiestomerge, by = "childcode") %>%
mutate(countrycode = substr(childcode, 1, 2))

#filter out non respondents
outcomes_long <- outcomes_long %>% filter(inround == "yes")

#for a few observations in round 5 in et dint is missing, despite giving all answers. We plug the mean
date of interview for ethopia in
outcomes_long <- outcomes_long %>% mutate(dint = ifelse(is.na(dint) == T, "2016-11-28", dint))

outcomes_long$last_tvet_year = years_into_dates(as.character(outcomes_long$last_tvet_year))

outcomes_long$year_graduated_upper_sec =
years_into_dates(as.character(outcomes_long$year_graduated_upper_sec))

#------------------------------------------------------------------------#
#calculating age
years_into_dates_age <- function(data) {
  data <- case_when(
    is.na(data) ~ NA_character_,
    data == 1992 ~ "1992-06-30",
    data == 1994 ~ "1994-06-30",
    data == 1995 ~ "1995-06-30",
    data == 1996 ~ "1996-06-30",
    data == 2000 ~ "2000-06-30",
    data == 2001 ~ "2001-06-30",
    data == 2002 ~ "2002-06-30",
    TRUE ~ "Not found"
  )
}

merge <- read.csv("proc_data/covariates.csv")
merge <- merge %>% dplyr::select(childcode, year_of_birth)

outcomes_long <- outcomes_long %>% left_join(merge, by = "childcode")
remove(merge)

outcomes_long$year_of_birth_date <- years_into_dates_age(outcomes_long$year_of_birth)

# Function to calculate age
calculate_age <- function(data, dint_col, yob_col) {
  # Ensure the date columns are in Date format
  data[[dint_col]] <- as.Date(data[[dint_col]])
```

```
  data[[yob_col]] <- as.Date(data[[yob_col]])

  # Calculate age in years
  data$age_years <- as.numeric(difftime(data[[dint_col]], data[[yob_col]], units = "days")) / 365.25

  data$age_years <- round(data$age_years, 0)

  return(data)
}

outcomes_long <- calculate_age(outcomes_long, "dint", "year_of_birth_date")
outcomes_long <- outcomes_long %>% dplyr::select(-c(year_of_birth, year_of_birth_date))

outcomes_long <- outcomes_long %>%
  mutate(
     last_tvet_year = as.Date(last_tvet_year),
     dint = as.Date(dint),
     year_graduated_upper_sec = as.Date(year_graduated_upper_sec)
    ) %>%

  #flagging if graduation was after interview or age < 18
  mutate(
     graduate_tvet_after = ifelse(dint < last_tvet_year, 1, ifelse(is.na(last_tvet_year) == T, NA, 0)),
     graduated_upper_sec_after = ifelse(dint < year_graduated_upper_sec, 1,
ifelse(is.na(year_graduated_upper_sec) == T, NA, 0)),
     age_below_18 = ifelse(age_years < 18, 1, 0)) %>%

#however the course is already finished if they reported not being enrolled during the last wave 5,
probably since the course is finished slightly earlier than anticipated. Thus enroll = "no" for wave 5
should override the above, by definition it's not possible that they started their education after the
interview, since then its data would not have been collected.
  mutate(graduate_tvet_after = case_when(
    enrol == "no" & call == "Call 5" ~ 0,
    TRUE ~ graduate_tvet_after),
    graduated_upper_sec_after = case_when(
    enrol == "no" & call == "Call 5" ~ 0,
    TRUE ~ graduated_upper_sec_after)) %>% #objection is 1 if there is a reason not to include that
row in the dataset (e.g. currently enrolled, or finishing TVET or upper secondary later)
    mutate(objection = case_when(
       enrol == "yes" ~ 1,
       graduate_tvet_after == 1 ~ 1,
       graduated_upper_sec_after == 1 ~ 1,
       age_below_18 == 1 ~ 1,
       TRUE ~ 0
     ))

# Now find the first column per childcode with no objection, thus making a wide df
outcomes_wide <- outcomes_long %>%
  filter(call != "Call 3", objection == 0) %>%
  arrange(childcode, dint) %>%
  group_by(childcode) %>%
```

```
  slice_head(n = 1) %>%
  ungroup() %>% dplyr::select(-c(last_tvet_year, year_graduated_upper_sec,
attended_vocational_secondary, completed_general_upper_secondary, countrycode,
graduate_tvet_after, graduated_upper_sec_after, age_below_18, objection))
```

Preparing outcome wide + truncating
Removing outliers in outcomes
I remove very clear data errors that are not possible, replacing them with NA

Then:For continuous variables not limited to 0-1 scale, I truncate everything above 99% at 99%, to correct for smaller data errors and outlier effects.

```{r}
library(dplyr)

# Define a function to truncate values at the 99th percentile
truncate_at_99 <- function(x, na_value = 10000) {
  x <- ifelse(x > na_value, NA_real_, x) # Replace extreme outliers with NA
  threshold <- quantile(x, 0.99, na.rm = TRUE)
  pmin(x, threshold, na.rm = TRUE)
}

#the values are the benchmark of irrealist -> then replaced with na
outcomes_wide <- outcomes_wide %>%
  mutate(
    week_hoursworked = truncate_at_99(week_hoursworked, 120),
    month_hoursworked = truncate_at_99(month_hoursworked, 480),
    full_time_weekly_income = truncate_at_99(full_time_weekly_income, 10000),
    hourly_income = truncate_at_99(hourly_income, 250),
    real_weekly_income = truncate_at_99(real_weekly_income, 10000)
  ) %>%
  mutate(
    year = substr(outcomes_wide$dint, 1, 4)) #calculates year

# Print the summary statistics and histograms for verification
summary(outcomes_wide$week_hoursworked)
hist(outcomes_wide$week_hoursworked)

summary(outcomes_wide$month_hoursworked)
hist(outcomes_wide$month_hoursworked)

summary(outcomes_wide$full_time_weekly_income)
hist(outcomes_wide$full_time_weekly_income)

summary(outcomes_wide$hourly_income)
hist(outcomes_wide$hourly_income)

summary(outcomes_wide$real_weekly_income)
hist(outcomes_wide$real_weekly_income)
```

```
#change to factors
outcomes_wide <- outcomes_wide %>%
  mutate_if(is.integer, as.factor) %>%
  mutate_if(is.character, as.factor)
```

##4.2. Calculating work experience
work_experience:
Defined as years since last year of study, starting counting from the age of 15.
I assume mid-school years end 30th June
and I assume full school years end 31th December

We then calculate months since last time working.
However there is still a problem: some people have started working and then later re-enrolled resulting in negative work_experiences

```{r}
merge <- read.csv("~/1. UU/Thesis/Coding/proc_data/edu_dummies.csv")
merge <- edudummies %>% dplyr::select(c(childcode, year_graduated_upper_sec, last_tvet_year))

merge2 <- read.csv("proc_data/covariates.csv")
merge2 <- merge2 %>% dplyr::select(childcode, year_of_birth)

outcomes_wide <- outcomes_wide %>% left_join(merge, by = "childcode") %>%
  left_join(merge2, by = "childcode")

#never corresponds to earliest (2000-2001)
outcomes_wide$year_graduated_upper_sec <-
years_into_dates(outcomes_wide$year_graduated_upper_sec)
outcomes_wide$last_tvet_year <- years_into_dates(outcomes_wide$last_tvet_year)

# Convert date columns to date objects
outcomes_wide <- outcomes_wide %>%
  mutate(
    last_tvet_year = as.Date(last_tvet_year),
    year_graduated_upper_sec = as.Date(year_graduated_upper_sec),
    dint = as.Date(dint)
  )

# Calculate the date when the person turned 16
outcomes_wide <- outcomes_wide %>%
  mutate(date_turned_16 = as.Date(paste0(as.numeric(as.character(year_of_birth)) + 16, "-01-01")))

# Calculate work experience in months
outcomes_wide <- outcomes_wide %>%
  mutate(work_experience = ifelse(dint >= date_turned_16,
                                  round(difftime(dint, pmax(last_tvet_year, year_graduated_upper_sec,
date_turned_16, na.rm = T), units = "days") / 30.5),  0)) %>%
  mutate(work_experience = ifelse(work_experience < 0, 0, work_experience)) %>% #correcting small
inaccuracies
```

```
  mutate(work_experience = work_experience/12) #in years

table(outcomes_wide$work_experience)

#and a grouped work_experience variable
# Define the breaks for the intervals
breaks <- seq(0, max(outcomes_wide$work_experience, na.rm = TRUE), by = 0.5)
# Add an upper bound for the maximum work experience
breaks <- c(breaks, Inf)

# Define the labels for the intervals
labels <- paste(breaks[-length(breaks)], "-",  breaks[-1], sep = "")
labels[length(labels)] <- paste(breaks[length(breaks) - 1], "+", sep = "")

# Correct the labels to ensure they correctly reflect intervals like "0 - 0.5", "0.5 - 1.0", etc.
for (i in 1:(length(labels) - 1)) {
  labels[i] <- paste0(breaks[i], "-", breaks[i+1])
}

# Apply the cut function to create the categories
outcomes_wide <- outcomes_wide %>%
  mutate(work_experience_grouped = cut(work_experience, breaks = breaks, labels = labels, right =
FALSE))

outcomes_wide <- outcomes_wide %>% dplyr::select(-c(year_graduated_upper_sec, last_tvet_year,
year_of_birth, date_turned_16))
remove(breaks, labels, merge, merge2)

```

##4.3. Preparing covariates + truncating
we change the integers and characters into factors,
then make sure the right variables are numeric
all numeric variables are truncated at 99% and when possible at 1% to remove outliers and errors.

```{r}
library(dplyr)
covariates <- read.csv("proc_data/covariates.csv")
age <- outcomes_wide %>% dplyr::select(childcode, age_years, work_experience,
work_experience_grouped)
covariates <- covariates %>% left_join(age, by = "childcode")
outcomes_wide <- outcomes_wide %>% dplyr::select(-c(age_years, work_experience,
work_experience_grouped))

covariates <- covariates %>%
  mutate_if(is.integer, as.factor) %>%
  mutate_if(is.character, as.factor) %>%
  mutate(
    noncog_friend = as.numeric(noncog_friend),
    dadage_atbirth = as.numeric(dadage_atbirth),
    momage_atbirth = as.numeric(momage_atbirth),
```

```
    careage_atbirth = as.numeric(careage_atbirth),
    hhsize = as.numeric(hhsize),
    male012 = as.numeric(male012),
    female012 = as.numeric(female012),
    bornbef = as.numeric(bornbef),
    bornaft = as.numeric(bornaft),
    total_children_household = as.numeric(total_children_household),
    subjective_health_13 = as.numeric(subjective_health_13),
    popsize = as.numeric(popsize),
    timecap = as.numeric(timecap)
  ) %>% dplyr::select(-momedu, -dadedu, -caredu)

#any numeric covariates are truncated at 99% for data errors
# Define a function to truncate at the 99th percentile

truncate_at_1_and_99 <- function(x) {
  quantiles <- quantile(x, c(0.01, 0.99), na.rm = TRUE)
  x[x < quantiles[1]] <- quantiles[1]
  x[x > quantiles[2]] <- quantiles[2]
  return(x)
}

# Apply the truncation function to all numeric variables
covariates <- covariates %>%
  mutate(across(where(is.numeric), truncate_at_1_and_99))

#for religion and ethnic group if less than 10 observations move to other category
# Get the counts of each level
counts <- table(covariates$chrel)
levels_to_combine <- names(counts[counts < 11])

# Update variable combining levels with fewer than 10 observations into "Other"
covariates$chrel <- as.character(covariates$chrel)
covariates$chrel[covariates$chrel_combined %in% levels_to_combine] <- "other"

# Convert the variable back to a factor
covariates$chrel <- factor(covariates$chrel)

counts <- table(covariates$chethnic)
levels_to_combine <- names(counts[counts < 11])

# Update variable combining levels with fewer than 10 observations into "Other"
covariates$chethnic <- as.character(covariates$chethnic)
covariates$chethnic[covariates$chethnic_combined %in% levels_to_combine] <- "other"

# Convert the variable back to a factor
covariates$chethnic <- factor(covariates$chethnic)

```

````
##4.4. Preparing edu-dummies
```{r}
edudummies <- read.csv("~/1. UU/Thesis/Coding/proc_data/edu_dummies.csv")

edudummies <- edudummies %>%
  mutate_if(is.integer, as.factor) %>%
  mutate_if(is.character, as.factor)

```



##4.5. Merge data
```{r}
data_notinputed <- outcomes_wide %>% left_join(edudummies, by = "childcode") %>%
left_join(covariates, by = "childcode") %>% rename(wi = wi.x, countrycode = countrycode.x) %>%
dplyr::select(-countrycode.y, wi.y)

remove(outcomes_long, outcomes_wide, covariates_inputed, covariates, edudummies, age,
edudummiestomerge)
```

##4.6. Create treatment variables
```{r}
data_notinputed <- data_notinputed %>% mutate(vocational_vs_general_secondary = case_when(
  attended_vocational_secondary == 1 ~ 1,
  attended_vocational_secondary == 0 & completed_general_upper_secondary == 1 ~ 0,
  TRUE ~ NA))

data_notinputed <- data_notinputed %>% mutate(vocational_vs_drop_out = case_when(
  attended_vocational_secondary == 1 ~ 1,
  attended_vocational_secondary == 0 & completed_general_upper_secondary == 0 &
completed_primary == 1 ~ 0, #graduated primary but not upper secondary
  TRUE ~ NA))

data_notinputed <- data_notinputed %>% mutate(attended_general_secondary = case_when(
  completed_general_upper_secondary == 1 ~ 1,
  TRUE ~ 0))

data_notinputed <- data_notinputed %>% mutate(treatment_3way = case_when(
  attended_vocational_secondary == 1 ~ "vocational secondary",
  attended_vocational_secondary == 0 & completed_general_upper_secondary == 0 &
completed_primary == 1 ~ "dropped out post-primary",
  attended_vocational_secondary == 0 & completed_general_upper_secondary == 1 ~
    "general secondary",
  TRUE ~ NA_character_))
```


##4.7 save data
```{r}
````
````

write.csv(data_notinputed, "~/1. UU/Thesis/Coding/proc_data/final_data_notinputed.csv", row.names = FALSE)

```
```

#5. Descriptives before Inputed Data
This chapter uses the not inputed data to test for NAs and descriptives,
after multiple imputation this would be a lot more difficult.
It can download the new data, and thus not it is not necessary to run the full code before.

##5.1. Grouping Variables

```{r}
yc <- "yc"
region <- "countrycode"
survey_dummies <- c(yc, region)

child_demographics_propensity <- c("chsex", "chethnic", "chrel")
#for propensity score calculations the above can be included, but sample size is insufficient to include high-factor variables in lm
child_demographics_noage <- c("chsex")

child_health <- c("zweight_8", "zheight_8", "zbmi_8", "chillness_8_13", "long_term_health_problem", "chdisability", "subjective_health_13")
child_time_use <- c("hsleep", "hcare", "hchore", "htask", "hwork", "hschool", "hstudy", "hplay", "chldwork_during_school", "missed_school")
child_cognitive_skills <- c("math_score_13", "math_score_improvement", "read_score_13", "ppvt_score_13", "ppvt_score_improvement")
child_non_cognitive_skills <- c("noncog_friend", "noncog_hardtalk", "noncog_incgame", "noncog_lead", "noncog_helpchld", "noncog_trust", "noncog_selfefficiacy", "noncog_selfesteem")
child_expectations <- c("expected_grade", "dreamjob_sector", "vocational_dreamjob_dummy", "academic_dreamjob_dummy")

family_demographics <- c("dadage_atbirth", "momage_atbirth", "careage_atbirth", "dadpassed", "mompassed", "primarycaregiver", "parent_sick")

family_size <- c("hhsize", "male012", "female012", "bornbef", "bornaft", "total_children_household")

family_education <- c("dadcantread", "momcantread", "carecantread", "mom_edu_attended_formaleducation",    "mom_edu_beyond_primaryeducation", "mom_edu_attended_postsecondary", "mom_edu_attended_vocational", "dad_edu_attended_formaleducation", "dad_edu_beyond_primaryeducation", "dad_edu_attended_postsecondary", "dad_edu_attended_vocational", "care_edu_attended_formaleducation", "care_edu_beyond_primaryeducation", "care_edu_attended_postsecondary", "care_edu_attended_vocational")

family_valuation_of_education <- c("formal_education_useful", "education_during_financial_hardship", "quality_primary_school")
```

```r
family_expectations <- c("expected_age_married", "expected_age_earning",
"expected_age_leaving_school", "realistic_expectations_parents", "parents_dreamjob_sector",
"parents_vocational_dreamjob_dummy", "parents_academic_dreamjob_dummy")

family_economics <- c("household_primary_job", "ownhouse", "hq", "sv", "cd", "debt")

family_shock <- c("shock_crime", "shock_household_job_loss", "shock_natural_disaster",
"shock_house_collapse")

community_type <- c("typesite_w1", "popsize", "timecap")
community_jobs <- c("agriculture_jobs", "factory_jobs", "craft_jobs")
community_education_available <- c("public_secondary_available", "private_secondary_available",
"lower_vocational_available", "public_higher_vocational_available",
"private_higher_vocational_available")

ivars <- c(survey_dummies, child_demographics_noage, child_health, child_time_use,
child_cognitive_skills, child_non_cognitive_skills, child_expectations, family_demographics,
family_size, family_economics, family_education, family_expectations,
family_valuation_of_education, community_type, community_jobs, community_education_available)
```

##5.2. Checking NAs and randomness
missing completely:
household primary job peru
formal education useful et and in mostly missing
hq, sv, cd country et and vn missing

```{r}
data_notinputed <- read_csv("proc_data/final_data_inputed.csv")

na_counts <- data_notinputed %>%
  dplyr::select(all_of(ivars)) %>%
  summarise(complete_cases = sum(complete.cases(.)),
            avg_missing_covariates = mean(rowSums(is.na(.))))

# View the result
print(na_counts)

# View the result
print(na_counts)
remove(na_counts)

#calculating NAs per country per cohort
na_counts_per_country <- data_notinputed %>%
  group_by(countrycode, yc) %>%
  dplyr::select(all_of(ivars)) %>%
  summarise_all(~ sum(is.na(.)) / n()) %>%
  ungroup() %>%
  mutate(countrycode = as.character(countrycode), yc = as.character(yc))
```

# Calculate the total average percentage missing
total_avg_missing <- na_counts_per_country %>%
  summarise_all(mean, na.rm = TRUE) %>%
  mutate(countrycode = "Total", yc = "Total")

# Combine the two data frames
na_counts_per_country <- bind_rows(na_counts_per_country, total_avg_missing)

# View the result
print(na_counts_per_country)
write.csv2(na_counts_per_country, "~/1. UU/Thesis/Coding/proc_data/missing_data.csv", row.names = FALSE)

remove(na_counts, na_counts_per_country)

library(naniar)
library(forcats)
library(finalfit)

#plotting missingness
data_notinputed %>% dplyr::select(all_of(ivars)) %>%
 missing_plot()

#checking missing at random assumption
temp <- data_notinputed %>%
  dplyr::select(all_of(ivars))
mcar_test(temp) #missing not at random

temp <- data_notinputed %>% dplyr::select(-c(read_score_13, household_primary_job, hq, sv, cd, dadcantread, momcantread, carecantread, typesite_w1, popsize, timecap, agriculture_jobs, factory_jobs, craft_jobs, public_secondary_available, private_secondary_available, lower_vocational_available, public_higher_vocational_available, private_higher_vocational_available)) %>% mutate(across(where(is.factor), ~ as.numeric(fct_relevel(.))))

test <- mcar_test(temp) #missing at random
summary(test)

missing_plot(temp)
remove(temp)

```

##5.3. Descriptives table
```{r}
data_summary <- data_notinputed

variables_to_split <- c("attended_higher_education", "neet","any_iga", "formal_iga", "self_emp", "nonfarm_iga", survey_dummies, child_demographics_noage, child_health, child_time_use, child_cognitive_skills, child_non_cognitive_skills, child_expectations, family_demographics,

family_size, family_economics, family_education, family_expectations,
family_valuation_of_education, community_type, community_jobs, community_education_available)

for (var in variables_to_split) {
  if (var %in% names(data_summary) && is.factor(data_summary[[var]])) {
    data_summary <- splitfactor(data_summary, var, drop.na = FALSE, drop.first = "if2")
  }
}

countrycode_table <- data_notinputed %>%
  group_by(treatment_3way, yc) %>%
  summarise(countrycode_table = list(table(countrycode)), .groups = 'drop')

countrycode_table
# Calculate summary statistics for numeric variables only

summary_stats <- data_summary %>%
  group_by(treatment_3way, `yc_Younger cohort`) %>%
  summarise(across(where(is.numeric),
    ~ round(mean(., na.rm = TRUE), 2)
  ), .groups = 'drop')

write.csv2(summary_stats, "~/1. UU/Thesis/Coding/proc_data/descriptives.csv", row.names = FALSE,
sep = ";")
```

# 6. Multiple Imputed Data
The following uses data_notinputed as input and creates a five-time imputed dataset, which will be
used for the actual analysis.

## 6.1. Compute imputed dataset
## Create inputed dataset
```{r include=FALSE}
library(mice)
library(dplyr)

data_notinputed <- read.csv("~/1. UU/Thesis/Coding/proc_data/final_data_notinputed.csv")

#dplyr::selecting variables to impute and prepping dataset
outcomes <- c("week_hoursworked", "hourly_income", "neet",
              "any_iga", "formal_iga", "self_emp", "nonfarm_iga", "attended_higher_education")

treatments <- c("attended_vocational_secondary", "vocational_vs_general_secondary",
"vocational_vs_drop_out")

ivars <- c("countrycode", "yc", "year", "work_experience", child_demographics_noage, child_health,
child_time_use, child_cognitive_skills, child_non_cognitive_skills, child_expectations,
family_demographics, family_size, family_economics, family_education, family_expectations,
family_valuation_of_education, community_type, community_jobs, community_education_available)

```
vars <- c(outcomes, treatments, ivars)

data_toimpute <- data_notinputed %>%
  dplyr::select(all_of(vars)) %>%
  mutate_if(is.character, as.factor) %>%
  mutate(across(where(is.integer), ~ if (n_distinct(.) == 2 || n_distinct(.) == 3 && any(is.na(.)))
as.factor(.) else .)) %>%
  mutate(year = as.factor(year))

#set prediction matrix, with parameters tuned to get 25 predictors per variable average
inlist <- c("yc", "countrycode", "chsex") #always to include
pred <- quickpred(data_toimpute, minpuc = 0.52, include = inlist)
mean(rowSums(pred)) #25 predictors on average which is perfect

#making sure treatments are not imputed and used as predictors
pred[,'attended_vocational_secondary'] = 0
pred[,'vocational_vs_general_secondary'] = 0
pred[,'vocational_vs_drop_out'] = 0

#set method
#run with zero repeats
imp <- mice(data_toimpute, maxit=0)
meth <- imp$method

#ordered categorical variables
poly <-  c("noncog_hardtalk", "noncog_incgame", "noncog_lead", "noncog_helpchld",
"formal_education_useful", "quality_primary_school", "agriculture_jobs",
"public_secondary_available", "private_secondary_available", "lower_vocational_available",
"public_higher_vocational_available", "private_higher_vocational_available")
#only the purely order categoricals are included

#reordering categorical variables if necessary
data_toimpute <- data_toimpute %>%
  mutate(
    noncog_hardtalk = factor(noncog_hardtalk, levels = c("never", "sometimes", "always")),
    noncog_incgame = factor(noncog_incgame, levels = c("never", "sometimes", "always")),
    noncog_lead = factor(noncog_lead, levels = c("never", "sometimes", "always")),
    noncog_helpchld = factor(noncog_helpchld, levels = c("never", "sometimes", "always")),
    quality_primary_school = factor(quality_primary_school, levels = c("strongly disagree", "disagree",
"more or less", "agree", "strongly agree")),
    agriculture_jobs = factor(agriculture_jobs, levels = c("not important", "somewhat important", "most
important"))
  )

#change methods
meth[poly] <- "polr"

#create imputational dataframe
data_mi <- mice(data_toimpute, pred = pred, method = meth, seed = 11062024, m = 5)
```

##6.2. Transforming MI Dataframing
Scaling and correcting to right data type: Calculating Propensity for General Secondary and Asess Sub Samples
Converting to right type

````
```{r}
#-------------------------------------------------------------#
#adjusting outcomes to numeric for later analyses
data_mi <- mice::complete(data_mi, action="long", include = TRUE)
data_mi$neet <- as.numeric(data_mi$neet)
data_mi$any_iga <- as.numeric(data_mi$any_iga)
data_mi$formal_iga <- as.numeric(data_mi$formal_iga)
data_mi$self_emp <- as.numeric(data_mi$self_emp)
data_mi$nonfarm_iga <- as.numeric(data_mi$nonfarm_iga)
data_mi$attended_higher_education <- as.numeric(data_mi$attended_higher_education)
data_mi <- as.mids(data_mi)

```
````

calculating SD

````
```{r}

#--------------------------------------------------------------------#
##calculate sd for continuous outcomes before standardizing
sd_list <- mice::complete(data_mi, action = "all", include = FALSE)
sd_variables <- c("week_hoursworked", "hourly_income")

# Initialize an empty list to store the results
sd_results <- list()

# Loop over each dataset and calculate the standard deviations
for (i in seq_along(sd_list)) {
  df <- sd_list[[i]]

  # Calculate standard deviations for each variable
  sd_values <- sapply(sd_variables, function(var) {
    sd(df[[var]], na.rm = TRUE)
  })

  # Store the results in the list with dataset index
  sd_results[[i]] <- sd_values
}

# Convert the list to a data frame for better readability
sd_df <- do.call(rbind, sd_results)
colnames(sd_df) <- sd_variables

# Print the results
print(sd_df)

#for sub-sample 2
````
````

```
# Initialize an empty list to store the results
sd_results <- list()

# Loop over each dataset and calculate the standard deviations
for (i in seq_along(sd_list)) {
  df <- sd_list[[i]]

  # Filter the data frame to exclude rows where vocational_vs_general_secondary is NA
  df_filtered <- df[!is.na(df$vocational_vs_general_secondary), ]

  # Calculate standard deviations for each variable
  sd_values <- sapply(sd_variables, function(var) {
    sd(df_filtered[[var]], na.rm = TRUE)
  })

  # Store the results in the list with dataset index
  sd_results[[i]] <- sd_values
}

# Convert the list to a data frame for better readability
sd_df <- do.call(rbind, sd_results)
colnames(sd_df) <- sd_variables

# Print the results
print(sd_df)

#dep 3
# Initialize an empty list to store the results
sd_results <- list()

# Loop over each dataset and calculate the standard deviations
for (i in seq_along(sd_list)) {
  df <- sd_list[[i]]

  # Filter the data frame to exclude rows where vocational_vs_drop_out is NA
  df_filtered <- df[!is.na(df$vocational_vs_drop_out), ]

  # Calculate standard deviations for each variable
  sd_values <- sapply(sd_variables, function(var) {
    sd(df_filtered[[var]], na.rm = TRUE)
  })

  # Store the results in the list with dataset index
  sd_results[[i]] <- sd_values
}

# Convert the list to a data frame for better readability
sd_df <- do.call(rbind, sd_results)
colnames(sd_df) <- sd_variables

# Print the results
```

print(sd_df)
remove(sd_df, sd_results, sd_list, sd_variables)
```

scaling data
```{r}

#scaling every numeric outcome and covariate
process_data <- function(data) {
  # Identify numeric columns
  numeric_columns <- sapply(data, is.numeric)

  # Exclude specific columns
  numeric_columns["work_experience"] <- FALSE
  numeric_columns[".id"] <- FALSE
  numeric_columns["ps_glm_dep4"] <- FALSE
  numeric_columns["ps_glm_dep4_strata"] <- FALSE

  # Exclude numeric columns with only two or three unique values
  numeric_columns <- numeric_columns & sapply(data, function(x) length(unique(x)) > 3)

  # Scale numeric columns to have mean 0 and standard deviation 1
  data[numeric_columns] <- lapply(data[numeric_columns], function(x) scale(x, center = TRUE, scale
= TRUE))
  data[numeric_columns] <- lapply(data[numeric_columns], function(x) as.numeric(x))
  return(data)
}

data_mi <- mice::complete(data_mi, action="long", include = TRUE)

# Process the data
data_mi <- data_mi %>%
  group_by(.imp) %>%
  do(process_data(.)) %>% ungroup()

data_mi <-as.mids(data_mi)
```

Calculate propensity scores and divide in subsamples
propensity score for general education is calculated using the sample with only primary school graduates thus dep2+dep3 combined
```{r}
temp <- mice::complete(data_mi, include = T, action = "long")

calculate_propensity_general <- function(df) {
  formula <- as.formula(paste("attended_general_secondary", paste(ivars, collapse = " + "), sep = "~"))
  df <- df %>% filter(!(is.na(vocational_vs_general_secondary) & is.na(vocational_vs_drop_out)))
  ps_general_secondary <- glm(formula, family = "binomial", data = df)
  df$ps_general_secondary <- predict(ps_general_secondary, type = "response")
  df$ps_general_secondary_strata <- cut(df$ps_general_secondary,

breaks = quantile(df$ps_general_secondary, probs = seq(0, 1, by = 0.2), na.rm = TRUE),
include.lowest = T, labels = F)
  return(df)
}

temp <- temp %>% mutate(attended_general_secondary = case_when(
    vocational_vs_general_secondary == 0 ~ 1,
    TRUE ~ 0))

# Calculate propensity scores for each group defined by .imp
ps_scores <- temp %>% filter(.imp != 0) %>%
  group_by(.imp) %>%
  calculate_propensity_general() %>%
  ungroup()

#merge back with .imp
temp <- temp %>% filter(.imp == 0) %>% bind_rows(ps_scores)

#create seperate datasets for dep 2 and dep 3 without missing treatment values
data_mi_dep2 <- temp %>% filter(is.na(vocational_vs_general_secondary) == F)
data_mi_dep2 <- as.mids(data_mi_dep2)

data_mi_dep3 <- temp %>% filter(is.na(vocational_vs_drop_out) == F)
data_mi_dep3 <- as.mids(data_mi_dep3)
remove(meth, imp, poly, pred, data_toimpute, temp, ps_scores)
```

## 6.3. Save MI data
```{r}
write.csv(complete(data_mi, action = "long", include = T), "~/1.
UU/Thesis/Coding/proc_data/final_data_dep1_mi.csv", row.names = FALSE)

write.csv(complete(data_mi_dep2, action = "long", include = T), "~/1.
UU/Thesis/Coding/proc_data/final_data_dep2_mi.csv", row.names = FALSE)

write.csv(complete(data_mi_dep3, action = "long", include = T), "~/1.
UU/Thesis/Coding/proc_data/final_data_dep3_mi.csv", row.names = FALSE)
```
```

# IV: Code for Data Analysis

````
---
title: "Multiple Inputed analysis"
author: "Xavier Friesen"
date: "`r Sys.Date()`"
output: html_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(include = FALSE)
```

Manual:
Chapters 1-6 cover the main results in the paper.
Chapter 7 covers the robustness checks, which is mainly the same code repeated with slightly different specifications/propensity scores

For the weighted regressions to work it is essential to run chapter 1 and chapter 3 before chapter 4/5/6, and section 7.1 before the rest of this chapter to get the balanced data sets
Apart from that, the regressions should work when run chapter by chapter

The data preparation file does not have to be run before this analysis. The final imputed datasets are loaded in section 1.1.

#1. Descriptives and Preperation
##1.1. Loading data
```{r}
library(dplyr)
library(mice)
library(miceadds)
library(readr)
data_mi <- read_csv("proc_data/final_data_dep1_mi.csv") %>% as.mids()
data_mi_dep2 <- read_csv("proc_data/final_data_dep2_mi.csv") %>% as.mids()
data_mi_dep3 <- read_csv("proc_data/final_data_dep3_mi.csv") %>% as.mids()
```

##1.2. Custom Helper Functions
```{r}
library(sandwich)
library(lmtest)
#compute robust SE
apply_robust_se <- function(model) {
  robust_vcov <- vcovHC(model, type = "HC1")
  coeftest(model, vcov = robust_vcov)
}
````
````

```
# Function to fill data frames to match the maximum number of rows
fill_to_max_rows <- function(df, max_rows) {
  n_rows <- nrow(df)
  if (n_rows < max_rows) {
    additional_rows <- max_rows - n_rows
    filler <- data.frame(matrix(NA, nrow = additional_rows, ncol = ncol(df)))
    colnames(filler) <- colnames(df)
    df <- rbind(df, filler)
  }
  return(df)
}

#calculate work experience strata
calculate_work_experience <- function(df){ #labels
# Define the breaks for the intervals
breaks <- seq(0, 5, by = 1)
# Add an upper bound for the maximum work experience
breaks <- c(breaks, Inf)

# Define the labels for the intervals
labels <- paste(breaks[-length(breaks)], "-",  breaks[-1], sep = "")
labels[length(labels)] <- paste(breaks[length(breaks) - 1], "+", sep = "")

# Correct the labels to ensure they correctly reflect intervals like "0 - 1", "1 - 2", etc.
for (i in 1:(length(labels) - 1)) {
  labels[i] <- paste0(breaks[i], "-", breaks[i+1])
}
# Apply the cut function to create the categories
df <- df %>%
  mutate(work_experience_grouped = cut(work_experience, breaks = breaks, labels = labels, right =
FALSE))
return(df)
}

# Write csv file
write_csv_mi <- function(data_list, csv_filename) {
  file_name <- paste(csv_filename, ".csv", sep = "")

  # Determine the maximum number of rows in the data frames
  max_rows <- max(sapply(data_list, nrow))

  # Extract the first column (variable names) from the first data frame
  first_column <- data_list[[1]][, 1]
  first_column_df <- data.frame(first_column)
  colnames(first_column_df) <- "Variable Name"
  first_column_df <- fill_to_max_rows(first_column_df, max_rows)
```

```
# Extract and combine the first two columns from each data frame
combined_columns <- lapply(data_list, function(df) {
  # Check if data frame has at least two columns
  if (ncol(df) >= 2) {
    # Round each column to three digits
    if (is.numeric(df[, 2])) {
  col1 <- round(df[, 2], 3)
} else {
  col1 <- df[, 2]
}
    col2 <- round(df[, 3], 3)

    # Initialize combined_col
    combined_col <- vector("character", length(col1))

    # Loop through each element to add significance and handle NA
    for (i in seq_along(col1)) {
      if (ncol(df) >= 6) {
        significance <- ifelse(df[i, 6] < 0.01, "***",
                        ifelse(df[i, 6] < 0.05, "**",
                               ifelse(df[i, 6] < 0.1, "*", "")))
      } else {
        significance <- ""
      }

      if (is.na(col2[i])) {
        combined_col[i] <- as.character(col1[i])
      } else {
        combined_col[i] <- paste0(col1[i], " (", col2[i], ")", significance)
      }
    }

    # Create a data frame for the combined column
    result_df <- data.frame(combined_col)
    # Ensure the data frame has the same number of rows as the maximum
    result_df <- fill_to_max_rows(result_df, max_rows)
    return(result_df)
  } else {
    # Handle cases with fewer than 2 columns
    warning("Data frame has fewer than 2 columns.")
    return(NULL)
  }
})

# Remove any NULL entries from the list
combined_columns <- combined_columns[!sapply(combined_columns, is.null)]
```

```
  # Rename columns to indicate their source
  combined_columns <- lapply(seq_along(combined_columns), function(i) {
    colnames(combined_columns[[i]]) <- paste0(names(data_list)[i], "_combined")
    combined_columns[[i]]
  })

  # Combine them side by side
  combined_df <- cbind(first_column_df, do.call(cbind, combined_columns))

  # Write the combined data frame to a CSV file
  write.csv2(combined_df, file = file_name, row.names = FALSE)

  # Print a message indicating that the file has been saved
  cat("The combined data frame has been saved as '", file_name, "'.\n", sep = "")
}

 # Calculate adjusted R-squared, AIC, and number of observations for each imputed dataset model
get_other_statistics <- function(individual_models, model_summary) {
  adj_r_squared <- sapply(individual_models, function(model) summary(model)$adj.r.squared)
  aic_values <- sapply(individual_models, AIC)
  num_obs <- sapply(individual_models, function(model) nobs(model))
  pooled_adj_r_squared <- mean(adj_r_squared)
  pooled_aic_values <- mean(aic_values)
  num_obs <- mean(num_obs)
  # Pool the values
  # Create a data frame for the pooled statistics
  stats_df <- data.frame(
    term = c("adj_r_squared", "AIC criterion", "Number of Observations"),
    estimate = c(as.numeric(round(pooled_adj_r_squared, 3)), as.numeric(round(pooled_aic_values,
3)), as.numeric(round(num_obs, 0))),
    std.error = NA,
    statistic = NA,
    df = NA,
    p.value = NA,
    stringsAsFactors = FALSE
  )

  model_summary <- bind_rows(stats_df, model_summary)
  return(model_summary)
}

#second function to get adj r2 aic and num obs
compute_model_stats <- function(model) {
  adj_r_squared <- summary(model)$adj.r.squared
  aic_value <- AIC(model)
  num_obs <- length(model$fitted.values)
```

```
  return(list(adj_r_squared = adj_r_squared, aic_value = aic_value, num_obs = num_obs))
}

#saving marginal output in a readable csv file for printing, this requires a slightly different function
then earlier
# Define the function
write_csv_mi_margin <- function(data_list, csv_filename) {
  file_name <- paste(csv_filename, ".csv", sep = "")

  # Determine the maximum number of rows in the data frames
  max_rows <- max(sapply(data_list, nrow))

  # Extract the first column (variable names) from the first data frame
  first_column <- data_list[[1]][, 1]
  first_column_df <- data.frame(first_column)
  colnames(first_column_df) <- "Variable Name"
  first_column_df <- fill_to_max_rows(first_column_df, max_rows)

  # Extract and combine column 2 and 3 from each data frame
  combined_columns <- lapply(data_list, function(df) {
    # Check if data frame has at least two columns
    if (ncol(df) >= 2) {
      # Round each column to three digits
      col1 <- round(df[,2], 3)
      col2 <- round(df[,3], 3)
      col3 <- round(df[,5], 3)

      # Initialize combined_col
      combined_col <- vector("character", length(col1))

      # Loop through each element to add significance and handle NA
      for (i in c(1:4)) {
        if (ncol(df) >= 5) {
          significance <- ifelse(col3$p.value[i] < 0.01, "***",
                        ifelse(col3$p.value[i] < 0.05, "**",
                              ifelse(col3$p.value[i] < 0.1, "*", "")))
        } else {
          significance <- ""
        }
          combined_col[i] <- paste0(col1$estimate[i], " (", col2$std.error[i], ")", significance)
      }

      # Create a data frame for the combined column
      result_df <- data.frame(combined_col)
      # Ensure the data frame has the same number of rows as the maximum
      result_df <- fill_to_max_rows(result_df, max_rows)
      return(result_df)
```

```
  } else {
    # Handle cases with fewer than 2 columns
    warning("Data frame has fewer than 2 columns.")
    return(NULL)
  }
})


# Remove any NULL entries from the list
combined_columns <- combined_columns[!sapply(combined_columns, is.null)]

# Rename columns to indicate their source
combined_columns <- lapply(seq_along(combined_columns), function(i) {
  colnames(combined_columns[[i]]) <- paste0(names(data_list)[i], "_combined")
  combined_columns[[i]]
})

# Combine them side by side
combined_df <- cbind(first_column_df, do.call(cbind, combined_columns))

# Write the combined data frame to a CSV file
write.csv2(combined_df, file = file_name, row.names = FALSE)

# Print a message indicating that the file has been saved
cat("The combined data frame has been saved as '", file_name, "'.\n", sep = "")
}

#and a few more small adjustment for heterogeneous effects
write_csv_mi_margin_htg <- function(data_list, csv_filename) {
  file_name <- paste(csv_filename, ".csv", sep = "")

  # Determine the maximum number of rows in the data frames
  max_rows <- max(sapply(data_list, nrow))

  first_column <- data_list[[1]]$Interaction_Var
  first_column_df <- data.frame(first_column)
  colnames(first_column_df) <- "Variable Name"
  first_column_df <- fill_to_max_rows(first_column_df, max_rows)

  # Extract the 'value' column from the first data frame
  second_column <- data_list[[1]]$value
  second_column_df <- data.frame(second_column)
  colnames(second_column_df) <- "Variable value"
  second_column_df <- fill_to_max_rows(second_column_df, max_rows)


  # Extract and combine column 2 and 3 from each data frame
```

```
combined_columns <- lapply(data_list, function(df) {
  # Check if data frame has at least two columns
  if (ncol(df) >= 2) {
    # Round each column to three digits
    col1 <- round(df[,2], 3)
    col2 <- round(df[,3], 3)
    col3 <- round(df[,5], 3)

    # Initialize combined_col
    combined_col <- vector("character", length(col1))

    # Loop through each element to add significance and handle NA
for (i in 1:length(col1)) {
  # Check if column 5 exists and if it does, assign significance based on p.value
  if (!is.na(col3[i])) {
    significance <- ifelse(col3[i] < 0.01, "***",
                    ifelse(col3[i] < 0.05, "**",
                         ifelse(col3[i] < 0.1, "*", "")))
  } else {
    significance <- ""
  }

  # Create the combined string for each row
  combined_col[i] <- paste0(col1[i], " (", col2[i], ")", significance)
 }

    # Create a data frame for the combined column
    result_df <- data.frame(combined_col)
    # Ensure the data frame has the same number of rows as the maximum
    result_df <- fill_to_max_rows(result_df, max_rows)
    return(result_df)
  } else {
    # Handle cases with fewer than 2 columns
    warning("Data frame has fewer than 2 columns.")
    return(NULL)
  }
 })


 # Remove any NULL entries from the list
 combined_columns <- combined_columns[!sapply(combined_columns, is.null)]

 # Rename columns to indicate their source
 combined_columns <- lapply(seq_along(combined_columns), function(i) {
  colnames(combined_columns[[i]]) <- paste0(names(data_list)[i], "_combined")
  combined_columns[[i]]
 })
```

```
  # Combine them side by side
  combined_df <- cbind(first_column_df, second_column_df, do.call(cbind, combined_columns))

  # Write the combined data frame to a CSV file
  write.csv2(combined_df, file = file_name, row.names = FALSE)

  # Print a message indicating that the file has been saved
  cat("The combined data frame has been saved as '", file_name, "'.\n", sep = "")
}

```

##1.3. Grouping Variables
Full specification
```{r}
yc <- "yc"
region <- "countrycode"
survey_dummies <- c(yc, region)

child_demographics_propensity <- c("chsex", "chethnic", "chrel")
#for propensity score calculations the above can be included, but sample size is insufficient to include high-factor variables in lm
child_demographics_noage <- c("chsex")

child_health <- c("zweight_8", "zheight_8", "zbmi_8", "chillness_8_13",
"long_term_health_problem", "chdisability", "subjective_health_13")
child_time_use <- c("hsleep", "hcare", "hchore", "htask", "hwork", "hschool", "hstudy", "hplay",
"chldwork_during_school", "missed_school")
child_cognitive_skills <- c("math_score_13", "math_score_improvement", "read_score_13",
"ppvt_score_13", "ppvt_score_improvement")
child_non_cognitive_skills <- c("noncog_friend", "noncog_hardtalk", "noncog_incgame",
"noncog_lead", "noncog_helpchld", "noncog_trust", "noncog_selfefficiacy", "noncog_selfesteem")
child_expectations <- c("expected_grade", "dreamjob_sector", "vocational_dreamjob_dummy",
"academic_dreamjob_dummy")

family_demographics <- c("dadage_atbirth", "momage_atbirth", "careage_atbirth", "dadpassed",
"mompassed", "primarycaregiver", "parent_sick")

family_size <- c("hhsize", "male012", "female012", "bornbef", "bornaft", "total_children_household")

family_education <- c("dadcantread", "momcantread", "carecantread",
"mom_edu_attended_formaleducation",    "mom_edu_beyond_primaryeducation",
"mom_edu_attended_postsecondary", "mom_edu_attended_vocational",
"dad_edu_attended_formaleducation", "dad_edu_beyond_primaryeducation",
"dad_edu_attended_postsecondary", "dad_edu_attended_vocational",
```

"care_edu_attended_formaleducation", "care_edu_beyond_primaryeducation",
"care_edu_attended_postsecondary", "care_edu_attended_vocational")

family_valuation_of_education <- c("formal_education_useful",
"education_during_financial_hardship", "quality_primary_school")

family_expectations <- c("expected_age_married", "expected_age_earning",
"expected_age_leaving_school", "realistic_expectations_parents", "parents_dreamjob_sector",
"parents_vocational_dreamjob_dummy", "parents_academic_dreamjob_dummy")

family_economics <- c("household_primary_job", "ownhouse", "hq", "sv", "cd", "debt")

family_shock <- c("shock_crime", "shock_household_job_loss", "shock_natural_disaster",
"shock_house_collapse")

community_type <- c("typesite_w1", "popsize", "timecap")
community_jobs <- c("agriculture_jobs", "factory_jobs", "craft_jobs")
community_education_available <- c("public_secondary_available", "private_secondary_available",
"lower_vocational_available", "public_higher_vocational_available",
"private_higher_vocational_available")

ivars <- c(survey_dummies, child_demographics_noage, child_health, child_time_use,
child_cognitive_skills, child_non_cognitive_skills, child_expectations, family_demographics,
family_size, family_economics, family_education, family_expectations,
family_valuation_of_education, community_type, community_jobs, community_education_available)
```

Grouping without multicollinearity
the below code identifies which variables are multicollinear but this can only be run after the first
weighted-regressions are completed

treatment = "vocational_vs_drop_out"
all_vars = c(treatment, ivars, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")

outcome = "self_emp"
formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))

vif <- lapply(seq_along(weighted_data), function(i) {
    model <- lm(formula = formula, data = data[[i]], weights = weights)
  })

vif(vif[[1]])

```{r}

```
yc <- "yc"
region <- "countrycode"
survey_dummies <- c(yc, region)

child_demographics_propensity <- c("chsex", "chethnic", "chrel")
#for propensity score calculations the above can be included, but sample size is insufficient to include
high-factor variables in lm
child_demographics_noage <- c("chsex")

child_health <- c("zheight_8", "zbmi_8", "chillness_8_13", "long_term_health_problem",
"chdisability", "subjective_health_13")
child_time_use <- c("hsleep", "hcare", "hchore", "htask", "hwork", "hschool", "hstudy", "hplay",
"chldwork_during_school", "missed_school")
child_cognitive_skills <- c("math_score_13", "math_score_improvement", "read_score_13",
"ppvt_score_13", "ppvt_score_improvement")
child_non_cognitive_skills <- c("noncog_friend", "noncog_hardtalk", "noncog_incgame",
"noncog_lead", "noncog_helpchld", "noncog_trust", "noncog_selfefficiacy", "noncog_selfesteem")
child_expectations <- c("expected_grade", "vocational_dreamjob_dummy",
"academic_dreamjob_dummy")

family_demographics <- c("dadage_atbirth", "momage_atbirth", "dadpassed", "mompassed",
"primarycaregiver", "parent_sick")

family_size <- c("hhsize", "bornbef", "bornaft")

family_education <- c("dadcantread", "momcantread", "mom_edu_attended_formaleducation",
"mom_edu_beyond_primaryeducation", "mom_edu_attended_postsecondary",
"mom_edu_attended_vocational", "dad_edu_attended_formaleducation",
"dad_edu_beyond_primaryeducation", "dad_edu_attended_postsecondary",
"dad_edu_attended_vocational")

family_valuation_of_education <- c("formal_education_useful",
"education_during_financial_hardship", "quality_primary_school")

family_expectations <- c("expected_age_married", "expected_age_earning",
"realistic_expectations_parents", "parents_vocational_dreamjob_dummy",
"parents_academic_dreamjob_dummy")

family_economics <- c("ownhouse", "hq", "sv", "cd", "debt")

family_shock <- c("shock_crime", "shock_household_job_loss", "shock_natural_disaster",
"shock_house_collapse")

community_type <- c("typesite_w1", "popsize", "timecap")
community_jobs <- c("agriculture_jobs", "factory_jobs", "craft_jobs")
community_education_available <- c("public_secondary_available", "lower_vocational_available",
"public_higher_vocational_available", "private_higher_vocational_available")
```

```
ivars_no_mc <- c(survey_dummies, child_demographics_noage, child_health, child_time_use,
child_cognitive_skills, child_non_cognitive_skills, child_expectations, family_demographics,
family_size, family_economics, family_education, family_expectations,
family_valuation_of_education, community_type, community_jobs, community_education_available)
```

##1.4. Compute Means

```{r}
temp <- complete(data_mi, action = "long")
outcome_variables <- c("week_hoursworked", "hourly_income", "any_iga", "formal_iga", "self_emp",
"nonfarm_iga", "attended_higher_education")

# Create a list to store mean results
mean_results <- list()

# Loop through each outcome variable and calculate the mean
for (var in outcome_variables) {
  # Calculate the mean for the current variable
  mean_value <- temp %>%
    summarise(mean = mean(.data[[var]], na.rm = TRUE)) %>%
    pull(mean)

  # Store the mean value in the list
  mean_results[[var]] <- mean_value
}

print(mean_results)


temp <- complete(data_mi_dep2, action = "long")
outcome_variables <- c("week_hoursworked", "hourly_income", "any_iga", "formal_iga", "self_emp",
"nonfarm_iga", "attended_higher_education")

# Create a list to store mean results
mean_results_dep2 <- list()

# Loop through each outcome variable and calculate the mean
for (var in outcome_variables) {
  # Calculate the mean for the current variable
  mean_value <- temp %>%
    summarise(mean = mean(.data[[var]], na.rm = TRUE)) %>%
    pull(mean)

  # Store the mean value in the list
  mean_results_dep2[[var]] <- mean_value
```

~~~
}

print(mean_results_dep2)
temp <- complete(data_mi_dep3, action = "long")
outcome_variables <- c("week_hoursworked", "hourly_income", "any_iga", "formal_iga", "self_emp",
"nonfarm_iga", "attended_higher_education")

# Create a list to store mean results
mean_results_dep3 <- list()

# Loop through each outcome variable and calculate the mean
for (var in outcome_variables) {
  # Calculate the mean for the current variable
  mean_value <- temp %>%
    summarise(mean = mean(.data[[var]], na.rm = TRUE)) %>%
    pull(mean)

  # Store the mean value in the list
  mean_results_dep3[[var]] <- mean_value
}

print(mean_results_dep3)

```



#2. Linear Regressions
##2.1. General Function
```{r}
#required to calculate robust
outcome_variables <- c("week_hoursworked", "hourly_income",
                       "any_iga", "formal_iga", "self_emp", "nonfarm_iga", "attended_higher_education")

treatment = "attended_vocational_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")

lm_regression_models_dep1 = list()
for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
~~~

```
  pooled_model <- pool(robust_mira)
  # Get the summary of the pooled model
  model_summary <- summary(pooled_model)

 # Calculate adjusted R-squared, AIC, and number of observations for each imputed dataset model
  adj_r_squared <- sapply(individual_models, function(model) summary(model)$adj.r.squared)
  aic_values <- sapply(individual_models, AIC)
  num_obs <- sapply(individual_models, function(model) nobs(model))

  pooled_adj_r_squared <- mean(adj_r_squared)
  pooled_aic_values <- mean(aic_values)
  num_obs <- mean(num_obs)
  # Pool the values
  # Create a data frame for the pooled statistics
  stats_df <- data.frame(
    term = c("adj_r_squared", "AIC criterion", "Number of Observations"),
    estimate = c(as.numeric(round(pooled_adj_r_squared, 3)), as.numeric(round(pooled_aic_values,
3)), as.numeric(round(num_obs, 3))),
    std.error = NA,
    statistic = NA,
    df = NA,
    p.value = NA,
    stringsAsFactors = FALSE
  )

  model_summary <- bind_rows(stats_df, model_summary)

  lm_regression_models_dep1[[outcome]] <- model_summary
}

#-------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")

lm_regression_models_dep2 = list()
for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
  pooled_model <- pool(robust_mira)
  # Get the summary of the pooled model
```

```
  model_summary <- summary(pooled_model)

   # Calculate adjusted R-squared, AIC, and number of observations for each imputed dataset model
  adj_r_squared <- sapply(individual_models, function(model) summary(model)$adj.r.squared)
  aic_values <- sapply(individual_models, AIC)
  num_obs <- sapply(individual_models, function(model) nobs(model))

  pooled_adj_r_squared <- mean(adj_r_squared)
  pooled_aic_values <- mean(aic_values)
  num_obs <- mean(num_obs)
  # Pool the values
  # Create a data frame for the pooled statistics
  stats_df <- data.frame(
    term = c("adj_r_squared", "AIC criterion", "Number of Observations"),
    estimate = c(as.numeric(round(pooled_adj_r_squared, 3)), as.numeric(round(pooled_aic_values,
3)), as.numeric(round(num_obs, 3))),
    std.error = NA,
    statistic = NA,
    df = NA,
    p.value = NA,
    stringsAsFactors = FALSE
  )

  model_summary <- bind_rows(stats_df, model_summary)
  lm_regression_models_dep2[[outcome]] <- model_summary
}

#--------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")

lm_regression_models_dep3 = list()
for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
  pooled_model <- pool(robust_mira)
  # Get the summary of the pooled model
  model_summary <- summary(pooled_model)
 # Calculate adjusted R-squared, AIC, and number of observations for each imputed dataset model
  adj_r_squared <- sapply(individual_models, function(model) summary(model)$adj.r.squared)
```

  aic_values <- sapply(individual_models, AIC)
  num_obs <- sapply(individual_models, function(model) nobs(model))

  pooled_adj_r_squared <- mean(adj_r_squared)
  pooled_aic_values <- mean(aic_values)
  num_obs <- mean(num_obs)
  # Pool the values
  # Create a data frame for the pooled statistics
  stats_df <- data.frame(
    term = c("adj_r_squared", "AIC criterion", "Number of Observations"),
    estimate = c(as.numeric(round(pooled_adj_r_squared, 3)), as.numeric(round(pooled_aic_values, 3)), as.numeric(round(num_obs, 3))),
    std.error = NA,
    statistic = NA,
    df = NA,
    p.value = NA,
    stringsAsFactors = FALSE
  )

  model_summary <- bind_rows(stats_df, model_summary)
  lm_regression_models_dep3[[outcome]] <- model_summary
}

```

Write CSV file for linear regressions

```{r}


write_csv_mi(lm_regression_models_dep1, csv_filename = "lm_regression_models_dep1")
write_csv_mi(lm_regression_models_dep2, csv_filename = "lm_regression_models_dep2")
write_csv_mi(lm_regression_models_dep3, csv_filename = "lm_regression_models_dep3")

```

##2.2 Horizontal regression per outcome
```{r}


models_list <- list(NULL,
      survey_dummies,
      c(survey_dummies, "countrycode:year"),
      c(survey_dummies, child_demographics_noage, child_health, child_time_use, child_cognitive_skills, child_non_cognitive_skills, child_expectations, "countrycode:year"),

```
    c(survey_dummies, child_demographics_noage, child_health, child_time_use,
child_cognitive_skills, child_non_cognitive_skills, child_expectations, family_demographics,
family_size, family_economics, family_education, family_expectations, "countrycode:year"),
     c(survey_dummies, child_demographics_noage, child_health, child_time_use,
child_cognitive_skills, child_non_cognitive_skills, child_expectations, family_demographics,
family_size, family_economics, family_education, family_expectations,
family_valuation_of_education, community_type, community_jobs, community_education_available,
"countrycode:year")
    )

horizontal_regression <- function(outcome, models_list, name_table, treatment) {
  output <- list()
  for (i in seq_along(models_list)) {
    ivars_no_mc <- models_list[[i]]
    ivars_no_mc <- c(treatment, ivars_no_mc)
    formula <- as.formula(paste(outcome, paste(ivars_no_mc, collapse = " + "), sep = "~"))
    model_estimated <- lm.mids(formula = formula, data = data_mi)
    individual_models <- model_estimated$analyses
    robust_models <- lapply(individual_models, apply_robust_se)
    robust_mira <- as.mira(robust_models)
    pooled_model <- pool(robust_mira)
    model_summary <- summary(pooled_model)
    model_summary <- get_other_statistics(individual_models = individual_models, model_summary =
model_summary)
    output[[paste("Model", i)]] <- model_summary
  }
  write_csv_mi(output, csv_filename = name_table)
}

#------------------------------------------------------------------------#
horizontal_regression(outcome = "week_hoursworked", name_table =
"lm_horizontal_dep1_week_hoursworked", treatment = "attended_vocational_secondary", models_list
= models_list)

horizontal_regression(outcome = "hourly_income", name_table =
"lm_horizontal_dep1_hourly_income", treatment = "attended_vocational_secondary", models_list =
models_list)

horizontal_regression(outcome = "neet", name_table = "lm_horizontal_dep1_neet", treatment =
"attended_vocational_secondary", models_list = models_list)

horizontal_regression(outcome = "any_iga", name_table = "lm_horizontal_dep1_any_iga", treatment
= "attended_vocational_secondary", models_list = models_list)

horizontal_regression(outcome = "formal_iga", name_table = "lm_horizontal_dep1_formal_iga",
treatment = "attended_vocational_secondary", models_list = models_list)
```

~~~~
horizontal_regression(outcome = "self_emp", name_table = "lm_horizontal_dep1_self_emp",
treatment = "attended_vocational_secondary", models_list = models_list)

horizontal_regression(outcome = "nonfarm_iga", name_table = "lm_horizontal_dep1_nonfarm_iga",
treatment = "attended_vocational_secondary", models_list = models_list)

horizontal_regression(outcome = "attended_higher_education", name_table =
"lm_horizontal_dep1_attended_higher_education", treatment = "attended_vocational_secondary",
models_list = models_list)

horizontal_regression(outcome = "week_hoursworked", name_table =
"lm_horizontal_dep2_week_hoursworked", treatment = "vocational_vs_general_secondary",
models_list = models_list)

horizontal_regression(outcome = "hourly_income", name_table =
"lm_horizontal_dep2_hourly_income", treatment = "vocational_vs_general_secondary", models_list =
models_list)

horizontal_regression(outcome = "week_hoursworked", name_table =
"lm_horizontal_dep3_week_hoursworked", treatment = "vocational_vs_drop_out", models_list =
models_list)

horizontal_regression(outcome = "hourly_income", name_table =
"lm_horizontal_dep3_hourly_income", treatment = "vocational_vs_drop_out", models_list =
models_list)

```

##2.3 Other Specifications
country heterogeneity
```{r}
treatment = "attended_vocational_secondary"
interaction = paste0("countrycode:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
lm_country_regression_models_dep1 = list()

for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
  pooled_model <- pool(robust_mira)
~~~~

```
  # Get the summary of the pooled model
  model_summary <- summary(pooled_model)
  lm_country_regression_models_dep1[[outcome]] <- model_summary
}

#--------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
interaction = paste0("countrycode:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
lm_country_regression_models_dep2 = list()

for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
  pooled_model <- pool(robust_mira)
  # Get the summary of the pooled model
  model_summary <- summary(pooled_model)
  lm_country_regression_models_dep2[[outcome]] <- model_summary
}

#--------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("countrycode:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
lm_country_regression_models_dep3 = list()
for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
  pooled_model <- pool(robust_mira)
  # Get the summary of the pooled model
  model_summary <- summary(pooled_model)
  lm_country_regression_models_dep3[[outcome]] <- model_summary
}
```

```

Including work experience
```{r}
interaction = c("work_experience:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")

treatment = "attended_vocational_secondary"
lm_work_regression_models_dep1 = list()
for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
  pooled_model <- pool(robust_mira)
  # Get the summary of the pooled model
  model_summary <- summary(pooled_model)
  lm_work_regression_models_dep1[[outcome]] <- model_summary
}

#--------------------------------------------------------------#
interaction = c("work_experience:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")

treatment = "vocational_vs_general_secondary"
lm_work_regression_models_dep2 = list()
for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
  pooled_model <- pool(robust_mira)
  # Get the summary of the pooled model
  model_summary <- summary(pooled_model)
  lm_work_regression_models_dep2[[outcome]] <- model_summary

```
}

#--------------------------------------------------------------#
interaction = c("work_experience:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")

treatment = "vocational_vs_drop_out"
lm_work_regression_models_dep3 = list()
for (outcome in outcome_variables) {
  formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))
  model_estimated <- lm.mids(formula = formula, data = data_mi)
  # Extract individual models
  individual_models <- model_estimated$analyses
  # Apply robust standard errors to each model
  robust_models <- lapply(individual_models, apply_robust_se)
  # Create a new mira object with the robust models
  robust_mira <- as.mira(robust_models)
  # Pool the results using mice's pool function
  pooled_model <- pool(robust_mira)
  # Get the summary of the pooled model
  model_summary <- summary(pooled_model)
  lm_work_regression_models_dep3[[outcome]] <- model_summary
}

```

#3. Estimating Propensity Scores and Balance
##3.1. GLM
The below does glm
It includes all variables, including the multicollineair ones to achieve maximum prediction. This does not bias the propensity score fitted values (https://www.sciencedirect.com/science/article/pii/S0022522315005085)

```{r}
library(MatchThem)
library(WeightIt)
library(cobalt)
library(ggplot2)

treatment = "attended_vocational_secondary"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))

mi_dep1_glm <- weightthem(datasets = data_mi, formula = formula, method = "glm", estimand = "ATT", link = "probit")
mi_dep1_glm <- trim(mi_dep1_glm, at = 15)
```

```
bal_tab <- bal.tab(mi_dep1_glm, un = T,
stats = c("mean.diffs", "variance.ratios", "ks.statistics"), thresholds = c(m = .1, v = 2), by =
mi_dep1_glm$countrycode)
print(bal_tab)

summary(bal_tab$Balance.Across.Imputations$Mean.Diff.Adj)
temp <- rownames(bal_tab$Balance.Across.Imputations)
print(temp[abs(bal_tab$Balance.Across.Imputations$Mean.KS.Adj > 0.1) == 1])
print(temp[abs(bal_tab$Balance.Across.Imputations$Mean.Diff.Adj > 0.1) == 1])

#removing variable names from plot to improve readiblity
names <- var.names(bal_tab)

df <- data.frame(
  old = names,
  new = seq(1, by = 1, to = 163))
df <- df[-1, ]

love.plot(bal_tab,
          var.names = df,
          drop.distance = T,
          var.order = "unadjusted",
          stats = c("mean.diffs", "ks.statistics"),
          abs = F,
          threshold = c(0.1, 0.1),
          shapes = c("circle filled", "circle"),
          col = c( "grey", "black"),
          alpha = 0.7,
          title = NULL,
          labels = F)

#----------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))

mi_dep2_glm <- weightthem(datasets = data_mi_dep2, formula = formula, method = "glm", estimand
= "ATT", link = "probit")
mi_dep2_glm <- trim(mi_dep2_glm, at = 15)

bal_tab <- bal.tab(mi_dep2_glm, un = T,
stats = c("mean.diffs", "variance.ratios", "ks.statistics"), thresholds = c(m = .1, v = 2), by =
mi_dep1_glm$countrycode)
print(bal_tab)

summary(bal_tab$Balance.Across.Imputations$Mean.Diff.Adj)
temp <- rownames(bal_tab$Balance.Across.Imputations)
print(temp[abs(bal_tab$Balance.Across.Imputations$Mean.KS.Adj > 0.1) == 1])
```

```
print(temp[abs(bal_tab$Balance.Across.Imputations$Mean.Diff.Adj > 0.1) == 1])

love.plot(bal_tab,
        var.names = df,
        drop.distance = T,
        var.order = "unadjusted",
        stats = c("mean.diffs", "ks.statistics"),
        abs = F,
        threshold = c(0.1, 0.1),
        shapes = c("circle filled", "circle"),
        col = c( "darkgrey", "black"),
        alpha = 0.7,
        title = NULL,
        labels = F)

#--------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))

mi_dep3_glm <- weightthem(datasets = data_mi_dep3, formula = formula, method = "glm", estimand
= "ATT", link = "probit")
mi_dep3_glm <- trim(mi_dep3_glm, at = 15)

bal_tab <- bal.tab(mi_dep3_glm, un = T,
stats = c("mean.diffs", "variance.ratios", "ks.statistics"), thresholds = c(m = .1, v = 2), by =
mi_dep1_glm$countrycode)
print(bal_tab)


summary(bal_tab$Balance.Across.Imputations$Mean.Diff.Adj)
temp <- rownames(bal_tab$Balance.Across.Imputations)
print(temp[abs(bal_tab$Balance.Across.Imputations$Mean.KS.Adj > 0.1) == 1])
print(temp[abs(bal_tab$Balance.Across.Imputations$Mean.Diff.Adj > 0.1) == 1])

df <- data.frame(
  old = names,
  new = seq(1, by = 1, to = 163))
df <- df[-1, ]

love.plot(bal_tab,
        var.names = df,
        drop.distance = T,
        var.order = "unadjusted",
        stats = c("mean.diffs", "ks.statistics"),
        abs = F,
        threshold = c(0.1, 0.1),
        shapes = c("circle filled", "circle"),
```

```
        col = c( "darkgrey", "black"),
        alpha = 0.7,
        title = NULL,
        labels = F)



#balance across all three are great
```

##3.2. Assessing Overlap

```{r}
library(ggplot2)
linearize_ps_variables <- function(data) {
   # Apply the linearization transformation: ln(e(x) / (1 - e(x)))
   data <- log(data/(1 - data))
  return(data)
}

treatment = "attended_vocational_secondary"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))

#only matchthem reports ps scores, same algorithm though then the above weightthem
mi_dep1_glm_match <- matchthem(datasets = data_mi, formula = formula, method = "subclass",
distance = "glm", link = "probit")

imp_glm_dep1 <- complete(mi_dep1_glm_match, action = "long")
imp_glm_dep1$distance_lin <- linearize_ps_variables(imp_glm_dep1$distance)

imp_glm_dep1 %>%
  mutate(drop = if_else(attended_vocational_secondary == 1, "Vocational Secondary", "Not Vocational
Secondary")) %>%
  ggplot(aes(x = distance_lin, fill = drop)) +
  geom_density(alpha = .5) +
  labs(x = "Linearized Propensity Scores", y = "Density", fill = "") +
  ggtitle("Dep 1: Area of Common Support") +
  facet_wrap(~ .imp, ncol = 2) +
  theme_bw()

max(imp_glm_dep1$distance_lin[imp_glm_dep1$attended_vocational_secondary == 0])
max(imp_glm_dep1$distance_lin[imp_glm_dep1$attended_vocational_secondary == 1])


#----------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))
```

````
#only matchthem reports ps scores, same algorithm though then the above weightthem
mi_dep2_glm_match <- matchthem(datasets = data_mi_dep2, formula = formula, method =
"subclass", distance = "glm", link = "probit")

imp_glm_dep2 <- complete(mi_dep2_glm_match, action = "long")
imp_glm_dep2$distance_lin <- linearize_ps_variables(imp_glm_dep2$distance)

imp_glm_dep2 %>%
  mutate(drop = if_else(vocational_vs_general_secondary == 1, "Vocational Secondary", "General
Secondary")) %>%
  ggplot(aes(x = distance_lin, fill = drop)) +
  geom_density(alpha = .5) +
  labs(x = "Linearized Propensity Scores", y = "Density", fill = "") +
  ggtitle("Dep 2: Area of Common Support") +
  facet_wrap(~ .imp, ncol = 2) +
  theme_bw()

#---#
treatment = "vocational_vs_drop_out"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))

#only matchthem reports ps scores, same algorithm though then the above weightthem
mi_dep3_glm_match <- matchthem(datasets = data_mi_dep3, formula = formula, method =
"subclass", distance = "glm", link = "probit")

imp_glm_dep3 <- complete(mi_dep3_glm_match, action = "long")

imp_glm_dep3$distance_lin <- linearize_ps_variables(imp_glm_dep3$distance)

imp_glm_dep3 %>%
  mutate(drop = if_else(vocational_vs_drop_out == 1, "Vocational Secondary", "Drop Out")) %>%
  ggplot(aes(x = distance_lin, fill = drop)) +
  geom_density(alpha = .5) +
  labs(x = "Linearized Propensity Scores", y = "Density", fill = "") +
  ggtitle("Dep 3: Area of Common Support") +
  facet_wrap(~ .imp, ncol = 2) +
  theme_bw()

```

#4. Weighted Regressions
##4.1. Function with only covariates
```{r}
library(car)
weighted_regressions <- function(data, all_vars, outcome_variables = c("week_hoursworked",
"hourly_income", "any_iga", "formal_iga", "self_emp", "nonfarm_iga",
"attended_higher_education")) {
````

```
  weighted_regression_models <- list()

  for (outcome in outcome_variables) {
    # Computing formula
    formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))

    # Running regressions and calculating statistics
    regression_results <- lapply(seq_along(data), function(i) {
      model <- lm(formula = formula, data = data[[i]], weights = weights)
      stats <- compute_model_stats(model)
      list(model = model, stats = stats)
    })

    # Extract models and statistics
    models <- lapply(regression_results, `[[`, "model")
    stats <- lapply(regression_results, `[[`, "stats")

    # Calculate mean statistics
    mean_adj_r_squared <- mean(sapply(stats, `[[`, "adj_r_squared"))
    mean_aic_values <- mean(sapply(stats, `[[`, "aic_value"))
    mean_num_obs <- mean(sapply(stats, `[[`, "num_obs"))

    # Apply robust standard errors
    robust_models <- lapply(models, apply_robust_se)
    robust_mira <- as.mira(robust_models)
    pooled_model <- pool(robust_mira)
    model_summary <- summary(pooled_model)

    # Create a data frame for the pooled statistics
    stats_df <- data.frame(
      term = c("adj_r_squared", "AIC criterion", "Number of Observations"),
      estimate = c(mean_adj_r_squared, mean_aic_values, mean_num_obs),
      std.error = NA,
      statistic = NA,
      df = NA,
      p.value = NA,
      stringsAsFactors = FALSE
    )


    # Combine the additional statistics with the model summary
    model_summary <- bind_rows(stats_df, model_summary)

name <- paste("countrycode*", treatment, sep = "")

# Check if name is in all_vars and perform linear hypothesis tests if true to assess p-values of
interactions
```

```
if (name %in% all_vars) {
  # Define the variable names based on treatment
nameT <- paste(treatment, "1", sep = "")
nameIN <- paste(treatment, "1:countrycodeIN", sep = "")
namePE <- paste(treatment, "1:countrycodePE", sep = "")
nameVN <- paste(treatment, "1:countrycodeVN", sep = "")
hIN <- paste(nameT, "+", nameIN, sep = "")
hPE <- paste(nameT, "+", namePE, sep = "")
hVN <- paste(nameT, "+", nameVN, sep = "")
 joint_significance <- lapply(models, function(model) {
   linearHypothesis(model, c(nameIN, namePE, nameVN), white.adjust = "hc1")
 })
 joint_significance <- mean(sapply(joint_significance, function(x) x$`Pr(>F)`[[2]]))
 joint_significance <- paste0("p = ", round(joint_significance, 3))

 IN <- lapply(models, function(model) {
   linearHypothesis(model, hIN, white.adjust = "hc1")
 })
 IN <- mean(sapply(IN, function(x) x$`Pr(>F)`[[2]]))
 IN <- paste0("p = ", round(IN, 3))

 PE <- lapply(models, function(model) {
   linearHypothesis(model, hPE, white.adjust = "hc1")
 })
 PE <- mean(sapply(PE, function(x) x$`Pr(>F)`[[2]]))
 PE <- paste0("p = ", round(PE, 3))

 VN <- lapply(models, function(model) {
   linearHypothesis(model, hVN, white.adjust = "hc1")
 })
 VN <- mean(sapply(VN, function(x) x$`Pr(>F)`[[2]]))
 VN <- paste0("p = ", round(VN, 3))
 # Create a data frame for the pooled statistics
 wald_tests <- data.frame(
    term = c("Joint Significance", "P-Wald India", "P-Wald Peru", "P_wald vietnam"),
    estimate = c(joint_significance, IN, PE, VN),
    std.error = NA,
    statistic = NA,
    df = NA,
    p.value = NA,
    stringsAsFactors = FALSE
  )
 #bind to summary
 model_summary$estimate <- round(model_summary$estimate, 3)
 model_summary$estimate <- as.character(model_summary$estimate)
 model_summary <- bind_rows(wald_tests, model_summary)
}
```

````
    # Store the summary
    weighted_regression_models[[outcome]] <- model_summary
  }

  return(weighted_regression_models)
}

treatment = "attended_vocational_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep1_glm, action = "all")

weighted_regression_models_dep1 = weighted_regressions(data = weighted_data, all_vars = all_vars)
write_csv_mi(data_list = weighted_regression_models_dep1, "weighted_regression_models_dep1")


#--------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "all")
weighted_regression_models_dep2 = list()

weighted_regression_models_dep2 = weighted_regressions(data = weighted_data, all_vars = all_vars)
write_csv_mi(data_list = weighted_regression_models_dep2, "weighted_regression_models_dep2")

#--------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")
weighted_regression_models_dep3 = list()

weighted_regression_models_dep3 = weighted_regressions(data = weighted_data, all_vars = all_vars)
write_csv_mi(data_list = weighted_regression_models_dep3, "weighted_regression_models_dep3")
```


##4.2 Country heterogeneity
```{r}
treatment = "attended_vocational_secondary"
interaction = paste0("countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "work_experience", "countrycode:year")

weighted_data <- complete(mi_dep1_glm, action = "all")

weighted_country_regression_models_dep1 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
````

write_csv_mi(data_list = weighted_country_regression_models_dep1,
"weighted_country_regression_models_dep1")

#---#
treatment = "vocational_vs_general_secondary"
interaction = paste0("countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "work_experience", "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "all")

weighted_country_regression_models_dep2 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_country_regression_models_dep2,
"weighted_country_regression_models_dep2")

#---#
treatment = "vocational_vs_drop_out"
interaction = paste0("countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")

#weighted_country_regression_models_dep3 = weighted_regressions(data = weighted_data, all_vars
= all_vars)
#write_csv_mi(data_list = weighted_country_regression_models_dep3,
"weighted_country_regression_models_dep3")

#this last regression only works with non-robust standarderrors

```

##4.3. Job experience
```{r}
treatment = "attended_vocational_secondary"
interaction = paste0("work_experience_grouped*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")

weighted_data <- complete(mi_dep1_glm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

weighted_work_regression_models_dep1 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_work_regression_models_dep1,
"weighted_work_regression_models_dep1")

#---------------------------------------------------------------#

~~~~
treatment = "vocational_vs_general_secondary"
interaction = paste0("work_experience_grouped*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

weighted_work_regression_models_dep2 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_work_regression_models_dep2,
"weighted_work_regression_models_dep2")

#---------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("work_experience_grouped*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

weighted_work_regression_models_dep3 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_work_regression_models_dep3,
"weighted_work_regression_models_dep3")

```


4.4. Propensity to Study General
```{r}
treatment = "attended_vocational_secondary"
interaction = paste0("ps_general_secondary_strata:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")

## Calculating propensity scores per mi data_set
weighted_data <- complete(mi_dep1_glm, action = "all")

weighted_prop_regression_models_dep1 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_prop_regression_models_dep1,
"weighted_prop_regression_models_dep1")

#---------------------------------------------------------------#
~~~~

~~~~
treatment = "vocational_vs_general_secondary"
interaction = paste0("ps_general_secondary_strata:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "all")

weighted_prop_regression_models_dep2 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_prop_regression_models_dep2,
"weighted_prop_regression_models_dep2")

#--------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("ps_general_secondary_strata:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")


weighted_prop_regression_models_dep3 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_prop_regression_models_dep3,
"weighted_prop_regression_models_dep3")
```

#5. Marginal Effects

##5.1. Country Heterogeneity
```{r}
library(marginaleffects)
library(modelsummary)
weighted_marginal_effects <- function(data, all_vars, outcome_variables = c("week_hoursworked",
"hourly_income", "any_iga", "formal_iga", "self_emp", "nonfarm_iga", "attended_higher_education"),
treatment, moderator) {
  weighted_regression_models <- list()

  for (outcome in outcome_variables) {
    # Computing formula
    formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))

    # Running regressions and calculating statistics
    regression_results <- lapply(seq_along(data), function(i) {
      model <- lm(formula = formula, data = data[[i]], weights = weights)
      stats <- compute_model_stats(model)
      newdata <- subset(data[[i]], data[[i]][[treatment]] == 1)
      slope <- avg_slopes(model, vcov = "HC1", variable = treatment, by = moderator,
                newdata = newdata)
      list(model = model, stats = stats, slope = slope)
    })
~~~~

```
    # Extract models and statistics
    models <- lapply(regression_results, `[[`, "model")
    results <- lapply(regression_results, `[[`, "slope")
    results <- lapply(results, function(res) as.data.frame(res))

    # Combine all results into one data frame
    results <- bind_rows(results)

    #pooling by avg
    final_df <- results %>%
      group_by(across(all_of(moderator))) %>%
     summarise(across(where(is.numeric), mean, na.rm = TRUE),
               across(where(~ !is.numeric(.)), ~ first(.)))

    # Store the output
    weighted_regression_models[[outcome]] <- final_df
  }

  return(weighted_regression_models)
}


#------------------------------------------------------------------#
weighted_data <- complete(mi_dep2_glm, action = "all")
treatment = "vocational_vs_general_secondary"
interaction = paste0("countrycode*", treatment)
all.vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")

marginal_weighted_country_regression_models_dep2 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "countrycode")

#------------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")

marginal_weighted_country_regression_models_dep3 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "countrycode")

write_csv_mi_margin(marginal_weighted_country_regression_models_dep2, csv_filename =
"marginal_weighted_country_regression_models_dep2")

write_csv_mi_margin(marginal_weighted_country_regression_models_dep3, csv_filename =
"marginal_weighted_country_regression_models_dep3")
```

````
```

creating a forest plot
```{r}
library(forestplot)
library(stringr)
outcome_names = c("Hours worked per week", "Hourly wage", "Any IGA", "Formal IGA", "Self-employment", "Non-farming IGA", "Attended higher education")
data <- list()
data <- lapply(seq_along(marginal_weighted_country_regression_models_dep2), function(i) {
df <- data.frame(
  outcome = outcome_names[[i]],
  group = marginal_weighted_country_regression_models_dep2[[i]]$countrycode,
  mean = marginal_weighted_country_regression_models_dep2[[i]]$estimate,
  lower = marginal_weighted_country_regression_models_dep2[[i]]$conf.low,
  upper = marginal_weighted_country_regression_models_dep2[[i]]$conf.high
)
}) %>% bind_rows(data) %>% mutate(outcome = str_wrap(outcome, width = 15))

n_per_group <- 1
n_groups <- length(unique(data$outcome))
hrzl_lines <- list()

#create horizontal lines in between outcomes
for (i in seq(n_per_group, n_per_group * (n_groups - 1), by = n_per_group)) {
 hrzl_lines[[as.character(i + 1)]] <- gpar(lty = "longdash", col = "black", lwd = 2)
}

data %>% group_by(group) %>% forestplot(labeltext = outcome,  boxsize = .1, line.margin = .1,
legend = c("Ethiopia", "India", "Peru", "Vietnam"), xlab = "Average Marginal Treatment Effect", clip
= c(-.41, 0.25),
                                        xticks = c(-.40, -.25, -0.15, -0.05, .05, 0.15, 0.25), lwd.zero = 5, lwd.ci = 2,
hrzl_lines = hrzl_lines) |>
  fp_set_style(box = c("green", "orange", "red", "yellow") |> lapply(function(x) gpar(fill = x, col =
"#555555")))

outcome_names = c("Hours worked per week", "Hourly wage", "Any IGA", "Formal IGA", "Self-employment", "Non-farming IGA", "Attended higher education")
data <- list()
data <- lapply(seq_along(marginal_weighted_country_regression_models_dep2), function(i) {
df <- data.frame(
  outcome = outcome_names[[i]],
  group = marginal_weighted_country_regression_models_dep3[[i]]$countrycode,
  mean = marginal_weighted_country_regression_models_dep3[[i]]$estimate,
  lower = marginal_weighted_country_regression_models_dep3[[i]]$conf.low,
  upper = marginal_weighted_country_regression_models_dep3[[i]]$conf.high
)
````

````
}) %>% bind_rows(data) %>% mutate(outcome = str_wrap(outcome, width = 15))

n_per_group <- 1
n_groups <- length(unique(data$outcome))
hrzl_lines <- list()

#create horizontal lines in between outcomes
for (i in seq(n_per_group, n_per_group * (n_groups - 1), by = n_per_group)) {
 hrzl_lines[[as.character(i + 1)]] <- gpar(lty = "longdash", col = "black", lwd = 2)
}

data %>% group_by(group) %>% forestplot(labeltext = outcome,  boxsize = .1, line.margin = .1,
legend = c("Ethiopia", "India", "Peru", "Vietnam"), xlab = "Average Marginal Treatment Effect", clip
= c(-.15, 0.45),
                                        xticks = c(-0.15, -0.05, .05, 0.15, 0.25, 0.35, 0.45), lwd.zero = 5, lwd.ci = 2,
hrzl_lines = hrzl_lines) |>
    fp_set_style(box = c("green", "orange", "red", "yellow") |> lapply(function(x) gpar(fill = x, col =
"#555555")))

```

##5.2. Work Experience
```{r}
#---------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
interaction = paste0("work_experience_grouped*", treatment, "+countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

marginal_weighted_work_regression_models_dep2 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "work_experience_grouped")


#---------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("work_experience_grouped*", treatment, "+countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})
````

````
marginal_weighted_work_regression_models_dep3 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "work_experience_grouped")

write_csv_mi_margin(marginal_weighted_work_regression_models_dep2, csv_filename =
"marginal_weighted_work_regression_models_dep2")

write_csv_mi_margin(marginal_weighted_work_regression_models_dep3, csv_filename =
"marginal_weighted_work_regression_models_dep3")

```

Plotting Job Experience
```{r}
# Define the function
create_interaction_plot <- function(model_data, sublist_name, term_pattern, job_experience_levels) {
  coefficients <- model_data$estimate
  lower_ci <- model_data$conf.low
  upper_ci <- model_data$conf.high

  coefficients_x <-  c(0.5, 1.5, 2.5, 3.5, 4.5, 5.5)
  # Prepare the data for plotting
  plot_data <- data.frame(
    job_experience_levels = coefficients_x,
    coefficients = coefficients,
    lower_ci = lower_ci,
    upper_ci = upper_ci
  )

p <- ggplot(plot_data, aes(x = job_experience_levels, y = coefficients)) +
  geom_hline(yintercept = 0, linetype = "solid", color = "darkgrey", linewidth = 0.5) +
  geom_point() +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.2) +
  geom_smooth(method = "loess", se = F, color = "black") +
  scale_x_continuous(breaks = coefficients_x, labels = job_experience_levels) +
  theme_minimal() +
  labs(
    x = "",
    y = ""
  )
  theme(
    axis.text.x = element_text(size = 12),  # Adjust size as needed
    axis.text.y = element_text(size = 12)
  )
  return(p)
}

# Job experience levels as a factor
````

```
job_experience_levels <- as.factor(c("0-1", "1-2", "2-3", "3-4", "4-5", "5+"))

# Pattern to match the interaction terms
term_pattern <- "vocational_vs_general_secondary1:work_experience"

# Use lapply to create a plot for each sublist
output <- lapply(names(marginal_weighted_work_regression_models_dep2), function(sublist_name) {
  model_data <- marginal_weighted_work_regression_models_dep2[[sublist_name]]
  create_interaction_plot(model_data, sublist_name, term_pattern, job_experience_levels)
})

library(patchwork)
plots_to_print <- wrap_plots(output[1:4], ncol = 2)
plots_to_print

# Use lapply to create a plot for each sublist
term_pattern <- "vocational_vs_drop_out1:work_experience"

output_dep3 <- lapply(names(marginal_weighted_work_regression_models_dep3),
function(sublist_name) {
  model_data <- marginal_weighted_work_regression_models_dep3[[sublist_name]]
  create_interaction_plot(model_data, sublist_name, term_pattern, job_experience_levels)
})

plots_to_print2 <- wrap_plots(output_dep3[1:4], ncol = 2)
plots_to_print2
```

##5.3. Propensity to study general education

```{r}
treatment = "vocational_vs_general_secondary"
interaction = paste0("ps_general_secondary_strata:", treatment, "+countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "all")
weighted_data <- lapply(weighted_data, function(df) {
  df$ps_general_secondary_strata <- as.factor(df$ps_general_secondary_strata)
  return(df)
})

#histogram for ps scores accross vocational secondary
hist(weighted_data[[1]]$ps_general_secondary[weighted_data[[1]]$attended_vocational_secondary
== 1], breaks = 50, main = "Propensity to Attend General Secondary accross TVE-students", xlab =
"Propensity Score attending General Education vs. Dropping Out", freq = F)

marginal_weighted_prop_regression_models_dep2 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "ps_general_secondary_strata")
```

```
#----------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("ps_general_secondary_strata:", treatment, "+countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  df$ps_general_secondary_strata <- as.factor(df$ps_general_secondary_strata)
  return(df)
})

marginal_weighted_prop_regression_models_dep3 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "ps_general_secondary_strata")

write_csv_mi_margin(marginal_weighted_prop_regression_models_dep2, csv_filename =
"marginal_weighted_prop_regression_models_dep2")

write_csv_mi_margin(marginal_weighted_prop_regression_models_dep3, csv_filename =
"marginal_weighted_prop_regression_models_dep3")


#------------------------------------------------------------------------------#
#some descriptives

temp <- complete(mi_dep3_glm)
#histogram for ps scores accross vocational secondary
hist(temp$ps_general_secondary[temp$attended_vocational_secondary == 1], breaks = 50, main =
"Vocational Students", xlab = "Propensity Score attending General Education vs. Dropping Out", freq
= T, cex.lab=1.4)

hist(temp$ps_general_secondary[temp$vocational_vs_drop_out == 0], breaks = 50, main = "Drop
Outs", xlab = "Propensity Score attending General Education vs. Dropping Out", freq = T,
cex.lab=1.4)

temp_data <-  complete(mi_dep3_glm, action = "all")
ps <- lapply(temp_data, function(df){
  df <- df %>% filter(attended_vocational_secondary == 1)
  sum <- sum(df$ps_general_secondary)
  return(sum)
})

ps <- bind_rows(ps) %>% rowMeans() ## 212 people would otherwise have attended general
secondary

ps_per_country <- lapply(temp_data, function(df){
  df <- df %>% filter(attended_vocational_secondary == 1) %>% group_by(countrycode) %>%
summarize(sum = sum(ps_general_secondary))
```

````
  return(df)
})

ps_per_country <- bind_rows(ps_per_country) %>% group_by(countrycode) %>% summarize(mean
= mean(sum))
print(ps_per_country$mean)
table(temp$countrycode, temp$attended_vocational_secondary)[,2]

#drop out is manually calculated, with (vocationals per country - ps_per_country)/(vocationals per)

ps_mean_country <- lapply(temp_data, function(df){
  df <- df %>% filter(attended_vocational_secondary == 1) %>% group_by(countrycode) %>%
summarize(mean = mean(ps_general_secondary))
  return(df)
})

ps_mean_country <- bind_rows(ps_mean_country) %>% group_by(countrycode) %>%
summarize(mean = mean(mean))
print(ps_mean_country$mean)

```
````

Plot Marginals Propensity

````
```{r}
create_interaction_plot <- function(model_data, sublist_name, term_pattern) {
  coefficients <- model_data$estimate
  lower_ci <- model_data$conf.low
  upper_ci <- model_data$conf.high

  coefficients_x <-  c(0.1, 0.3, 0.5, 0.7, 0.9)
  # Prepare the data for plotting
  plot_data <- data.frame(
    job_experience_levels = coefficients_x,
    coefficients = coefficients,
    lower_ci = lower_ci,
    upper_ci = upper_ci
  )

labels <- c("0-0.2", "0.2-0.4", "0.4-0.6", "0.6-0.8", "0.8-1.0")
p <- ggplot(plot_data, aes(x = coefficients_x, y = coefficients)) +
  geom_hline(yintercept = 0, linetype = "solid", color = "darkgrey", linewidth = 0.5) +
  geom_point() +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.05) +
  geom_smooth(method = "loess", se = T, color = "black", level = 0.9) +
  scale_x_continuous(breaks = coefficients_x, labels = labels) +
  theme_minimal() +
  labs(
````

```
    x = "",
    y = ""
  )
  theme(
    axis.text.x = element_text(size = 13),  # Adjust size as needed
    axis.text.y = element_text(size = 13)
  )
  return(p)
}

# Pattern to match the interaction terms
term_pattern <- "vocational_vs_general_secondary1:ps"

# Use lapply to create a plot for each sublist
output_prop <- lapply(names(marginal_weighted_prop_regression_models_dep2),
function(sublist_name) {
  model_data <- marginal_weighted_prop_regression_models_dep2[[sublist_name]]
  create_interaction_plot(model_data, sublist_name, term_pattern)
})


library(patchwork)
plots_to_print <- wrap_plots(output_prop[1:4], ncol = 2)
plots_to_print

term_pattern <- "vocational_vs_drop_out1:ps"
# Use lapply to create a plot for each sublist
output_dep3_prop <- lapply(names(marginal_weighted_prop_regression_models_dep3),
function(sublist_name) {
  model_data <- marginal_weighted_prop_regression_models_dep3[[sublist_name]]
  create_interaction_plot(model_data, sublist_name, term_pattern)
})

plots_to_print2 <- wrap_plots(output_dep3_prop[1:4], ncol = 2)
plots_to_print2
sub_group = "countrycode"
```

#6. Heterogeneous Effects
##6.1. Checking Balance Within Sub Groups
function to create balance plots per category

```{r}
library(patchwork)
balance_within_sub_group_plots <- function(sub_group){
bal_tab_dep2 <- bal.tab(mi_dep2_glm, un = T,
stats = c("mean.diffs"), thresholds = c(m = .25), cluster = sub_group)
```

```
bal_tab_dep3 <- bal.tab(mi_dep3_glm, un = T,
stats = c("mean.diffs"), thresholds = c(m = .25), cluster = sub_group)

#removing variable names from plot to improve readiblity
bal_tab <- bal.tab(mi_dep2_glm, un = T,
                 stats = c("mean.diffs"), thresholds = c(m = .1),)

check_names <- var.names(bal_tab)

df <- data.frame(old = var.names(bal_tab_dep2))

if (any(grepl(paste0("^", sub_group), check_names))) {
   pattern <- paste0("^", sub_group)
   df <- df[!grepl(pattern, df$old), , drop = FALSE]
   levels <- length(unique(temp[[sub_group]]))
} else {
   levels <- 0
  }

if (levels == 2) {levels = 1}
df <- data.frame(old = df,
                new = seq(1, by = 1, to = 163-levels))

dep2 <- love.plot(bal_tab_dep2,
         var.names = df,
         covs.list = names(bal_tab_dep2),
         drop.distance = T,
         limits = list(m = c(-.5, .5)),
         var.order = "unadjusted",
         stats = c("mean.diffs"),
         cluster = T,
         abs = F,
         threshold = c(0.1, 0.1),
         shapes = c("circle filled", "circle"),
         col = c( "grey", "black"),
         alpha = 0.7,
         title = NULL,
         labels = F)

dep3 <- love.plot(bal_tab_dep3,
         var.names = df,
         drop.distance = T,
         var.order = "unadjusted",
         stats = c("mean.diffs"),
         cluster = T,
         abs = F,
         threshold = c(0.1, 0.1),
```

```
        limits = list(m = c(-.5, .5)),
        shapes = c("circle filled", "circle"),
        col = c( "grey", "black"),
        alpha = 0.7,
        title = NULL,
        labels = F)

 # Combine the plots using patchwork
plots_to_print <- dep2 + dep3 + plot_layout(ncol = 2)
return(plots_to_print)
}

p1 <- balance_within_sub_group_plots(sub_group = "countrycode")
p2 <- balance_within_sub_group_plots(sub_group = "ps_general_secondary_strata")
p3 <- balance_within_sub_group_plots(sub_group = "chsex")
p4 <- balance_within_sub_group_plots(sub_group = "vocational_dreamjob_dummy")
p5 <- balance_within_sub_group_plots(sub_group = "agriculture_jobs")
p6 <- balance_within_sub_group_plots(sub_group = "factory_jobs")
p7 <- balance_within_sub_group_plots(sub_group = "craft_jobs")
p8 <- balance_within_sub_group_plots(sub_group = "noncog_lead")

balance_within_sub_group_numeric <- function(sub_group){
  # Divide numerical interactions into 4 quantiles before checking balance
  temp <- complete(mi_dep2_glm, action = "long")
  temp_quantiles_dep2 <- temp
  numeric_vars <- sapply(temp_quantiles_dep2, is.numeric)
  temp_quantiles_dep2[, numeric_vars] <- lapply(temp_quantiles_dep2[, numeric_vars], function(x) {
    cut(x, breaks = 4, labels = FALSE)
  })

  temp <- complete(mi_dep3_glm, action = "long")
  temp_quantiles_dep3 <- temp
  numeric_vars <- sapply(temp_quantiles_dep3, is.numeric)
  temp_quantiles_dep3[, numeric_vars] <- lapply(temp_quantiles_dep3[, numeric_vars], function(x) {
    cut(x, breaks = 4, labels = FALSE)
  })

  # Recalculate balance tables with quantiles
  bal_tab_dep2 <- bal.tab(mi_dep2_glm, un = T,
                        stats = c("mean.diffs"), thresholds = c(m = .25), cluster = sub_group, data =
temp_quantiles_dep2)

  bal_tab_dep3 <- bal.tab(mi_dep3_glm, un = T,
                        stats = c("mean.diffs"), thresholds = c(m = .25), cluster = sub_group, data =
temp_quantiles_dep3)
```

```
 # Remove variable names from plot to improve readability
 bal_tab <- bal.tab(mi_dep2_glm, un = T,
                    stats = c("mean.diffs"), thresholds = c(m = .1))
check_names <- var.names(bal_tab)

 df <- data.frame(old = var.names(bal_tab_dep2),
                  new = seq(1, by = 1, length.out = 163))

 # Create love plots
 dep2 <- love.plot(bal_tab_dep2,
                   var.names = df,
                   covs.list = names(bal_tab_dep2),
                   drop.distance = T,
                   limits = list(m = c(-.5, .5)),
                   var.order = "unadjusted",
                   stats = c("mean.diffs"),
                   cluster = T,
                   abs = F,
                   threshold = c(0.1, 0.1),
                   shapes = c("circle filled", "circle"),
                   col = c("grey", "black"),
                   alpha = 0.7,
                   title = NULL,
                   labels = F)

 dep3 <- love.plot(bal_tab_dep3,
                   var.names = df,
                   drop.distance = T,
                   var.order = "unadjusted",
                   stats = c("mean.diffs"),
                   cluster = T,
                   abs = F,
                   threshold = c(0.1, 0.1),
                   limits = list(m = c(-.5, .5)),
                   shapes = c("circle filled", "circle"),
                   col = c("grey", "black"),
                   alpha = 0.7,
                   title = NULL,
                   labels = F)

 # Combine the plots using patchwork
 plots_to_print <- dep2 + dep3 + plot_layout(ncol = 2)
 return(plots_to_print)
}

n1 <- balance_within_sub_group_numeric(sub_group = "work_experience")
n2 <- balance_within_sub_group_numeric(sub_group = "hstudy")
```

~~~
n3 <- balance_within_sub_group_numeric(sub_group = "math_score_13")
n4 <- balance_within_sub_group_numeric(sub_group = "ppvt_score_13")
n5 <- balance_within_sub_group_numeric(sub_group = "noncog_selfefficiacy")
n6 <- balance_within_sub_group_numeric(sub_group = "noncog_selfesteem")
n7 <- balance_within_sub_group_numeric(sub_group = "popsize")

```


```{r}
p1
p2
p3
p4
p5
p6
p7
p8
n1
n2
n3
n4
n5
n6
n7
n8
```


##6.2. Calculating Contrasts
```{r}
library(marginaleffects)
library(modelsummary)

#for categorical predictors below
#for continuous predictors a different technique is needed due to the multiple imputed data not being able to average
weighted_slopes <- function(data, all_vars, outcome_variables = c("week_hoursworked", "hourly_income", "any_iga", "formal_iga"), treatment, interactions_factor) {
  combined_outcomes <- list()

  for (outcome in outcome_variables) {
    combined_interactions <- data.frame()

    for (interaction_var in interactions_factor) {
      # Construct the formula for each interaction
~~~

```
      formula_str <- paste(outcome, "~", treatment, "+", paste(interaction_var, treatment, sep = "*"),
"+", paste(all_vars, collapse = " + "))
      formula <- as.formula(formula_str)

      # Running regressions and calculating statistics in parallel
      i = 1
      regression_results <- lapply(seq_along(data), function(i) {
        model <- lm(formula = formula, data = data[[i]], weights = data[[i]]$weights)
        newdata <- subset(data[[i]], data[[i]][[treatment]] == 1)
        slope <- avg_slopes(model, vcov = "HC1", variable = treatment, by = interaction_var,
                  newdata = newdata)
        slope <- slope %>%
          mutate(Interaction_Var = interaction_var, row_id = seq(from = 1, by = 1, to = nrow(slope)))
        list(model = model, slope = slope)
      })


    # Extract model
    results <- lapply(regression_results, `[[`, "slope")
    results <- lapply(results, function(res) as.data.frame(res))

    # Combine all results into one data frame
    results <- bind_rows(results)

    #pooling by avg
    final_df <- results %>%
      group_by(across(all_of(interaction_var))) %>%
     summarise(across(where(is.numeric), mean, na.rm = TRUE),
              across(where(~ !is.numeric(.)), ~ first(.)))

    combined_interactions <- bind_rows(combined_interactions, final_df)
 }
    combined_outcomes[[outcome]] <- combined_interactions
  }
  return(combined_outcomes)
}


#-----------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "all")

interactions_factor = c("chsex", "vocational_dreamjob_dummy", "agriculture_jobs", "factory_jobs",
"craft_jobs", "noncog_lead")

#calculate and collapse
```

```
marginal_slopes_dep2 = weighted_slopes(data = weighted_data, all_vars = all_vars, treatment =
treatment, interactions_factor = interactions_factor)

columns_with_na <- names(marginal_slopes_dep2[[1]])[sapply(marginal_slopes_dep2[[1]],
function(col) any(is.na(col)))]


marginal_slopes_dep2 <- lapply(marginal_slopes_dep2, function(df){
  df <- df %>% mutate(value = coalesce(!!!syms(columns_with_na)))
  return(df)
})

#--------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")

marginal_slopes_dep3 = weighted_slopes(data = weighted_data, all_vars = all_vars, treatment =
treatment, interactions_factor = interactions_factor)

marginal_slopes_dep3 <- lapply(marginal_slopes_dep3, function(df){
  df <- df %>% mutate(value = coalesce(!!!syms(columns_with_na)))
  return(df)
})

write_csv_mi_margin_htg(marginal_slopes_dep2, csv_filename = "marginal_slopes_dep2")
write_csv_mi_margin_htg(marginal_slopes_dep3, csv_filename = "marginal_slopes_dep3")
```

##6.3 Calculating numeric effects
estimates

```{r}
library(marginaleffects)

weighted_slopes_numeric <- function(data_list, all_vars, outcome_variables =
c("week_hoursworked", "hourly_income", "any_iga", "formal_iga"), treatment, interactions_numeric)
{
  combined_outcomes <- list()

  for (outcome in outcome_variables) {
    combined_interactions <- data.frame()

    avg_slope_result <- list()
    for (interaction_var in interactions_numeric) {
      # Construct the formula for each interaction
      formula_str <- paste(outcome, "~", treatment, "+", paste(interaction_var, treatment, sep = "*"),
"+", paste(all_vars, collapse = " + "))
```

```
    formula <- as.formula(formula_str)

    # Running regressions and calculating statistics in parallel
   regression_results <- lapply(seq_along(data), function(i) {
      model <- lm(formula = formula, data = data[[i]], weights = weights)
      newdata <- subset(data[[i]], data[[i]][[treatment]] == 1)

      data_mean <- newdata
      data_plus_sd <- newdata
      data_min_sd <- newdata

      data_mean[[interaction_var]] <- 0
      data_plus_sd[[interaction_var]] <- 1
      data_min_sd[[interaction_var]] <- -1

      counterfactualdata <- bind_rows(data_min_sd, data_mean, data_plus_sd)

      slope <- avg_comparisons(model,
                      newdata = counterfactualdata,
                      variables = treatment,
                      by = interaction_var,
                      vcov = "HC1")
      slope <- slope %>%
        mutate(Interaction_Var = interaction_var, row_id = seq(from = 1, by = 1, to = nrow(slope)))
      list(model = model, slope = slope)
    })

  # Extract model
  results <- lapply(regression_results, `[[`, "slope")
  results <- lapply(results, function(res) as.data.frame(res))

  # Combine all results into one data frame
  results <- bind_rows(results)

  #pooling by avg
  final_df <- results %>% group_by(row_id) %>%
    summarise(across(where(is.numeric), mean, na.rm = TRUE),
          across(where(~ !is.numeric(.)), ~ first(.))) %>% ungroup()

  combined_interactions <- bind_rows(combined_interactions, final_df)
}
  combined_outcomes[[outcome]] <- combined_interactions
 }
 return(combined_outcomes)
}
```

```
#----------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "all")

interactions_numeric = c("hstudy", "math_score_13", "ppvt_score_13", "noncog_selfefficiacy",
"noncog_selfesteem", "popsize")

marginal_slopes_dep2_numeric = weighted_slopes_numeric(data = weighted_data, all_vars =
all_vars, treatment = treatment, interactions_numeric = interactions_numeric)


#creates a value row, instead of different columns with the values
columns_with_na <-
names(marginal_slopes_dep2_numeric[[1]])[sapply(marginal_slopes_dep2_numeric[[1]],
function(col) any(is.na(col)))]

marginal_slopes_dep2_numeric <- lapply(marginal_slopes_dep2_numeric, function(df){
  df <- df %>% mutate(value = coalesce(!!!syms(columns_with_na)))
  return(df)
})

#change order of variable
marginal_slopes_dep2_numeric <- lapply(marginal_slopes_dep2_numeric, function(df) {
  df = df %>% dplyr::select(-hstudy, hstudy)
 return(df)
})

write_csv_mi_margin_htg(marginal_slopes_dep2_numeric, "marginal_slopes_dep2_numeric")

#----------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "all")

interactions_numeric = c("hstudy", "math_score_13", "ppvt_score_13", "noncog_selfefficiacy",
"noncog_selfesteem", "popsize")

marginal_slopes_dep3_numeric = weighted_slopes_numeric(data = weighted_data, all_vars =
all_vars, treatment = treatment, interactions_numeric = interactions_numeric)

#creates a value row, instead of different columns with the values
marginal_slopes_dep3_numeric <- lapply(marginal_slopes_dep3_numeric, function(df){
  df <- df %>% mutate(value = coalesce(!!!syms(columns_with_na)))
  return(df)
})
```

````
marginal_slopes_dep3_numeric <- lapply(marginal_slopes_dep3_numeric, function(df) {
  df = df %>% dplyr::select(-hstudy, hstudy)
return(df)
})

write_csv_mi_margin_htg(marginal_slopes_dep3_numeric, "marginal_slopes_dep3_numeric")

```



plots
```{r}
library(marginaleffects)
library(modelsummary)

weighted_slopes_numeric_plots <- function(data, all_vars, outcome_variables =
c("week_hoursworked", "hourly_income", "any_iga", "formal_iga"), treatment, interactions_numeric)
{
  plots_all <- list()

  for (outcome in outcome_variables) {
    plots <- list()
     for (interaction_var in interactions_numeric) {
      # Construct the formula for each interaction
      formula_str <- paste(outcome, "~", treatment, "+", paste(interaction_var, treatment, sep = "*"),
"+", paste(all_vars, collapse = " + "))
      formula <- as.formula(formula_str)

      # Running regressions and calculating statistics in parallel
      regression_results <- lm(formula = formula, data = data, weights = weights / 5) # divided by 5
because I'm using a long data frame

      plot <- plot_comparisons(regression_results, variable = treatment, vcov = "HC1", by =
interaction_var, wts = regression_results$weights) + labs(y = "") +
        geom_hline(yintercept = 0, linetype = "solid", color = "red", linewidth = 0.5) +
        theme_minimal() +
        theme(
          axis.text.x = element_text(size = 13),  # Adjust size as needed
          axis.text.y = element_text(size = 13)
        )

      # Append the result to the slopes list
      plots[[interaction_var]] <- plot
      }
    # Append the slopes list to the plots list
    plots_all[[outcome]] <- plots }
````

```
    return(plots_all) }


#----------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_glm, action = "long")

interactions = c("countrycode", "chsex", "hstudy", "math_score_13", "ppvt_score_13",
"noncog_friend", "noncog_selfefficiacy", "noncog_selfesteem", "vocational_dreamjob_dummy",
"typesite_w1", "popsize", "factory_jobs", "craft_jobs")

interactions_numeric = c("hstudy", "math_score_13", "ppvt_score_13", "noncog_friend",
"noncog_selfefficiacy", "noncog_selfesteem", "popsize")

marginal_slopes_dep2_numeric_plots = weighted_slopes_numeric_plots(data = weighted_data,
all_vars = all_vars, treatment = treatment, interactions_numeric = interactions_numeric)


#----------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_glm, action = "long")

marginal_slopes_dep3_numeric_plots = weighted_slopes_numeric_plots(data = weighted_data,
all_vars = all_vars, treatment = treatment, interactions_numeric = interactions_numeric)

```

#7. Robustness checks

##7.1. GBM Specification

I use the exact same methodology and coding as above, but now using GBM calculated propensity weights.

```{r eval=FALSE, include=FALSE}
treatment = "attended_vocational_secondary"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))

mi_dep1_gbm <- weightthem(datasets = data_mi, formula = formula,  method = "gbm",
interaction.depth = c(1:3), start.tree = 1, n.trees = 12000, estimand = "ATT", criterion = "smd.mean",
                    distribution = "bernoulli",
                    shrinkage = 0.05)

#----------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))
```

```
mi_dep2_gbm <- weightthem(datasets = data_mi_dep2, formula = formula,  method = "gbm",
interaction.depth = c(1:3), start.tree = 1, n.trees = 12000, estimand = "ATT", criterion = "smd.mean",
                    distribution = "bernoulli",
                    shrinkage = 0.05)

treatment = "vocational_vs_drop_out"
formula = as.formula(paste(treatment, paste0(ivars, collapse = " + "), sep = "~"))

mi_dep3_gbm <- weightthem(datasets = data_mi_dep3, formula = formula,  method = "gbm",
interaction.depth = c(1:3), start.tree = 1, n.trees = 12000, estimand = "ATT", criterion = "smd.mean",
                    distribution = "bernoulli",
                    shrinkage = 0.05)
```

###7.1.1 Weighted Regressions

```{r}
library(car)
library(sandwich)
library(lmtest)

weighted_regressions <- function(data, all_vars, outcome_variables = c("week_hoursworked",
"hourly_income", "any_iga", "formal_iga", "self_emp", "nonfarm_iga",
"attended_higher_education")) {
  weighted_regression_models <- list()

  for (outcome in outcome_variables) {
    # Computing formula
    formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))

    # Running regressions and calculating statistics
    regression_results <- lapply(seq_along(data), function(i) {
      model <- lm(formula = formula, data = data[[i]], weights = weights)
      stats <- compute_model_stats(model)
      list(model = model, stats = stats)
    })

    # Extract models and statistics
    models <- lapply(regression_results, `[[`, "model")
    stats <- lapply(regression_results, `[[`, "stats")

    # Calculate mean statistics
    mean_adj_r_squared <- mean(sapply(stats, `[[`, "adj_r_squared"))
    mean_aic_values <- mean(sapply(stats, `[[`, "aic_value"))
    mean_num_obs <- mean(sapply(stats, `[[`, "num_obs"))
```

```
  # Apply robust standard errors
  robust_models <- lapply(models, apply_robust_se)
  robust_mira <- as.mira(robust_models)
  pooled_model <- pool(robust_mira)
  model_summary <- summary(pooled_model)

  # Create a data frame for the pooled statistics
  stats_df <- data.frame(
    term = c("adj_r_squared", "AIC criterion", "Number of Observations"),
    estimate = c(mean_adj_r_squared, mean_aic_values, mean_num_obs),
    std.error = NA,
    statistic = NA,
    df = NA,
    p.value = NA,
    stringsAsFactors = FALSE
  )


  # Combine the additional statistics with the model summary
  model_summary <- bind_rows(stats_df, model_summary)

name <- paste("countrycode*", treatment, sep = "")

# Check if name is in all_vars and perform linear hypothesis tests if true to assess p-values of
interactions
if (name %in% all_vars) {
  # Define the variable names based on treatment
nameT <- paste(treatment, "1", sep = "")
nameIN <- paste(treatment, "1:countrycodeIN", sep = "")
namePE <- paste(treatment, "1:countrycodePE", sep = "")
nameVN <- paste(treatment, "1:countrycodeVN", sep = "")
hIN <- paste(nameT, "+", nameIN, sep = "")
hPE <- paste(nameT, "+", namePE, sep = "")
hVN <- paste(nameT, "+", nameVN, sep = "")
 joint_significance <- lapply(models, function(model) {
   linearHypothesis(model, c(nameIN, namePE, nameVN), white.adjust = "hc1")
 })
 joint_significance <- mean(sapply(joint_significance, function(x) x$`Pr(>F)`[[2]]))
 joint_significance <- paste0("p = ", round(joint_significance, 3))

 IN <- lapply(models, function(model) {
   linearHypothesis(model, hIN, white.adjust = "hc1")
 })
 IN <- mean(sapply(IN, function(x) x$`Pr(>F)`[[2]]))
 IN <- paste0("p = ", round(IN, 3))

 PE <- lapply(models, function(model) {
```

```
    linearHypothesis(model, hPE, white.adjust = "hc1")
  })
  PE <- mean(sapply(PE, function(x) x$`Pr(>F)`[[2]]))
  PE <- paste0("p = ", round(PE, 3))

  VN <- lapply(models, function(model) {
    linearHypothesis(model, hVN, white.adjust = "hc1")
  })
  VN <- mean(sapply(VN, function(x) x$`Pr(>F)`[[2]]))
  VN <- paste0("p = ", round(VN, 3))
  # Create a data frame for the pooled statistics
  wald_tests <- data.frame(
      term = c("Joint Significance", "P-Wald India", "P-Wald Peru", "P_wald vietnam"),
      estimate = c(joint_significance, IN, PE, VN),
      std.error = NA,
      statistic = NA,
      df = NA,
      p.value = NA,
      stringsAsFactors = FALSE
    )
  #bind to summary
  model_summary$estimate <- round(model_summary$estimate, 3)
  model_summary$estimate <- as.character(model_summary$estimate)
  model_summary <- bind_rows(wald_tests, model_summary)
}

    # Store the summary
    weighted_regression_models[[outcome]] <- model_summary
  }

  return(weighted_regression_models)
}

treatment = "attended_vocational_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep1_gbm, action = "all")

weighted_regression_models_dep1 = weighted_regressions(data = weighted_data, all_vars = all_vars)
write_csv_mi(data_list = weighted_regression_models_dep1,
"gbm_weighted_regression_models_dep1")


#--------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_gbm, action = "all")
weighted_regression_models_dep2 = list()
```

weighted_regression_models_dep2 = weighted_regressions(data = weighted_data, all_vars = all_vars)
write_csv_mi(data_list = weighted_regression_models_dep2,
"gbm_weighted_regression_models_dep2")

#---#
treatment = "vocational_vs_drop_out"
all_vars = c(treatment, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_gbm, action = "all")
weighted_regression_models_dep3 = list()

weighted_regression_models_dep3 = weighted_regressions(data = weighted_data, all_vars = all_vars)
write_csv_mi(data_list = weighted_regression_models_dep3,
"gbm_weighted_regression_models_dep3")
```

Country heterogeneity
```{r}
treatment = "attended_vocational_secondary"
interaction = paste0("countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "work_experience", "countrycode:year")

weighted_data <- complete(mi_dep1_gbm, action = "all")

weighted_country_regression_models_dep1 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_country_regression_models_dep1,
"gbm_weighted_country_regression_models_dep1")

#---------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
interaction = paste0("countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "work_experience", "countrycode:year")
weighted_data <- complete(mi_dep2_gbm, action = "all")

weighted_country_regression_models_dep2 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_country_regression_models_dep2,
"gbm_weighted_country_regression_models_dep2")

#---------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_gbm, action = "all")

````
#weighted_country_regression_models_dep3 = weighted_regressions(data = weighted_data, all_vars
= all_vars)
#write_csv_mi(data_list = weighted_country_regression_models_dep3,
"gbm_weighted_country_regression_models_dep3")

#this last regression only works with non-robust standarderrors

```

Job experience
```{r}
treatment = "attended_vocational_secondary"
interaction = paste0("work_experience_grouped*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")

calculate_work_experience <- function(df){ #labels
# Define the breaks for the intervals
breaks <- seq(0, 5, by = 1)
# Add an upper bound for the maximum work experience
breaks <- c(breaks, Inf)

# Define the labels for the intervals
labels <- paste(breaks[-length(breaks)], "-",  breaks[-1], sep = "")
labels[length(labels)] <- paste(breaks[length(breaks) - 1], "+", sep = "")

# Correct the labels to ensure they correctly reflect intervals like "0 - 1", "1 - 2", etc.
for (i in 1:(length(labels) - 1)) {
  labels[i] <- paste0(breaks[i], "-", breaks[i+1])
}
# Apply the cut function to create the categories
df <- df %>%
  mutate(work_experience_grouped = cut(work_experience, breaks = breaks, labels = labels, right =
FALSE))
return(df)
}

weighted_data <- complete(mi_dep1_gbm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

weighted_work_regression_models_dep1 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_work_regression_models_dep1,
"gbm_weighted_work_regression_models_dep1")
````

```
#--------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
interaction = paste0("work_experience_grouped*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_gbm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

weighted_work_regression_models_dep2 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_work_regression_models_dep2,
"gbm_weighted_work_regression_models_dep2")

#--------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("work_experience_grouped*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_gbm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

weighted_work_regression_models_dep3 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_work_regression_models_dep3,
"gbm_weighted_work_regression_models_dep3")

```

Propensity to Study General

```{r}
treatment = "attended_vocational_secondary"
interaction = paste0("ps_general_secondary_strata:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")

## Calculating propensity scores per mi data_set
weighted_data <- complete(mi_dep1_gbm, action = "all")

weighted_prop_regression_models_dep1 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_prop_regression_models_dep1,
"gbm_weighted_prop_regression_models_dep1")
```

```
#-----------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
interaction = paste0("ps_general_secondary_strata:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_gbm, action = "all")

weighted_prop_regression_models_dep2 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_prop_regression_models_dep2,
"gbm_weighted_prop_regression_models_dep2")

#-----------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("ps_general_secondary_strata:", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_gbm, action = "all")


weighted_prop_regression_models_dep3 = weighted_regressions(data = weighted_data, all_vars =
all_vars)
write_csv_mi(data_list = weighted_prop_regression_models_dep3,
"gbm_weighted_prop_regression_models_dep3")
```

###7.1.2. Marginal Effects
Country Heterogeneity
```{r}
library(marginaleffects)
library(modelsummary)
weighted_marginal_effects <- function(data, all_vars, outcome_variables = c("week_hoursworked",
"hourly_income", "any_iga", "formal_iga", "self_emp", "nonfarm_iga", "attended_higher_education"),
treatment, moderator) {
  weighted_regression_models <- list()

  for (outcome in outcome_variables) {
    # Computing formula
    formula <- as.formula(paste(outcome, paste(all_vars, collapse = " + "), sep = "~"))

    # Running regressions and calculating statistics
    regression_results <- lapply(seq_along(data), function(i) {
      model <- lm(formula = formula, data = data[[i]], weights = weights)
      stats <- compute_model_stats(model)
      newdata <- subset(data[[i]], data[[i]][[treatment]] == 1)
      slope <- avg_slopes(model, vcov = "HC1", variable = treatment, by = moderator,
                newdata = newdata)
      list(model = model, stats = stats, slope = slope)
    })
```

```
    # Extract models and statistics
    models <- lapply(regression_results, `[[`, "model")
    results <- lapply(regression_results, `[[`, "slope")
    results <- lapply(results, function(res) as.data.frame(res))

    # Combine all results into one data frame
    results <- bind_rows(results)

    #pooling by avg
    final_df <- results %>%
      group_by(across(all_of(moderator))) %>%
     summarise(across(where(is.numeric), mean, na.rm = TRUE),
           across(where(~ !is.numeric(.)), ~ first(.)))

    # Store the output
    weighted_regression_models[[outcome]] <- final_df
  }

  return(weighted_regression_models)
}

#-------------------------------------------------------------#
weighted_data <- complete(mi_dep2_gbm, action = "all")
treatment = "vocational_vs_general_secondary"
interaction = paste0("countrycode*", treatment, "+countrycode*", treatment)
all_var = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_gbm, action = "all")

marginal_weighted_country_regression_models_dep2 = weighted_marginal_effects(data =
weighted_data, all_vars = all_var, treatment = treatment, moderator = "countrycode")

#-------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("countrycode*", treatment, "+countrycode*", treatment)
all_var = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_gbm, action = "all")

marginal_weighted_country_regression_models_dep3 = weighted_marginal_effects(data =
weighted_data, all_vars = all_var, treatment = treatment, moderator = "countrycode")

write_csv_mi_margin(marginal_weighted_country_regression_models_dep2, csv_filename =
"gbm_marginal_weighted_country_regression_models_dep2")

write_csv_mi_margin(marginal_weighted_country_regression_models_dep3, csv_filename =
"gbm_marginal_weighted_country_regression_models_dep3")
```

```

Work Experience

```{r}
#--------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
interaction = paste0("work_experience_grouped*", treatment, "+countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_gbm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

marginal_weighted_work_regression_models_dep2 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "work_experience_grouped")


#--------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("work_experience_grouped*", treatment, "+countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_gbm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  weighted_data <- calculate_work_experience(df)
})

marginal_weighted_work_regression_models_dep3 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "work_experience_grouped")

write_csv_mi_margin(marginal_weighted_work_regression_models_dep2, csv_filename =
"gbm_marginal_weighted_work_regression_models_dep2")

write_csv_mi_margin(marginal_weighted_work_regression_models_dep3, csv_filename =
"gbm_marginal_weighted_work_regression_models_dep3")

```

Plotting Job Experience

```{r}
# Define the function
create_interaction_plot <- function(model_data, sublist_name, term_pattern, job_experience_levels) {
  coefficients <- model_data$estimate
  lower_ci <- model_data$conf.low
  upper_ci <- model_data$conf.high
```
```

```
  coefficients_x <-  c(0.5, 1.5, 2.5, 3.5, 4.5, 5.5)
  # Prepare the data for plotting
  plot_data <- data.frame(
    job_experience_levels = coefficients_x,
    coefficients = coefficients,
    lower_ci = lower_ci,
    upper_ci = upper_ci
  )

p <- ggplot(plot_data, aes(x = job_experience_levels, y = coefficients)) +
  geom_hline(yintercept = 0, linetype = "solid", color = "grey", linewidth = 0.5) +
  geom_point() +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.2) +
  geom_smooth(method = "loess", se = F, color = "black") +
  scale_x_continuous(breaks = coefficients_x, labels = job_experience_levels) +
  theme_minimal() +
  labs(
    x = "",
    y = ""
  )
  theme(
    axis.text.x = element_text(size = 12),  # Adjust size as needed
    axis.text.y = element_text(size = 12)
  )
  return(p)
}

# Job experience levels as a factor
job_experience_levels <- as.factor(c("0-1", "1-2", "2-3", "3-4", "4-5", "5+"))

# Pattern to match the interaction terms
term_pattern <- "vocational_vs_general_secondary1:work_experience"

# Use lapply to create a plot for each sublist
gbm_output <- lapply(names(marginal_weighted_work_regression_models_dep2),
function(sublist_name) {
  model_data <- marginal_weighted_work_regression_models_dep2[[sublist_name]]
  create_interaction_plot(model_data, sublist_name, term_pattern, job_experience_levels)
})

library(patchwork)
plots_to_print <- wrap_plots(gbm_output[1:4], ncol = 2)
plots_to_print

# Use lapply to create a plot for each sublist
term_pattern <- "vocational_vs_drop_out1:work_experience"
```

```
gbm_output_dep3 <- lapply(names(marginal_weighted_work_regression_models_dep3), function(sublist_name) {
  model_data <- marginal_weighted_work_regression_models_dep3[[sublist_name]]
  create_interaction_plot(model_data, sublist_name, term_pattern, job_experience_levels)
})

plots_to_print2 <- wrap_plots(gbm_output_dep3[1:4], ncol = 2)
plots_to_print2
```

Propensity to study general education

```{r}
#-----------------------------------------------------------------#
treatment = "vocational_vs_general_secondary"
interaction = paste0("ps_general_secondary_strata:", treatment, "+countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep2_gbm, action = "all")
weighted_data <- lapply(weighted_data, function(df) {
  df$ps_general_secondary_strata <- as.factor(df$ps_general_secondary_strata)
  return(df)
})

#histogram for ps scores accross vocational secondary
hist(weighted_data[[1]]$ps_general_secondary[weighted_data[[1]]$attended_vocational_secondary == 1], breaks = 50, main = "Propensity to Attend General Secondary accross TVE-students", xlab = "Propensity Score attending General Education vs. Dropping Out", freq = F)

marginal_weighted_prop_regression_models_dep2 = weighted_marginal_effects(data = weighted_data, all_vars = all_vars, treatment = treatment, moderator = "ps_general_secondary_strata")

#-----------------------------------------------------------------#
treatment = "vocational_vs_drop_out"
interaction = paste0("ps_general_secondary_strata:", treatment, "+countrycode*", treatment)
all_vars = c(treatment, interaction, ivars_no_mc, "countrycode:year")
weighted_data <- complete(mi_dep3_gbm, action = "all")

weighted_data <- lapply(weighted_data, function(df) {
  df$ps_general_secondary_strata <- as.factor(df$ps_general_secondary_strata)
  return(df)
})

temp <- complete(mi_dep3_gbm)
#histogram for ps scores accross vocational secondary
hist(temp$ps_general_secondary[temp$attended_vocational_secondary == 1], breaks = 50, main = "Vocational Students", xlab = "Propensity Score attending General Education vs. Dropping Out", freq = T)
```

```
hist(temp$ps_general_secondary[temp$vocational_vs_drop_out == 0], breaks = 50, main = "Drop
Outs", xlab = "Propensity Score attending General Education vs. Dropping Out", freq = T)

marginal_weighted_prop_regression_models_dep3 = weighted_marginal_effects(data =
weighted_data, all_vars = all_vars, treatment = treatment, moderator = "ps_general_secondary_strata")

write_csv_mi_margin(marginal_weighted_prop_regression_models_dep2, csv_filename =
"gbm_marginal_weighted_prop_regression_models_dep2")

write_csv_mi_margin(marginal_weighted_prop_regression_models_dep3, csv_filename =
"gbm_marginal_weighted_prop_regression_models_dep3")
```

Plot Marginals Propensity

```{r}
create_interaction_plot <- function(model_data, sublist_name, term_pattern) {
  coefficients <- model_data$estimate
  lower_ci <- model_data$conf.low
  upper_ci <- model_data$conf.high

  coefficients_x <-  c(0.1, 0.3, 0.5, 0.7, 0.9)
  # Prepare the data for plotting
  plot_data <- data.frame(
    job_experience_levels = coefficients_x,
    coefficients = coefficients,
    lower_ci = lower_ci,
    upper_ci = upper_ci
  )

labels <- c("0-0.2", "0.2-0.4", "0.4-0.6", "0.6-0.8", "0.8-1.0")
p <- ggplot(plot_data, aes(x = coefficients_x, y = coefficients)) +
  geom_hline(yintercept = 0, linetype = "solid", color = "grey", linewidth = 0.5) +
  geom_point() +
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci), width = 0.05) +
  geom_smooth(method = "loess", se = T, color = "black", level = 0.9) +
  scale_x_continuous(breaks = coefficients_x, labels = labels) +
  theme_minimal() +
  labs(
    x = "",
    y = ""
  )
  theme(
    axis.text.x = element_text(size = 13),  # Adjust size as needed
    axis.text.y = element_text(size = 13)
  )
  return(p)
```

```
}

# Pattern to match the interaction terms
term_pattern <- "vocational_vs_general_secondary1:ps"

# Use lapply to create a plot for each sublist
gbm_output_prop <- lapply(names(marginal_weighted_prop_regression_models_dep2),
function(sublist_name) {
  model_data <- marginal_weighted_prop_regression_models_dep2[[sublist_name]]
  create_interaction_plot(model_data, sublist_name, term_pattern)
})


library(patchwork)
plots_to_print <- wrap_plots(gbm_output_prop[1:4], ncol = 2)
plots_to_print

term_pattern <- "vocational_vs_drop_out1:ps"
# Use lapply to create a plot for each sublist
gbm_output_dep3_prop <- lapply(names(marginal_weighted_prop_regression_models_dep3),
function(sublist_name) {
  model_data <- marginal_weighted_prop_regression_models_dep3[[sublist_name]]
  create_interaction_plot(model_data, sublist_name, term_pattern)
})

plots_to_print2 <- wrap_plots(gbm_output_dep3_prop[1:4], ncol = 2)
plots_to_print2
```