Utrecht University

Department of Information and Computing Science

Applied Data Science Master Thesis

Predictive Analysis of Subjective Well-Being in South America using Machine Learning Techniques

First Examiner:
Yolanda Grift

Candidate:
Rodi Doesburg

Second Examiner:
Linda Keijzer

In Cooperation with:
University School of Economics

July 27, 2024

Abstract

In the realm of socio-economic research, understanding the factors influencing life satisfaction across diverse populations is crucial for informing policy and improving societal well-being. This study explores the relationship between income and life satisfaction in South American countries using comprehensive data analysis techniques, integrating both cluster and factor analyses to dissect how various socio-economic variables contribute to subjective well-being. The study also addresses the challenge of identifying the most influential factors on life satisfaction amidst diverse economic and cultural contexts within South America.

The findings from the cluster analysis demonstrate a moderate positive correlation between income and life satisfaction across South American countries. Each country exhibits unique economic conditions, social structures, and cultural factors that influence this relationship, thereby potentially diluting the impact of income on life satisfaction within individual countries. The clustering method used is k-means, which assigns each data point to a specific cluster with hard boundaries, ensuring that each observation belongs to only one cluster.

The factor analysis highlights income, age, and institutional trust as significant predictors of subjective well-being, with additional variables marginally improving predictive accuracy. Models using the top 10 variables showed commendable accuracy, but including all 74 variables resulted in a marginal 0-6% improvement in model accuracy, highlighting the incremental benefit of incorporating additional variables for enhancing predictive power.

Understanding these relationships informs policymakers about the multifaceted nature of life satisfaction, prompting considerations beyond income to enhance societal well-being effectively. Moreover, these findings underscore the universal relevance of diverse socio-economic factors in shaping life satisfaction. Future research could expand methodologies to incorporate additional cultural and health-related variables, offering deeper insights into subjective well-being across different global regions.

Contents

1. Introduction

1.1 Objective and Context

The primary objective of this scientific article is to investigate the determinants of subjective well-being (SWB) in South America, with a focus on understanding the factors that influence overall life satisfaction. Subjective well-being is a broad concept encompassing various aspects of how individuals experience and evaluate their lives. In this article, life satisfaction will be used as a substitute for SWB. Although life satisfaction is only one component of SWB, it provides a relevant and practical measure for our analysis due to its availability in the dataset [1].

To achieve this, data from the Latin American Public Opinion Project (LAPOP) [2] has been utilized. The outcome variable for this study is derived from the LAPOP survey question coded as LS3, which asks respondents about their general life satisfaction: "To begin, in general how satisfied are you with your life?" This question serves as the measure of SWB.

To ensure manageability and reduce the number of countries analyzed, South America was selected as the focus region due to its inclusion of major countries like Brazil and Argentina. The countries included in this analysis are Colombia, Ecuador, Peru, Bolivia, Paraguay, Chile, Uruguay, Argentina, Venezuela, and Brazil. These ten countries are classified as South America in the technical documentation of LAPOP and are presented in this specific order.

1.2 Research Question and Analysis Overview

The research question guiding this study is:

"Which variables from the LAPOP data are most influential on life satisfaction in South American countries?"

This question is central to the investigation as it aims to uncover the key determinants of life satisfaction, providing insights into subjective well-being across different socioeconomic contexts.

To address this question, the analysis is divided into two main sections. The first section involves a cluster analysis to examine the relationship between income and life satisfaction. This method identifies patterns and groups countries with similar characteristics. It provides a nuanced understanding of how income influences life satisfaction while accounting for heterogeneity between countries. Economic studies often emphasize the role of income in predicting life satisfaction as stated in this study [3], and this analysis aims to validate or challenge this assumption in the South American context.

The second section, which involves a factor analysis, broadens the scope to include a comprehensive range of variables from the LAPOP dataset. Given that the dataset includes over 180 questions [2], the objective is to identify which of these variables most significantly influence life satisfaction using various machine learning models. This analysis will touch more upon the data scientific working field.

The analysis focuses on the years 2016 to 2017. This period was chosen to avoid the economic influences of the 2008 financial crisis and the onset of the COVID-19 pandemic in late 2019. Additionally, the years 2016 and 2017 had the most available data, making this period suitable for a comprehensive analysis.

1.3 Variable Selection and Categorization

To guide the selection of variables, four key articles on the determinants of well-being were reviewed [4] [5] [6] [7]. A comprehensive list of determinants from these articles was compiled and categorized into seven major groups that will be used in this analysis. The complete list is provided in Appendix A. Below, the summary of the findings is displayed:

1. **Income**
2. **Personal Characteristics**: age, gender, ethnicity, personality
3. **Socially Developed Characteristics**: education, health, type of work, unemployment
4. **How Time is Spent**: hours worked, commuting time, caring for others, community involvement and volunteering, exercise, religious activities
5. **Attitude and Beliefs towards Self, Others, Life**: attitude towards our circumstances, social/institutional trust, political persuasion, religion
6. **Relationships**: marriage and intimate relationship, having children, seeing family and friends
7. **Wider Economic, Social, and Political Environment**: income equality, unemployment rates, inflation, welfare system and public insurance, degree of democracy, climate and the natural environment, safety and deprivation of the area, urbanization, internet access, housing conditions

The LAPOP data's core questionnaire questions are coded, such as LS3 for the life satisfaction question [2]. Each code corresponds to a column in the raw data and falls under one of the seven main determinants. Not all aspects of these seven determinants were addressed by specific questions within the 180+ questions asked in the LAPOP dataset. After conducting the two analyses, the most influential variables will be identified, facilitating the determination of which sub-categories within the seven determinants are most significant. This will enable clear conclusions about the key factors influencing life satisfaction.

2. Data and Methods

2.1 Data Pre-processing

In preparation for the two analyses, the datasets undergo a five-step pre-processing procedure:

1. *Transformation of the Life Satisfaction Scale:* The current scale for life satisfaction is counterintuitive, with lower categories indicating higher satisfaction. To enhance clarity and interpretability, the scale is inverted so that higher categories reflect higher life satisfaction.
2. *Refinement of Country-Specific Datasets:* Each country's dataset is initially processed individually. To retain only the variables pertinent to the analyses, country-specific columns will be excluded and only the 'core questionnaire' questions applicable to all ten South American countries are retained.
3. *Conversion of the Income Variable:* The income data, originally presented as categorical factors, is first converted to local currencies. Subsequently, these amounts are converted to USD and standardized using Purchasing Power Parity (PPP) for comparability across countries. This conversion process ensures that income data is consistent and comparable across different countries.
4. *Unification of Similar Variables:* Similar variables are identified and consolidated into composite factors to reduce complexity and variance in the analyses. This step helps streamline the dataset and improves the readability of the results in the subsequent analyses.
5. *Compilation and Integration of Datasets:* After completing the above steps for each country, all country-specific datasets are merged into a single comprehensive dataset. A thorough check for missing values is then conducted to ensure sufficient samples for each variable. The dataset is then subsetted to include only the necessary variables for analysis, retaining redundant columns temporarily for verification purposes. This cautious approach ensures data integrity throughout the pre-processing phase.

Further coding details are provided in Appendix B: R Report.

2.2 Examining the Influence of Income on Life Satisfaction

This section investigates the impact of income on life satisfaction using income variables derived from the q10g question in the questionnaire. The q10g question asks respondents to identify the range that best represents their monthly personal income from work, retirement, or pension, excluding the income of other household members.

From the responses to the q10g question, three income factors were calculated:

- Local currency (local_currency)
- Income in USD (income_usd)
- PPP-adjusted income in USD (income_ppp)

The primary objective of this analysis is to determine whether income significantly influences life satisfaction. Instead of a traditional regression analysis, which is unsuitable due to the categorical nature of the outcome variable and the predictor(s), a cluster analysis will be conducted. This analysis aims to identify patterns and group countries based on income levels and life satisfaction scores, providing a nuanced understanding of their relationship in South America.

Additionally, a summary statistics table will be used to examine the correlation between income and life satisfaction within each country. This detailed overview will offer insights into how income impacts life satisfaction at the national level, assessing whether higher income levels correlate with increased life satisfaction or if the impact of income is minimal.

2.2.1 Assumptions for Ordinal Categorical Variables

Outlined below are the four key assumptions necessary for ensuring the validity of the data analysis, particularly regarding the classification of the outcome variable. The outcome variable is a multi-category factor arranged in ordinal order.

Level of Measurement: The variables used in this analysis must be truly ordinal, which means they should exhibit a clear, ordered relationship among categories. For example, categories like "Very Dissatisfied," "Dissatisfied," "Satisfied," and "Very Satisfied" should reflect a meaningful order.

Appropriate Measures: For the summary statistics table, where ordinal categorical variables will be used, an appropriate measure must be utilized. Specifically, Spearman's Rank Correlation will be employed to examine relationships between ordinal variables, as it is suitable for such data types.

Independence: Each observation within a category must be independent of others. This means the response of one participant should not influence the response of another, thereby preventing biased results.

Sample Size: To support meaningful analysis, there must be an adequate number of observations in each category. Small sample sizes in ordinal categories can lead to unreliable and unstable estimates, affecting the validity of the results.

2.2.2 Calculating Mean Life Satisfaction and Income Values

To facilitate the comparison of income and life satisfaction across different countries, the mean values of the three key variables (life satisfaction, income in USD, and PPP in USD) will be calculated for each country. This step ensures that the analysis accounts for variations in average income and life satisfaction levels across the ten countries, resulting in one observation per country per variable.

The "income datasets" for each country, as referred to in the pre-processing section, where only the income variable q10g has been converted to local currency, USD, and PPP, will be used for these calculations. The exact dataset and codes can be found in Appendix A. This approach enhances the reliability of the analysis by minimizing the potential for errors introduced through extensive data manipulation.

The following steps outline the process for calculating the mean values:

Data Preparation: The income datasets will be used to ensure accurate and consistent calculations.

Mean Calculation: For each country's dataset, the mean values of the life satisfaction variable (ls3) and the income variables (income in USD and PPP in USD) will be computed.

Descriptive Table: The calculated mean values for each country will be compiled into a descriptive table, providing a clear comparison of the average life satisfaction and income levels across all ten countries.

By calculating and comparing these mean values, the analysis will gain insights into the general patterns of income and life satisfaction across different countries and correlations within countries.

2.2.3 K-Means Clustering

K-means clustering is employed in this study to identify patterns and groupings based on life satisfaction and income variables across the countries included in the analysis [8]. This clustering approach facilitates the exploration of similarities among countries and assesses the effectiveness of different income variables—specifically USD and PPP—in forming distinct clusters. The income variable that produces the most distinct clustering patterns will be utilized in subsequent analyses.

Clusters are formed based on similarities in life satisfaction and income levels, initially exploring different numbers of clusters, starting with 3 and extending to 5 clusters. Although clustering typically requires larger datasets to ensure meaningful and statistically robust groupings, this analysis proceeds with 10 data points due to the limited number of countries. The optimal number of clusters (K) will be selected for further analysis based on clustering performance metrics. To ensure consistency and reproducibility of the results, seed codes are set at the outset since cluster assignments are initially randomized.

2.2.4 Clustering Process with 3 and 5 Clusters

Initially, K=3 clusters are chosen to identify broad groupings of countries. The same code structure is then employed for K=5, where the algorithm iteratively assigns each country to one of the clusters based on its average life satisfaction and income levels.

For each country, the clustering algorithm considers two main variables: mean life satisfaction (*mean_ls3*) and either mean income in USD (*mean_income_usd*) or mean PPP (*mean_income_ppp*). These variables are crucial in determining which cluster a country belongs to, based on their proximity to the cluster centers calculated during the iterative process.

Once the clustering algorithm completes, each country is categorized into one of the clusters. These clusters represent groups of countries that share similar characteristics in terms of their reported life satisfaction and income levels. In the results, the clusters will be visualized through plots to explore how different income metrics impact life satisfaction across varied economic contexts.

2.2.5 Cluster Patterns

After the clusters are formed, a choice will be made to determine which clustering approach best fits the analysis. The aim is to analyze the pattern between life satisfaction and income variables within the selected clusters, enabling a comparative analysis across countries.

The pattern will be identified by calculating the correlation between life satisfaction and the two income variables.

2.2.6 Average Silhouette Width

Due to the small number of data points (only ten countries), not all assumptions for clustering will be examined. The primary assumption considered will be the average silhouette width, which will be displayed in a silhouette plot.

The average silhouette width assesses the quality of the clusters by measuring how well each data point fits within its cluster. Higher silhouette widths indicate better-defined clusters. The silhouette analysis also helps study the separation distance between the resulting clusters.

The silhouette plot shows how close each point in one cluster is to points in neighboring clusters, providing a visual way to assess parameters like the number of clusters. The silhouette measure ranges from -1 to 1. Silhouette coefficients near +1 indicate that the country is far from neighboring clusters, a value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters, and negative values suggest that samples might have been assigned to the wrong cluster [9].

2.2.7 Summary Statistics Table

Based on the average silhouette width, one of the two income variables (either income in USD or PPP) will be selected for further analysis. This chosen variable will be featured in a summary statistics table, where the correlation between the selected income variable and life satisfaction will be displayed for each individual country.

This approach allows for a detailed examination of the influence of income on life satisfaction within each South American country, highlighting the significance of income in each specific context. The results section will provide further details regarding the construction and insights derived from the summary statistics table.

2.3 Identifying the Key Predictors of Life Satisfaction

This section focuses on identifying the most significant predictors of life satisfaction using factor analysis and machine learning techniques. Through this comprehensive approach, the analysis aims to uncover the key factors influencing life satisfaction and provide insights that can inform policy and decision-making.

2.3.1 Checking for Multicollinearity

One of the critical assumptions underlying machine learning models is the absence of multicollinearity. Multicollinearity occurs when variables are highly correlated with each other, which can bias the model's results. Therefore, it is essential to examine the correlation between variables in this section [10].

- *No Perfect Multicollinearity:* Ensuring there are no aliased variables.
- *Variance Inflation Factors:* Checking for high variance inflation factors (VIF) among variables.

These checks are crucial to maintain the integrity and reliability of the machine learning models used in this analysis.

No Perfect Multicollinearity

Addressing the issue of perfect multicollinearity is essential to ensure the robustness of machine learning models. Perfect multicollinearity occurs when one variable can be perfectly predicted from a combination of other variables, which can distort the model's estimates and reduce its predictive power [10].

In this step, code is executed to detect any variables that exhibit perfect multicollinearity. If any such variables are identified, necessary adjustments will be made to the model. This may involve removing or combining variables to mitigate the effects of multicollinearity and ensure the reliability of the model's predictions.

VIF Check

After addressing perfect multicollinearity, the next step involves assessing the multicollinearity among the remaining variables using the VIF. VIF measures how much the variance of an estimated regression coefficient increases if predictors are correlated [10].

The following thresholds are used to interpret VIF values and identify variables that may exhibit multicollinearity:

- **VIF > 10:** Indicates severe multicollinearity. Variables with VIF values above this threshold show a very high degree of correlation, which can significantly distort model estimates and should be considered for removal or adjustment.

- **VIF > 5:** Suggests high multicollinearity. While not as severe as a VIF above 10, these variables still demonstrate substantial correlation that could impact model reliability and requires careful evaluation.
- **VIF > 2.5:** Points to moderate multicollinearity. Although less critical than the higher thresholds, VIF values above 2.5 indicate moderate correlation that might affect the model's performance and should be reviewed to ensure it does not adversely impact the analysis.

By applying these thresholds, the analysis aims to ensure that the remaining variables in the model are free from problematic levels of multicollinearity.

2.3.2 Multiple Imputation

To address missing data issues, various single imputation methods were considered, such as mean imputation, regression imputation, and stochastic imputation. Initially, listwise deletion was attempted, as this method removes unusable data without altering the existing data, adhering to the preferred approach in economy where creating data through imputation is generally avoided. However, this method resulted in only 4 complete rows out of 16,731 observations, rendering it impractical for use in machine learning models. Consequently, given the extreme reduction in data size with listwise deletion, imputing missing values became necessary to retain a sufficient dataset for analysis.

Multiple imputation emerged as the most effective approach with minimal error. This method involves creating multiple complete datasets, each containing plausible values for missing data points drawn from the observed data's distribution. These multiple datasets are analyzed separately and the results are combined to produce overall estimates and standard errors. Multiple imputation reduces bias, improves efficiency, and better reflects the uncertainty due to missing data, offering a robust solution compared to single imputation methods. By employing multiple imputation, the analysis ensures enhanced integrity, reliability, and accuracy of the results [11].

2.3.3 Training and Test Set

Effective machine learning models rely on the use of separate training and test sets. The training set is used to develop and fine-tune the models, capturing underlying data patterns and minimizing prediction errors.

The test set evaluates model performance on new, unseen data. This provides an unbiased measure of predictive accuracy and assesses the model's real-world applicability. An 80%-20% split between the training and test sets is commonly used to ensure proper model validation and prevent overfitting [12].

By implementing this structured approach, the analysis ensures the development of reliable models capable of making accurate predictions on new data.

2.3.4 Machine Learning Models

Four machine learning models will be utilized in this analysis: multinomial logistic regression, decision tree, random forest, and gradient boosting (XGBoost). Each model has been selected based on its suitability for handling the ordinal-level categorical variable, which is crucial for explaining life satisfaction across multiple categories.

- *Multinomial Logistic Regression:* Unlike binary logistic regression, multinomial logistic regression accommodates multiple categories in the outcome variable. It estimates the probabilities of each category of the dependent variable and provides insights into the direction and size of the effect of each feature on life satisfaction. This makes it well-suited for this analysis, where life satisfaction levels are ordinal and the aim is to understand and explain the underlying relationships and factors influencing life satisfaction [13].
- *Decision Tree:* This model is known for its simplicity and interpretability. It partitions the data based on features into hierarchical structures resembling a tree. It handles both categorical and continuous data and is effective in identifying important variables [13].
- *Random Forest:* A more advanced ensemble learning method, random forest builds upon decision trees by creating a multitude of them and averaging their outcomes. This approach improves accuracy and reduces overfitting compared to a single decision tree [13].
- *Gradient Boosting (XGBoost):* Among the most sophisticated models, XGBoost sequentially builds trees in an additive manner. It corrects errors from previous iterations by giving more weight to misclassified observations, resulting in higher explanatory performance. XGBoost is particularly effective for complex datasets with non-linear relationships [13].

These models are selected in order of increasing technical complexity, enabling a comprehensive evaluation of their explanatory power for life satisfaction. Even with a reduced number of variables, such as the top 10 identified from the correlation analysis, these models can effectively capture the underlying patterns and relationships in the data. Performance evaluation will be based on accuracy metrics derived from confusion matrices.

Each model follows a two-step process to ensure robustness and thorough evaluation:

1. Initial Training: The model is trained using the top 10 variables identified from the correlation matrix. Cross-validation is used to ensure robustness, and the model's accuracy is assessed using a confusion matrix on the test data subset.

2. Comprehensive Training: The model is then applied to the full dataset, which includes all 74 variables. This step determines if incorporating all available variables improves accuracy compared to using only the top 10 variables.

By following this two-step process, the analysis aims to identify whether the use of all variables enhances the explanatory performance of the machine learning models compared to using only the top 10 selected variables.

2.3.5 Evaluation of Model Performance

The performance of the models will be evaluated based on accuracy, with higher accuracy indicating better model performance. The evaluation will proceed as follows:

1. *Top 10 Variables:* Initially, the machine learning models will be executed using the top 10 variables identified through a correlation matrix. This matrix is analyzed multiple times to ensure consistency, and variables consistently appearing in the top 10 are used for model evaluation. The results will be presented in a table showing the accuracy of models using these variables.
2. *All Variables:* Subsequently, the models will be run using all available variables. This step aims to determine if including the complete dataset enhances model accuracy.
3. *Performance Difference:* Finally, the accuracy of models using the top 10 variables will be compared to those using all variables. This comparison will highlight whether including all variables improves or detracts from the model's accuracy, providing insight into the effectiveness of variable selection.

2.3.6 Selection of Top Features

Understanding the key predictors of life satisfaction is crucial for deriving meaningful insights from the data. This will be accomplished using the best-performing model selected based on overall accuracy and performance improvement.

It is insightful to compare the significant features for predicting life satisfaction identified by the best model with those identified by the correlation matrix. While the correlation matrix examines individual relationships between variables and life satisfaction, the machine learning models account for complex interactions among multiple variables. This comparison elucidates the added value of capturing multivariate interactions in feature importance [14].

The feature importance of the best-performing model will be presented in a table for clarity. Additionally, to enhance visibility and comprehension, a plot will visualize the relative importance of each feature in the model.

3. Results

3.1 Examining the Influence of Income on Life Satisfaction

This section presents the findings from the cluster analysis and other statistical evaluations conducted to understand the influence of income on life satisfaction.

3.1.1 Assumptions for Ordinal Categorical Variables

The results of the four key assumptions necessary for ensuring the validity of the data analysis are presented here. First, the output of each assumption will be displayed, followed by a discussion on whether the assumption is met and if any changes are needed.

Level of Measurement:

Variable	Levels of Measurement
ls3	Factor with 4 levels: "1", "2", "3", "4"
income_usd	Factor with 161 levels
income_ppp	Factor with 161 levels

Table 3.1: Levels of Measurement for Variables

The outcome, ls3 and income variables, income_usd, and income_ppp in our dataset exhibit ordinal levels of measurement.

Life Satisfaction (ls3): This variable is an ordinal factor with categories ranging from 1 to 4. Each level indicates a different degree of life satisfaction, with higher numbers representing greater satisfaction. The differences between these levels are not uniform.

Income (income_usd): This variable is an ordinal factor with income levels categorized into specific brackets in USD, indicating a ranking from low to high income. The exact differences between income levels are not consistent.

Income (income_ppp): Similarly, this variable is an ordinal factor representing income levels categorized into specific brackets adjusted for PPP, ranking from lower to higher income levels with non-uniform differences between levels.

These observations confirm that the assumption of ordinal level measurement for these variables is met.

Appropriate Measures:

Spearman's rank correlation measures the association between ranked variables. It is particularly suited for ordinal variables or continuous variables that do not meet the assumptions of parametric tests. The Spearman's rank correlation coefficient ranges from -1 to 1 [15]:

- 1 indicates a perfect positive correlation,
- -1 indicates a perfect negative correlation,
- 0 indicates no correlation.

Measure	**Correlation**
Correlation between ls3 and income_usd	0.106
Correlation between ls3 and income_ppp	0.105
Correlation between income_usd and income_ppp	0.971

Table 3.2: Correlations between Variables (n=10)

Spearman's rank correlation between ls3 and income_usd: 0.106

Interpretation: There is a weak positive correlation between average life satisfaction (ls3) and personal income (income_usd). This suggests that as personal income increases, there is a slight tendency for life satisfaction to also increase. However, the correlation is weak, indicating that income is not a strong predictor of life satisfaction in this dataset. Because life satisfaction is an ordinal-level categorical variable, calculating precise p-values for the correlation is more complex. Therefore, the statistical significance of this relationship cannot be accurately assessed from the correlation values alone.

Spearman's rank correlation between ls3 and income_ppp: 0.105

Interpretation: Similarly, there is a weak positive correlation between life satisfaction (ls3) and income observed for purchasing power parity (income_ppp). This suggests a slight tendency for higher life satisfaction with increased income, but the relationship remains weak. As with the previous correlation, the absence of p-values means the statistical significance of this relationship cannot be determined.

The weak correlations between life satisfaction and the two income variables suggest that factors other than income may play a more significant role in determining life satisfaction. Therefore, it is advisable to retain only one income variable in the analysis. This choice will be made in the results of the average silhouette width.

Independence:

For the other assumptions, the combined dataset is used. However, for the assumption of independence, individual country-specific datasets are compared to each other. This approach is more practical compared to extracting and analyzing each specific country's data from the combined dataset.

Each observation in the dataset (n=16731) is assumed to be independent. This assumption was evaluated by analyzing the distribution of the ls3 variable between different countries using boxplots.

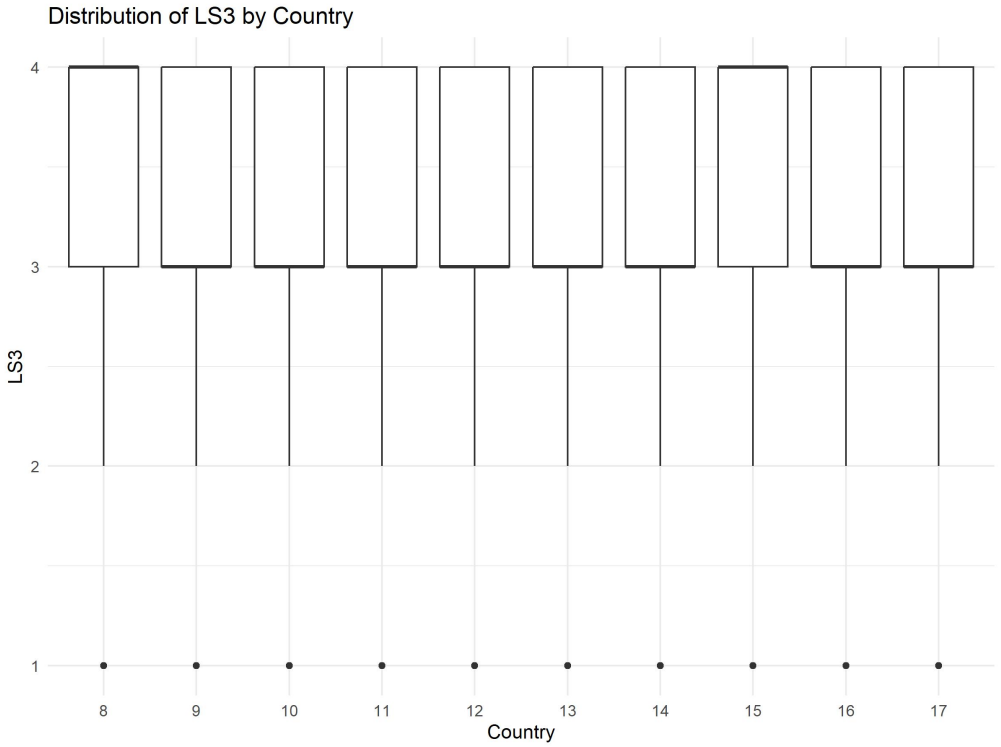


Figure 3.1: Independence Between Countries

The boxplots showed no significant patterns or dependencies between countries. The distributions were similar, with only minor shifts in central tendencies. This supports the assumption of independence.

Sample Size:

To facilitate measurements, the original income variable (q10g) will be utilized. This decision is based on q10g having only 17 categories (ranging from 0 to 16). The converted variables, such as income_usd and income_ppp, result in at least 161 data points (as seen in the level of measurement). While this constitutes a substantial dataset, using the original q10g variable for this assumption does not affect the distribution of sample sizes.

ls3	Sample Size
1	724
2	1674
3	6660
4	7261

Table 3.3: Sample Sizes for ls3 Categories

ls3 (outcome variable): Each category (1 to 4) contains a substantial number of observations, with counts of 724, 1674, 6660, and 7261 respectively. This ensures a robust representation across all levels of life satisfaction.

q10g	Sample Size	q10g	Sample Size
0	191	10	485
1	1362	11	366
2	1149	12	409
3	716	13	355
4	663	14	367
5	570	15	277
6	640	16	306
7	569		
8	539		
9	481		

Table 3.4: Sample Sizes for q10g

Country	Sample Size
8	1563
9	1545
10	1691
11	2647
12	1528
13	1625
14	1514
15	1532
16	1558
17	1528

Table 3.5: Sample Sizes for Country

q10g (income variable): The income variable shows varying counts within its 17 categories. Categories 0, 13, 14, 15, and 16 have fewer observations compared to others, but overall, there are sufficient data points (ranging from 191 to 1362) to conduct reliable statistical analysis for each income level.

Countries: The count of observations for each country ranges from 1528 to 2647, well above the minimum threshold of 1500 as assumed by LAPOP's technical guidelines [16]. This ensures a solid foundation for making valid comparisons between countries.

The sample size assumption is met due to having a sufficient number of data points available across all variables and categories.

3.1.2 K-Means Clustering

The generated plots from the k-means algorithm, as described in the data & methods section, are presented here. The clustering was performed using n=10 countries.

K-Means Clustering (K=3)

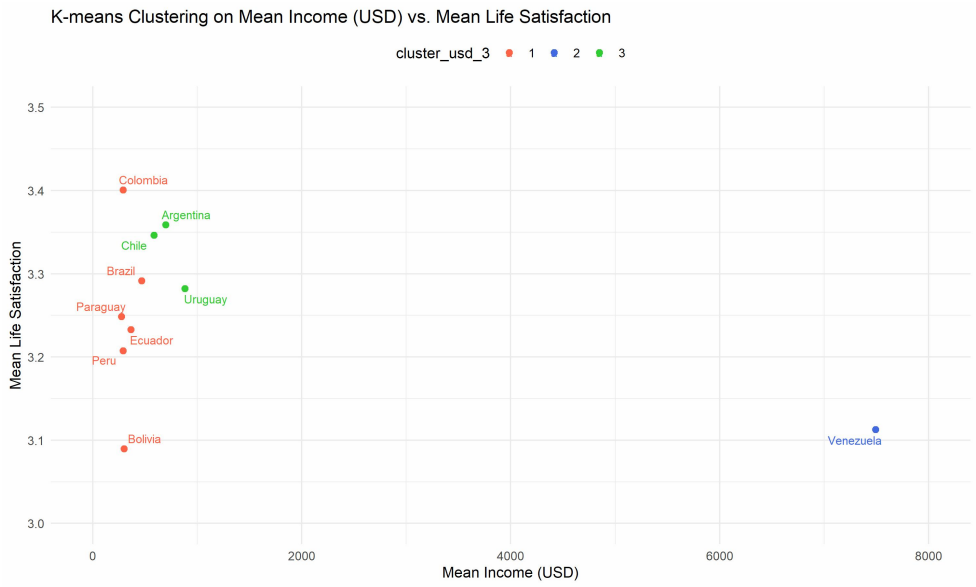


Figure 3.2: K-Means Clustering on Mean Income (USD) vs. Mean Life Satisfaction (k=3)

The clusters for income in USD and life satisfaction are categorized as follows:

- Cluster 1: Bolivia, Peru, Ecuador, Paraguay, Brazil, Colombia
- Cluster 2: Venezuela
- Cluster 3: Uruguay, Chile, Argentina

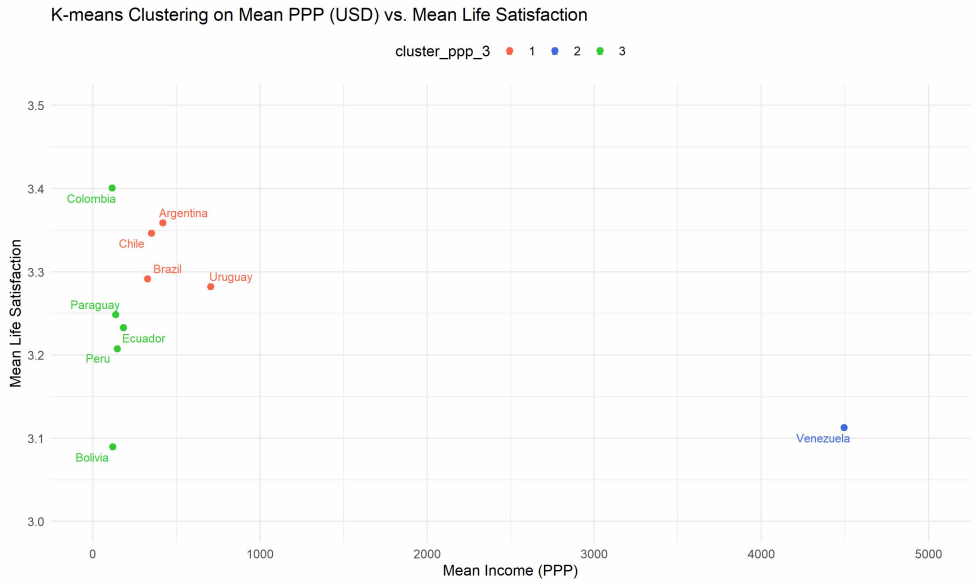


Figure 3.3: K-Means Clustering on Mean PPP (USD) vs. Mean Life Satisfaction (k=3)

The clusters for PPP in USD and life satisfaction are categorized as follows:

- Cluster 1: Uruguay, Brazil, Chile, Argentina
- Cluster 2: Venezuela
- Cluster 3: Bolivia, Peru, Ecuador, Paraguay, Colombia

Two key observations can be made from these results. First, Venezuela is identified as a distinct outlier. Second, the primary difference between the two clustering methods is that Brazil shifts to a different cluster when using PPP-adjusted income instead of the direct conversion from local currency to income in USD.

K-Means Clustering (K=5)

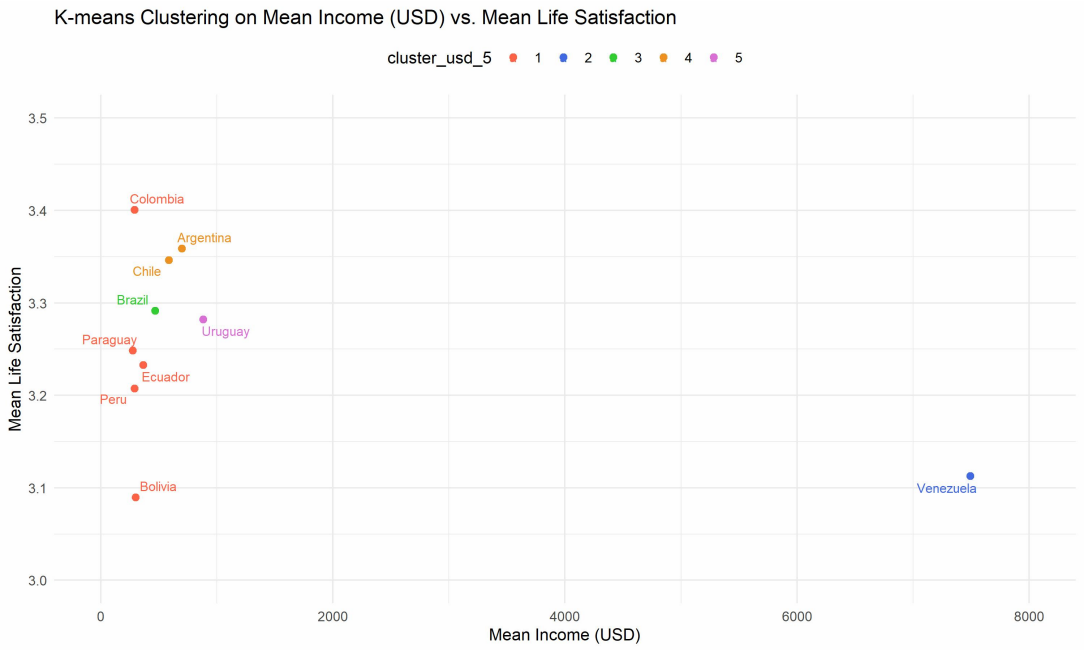


Figure 3.4: K-Means Clustering on Mean Income (USD) vs. Mean Life Satisfaction (k=5)

For the cluster of life satisfaction with income in USD for k=5, the clustering is as follows:

- Cluster 1: Bolivia, Peru, Ecuador, Paraguay, Colombia
- Cluster 2: Venezuela
- Cluster 3: Brazil
- Cluster 4: Chile, Argentina
- Cluster 5: Uruguay

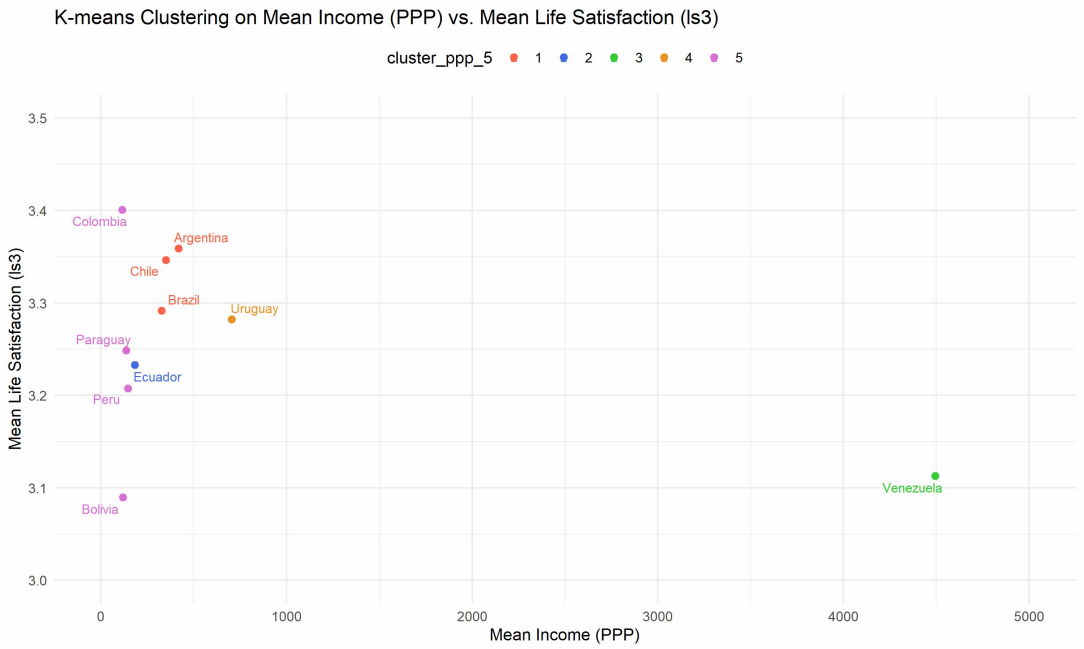


Figure 3.5: K-Means Clustering on Mean PPP (USD) vs. Mean Life Satisfaction (k=5)

For the clusters of life satisfaction with PPP in USD for k=5, the clustering is as follows:

- Cluster 1: Brazil, Chile, Argentina
- Cluster 2: Ecuador
- Cluster 3: Venezuela
- Cluster 4: Uruguay
- Cluster 5: Bolivia, Peru, Paraguay, Colombia

In this clustering with k=5, there are more significant changes compared to the clustering with k=3. Firstly, Venezuela remains a clear outlier. Secondly, only Venezuela and Uruguay stay in the same clusters as before. Brazil moves to the cluster with Chile and Argentina. Ecuador shifts to its own cluster, leaving the largest cluster of Bolivia, Peru, Paraguay, and Colombia.

K-Means Clustering (Excluding Outlier)

To achieve clearer visualization in the clustering analysis, the outlier Venezuela will be excluded. Consequently, the clustering will be simplified using k=3, given that only nine points remain in the dataset.

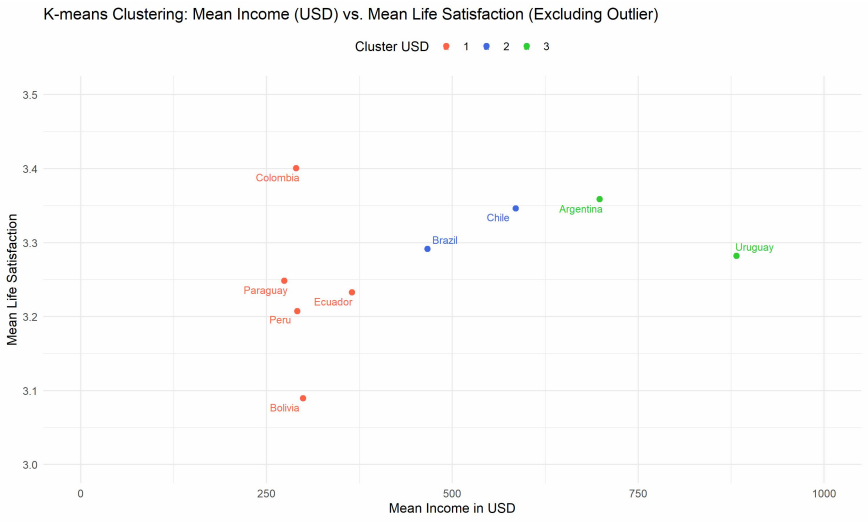


Figure 3.6: K-Means Clustering on Mean Income (USD) vs. Mean Life Satisfaction

For the income in USD cluster:

- Cluster 1: Bolivia, Peru, Ecuador, Paraguay, Colombia
- Cluster 2: Brazil, Chile
- Cluster 3: Uruguay, Argentina

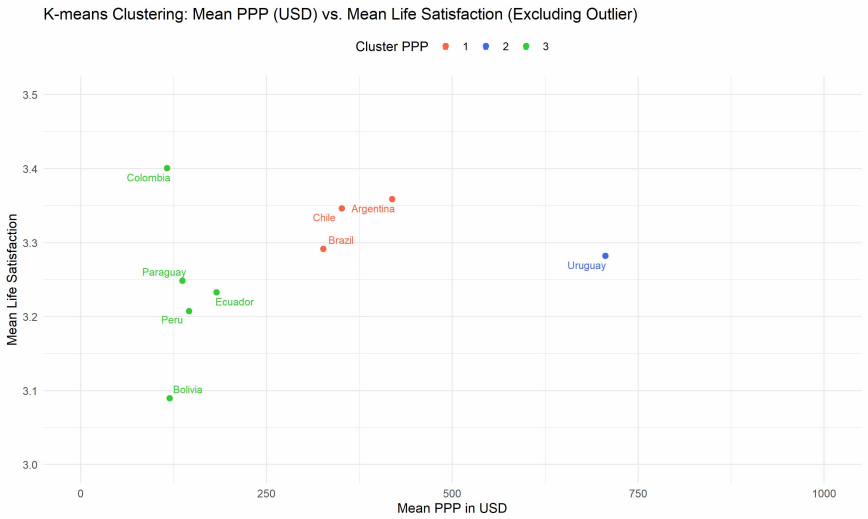


Figure 3.7: K-Means Clustering on Mean Income (USD) vs. Mean Life Satisfaction

For the PPP cluster:

- Cluster 1: Brazil, Chile, Argentina
- Cluster 2: Uruguay
- Cluster 3: Bolivia, Peru, Ecuador, Paraguay, Colombia

Excluding the outlier (Venezuela) reveals a clearer pattern in the clusters. Higher income in USD correlates positively with higher life satisfaction, a pattern that remains consistent even when considering PPP, despite the lower overall income levels.

A notable difference is that Argentina has moved to the Brazil-Chile cluster. Uruguay appears to have the highest income overall. The largest cluster, which has the lowest income, shows considerable variation in life satisfaction, ranging from approximately 3.1 to 3.4 on a scale of 1 to 4. While this numerical difference may seem minor, it is visually significant in the plot.

3.1.3 Patterns of Best Cluster

This section examines the patterns of life satisfaction and income variables among South American countries, with Venezuela excluded as an outlier. By removing the outlier, the aim is to obtain a clearer understanding of how income impacts life satisfaction among these countries, providing a more accurate between-country comparison.

Measure	**Correlation**
Mean Income (USD)	0.374
Mean PPP (USD)	0.317

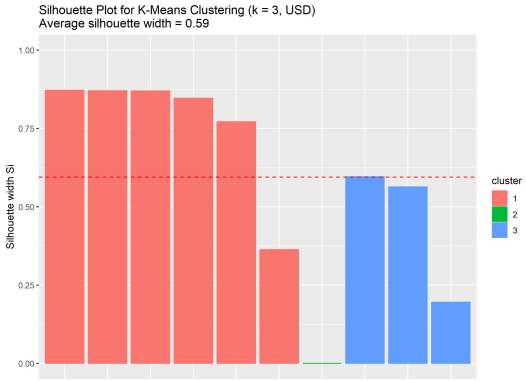
Table 3.6: Patterns of Clusters (Excluding Outlier).

These correlations are calculated based on the 9 countries in the cluster after excluding Venezuela. The coefficient of 0.374 for mean income (USD) suggests a moderate positive pattern, indicating that, on average, higher income is associated with higher life satisfaction. Similarly, a correlation of 0.317 for mean PPP (USD) reflects a positive but slightly weaker relationship. The difference between these correlations is not substantial enough to decisively prioritize one variable over the other in subsequent analyses. Further insights will be drawn from the average silhouette width.

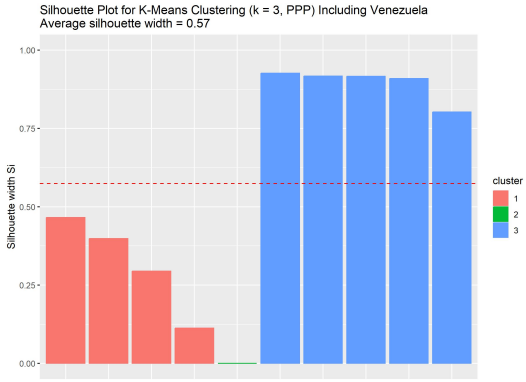
In contrast, the highest individual correlation previously observed in the assumptions section was 0.106, reflecting a weaker positive relationship between life satisfaction and personal income across the entire dataset. This lower correlation was noted before the exclusion of Venezuela. The higher aggregated correlation of 0.374 for the 9 countries collectively indicates a clearer pattern. This suggests that removing Venezuela has improved the detection of a more consistent relationship between income and life satisfaction.

3.1.4 Average Silhouette Width

The average silhouette width adheres to the same sequence as the clustering plots: first with k=3, then k=5, and finally k=3 without the outlier. Each bar in the silhouette plots represents a country randomly assigned to clusters.



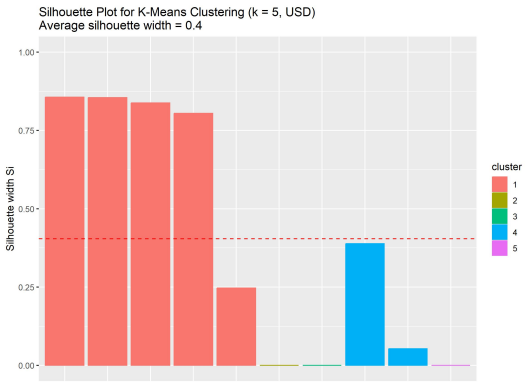
(a) Average Silhouette Width (K=3, USD)



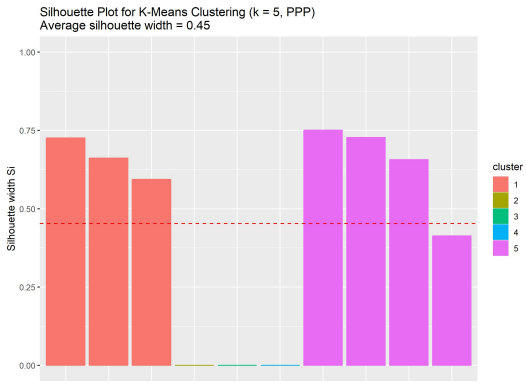
(b) Average Silhouette Width (K=3, PPP)

Figure 3.8: Silhouette Plots for USD and PPP with K=3

The clustering results with k=3, including the outlier, yield Average Silhouette Width (ASW) scores of approximately 0.59 for USD and 0.57 for PPP. While these scores are acceptable, further evaluation of other metrics is necessary to gain a comprehensive understanding.



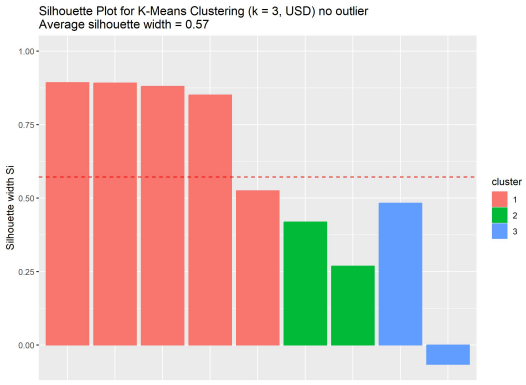
(a) Average Silhouette Width (K=5, USD)



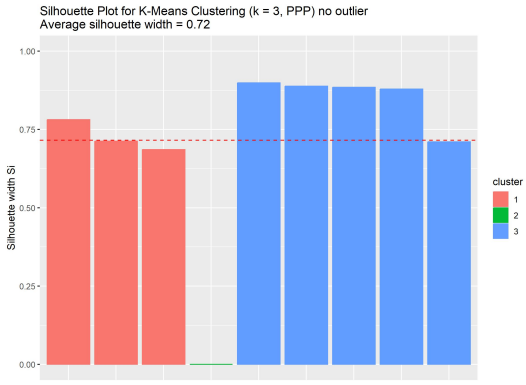
(b) Average Silhouette Width (K=5, PPP)

Figure 3.9: Silhouette Plots for USD and PPP with K=5

The clustering with k=5 produces a lower score than the k=3, around 0.4 for USD and 0.45 for PPP. Primarily due to the presence of single points in most clusters, which hinders proper ASW calculation.



(a) Average Silhouette Width (No Outlier, USD)



(b) Average Silhouette Width (No Outlier, PPP)

Figure 3.10: Silhouette Plots for USD and PPP with No Outlier

The optimal ASW is observed in the plot without the outlier, with scores of 0.57 for USD and 0.72 for PPP.

In conclusion, using fewer clusters is more appropriate for plots with limited data points, and removing outliers improves clustering performance. On average, the PPP variable provides better clustering results compared to income USD, as indicated by higher silhouette scores. Since the average silhouette width (ASW) scores for most silhouette plots are around or above 0.5, the assumptions for clustering are considered to be satisfactorily met.

The k=3 clustering model, excluding the outlier, demonstrates the highest ASW, suggesting the best cluster quality. The PPP-adjusted income is chosen for further analysis because it offers a more consistent and comparable measure of income across different countries by accounting for purchasing power.

3.1.5 Summary Statistics Table

Now that the average pattern between countries has been identified, the next step is to examine the correlation between income and life satisfaction within each South American country individually. This within-country analysis aims to provide a more detailed understanding of how income impacts life satisfaction on a country-specific level.

The results will be presented in a summary statistics table, which will include the correlation coefficients between income (measured in PPP) and life satisfaction for each South American country. This table will offer insights into variations that may not be apparent in the broader regional analysis.

Country	Mean LS	SE	Mean PPP (USD)	SE	Corr.
Argentina	3.36	(0.02)	418.69	(6.98)	0.130
Bolivia	3.09	(0.02)	119.65	(2.08)	0.119
Brazil	3.29	(0.02)	326.41	(6.49)	0.085
Chile	3.35	(0.02)	351.13	(5.76)	0.097
Colombia	3.40	(0.02)	115.93	(2.14)	0.074
Ecuador	3.23	(0.02)	182.43	(3.06)	0.091
Paraguay	3.25	(0.02)	136.82	(2.68)	0.182
Peru	3.21	(0.02)	145.69	(1.91)	0.104
Uruguay	3.28	(0.02)	705.56	(13.62)	0.202
Venezuela	3.11	(0.03)	4494.41	(65.78)	0.094

Table 3.7: Summary Statistics Table using Correlation

The correlations between PPP and life satisfaction within each country range from 0.074 to 0.202, indicating a relatively modest relationship between income and life satisfaction at the country level.

The average correlation between all countries is approximately 0.118, somewhat similar to the correlation observed when considering the data collectively. However, this table looks at each correlation separately, allowing for an understanding of how the relationship between income and life satisfaction varies by country:

- The highest correlation is observed in Uruguay (0.202), suggesting a stronger association between higher income and greater life satisfaction compared to other countries in the region.

- On the lower end, Colombia shows the weakest correlation (0.074), implying a less pronounced link between income and life satisfaction.

It is important to note that the pattern of the between-country correlation, which can exceed 1, cannot be directly compared to individual correlations that have a maximum of 1. Both within-country and between-country analyses suggest that other factors may play a more substantial role in determining individual life satisfaction levels.

3.2 Identifying the Key Predictors of Life Satisfaction

This section presents the findings from the factor analysis using machine learning models. The aim is to determine which variables most significantly impact life satisfaction and to validate whether the cluster analysis correctly identified income (PPP) as an influential variable.

3.2.1 Multicollinearity

The results of the multicollinearity analysis are presented in a table. Detailed information on the adjustments made to address perfect multicollinearity can be found in Appendix B. If significant multicollinearity is detected, adjustments will be made to improve the performance of the machine learning models.

VIF Check

After removing the perfect multicollinearity columns *q12f* and *q12m*, the Variance Inflation Factor (VIF) check was performed.

Threshold	Variables
VIF > 10	
VIF > 5	income_usd, income_ppp
VIF > 2.5	q10g, q10new, q2, vb11

Table 3.8: Variables Exceeding VIF Thresholds

Upon reviewing the VIF results, no variables exhibit VIF values above 10, indicating no severe multicollinearity. Two variables show VIF values above 5, which is expected given their similar nature. To address this, only one of these variables will be retained in the machine learning models to mitigate multicollinearity challenges.

With the removal of the redundant income variables and the *q12f* and *q12m* columns, the multicollinearity assumption is considered met. This ensures that the machine learning models will perform accurately and reliably, allowing for effective evaluation of feature importance.

3.2.2 Multiple Imputation

If the *q12m* (number of boys) and *q12f* (number of girls) columns are still included, this will cause problems during the multiple imputation process. The imputation method sometimes fills in observations randomly, leading to inconsistencies where *q12m* plus *q12f* does not equal *q12* (total number of children). To resolve this, both the perfectly collinear variables (*q12f* and *q12m*) and the redundant income variables (*q10g*, *local_currency*, *income_usd*) were removed.

By eliminating these variables, concerns about the imputation method assigning random values and disrupting variable relationships are alleviated. The remaining variables are neither bound to each other nor excessively correlated, ensuring that the dataset is more robust and suitable for further analysis.

3.2.3 Training and Test Set

The dimensions of both the training and test sets are verified to ensure the split is accurate and confirm the number of variables in each set.

Dataset	Observations	Variables
Training set	13,384	75
Test set	3,347	75

Table 3.9: The Data Training/Test Split

The table confirms that the data split was conducted successfully. The training set contains 13,384 observations, while the test set contains 3,347 observations. The total number of observations across both sets is 16,731, which matches the total number of observations in the dataset. Both sets include 75 variables in total, comprising the outcome variable and 74 predictor variables.

3.2.4 Correlation Matrix

With the improved multiple imputation method and the removal of income variables and the `q12` variables, the top 10 variable selection has become more reliable. These adjustments eliminate the need to consider a broader range of top 15 selections, ensuring that the selection of variables is less influenced by random seeds and multicollinearity assumptions. Consequently, a more stable and dependable top 10 variable list is achieved, providing a solid foundation for further analysis using the correlation matrix.

To identify the top 10 variables, the correlation matrix was run multiple times. Through this iterative process, the variables that appeared most frequently in the top positions were selected. While the top 5 variables consistently showed the highest correlation with the outcome variable, the last 3-5 variables were determined based on their frequent participation in the top 10 across different runs.

Variable	Description	Correlation with ls3
R	Housing Conditions	0.135
q10new	Monthly Income Household	0.118
ed2	Education Level Mother	0.098
B	Institutional Trust	0.094
LIB	Freedom of Expression	0.074
vic1ext	Victim of Crime	0.063
ed	Education Level Individual	0.061
a4	Most Serious Problem in the Country	0.060
q14	Willingness to Work/Live in Another Country	0.059
income_ppp	Monthly Income Individual	0.057

Table 3.10: Correlation of Variables with Outcome Variable (*ls3*)

The table above displays the top 10 variables most strongly correlated with the outcome variable, *ls3*. These variables represent the most influential predictors of life satisfaction identified through the correlation matrix and will serve as key inputs in the subsequent machine learning models.

3.2.5 Machine Learning Models Accuracy

This section assesses the predictive accuracy of four machine learning models: Multinomial Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting (XGBoost). The objective is to determine if incorporating all available variables improves the models' predictive performance compared to utilizing only the top 10 selected variables.

ML Model	**Acc. (Top 10 Variables)**	**Acc. (All Variables)**	**Difference (%)**
Mult. Log. Regression	0.480	0.489	0.90
Decision Tree	0.469	0.475	0.60
Random Forest	0.466	0.531	6.42
XGBoost	0.264	0.264	0.09

Table 3.11: Comparison of Accuracy Machine Learning Models

After executing the machine learning models using the top 10 variables, their respective accuracies are as follows:

- Multinomial Logistic Regression: 0.480
- Decision Tree: 0.469
- Random Forest: 0.466
- XGBoost: 0.264

The multinomial logistic regression model demonstrated the highest accuracy among the top 10 predictors, outperforming both the decision tree and random forest models. In contrast, the XGBoost model exhibited the lowest accuracy.

The models were then applied to the full dataset, using all variables, and their accuracies were measured:

- Multinomial Logistic Regression: 0.489
- Decision Tree: 0.475
- Random Forest: 0.531
- XGBoost: 0.264

The accuracies increased slightly, with the most notable improvement observed in the random forest model. The percentage improvements in accuracy were as follows:

- Multinomial Logistic Regression: 0.90
- Decision Tree: 0.60
- Random Forest: 6.42
- XGBoost: 0.09

Among the four models, the random forest model showed a significant improvement of over 6%, indicating its suitability for this analysis. Although the multinomial logistic regression model showed a slight improvement with the inclusion of all features, the enhancement was not substantial.

The random forest model emerges as the best overall model, both in terms of improvement and overall accuracy when using all features. This suggests that the random forest model is particularly effective at utilizing the full feature set to make accurate predictions, making it the preferred model for further analysis.

3.2.6 Definitive Selection of Top Features

This section presents the top 10 most important features identified by the best-performing machine learning model. This is followed by a plot ranking all variables from most to least important.

Top 16 Feature Importance

Feature	Importance	Feature	Importance
income_ppp	259	ed	185
B	258	MIL10	178
q2	247	ROS	177
D	236	R	176
vb11	220	SDNEW2	171
POL	212	IAREA	171
vb3n	203	MEDIA	169
q10new	192	a4	169

Table 3.12: Top 16 Feature Importance Values

It is important to note that the ranking of features may vary slightly with different random seeds. However, the top 16 features generally remain consistent across runs. There is a significant drop in importance after the top 9 features, which range from 185 to 178 in importance. Interestingly, the difference in importance between the 9th and 10th ranked features is almost equivalent to the difference between the 10th and 16th ranked features.

Category	Features
Income	income_ppp, q10new
Age	q2
Education	ed
Attitude Towards our Circumstances	D
Social/Institutional Trust	B, POL, MIL10
Political Persuasion	vb3n, vb11

Table 3.13: Categories and Corresponding Features (Top 10)

The table above categorizes the variables based on their corresponding 'question codes' within the subcategories of the seven main determinants outlined at the beginning of the article. This classification of features into categories such as income, education, and social trust aligns with existing economic literature on life satisfaction. Previous studies have consistently highlighted these factors as crucial in understanding life satisfaction. For example, income [17] [18] and education [19] are well-documented determinants of well-being, while social trust [20] and political engagement [21] have been shown to impact life satisfaction significantly. This categorization not only validates the importance of these features but also helps to contextualize them within the broader framework of life satisfaction research.

The following description will explain the most important feature codes and their meanings:

1. **Income (income_ppp, q10new)**
    - **income_ppp**: Reflects the respondent's monthly income converted to PPP. This allows for a more accurate comparison of income across different countries by accounting for cost of living differences. Keep in mind that the income_ppp variable is derived from q10g, which is the individual's monthly income, not from q10new.
    - **q10new**: Represents the total monthly income of the respondent's household, including remittances and income from all working members. Household income provides a broader view of the financial stability and resources available to an individual.
2. **Age (q2)**
    - **q2**: The age of the respondent. Age can influence life satisfaction through various stages of life, as individuals may experience different levels of contentment and fulfillment at different ages [22].
3. **Education (ed)**
    - **ed**: Indicates the number of years of schooling the respondent has completed. Education often correlates with higher income potential, better health, and more opportunities, which can contribute positively to life satisfaction [23].

4. **Attitude Towards Our Circumstances (D)**
    - **D1 to D6**: Measures the respondent's approval of democratic rights and freedoms, such as voting rights, the right to demonstrate, and the rights of minority groups. These questions gauge the individual's perception of political and social freedoms, which can impact their overall life satisfaction.
5. **Social/Institutional Trust (B, POL, MIL10)**
    - **B1 to B47A**: Assesses the respondent's trust in various political institutions, including the courts, the armed forces, the police, and political parties. Trust in these institutions can influence how secure and supported individuals feel within their society [20].
    - **POL (ING4, EFF1, EFF2, AOJ22NEW)**: Measures the respondent's views on democracy, political efficacy, and crime punishment. Positive views in these areas can lead to higher life satisfaction by fostering a sense of involvement and fairness in society [20].
    - **MIL10**: Evaluates the trustworthiness of various international governments and organizations. Trust in these entities can reflect broader geopolitical stability and personal security.
6. **Political Persuasion (vb3n, vb11)**
    - **vb3n**: Identifies who the respondent voted for in the last presidential election. This can indicate political alignment and satisfaction with political representation.
    - **vb11**: Indicates the political party the respondent identifies with. Political identity can be a significant factor in personal happiness and satisfaction, as it reflects shared values and community belonging.

These variables collectively cover a wide range of personal, social, and political factors that are crucial in understanding and predicting life satisfaction. The importance of these factors lies in their comprehensive representation of an individual's financial stability, educational background, age, personal beliefs, trust in institutions, and political engagement.

Features Plot

The plot below ranks all 74 variables by their importance, as determined by the Random Forest model. Each bar represents a variable, with darker shades of blue indicating higher importance and lighter shades denoting lower importance.

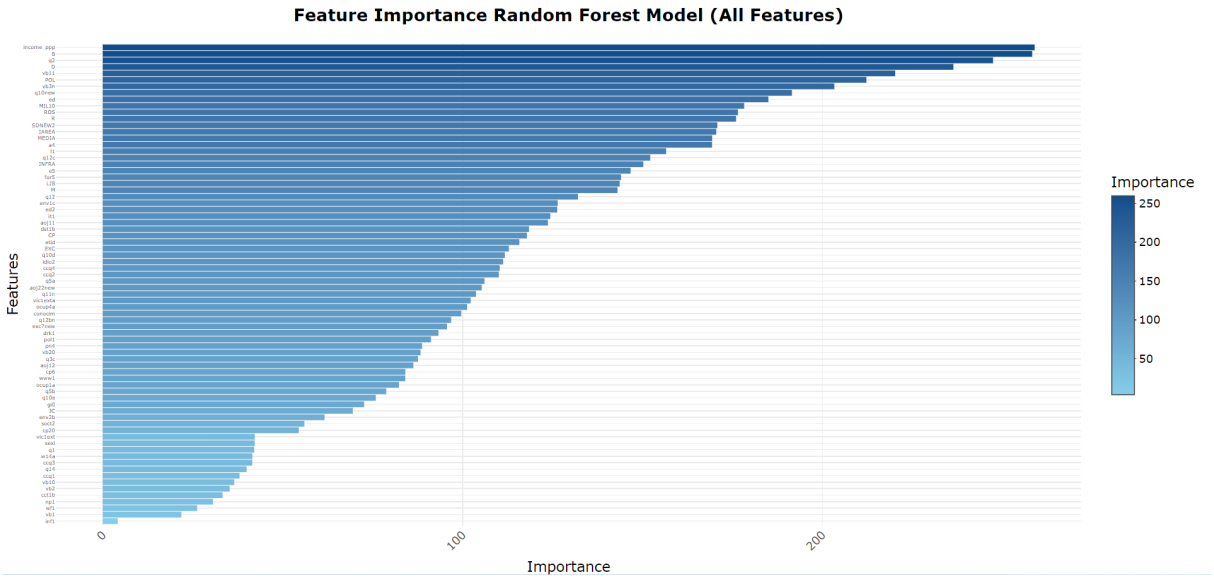


Figure 3.11: Feature Importance using Random Forest

The most influential variables are clearly distinguishable in the plot, demonstrating their significant impact on predicting life satisfaction (*ls3*). When comparing the top 10 variables identified through the correlation matrix with those highlighted by the Random Forest model, minimal differences are observed. The top 10 variables from the correlation matrix were: R, q10new, ed2, B, LIB, vic1ext, ed, a4, q14, and income_ppp. Notably, the variables not present in the top 16 of the Random Forest model are: ed2 (although a different education variable, ed, is included), LIB, vic1ext, and q14. This reveals that six variables are common between both lists, indicating strong consistency in their influence.

4. Conclusion

4.1 Conclusion of Cluster Analysis

The conclusion of the cluster analysis synthesizes an in-depth examination of the relationship between income and life satisfaction across South American countries. By analyzing correlations both between and within countries, the findings reveal how income influences life satisfaction in diverse socio-economic contexts. Key observations at both broad (macro) and specific (micro) levels are highlighted.

4.1.1 Comparing Between Countries and Within Countries

- **Comparing Between Countries:** When analyzing the correlation across the entire cluster of countries, the broader perspective captures greater variability and highlights overarching patterns that may not be apparent within smaller subsets.

  The between-countries correlation (0.374 for mean income in USD and 0.317 for mean PPP) suggests a moderate positive relationship between income and life satisfaction when considering all countries together. This pattern indicates that, generally, higher income is associated with higher life satisfaction across South America.

- **Within Each Country:** For individual countries, the dataset is smaller and potentially more homogeneous, which can lead to weaker correlations. There is less variability to capture a strong pattern, but country-specific factors become more pronounced.

  The highest individual country correlation of 0.202 indicates that within any given country, the relationship between income and life satisfaction is approximately weak. This weak yet positive correlation can be attributed to various country-specific factors, including cultural differences, social safety nets, cost of living, economic stability, and other non-income-related factors affecting life satisfaction.

4.1.2 Factors Behind Lower Correlations

- **Country-Specific Differences:** Each country has unique economic conditions and cultural factors that can dilute the impact of income on life satisfaction.
- **Sample Size:** Smaller sample sizes within each country may lead to less reliable estimates of correlation.
- **Income Distribution:** Variations in income distribution can affect the strength of the correlation. In more equitable countries, the correlation might be weaker compared to those with greater income inequality.
- **Non-Linear Relationships:** The relationship between income and life satisfaction may not be linear. Beyond a certain income level, further increases might not significantly boost life satisfaction.
- **Other Influencing Factors:** Factors like healthcare, education, social support, and political stability also play crucial roles in life satisfaction, potentially overshadowing the influence of income.

4.1.3 Understanding the Differences

The differences in correlations indicate that while income is an important factor in life satisfaction, its impact varies significantly based on specific conditions within each country. Broad patterns observed between countries might not be as evident within individual countries. This highlights the importance of considering both macro (regional) and micro (individual country) perspectives. Although the correlations between countries and within countries cannot be directly compared, the moderate positive correlation suggests that income does have an impact on life satisfaction.

4.1.4 Summary of the Cluster Analysis

In conclusion, the cluster analysis shows that income has a moderate correlation with life satisfaction across South American countries. Within individual countries, the correlation remains positive but weak, indicating that other factors significantly influence life satisfaction. This nuanced understanding is crucial for policymakers and researchers aiming to improve life satisfaction through economic measures.

4.2 Conclusion of Factor Analysis

Predicting an individual's well-being using 74 variables is inherently challenging. Achieving an accuracy of around 50% with the use of machine learning models is considered quite high and acceptable in this context.

4.2.1 Integrating Insights from Cluster Analysis to Factor Analysis

Building on the cluster analysis, a comprehensive examination explored additional variables impacting life satisfaction. This detailed exploration identified the top 10 most influential variables using a bivariate approach, such as a correlation matrix. While models using only these top 10 variables showed commendable accuracy, an interesting pattern emerged when the full set of 74 variables was used. The overall model accuracy improved by a marginal 0-6%, underscoring a crucial insight: although the top 10 variables are significant predictors of life satisfaction, the remaining 64 variables collectively enhance the model's predictive power. This highlights the importance of considering a broader range of variables to capture the multifaceted nature of life satisfaction.

Using the best-performing machine learning model, which employs a multivariate approach instead of a bivariate one, the importance of different variables shifted, altering the top 10 influential factors. These variables fall into categories such as income, age, education, attitude towards circumstances, social/institutional trust, and political persuasion. While income remains a significant factor, the inclusion of additional variables emphasizes the complex and multifaceted nature of life satisfaction. This validates both the cluster and factor analyses, showing the importance of a broad spectrum of factors in understanding life satisfaction.

4.2.2 Categorizing and Validating Variables

To better understand the impact of various factors on life satisfaction, variables were categorized into seven distinct categories based on findings from four referenced articles. Despite reducing the initial pool of over 180 variables to 74 through factor analysis, significant factors such as relaxation, recreation, and the nexus of nutrition, exercise, and sleep were not included in the dataset.

Furthermore, comprehensive health-related questions, encompassing both physical and mental health aspects, were notably absent. This gap is also evident in a survey about health in the Netherlands from the LISS database [24], which included many of these health-related variables. The omission of these factors highlights a limitation in the current dataset, suggesting that incorporating such additional variables could provide a more holistic understanding of the determinants of life satisfaction. Addressing these omissions in future research could achieve a more comprehensive understanding of the factors influencing life satisfaction. Integrating variables related to health, relaxation, and recreation would enrich the dataset and potentially lead to even more accurate predictions of life satisfaction.

4.3 Answering the Research Question

To provide clarity, the research question is restated as follows:

"Which variables from the LAPOP data are most influential on life satisfaction in South American countries?"

Addressing the Research Question: The analysis identifies key variables influencing life satisfaction in South American countries, focusing on income, age, education, personal attitudes, and institutional trust. The results reveal that these factors are significant predictors of life satisfaction. The top 10 variables provide substantial insights, but incorporating a broader range of factors from the dataset enhances the model's predictive accuracy, reflecting the multifaceted nature of life satisfaction.

Scope of Analysis: Currently, the analysis utilizes 74 variables, emphasizing the top 10-16 most influential ones. Other potential variables were not included in this analysis. The current model highlights the key predictors of life satisfaction but does not consider additional variables that could offer further insights.

Country-Specific Considerations: The analysis aggregates data across South American countries, providing a general overview of life satisfaction trends. No distinction is made between individual countries at this stage. Future analyses could benefit from country-specific insights to better understand regional variations in life satisfaction.

In summary, the analysis effectively addresses the research question by identifying key predictors of life satisfaction and demonstrating the importance of considering a broad range of factors. While the focus has been on the top variables, future work could explore additional factors and country-specific analyses to deepen the understanding of life satisfaction in South America.

4.4 Recommendations

To enhance the accuracy of predicting subjective well-being (SWB), it is essential to incorporate additional variables, particularly those related to health. Studies using data from the LISS database underscore the critical role of health in determining SWB [24]. Factors such as physical and mental health, as well as issues like addiction, are vital considerations that could significantly improve the model's predictive power.

Including these health-related variables could potentially elevate the model's accuracy to 70% or even 80%, signifying substantial and statistically meaningful performance improvements. While this projection is speculative and requires further validation, it highlights the importance of comprehensive data collection. This recommendation is particularly relevant for LAPOP interviewers, emphasizing the necessity of gathering detailed health-related data to enhance the accuracy of SWB predictions.

4.4.1 Follow-Up

To further advance the research, two follow-up studies are recommended. These follow-ups aim to refine the methodologies and enhance the robustness of the findings:

- **Reversing the Analytical Sequence:** One approach is to begin with a factor analysis to identify the most influential variables in South American Countries and subsequently perform a cluster analysis for each variable. This method could yield more distinct clusters per variable or allow for a combined cluster analysis that addresses the limitations posed by the small sample size (only 10 datapoints). This approach would enable the control of additional assumptions, such as the spherical shape of clusters.
- **Focusing Solely on Factor Analysis:** Another option is to eliminate the cluster analysis entirely and concentrate exclusively on the factor analysis. This approach would provide a deeper understanding of which variables are most significant in predicting life satisfaction within each South American country separately. By avoiding between-country comparisons through clustering, the research can uncover more nuanced insights into how cultural differences influence the importance of various variables in each country.

These follow-up studies could provide a more detailed understanding of the variables influencing life satisfaction and address the limitations encountered in the current research. By exploring these alternative approaches, future studies can offer more precise and culturally relevant insights into the factors that contribute to life satisfaction across South American countries.

4.5 Reflections and Insights

This research highlights the intersection of data science and economics in predicting life satisfaction. Several key findings emerged from this interdisciplinary approach:

Data Science vs. Economics Perspectives: The methodologies and analytical techniques from data science offered a unique perspective for examining economic phenomena. Traditional economic models can utilize continuous trend analyses and may not fully account for irregularities and instabilities in economic data, as discussed in the literature [25]. In contrast, data science approaches can provide a more granular and flexible analysis of individual predictors of life satisfaction.

Commonalities and Differences: Both disciplines underscored the importance of variables such as income, education, and age. However, data science methodologies revealed additional significant predictors, including attitude towards circumstances, institutional trust and political persuasion, which are often overlooked in traditional economic research [26].

Collaborative Experience: Working at the intersection of data science and economics was a valuable experience. It demonstrated the potential of combining quantitative techniques from data science with economic theories to derive more comprehensive insights. The collaboration highlighted the importance of interdisciplinary research in addressing complex socio-economic issues.

Learning Outcomes: The project offered valuable learning experiences, particularly in applying machine learning models to socio-economic data. Key insights gained include the significance of data quality and completeness, as well as the ethical considerations involved in data handling and interpretation.

These reflections emphasize the potential benefits of continued interdisciplinary collaboration and the integration of advanced data science techniques into economic research.

Bibliography

[1] E. Angner, "Subjective well-being," *The Journal of Socio-Economics*, vol. 39, no. 3, pp. 361–368, 2010, Financial Crisis: Capabilities and Happiness, ISSN: 1053-5357. DOI: https://doi.org/10.1016/j.socec.2009.12.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053535709001590.

[2] V. University, *Americasbarometer 2017 master questionnaire version # 18.0*, 2017. [Online]. Available: https://www.vanderbilt.edu/lapop/ab2016/AB2017-v18.0-Eng-170523_W.pdf.

[3] E. Diener and R. Biswas-Diener, "Will money increase subjective well-being?: A literature review and guide to needed research," in *Social indicators research series*, Springer, 2009, pp. 119–154. DOI: 10.1007/978-90-481-2350-6_6.

[4] P. Dolan, T. Peasgood, and M. White, "Do we really know what makes us happy? a review of the economic literature on the factors associated with subjective well-being," *Journal of Economic Psychology*, vol. 29, no. 1, pp. 94–122, 2008. DOI: 10.1016/j.joep.2007.09.001.

[5] R. J. Estes and M. J. Sirgy, *The pursuit of human Well-Being* (International handbooks of quality-of-life). Springer, 2017. DOI: 10.1007/978-3-319-39101-4.

[6] N. Pontarollo, M. Orellana, J. Segovia, *et al.*, "The determinants of subjective well-being in a developing country: The ecuadorian case," *Journal of Happiness Studies*, 2019. DOI: 10.1007/s10902-019-00211-w.

[7] "Quality of life in the european union: A multidimensional analysis," *Soc Indic Res*, vol. 141, pp. 765–789, 2018. DOI: 10.1007/s11205-018-1854-y.

[8] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, IEEE, 2010, pp. 63–67. DOI: 10.1109/iitsi.2010.74.

[9] F. Batool and C. Hennig, "Clustering with the average silhouette width," *Computational Statistics & Data Analysis*, vol. 158, p. 107 190, 2021. DOI: 10.1016/j.csda.2021.107190.

[10] J. I. Daoud, "Multicollinearity and regression analysis," *Journal of Physics: Conference Series*, vol. 949, p. 012 009, 2017. DOI: 10.1088/1742-6596/949/1/012009.

[11] P. C. Austin, I. R. White, D. S. Lee, and S. Van Buuren, "Missing data in clinical research: A tutorial on multiple imputation," *Canadian Journal of Cardiology*, vol. 37, no. 9, pp. 1322–1331, 2021. DOI: 10.1016/j.cjca.2020.11.010.

[12] B. Vrigazova, "The proportion for splitting data into training and test set for the bootstrap in classification problems," *Business Systems Research*, vol. 12, no. 1, pp. 228–242, 2021. DOI: 10.2478/bsrj-2021-0015.

[13] O. Almomani, M. A. Almaiah, A. Alsaaidah, S. Smadi, A. H. Mohammad, and A. Althunibat, "Machine learning classifiers for network intrusion detection system: Comparative study," in *2021 International Conference on Information Technology (ICIT)*, IEEE, 2021, pp. 70–75. DOI: 10.1109/icit52682.2021.9491770.

[14] R. M. Warner, *Applied Statistics: From Bivariate through Multivariate Techniques*. SAGE Publications, Inc., 2007. [Online]. Available: http://ci.nii.ac.jp/ncid/BA85186840.

[15] P. Sedgwick, "Spearman's rank correlation coefficient," *BMJ: British Medical Journal*, vol. 349, g7327, 2014. DOI: 10.1136/bmj.g7327.

[16] LAPOP, *AmericasBarometer, 2016/17*, In LAPOP AmericasBarometer 2016/17 Round of Surveys, 2016. [Online]. Available: https://www.vanderbilt.edu/lapop/ab2016/AmericasBarometer_2016-17_Sample_Design.pdf.

[17] E. Diener and M. E. P. Seligman, "Beyond money: Toward an economy of well-being," *Psychological Science in the Public Interest*, vol. 5, no. 1, pp. 1–31, 2004.

[18] R. A. Easterlin, *Does Economic Growth Improve the Human Lot? Some Empirical Evidence*, P. A. David and M. W. Reder, Eds. Academic Press, 1974.

[19] J. F. Helliwell, R. Layard, and J. D. Sachs, "World happiness report 2020," Sustainable Development Solutions Network, Tech. Rep., 2020.

[20] R. Inglehart and P. Norris, *The Growth of Secularism, the Decline of the Religious Factor, and the Emergence of New Political Attitudes*, P. Norris, Ed. Princeton University Press, 2003.

[21] J. F. Helliwell and R. D. Putnam, "The social context of well-being," in *The Science of Well-Being*, John Wiley & Sons, 2004.

[22] D. G. Blanchflower and A. J. Oswald, *Happiness over Time: A Review*, R. A. Easterlin, Ed. MIT Press, 2008.

[23] P. Oreopoulos and U. Petronijevic, "Making college worth it: A review of the returns to higher education," *The Future of Children*, vol. 23, no. 1, pp. 41–65, 2013.

[24] M. Marchand and CentERdata, *Health*, 2019.

[25] Y. Zhukova and O. Sobolieva-Tereshchenko, "Modeling macroeconomic indicators in unstable economies," *Journal of International Studies*, vol. 14, no. 2, pp. 128–148, 2021. DOI: 10.14254/2071-8330.2021/14-2/9.

[26] T. Carver and A. Grimes, "Income or consumption: Which better predicts subjective well-being?" *The Review of Income and Wealth*, vol. 65, no. S1, 2019. DOI: 10.1111/roiw.12414.

Appendices

Appendix A: Categorization 7 Determinants SWB

This appendix contains the question codes from the LAPOP core questionnaire, categorized into seven main determinants. Additionally, it includes information on which columns are present in each country.

Below begins the categorization of each question code into the seven main determinants.

1. Income:

- Q10G
- Q10NEW
- Q10D
- Q10E

2. Personal Characteristics:

- Age: Q2Y, Q2
- Gender: Q1, SEXI
- Ethnicity/cultural/indigenous: A4L, INF1, ETID, COLORR, COLORI
- Personality: VB1, VB2
- INTID = interviewer ID

3. Socially developed characteristics:

- Education: ED, ED2, CONOCIM
- Health: VAC1
- Type of work: OCUP1A
- Unemployment: (OCUP4A - 3, 4, 6, 7)

4. How we spend our time:

- Hours worked: (OCUP4A – 1, 2)
- Commuting time:
- Caring for others: (OCUP4A - 5)
- Community involvement and volunteering: NP1, CP7, CP8, CP13, CP20, PROT3
- Exercise:
- Religious activities: CP6, Q5A

5. Attitude and beliefs towards: self, others, life

- Attitude towards our circumstances: IT1, IDIO2, W14A, E5, D1, D2, D3, D4, D5, D6, LIB1, LIB2B, LIB2C, LIB4, EXC2, EXC6, EXC20, EXC11, EXC13, EXC14, EXC15, EXC16, EXC18, EXC7NEW
- (Social/institutional) trust: AOJ12, B1, B2, B3, B4, B6, B43, B12, B13, B18, B21, B21A, B32, B37, B47A, INFRAX, INFRA3, MIL10A1, MIL10E1, MIL10OAS1, MIL10UN1, MILOAS2, MIL10UN2, MIL10A2, MIL10E2, GI0
- Political persuasion: L1, L1B, JC10, JC13, JC15A, M1, M2, ROS1, ROS4, ING4, EFF1, EFF2, AOJ22NEW, MEDIA3, MEDIA4, VB3N, VB10, VB11, POL1, VB20
- Religion: Q5B, Q3C

6. Relationships:

- Marriage and intimate relationship: Q11n
- Having children: Q12, Q12M, Q12F
- Seeing family and friends: Q12C, Q12Bn

7. Wider economic, social and political environment:

- Income inequality: A4, SOCT2
- Unemployment rates: A4, SOCT2
- Inflation: A4, SOCT2
- Welfare system and public insurance: A4, WF1, CCT1B
- Degree of democracy: A4, PN4, FOR5N
- Climate and the natural environment: A4, DRK11, ENV1C1, ENV2B1, ENV1C2, ENV2B2, DRK12
- Safety and deprivation of the area: A4, VIC1EXT, VIC1EXTA, AOJ11, DST1B1, DST1B2, IAREA1, IAREA2, IAREA3, IAREA4, IAREA6, IAREA7
- Urbanization (living area): A4, SD2NEW2, SD3NEW2, SD6NEW2, CCQ1, CCQ2, CCQ3, CCQ4, Q14
- Internet access: WWW1
- Housing conditions: A4, R3, R4, R4A, R5, R6, R7, R8, R12, R14, R15, R18, R1, R16

- This section identifies the columns that are missing for each country. The named columns are those present in the data file.

Argentina missing columns: q2y, a4l, intid, np1, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, aoj22new, q12m, q12f, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2

Brazil missing columns: q2y, a4l, intid, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2

Colombia: a4l, vb1, intid, vac1, e5, b43, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, jc13, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2, ccq1, ccq2, ccq3, ccq4

Ecuador: a4l, intid, vac1, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2, ccq1, ccq2, ccq3, ccq4

Peru: q2y, a4l, inf1, intid, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2

Bolivia: q2y, a4l, intid, np1, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, q12m, q12f, cct1b, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2

Paraguay: a4l, intid, vac1, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2, ccq1, ccq2, ccq3, ccq4

Chile: q2y, a4l, vb1, intid, vac1, np1, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, aoj22new, q12m, q12f, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2

Uruguay: q2y, a4l, intid, np1, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, aoj22new, q12m, q12f, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2

Venezuela: a4l, intid, vac1, mil10a1, mil10e1, mil10oas1, mil10un1, miloas2, mil10un2, mil10a2, mil10e2, l1b, for5n, drk11, env1c1, env2b1, env1c2, env2b2, drk12, dst1b1, dst1b2, ccq1, ccq2, ccq3, ccq4

Appendix B: R Report

This appendix presents an R Markdown file containing all the code used in this article, rendered to HTML and subsequently converted to PDF for inclusion.

Predictive Analysis of Subjective Well-Being in South America Using Machine Learning Techniques

AUTHOR
Rodi Doesburg

PUBLISHED
July 27, 2024

Introduction

Alternative title: “Uncovering Major Influences on South American Subjective Well-Being through Machine Learning”

The primary objective of this report is to investigate the determinants of subjective well-being (SWB) in South America, with a focus on understanding the factors that influence overall life satisfaction. To achieve this, data from the Latin American Public Opinion Project (LAPOP) is utilized, specifically concentrating on South American countries. The countries included in this analysis are Colombia, Ecuador, Peru, Bolivia, Paraguay, Chile, Uruguay, Argentina, Venezuela, and Brazil. These are the ten countries classified as South America in the technical information of LAPOP.

The outcome variable for this study is derived from the LAPOP survey question LS3, which asks respondents about their general life satisfaction. The specific question is: “To begin, in general how satisfied are you with your life?” This question serves as the measure of SWB.

The analysis focuses on the years 2016 to 2017. This period was chosen to avoid the economic influences of the 2008 financial crisis and the onset of the COVID-19 pandemic in late 2019. The year 2016/2017 also had the most available data, making it a suitable period for a comprehensive analysis.

The analysis is divided into two main sections. The first section focuses on the relationship between income and life satisfaction. Economic studies often emphasize the role of income in predicting life satisfaction, and this analysis aims to validate or challenge this assumption in the South American context.

The second section broadens the scope to include a comprehensive range of variables from the LAPOP dataset. Given that the dataset includes over 180 questions, the objective is to identify which of these variables most significantly influence life satisfaction using various machine learning models. This analysis will touch more upon the data scientific working field.

To guide the selection of variables, four key articles on determinants of ll-being re reviewed. A combination of all used determinants in the four articles was made. The variables discussed in these articles were categorized into seven broad groups, which will be used in the analysis:

1. Income
2. Personal Characteristics: age, gender, ethnicity, personality
3. Socially Developed Characteristics: education, health, type of work, unemployment
4. How Time is Spent: hours worked, commuting, caring for others, community involvement, volunteering, exercise, religious activities
5. Attitudes and Beliefs: attitudes towards circumstances, trust, political persuasion, religion
6. Relationships: marriage, intimate relationships, having children, seeing family and friends

7. Wider Economic, Social, and Political Environment: income equality, unemployment rates, inflation, welfare system, public insurance, democracy, climate, natural environment, safety, urbanization

The questions from the core questionnaire of the LAPOP data all have codes. For example, LS3 is the question about life satisfaction. These codes, each corresponding to a specific column in the raw data files, are categorized within the seven main determinants. After conducting the two analyses, it will be straightforward to see which code corresponds to which sub-categories in the seven main categories. This categorization will facilitate making conclusions about which determinants are most influential on the life satisfaction outcome variable.

Combining traditional economic perspectives with a comprehensive exploration of additional variables, this report aims to provide a nuanced understanding of the factors that shape subjective well-being in South America. The analysis bridges the worlds of economy and data science, offering a holistic view of well-being determinants.

Libraries & Packages

Below the libraries currently loaded in the R session are displayed. The code is structured to first check if each package needs to be installed. If it does, the package is installed and then immediately loaded into the session.

▼ Code

```
if (!require(broom)) {
  install.packages("broom")
  library(broom)
}

if (!require(car)) {
  install.packages("car")
  library(car)
}

if (!require(caret)) {
  install.packages("caret")
  library(caret)
}

if (!require(cluster)) {
  install.packages("cluster")
  library(cluster)
}

if (!require(DescTools)) {
  install.packages("DescTools")
  library(DescTools)
}

if (!require(doParallel)) {
  install.packages("doParallel")
  library(doParallel)
```

```
}

if (!require(dplyr)) {
  install.packages("dplyr")
  library(dplyr)
}

if (!require(factoextra)) {
  install.packages("factoextra")
  library(factoextra)
}

if (!require(ggplot2)) {
  install.packages("ggplot2")
  library(ggplot2)
}

if (!require(ggrepel)) {
  install.packages("ggrepel")
  library(ggrepel)
}

if (!require(glmnet)) {
  install.packages("glmnet")
  library(glmnet)
}

if (!require(gridExtra)) {
  install.packages("gridExtra")
  library(gridExtra)
}

if (!require(haven)) {
  install.packages("haven")
  library(haven)
}

if (!require(htmlwidgets)) {
  install.packages("htmlwidgets")
  library(htmlwidgets)
}

if (!require(kableExtra)) {
  install.packages("kableExtra")
  library(kableExtra)
}

if (!require(knitr)) {
  install.packages("knitr")
  library(knitr)
}

if (!require(lme4)) {
  install.packages("lme4")
```

```
  library(lme4)
}

if (!require(Matrix)) {
  install.packages("Matrix")
  library(Matrix)
}

if (!require(MASS)) {
  install.packages("MASS")
  library(MASS)
}

if (!require(mice)) {
  install.packages("mice")
  library(mice)
}

if (!require(nnet)) {
  install.packages("nnet")
  library(nnet)
}

if (!require(plotly)) {
  install.packages("plotly")
  library(plotly)
}

if (!require(pROC)) {
  install.packages("pROC")
  library(pROC)
}

if (!require(psych)) {
  install.packages("psych")
  library(psych)
}

if (!require(randomForest)) {
  install.packages("randomForest")
  library(randomForest)
}

if (!require(readr)) {
  install.packages("readr")
  library(readr)
}

if (!require(rpart)) {
  install.packages("rpart")
  library(rpart)
}

if (!require(tibble)) {
```

```
  install.packages("tibble")
  library(tibble)
}

if (!require(tidyverse)) {
  install.packages("tidyverse")
  library(tidyverse)
}

if (!require(vcd)) {
  install.packages("vcd")
  library(vcd)
}

if (!require(xgboost)) {
  install.packages("xgboost")
  library(xgboost)
}
```

Data Import

The LAPOP survey data will be imported in accordance with the technical information outlined in the "Table 2: Fieldwork dates by country 2016/17 AmericasBarometer" from the AmericasBarometer series.

The datasets used in this analysis are sourced from the Latin American Public Opinion Project (LAPOP) and can be accessed via the official LAPOP website:

https://www.vanderbilt.edu/lapop/core-surveys.php

▼ Code

```
colombia_16 <- read_dta("1966987763Colombia LAPOP AmericasBarometer 2016 V1.0_W.dta")

ecuador_16_17 <- read_dta("1061044693Ecuador LAPOP AmericasBarometer 2016-17 V1.0_W.dta")

peru_17 <- read_dta("925581536Peru LAPOP AmericasBarometer 2017 V1.0_W.dta")

bolivia_17 <- read_dta("142790203Bolivia LAPOP AmericasBarometer 2017 V1.0_W.dta")

paraguay_16 <- read_dta("1311870907Paraguay LAPOP AmericasBarometer 2016 V1.0 W.dta")

chile_17 <- read_dta("336280178Chile LAPOP AmericasBarometer 2017 V1.0_W.dta")

uruguay_17 <- read_dta("646644681Uruguay LAPOP AmericasBarometer 2017 V1.0_W.dta")

brazil_17 <- read_dta("780314464Brazil LAPOP AmericasBarometer 2017 V1.0_W.dta")

venezuela_16_17 <- read_dta("25258094Venezuela LAPOP AmericasBarometer 2016-17 V1.0_W.dta")

argentina_17 <- read_dta("42076439Argentina LAPOP AmericasBarometer 2017 V1.0_W.dta")
```

Data Pre-Processing

In the data pre-processing chapter, a systematic approach is adopted to prepare the datasets for the two analyses. The following five steps outline the necessary transformations and adjustments:

- Transformation of the Life Satisfaction Scale (ls3):

The current scale for life satisfaction (ls3) is counterintuitive, with lower categories indicating higher satisfaction (e.g., category 1 = very satisfied, category 4 = very dissatisfied). To enhance clarity and interpretability, this scale will be inverted so that higher categories correspond to higher life satisfaction.

- Refinement of Country-Specific Datasets:

Each individual country dataset will be refined to include only the variables pertinent to the analyses. This involves excluding columns specific to individual countries (e.g., questions about Colombian politics) and retaining only the core questionnaire questions applicable to all ten South American countries.

- Conversion of the Income variable (q10g):

The individual income variable, located in column q10g in each dataset, will undergo a multi-step conversion process. The income variable q10g are displayed in local currencies in each individual country dataset. This column will be converted to the local currency per country. Following this, exchange rates will be applied to convert these amounts into USD. Finally, the income variable will be standardized using purchasing power parity (PPP) to ensure comparability across different countries. This process will utilize data from the World Bank.

- Unification of Similar Variables:

To reduce the complexity and variance in the machine learning models, similar variables will be compiled into composite factors. This involves identifying questionnaire codes that pertain to the same subject and consolidating them into a single factor. This step will streamline the analyses and improve the readability of the results in Analysis 2.

- Compilation and Integration of Datasets:

After performing the steps above, all ten country-specific datasets will be compiled into a single dataset. This unified dataset will facilitate the application of machine learning models, many of which require a single dataset.

Additionally, a thorough check for missing values will be conducted to ensure sufficient samples for each variable. The dataset will then be subsetted to include only the necessary variables for analysis, with the redundant columns being retained temporarily for verification purposes. This cautious approach ensures data integrity throughout the pre-processing phase.

Transforming the Outcome Variable

In the code below the categories for the ls3 column will be turned around. In that case during the analyses there will be a positive correlation and positive trend instead of a negative one.

▼ Code

```
# Transform ls3 for each country
colombia_16 <- colombia_16 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

ecuador_16_17 <- ecuador_16_17 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

peru_17 <- peru_17 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

bolivia_17 <- bolivia_17 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

paraguay_16 <- paraguay_16 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

chile_17 <- chile_17 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

uruguay_17 <- uruguay_17 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
```

```
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

brazil_17 <- brazil_17 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

venezuela_16_17 <- venezuela_16_17 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))

argentina_17 <- argentina_17 %>%
  mutate(ls3 = case_when(
    ls3 == 1 ~ 4,
    ls3 == 2 ~ 3,
    ls3 == 3 ~ 2,
    ls3 == 4 ~ 1
  ))
```

Subset Columns per Country

- Identify Missing Columns

The following test code identifies all missing columns in the country-specific datasets. This ensures that the code for selecting the columns to be used can be tailored for each country, preventing errors caused by non-existent columns.

▼ Code

```
# Create a vector of expected columns
expected_columns <- c("pais", "uniq_id",
                                      "ls3",
                                      "q10g", "q10new", "q10d", "q10e",
                                      "q2y", "q2", "q1", "sexi", "inf1",
                      "etid", "colorr", "colori", "vb1", "vb2", "intid",
                                      "ed", "ed2", "conocim", "vac1", "ocup1a",
                      "ocup4a",
                                      "np1", "cp7", "cp8", "cp13", "cp20",
                      "prot3", "cp6", "q5a",
                                      "it1", "idio2", "w14a", "e5", "d1",
                      "d2", "d3", "d4", "d5", "d6", "lib1", "lib2b", "lib2c",
                      "lib4", "exc2", "exc6", "exc20", "exc11", "exc13", "exc14",
```

```
                      "exc15", "exc16", "exc18", "exc7new", "aoj12", "b1", "b2",
                      "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
                      "b32", "b37", "b47a", "infrax", "infra3", "mil10a1",
                      "mil10e1", "mil10oas1", "mil10un1", "miloas2", "mil10un2",
                      "mil10a2", "mil10e2", "gi0", "l1", "jc10", "jc13", "jc15a",
                      "m1", "m2", "ros1", "ros4", "ing4", "eff1", "eff2",
                      "aoj22new", "media3", "media4", "vb3n", "vb10", "vb11",
                      "pol1", "vb20", "q5b", "q3c",
                                      "q11n", "q12", "q12m", "q12f", "q12c",
                      "q12bn",
                                      "a4", "soct2", "wf1", "cct1b", "pn4",
                      "for5n", "drk11", "env1c1", "env2b1", "env1c2", "env2b2",
                      "drk12", "vic1ext", "vic1exta", "aoj11", "dst1b1", "dst1b2",
                      "iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7",
                      "sd2new2", "sd3new2", "sd6new2", "ccq1", "ccq2", "ccq3",
                      "ccq4", "q14", "www1", "r3", "r4", "r4a", "r5", "r6", "r7",
                      "r8", "r12", "r14", "r15", "r18", "r1", "r16")

# Get the actual column names from the dataset
actual_columns <- colnames(colombia_16) # use the dataset needed, this is an example

# Identify missing columns
missing_columns <- setdiff(expected_columns, actual_columns)

# Print all missing columns
if(length(missing_columns) > 0) {
  print(paste("The following columns are missing from the dataset:",
              paste(missing_columns, collapse = ", ")))
} else {
  print("All expected columns are present in the dataset.")
}
```

- Subset Columns per Country

The following codes subsets the raw data for each country to include only the columns that will be used in the analyses. The selection of columns for each country is based on the results of the missing columns identification process.

▼ Code

```
colombia_subset <- colombia_16[, c("pais", "uniq_id",
                                   "ls3",
                                   "q10g", "q10new", "q10d", "q10e",
                                   "q2y", "q2", "q1", "sexi", "inf1",
                                 "etid", "colorr", "colori", "vb2",
                                   "ed", "ed2", "conocim", "ocup1a",
                                 "ocup4a",
                                   "np1", "cp7", "cp8", "cp13", "cp20",
                                 "prot3", "cp6", "q5a",
                                   "it1", "idio2", "w14a", "d1", "d2", "d3",
                                 "d4", "d5", "d6", "lib1", "lib2b", "lib2c",
                                 "lib4", "exc2", "exc6", "exc20", "exc11",
                                 "exc13", "exc14", "exc15", "exc16", "exc18",
                                 "exc7new", "aoj12", "b1", "b2", "b3", "b4",
```

```
"b6", "b12", "b13", "b18", "b21", "b21a",
"b32", "b37", "b47a", "infrax", "infra3",
"mil10a", "mil10e", "mil10oas", "mil10un",
"gi0", "l1", "jc10", "jc15a", "m1", "m2",
"ros1", "ros4", "ing4", "eff1", "eff2",
"aoj22new", "media3", "media4", "vb3n",
"vb10", "vb11", "pol1", "vb20", "q5b", "q3c",
    "q11n", "q12", "q12m", "q12f", "q12c",
"q12bn",
    "a4", "soct2", "wf1", "cct1b", "pn4",
"for5", "drk1", "env1c", "env2b", "vic1ext",
"vic1exta", "aoj11", "dst1b", "iarea1",
"iarea2", "iarea3", "iarea4", "iarea6",
"iarea7", "sd2new2", "sd3new2", "sd6new2",
"q14", "www1", "r3", "r4", "r4a", "r5", "r6",
"r7", "r8", "r12", "r14", "r15", "r18", "r1",
"r16")]
```

▼ Code

```
ecuador_subset <- ecuador_16_17[, c("pais", "uniq_id",
                                    "ls3",
                                    "q10g", "q10new", "q10d", "q10e",
                                    "q2y", "q2", "q1", "sexi", "inf1",
                                   "etid", "colorr", "colori", "vb1", "vb2",
                                    "ed", "ed2", "conocim", "ocup1a",
                                   "ocup4a",
                                    "np1", "cp7", "cp8", "cp13", "cp20",
                                   "prot3", "cp6", "q5a",
                                    "it1", "idio2", "w14a", "e5", "d1",
                                   "d2", "d3", "d4", "d5", "d6", "lib1",
                                   "lib2b", "lib2c", "lib4", "exc2", "exc6",
                                   "exc20", "exc11", "exc13", "exc14", "exc15",
                                   "exc16", "exc18", "exc7new", "aoj12", "b1",
                                   "b2", "b3", "b4", "b6", "b43", "b12", "b13",
                                   "b18", "b21", "b21a", "b32", "b37", "b47a",
                                   "infrax", "infra3", "mil10a", "mil10e",
                                   "mil10oas", "mil10un", "gi0", "l1", "jc10",
                                   "jc13", "jc15a", "m1", "m2", "ros1", "ros4",
                                   "ing4", "eff1", "eff2", "aoj22new", "media3",
                                   "media4", "vb3n", "vb10", "vb11", "pol1",
                                   "vb20", "q5b", "q3c",
                                    "q11n", "q12", "q12m", "q12f", "q12c",
                                   "q12bn",
                                    "a4", "soct2", "wf1", "cct1b", "pn4",
                                   "for5", "drk1", "env1c", "env2b", "vic1ext",
                                   "vic1exta", "aoj11", "dst1b", "iarea1",
                                   "iarea2", "iarea3", "iarea4", "iarea6",
                                   "iarea7", "sd2new2", "sd3new2", "sd6new2",
                                   "q14", "www1", "r3", "r4", "r4a", "r5", "r6",
                                   "r7", "r8", "r12", "r14", "r15", "r18", "r1",
                                   "r16")]
```

▼ Code

```
peru_subset <- peru_17[, c("pais", "uniq_id",
                                    "ls3",
                                    "q10g", "q10new", "q10d", "q10e",
                                    "q2", "q1", "sexi", "etid", "colorr",
                           "colori", "vb1", "vb2",
                                    "ed", "ed2", "conocim", "vac1", "ocup1a",
                           "ocup4a",
                                    "np1", "cp7", "cp8", "cp13", "cp20",
                           "prot3", "cp6", "q5a",
                                    "it1", "idio2", "w14a", "e5", "d1", "d2",
                           "d3", "d4", "d5", "d6", "lib1", "lib2b", "lib2c",
                           "lib4", "exc2", "exc6", "exc20", "exc11", "exc13",
                           "exc14", "exc15", "exc16", "exc18", "exc7new",
                           "aoj12", "b1", "b2", "b3", "b4", "b6", "b43", "b12",
                           "b13", "b18", "b21", "b21a", "b32", "b37", "b47a",
                           "infrax", "infra3", "mil10a", "mil10e", "gi0", "l1",
                           "jc10", "jc13", "jc15a", "m1", "m2", "ros1", "ros4",
                           "ing4", "eff1", "eff2", "aoj22new", "media3",
                           "media4", "vb3n", "vb10", "vb11", "pol1", "vb20",
                           "q5b", "q3c",
                                    "q11n", "q12", "q12m", "q12f", "q12c",
                           "q12bn",
                                    "a4", "soct2", "wf1", "cct1b", "pn4",
                           "for5", "drk1", "env1c", "env2b", "vic1ext",
                           "vic1exta", "aoj11", "dst1b", "iarea1", "iarea2",
                           "iarea3", "iarea4", "iarea6", "iarea7", "sd2new2",
                           "sd3new2", "sd6new2", "ccq1", "ccq2", "ccq3", "ccq4",
                           "q14", "www1", "r3", "r4", "r4a", "r5", "r6", "r7",
                           "r8", "r12", "r14", "r15", "r18", "r1", "r16")]
```

▼ Code

```
bolivia_subset <- bolivia_17[, c("pais", "uniq_id",
                                    "ls3",
                                    "q10g", "q10new", "q10d", "q10e",
                                    "q2", "q1", "sexi", "inf1", "etid",
                                 "colorr", "colori", "vb1", "vb2",
                                    "ed", "ed2", "conocim", "vac1", "ocup1a",
                                 "ocup4a",
                                    "cp7", "cp8", "cp13", "cp20", "prot3",
                                 "cp6", "q5a",
                                    "it1", "idio2", "w14a", "e5", "d1", "d2",
                                 "d3", "d4", "d5", "d6", "lib1", "lib2b",
                                 "lib2c", "lib4", "exc2", "exc6", "exc20",
                                 "exc11",  "exc13", "exc14", "exc15", "exc16",
                                 "exc18",    "exc7new", "aoj12", "b1", "b2",
                                 "b3", "b4", "b6", "b43", "b12", "b13", "b18",
                                 "b21", "b21a",    "b32", "b37", "b47a",
                                 "infrax", "infra3",  "mil10a", "mil10e",
                                 "mil10oas", "mil10un", "gi0", "l1", "jc10",
                                 "jc13", "jc15a", "m1", "m2", "ros1", "ros4",
                                 "ing4", "eff1", "eff2",  "aoj22new", "media3",
                                 "media4", "vb3n", "vb10",
                                 "vb11", "pol1", "vb20", "q5b", "q3c",
```

```
                                    "q11n", "q12", "q12c", "q12bn",
                                    "a4", "soct2", "wf1", "pn4", "for5",
                                "drk1", "env1c", "env2b", "vic1ext", "vic1exta",
                                "aoj11", "dst1b", "iarea1", "iarea2", "iarea3",
                                "iarea4", "iarea6", "iarea7", "sd2new2",
                                "sd3new2", "sd6new2", "ccq1", "ccq2", "ccq3",
                                "ccq4", "q14", "www1", "r3", "r4", "r4a", "r5",
                                "r6", "r7", "r8", "r12", "r14", "r15", "r18",
                                "r1", "r16")]
```

▼ Code

```
paraguay_subset <- paraguay_16[, c("pais", "uniq_id",
                                    "ls3",
                                    "q10g", "q10new", "q10d", "q10e",
                                    "q2y", "q2", "q1", "sexi", "inf1",
                                  "etid", "colorr", "colori", "vb1", "vb2",
                                    "ed", "ed2", "conocim", "ocup1a",
                                  "ocup4a",
                                    "np1", "cp7", "cp8", "cp13", "cp20",
                                  "prot3", "cp6", "q5a",
                                    "it1", "idio2", "w14a", "e5", "d1",
                                  "d2", "d3", "d4", "d5", "d6", "lib1", "lib2b",
                                  "lib2c", "lib4", "exc2", "exc6", "exc20",
                                  "exc11", "exc13", "exc14", "exc15", "exc16",
                                  "exc18", "exc7new", "aoj12", "b1", "b2", "b3",
                                  "b4", "b6", "b43", "b12", "b13", "b18", "b21",
                                  "b21a", "b32", "b37", "b47a", "infrax",
                                  "infra3", "mil10a", "mil10e", "mil10oas",
                                  "mil10un", "gi0", "l1", "jc10", "jc13",
                                  "jc15a", "m1", "m2", "ros1", "ros4", "ing4",
                                  "eff1", "eff2", "aoj22new", "media3",
                                  "media4", "vb3n", "vb10", "vb11", "pol1",
                                  "vb20", "q5b", "q3c",
                                    "q11n", "q12", "q12m", "q12f", "q12c",
                                  "q12bn",
                                    "a4", "soct2", "wf1", "cct1b", "pn4",
                                  "for5", "drk1", "env1c", "env2b", "vic1ext",
                                  "vic1exta", "aoj11", "dst1b", "iarea1",
                                  "iarea2", "iarea3", "iarea4", "iarea6",
                                  "iarea7", "sd2new2", "sd3new2", "sd6new2",
                                  "q14", "www1", "r3", "r4", "r4a", "r5", "r6",
                                  "r7", "r8", "r12", "r14", "r15", "r18", "r1",
                                  "r16")]
```

▼ Code

```
chile_subset <- chile_17[, c("pais", "uniq_id",
                                    "ls3",
                                    "q10g", "q10new", "q10d", "q10e",
                                    "q2", "q1", "sexi", "inf1", "etid",
                               "colorr", "colori", "vb2",
                                    "ed", "ed2", "conocim", "ocup1a",
                               "ocup4a",
```

```
                                    "cp7", "cp8", "cp13", "cp20", "prot3",
                                "cp6", "q5a",
                                    "it1", "idio2", "w14a", "e5", "d1",
                                "d2", "d3", "d4", "d5", "d6", "lib1", "lib2b",
                                "lib2c", "lib4", "exc2", "exc6", "exc20", "exc11",
                                "exc13", "exc14", "exc15", "exc16", "exc18",
                                "exc7new", "aoj12", "b1", "b2", "b3", "b4", "b6",
                                "b43", "b12", "b13", "b18", "b21", "b21a", "b32",
                                "b37", "b47a", "infrax", "infra3", "mil10a",
                                "mil10e", "mil10oas", "mil10un", "gi0", "l1",
                                "jc10", "jc13", "jc15a", "m1", "m2", "ros1", "ros4",
                                "ing4", "eff1", "eff2", "media3", "media4", "vb3n",
                                "vb10", "vb11", "pol1", "vb20", "q5b", "q3c",
                                    "q11n", "q12", "q12c", "q12bn",
                                    "a4", "soct2", "wf1", "cct1b", "pn4",
                                "for5", "drk1", "env1c", "env2b", "vic1ext",
                                "vic1exta", "aoj11", "dst1b", "iarea1", "iarea2",
                                "iarea3", "iarea4", "iarea6", "iarea7", "sd2new2",
                                "sd3new2", "sd6new2", "ccq1", "ccq2", "ccq3",
                                "ccq4", "q14", "www1", "r3", "r4", "r4a", "r5",
                                "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
                                "r16")]
```

▼ Code

```
uruguay_subset <- uruguay_17[, c("pais", "uniq_id",
                                    "ls3",
                                    "q10g", "q10new", "q10d", "q10e",
                                    "q2", "q1", "sexi", "inf1", "etid",
                                "colorr", "colori", "vb1", "vb2",
                                    "ed", "ed2", "conocim", "vac1", "ocup1a",
                                "ocup4a",
                                    "cp7", "cp8", "cp13", "cp20", "prot3",
                                "cp6", "q5a",
                                    "it1", "idio2", "w14a", "e5", "d1",
                                "d2", "d3", "d4", "d5", "d6", "lib1", "lib2b",
                                "lib2c", "lib4", "exc2", "exc6", "exc20",
                                "exc11", "exc13", "exc14", "exc15", "exc16",
                                "exc18", "exc7new", "aoj12", "b1", "b2", "b3",
                                "b4", "b6", "b43", "b12", "b13", "b18", "b21",
                                "b21a", "b32", "b37", "b47a", "infrax",
                                "infra3", "mil10a", "mil10e", "mil10oas",
                                "mil10un", "gi0", "l1", "jc10", "jc13",
                                "jc15a", "m1", "m2", "ros1", "ros4", "ing4",
                                "eff1", "eff2", "media3", "media4", "vb3n",
                                "vb10", "vb11", "pol1", "vb20", "q5b", "q3c",
                                    "q11n", "q12", "q12c", "q12bn",
                                    "a4", "soct2", "wf1", "cct1b", "pn4",
                                "for5", "drk1", "env1c", "env2b", "vic1ext",
                                "vic1exta", "aoj11", "dst1b", "iarea1",
                                "iarea2", "iarea3", "iarea4", "iarea6",
                                "iarea7", "sd2new2", "sd3new2", "sd6new2",
                                "ccq1", "ccq2", "ccq3", "ccq4", "q14", "www1",
```

```
                                  "r3", "r4", "r4a", "r5", "r6", "r7", "r8",
                                  "r12", "r14", "r15", "r18", "r1", "r16")]
```

▼ Code

```
brazil_subset <- brazil_17[, c("pais", "uniq_id",
                                       "ls3",
                                       "q10g", "q10new", "q10d", "q10e",
                                       "q2", "q1", "sexi", "inf1", "etid",
                                "colorr", "colori", "vb1", "vb2",
                                       "ed", "ed2", "conocim", "vac1", "ocup1a",
                                "ocup4a",
                                       "np1", "cp7", "cp8", "cp13", "cp20",
                                "prot3", "cp6", "q5a",
                                       "it1", "idio2", "w14a", "e5", "d1",
                                "d2", "d3", "d4", "d5", "d6", "lib1", "lib2b",
                                "lib2c", "lib4", "exc2", "exc6", "exc20", "exc11",
                                "exc13", "exc14", "exc15", "exc16", "exc18",
                                "exc7new", "aoj12", "b1", "b2", "b3", "b4", "b6",
                                "b43", "b12", "b13", "b18", "b21", "b21a", "b32",
                                "b37", "b47a", "infrax", "infra3", "mil10a",
                                "mil10e", "mil10oas", "mil10un", "gi0", "l1",
                                "jc10", "jc13", "jc15a", "m1", "m2", "ros1",
                                "ros4", "ing4", "eff1", "eff2", "aoj22new",
                                "media3", "media4", "vb3n", "vb10", "vb11",
                                "pol1", "vb20", "q5b", "q3c",
                                       "q11n", "q12", "q12m", "q12f", "q12c",
                                "q12bn",
                                       "a4", "soct2", "wf1", "cct1b", "pn4",
                                "for5", "drk1", "env1c", "env2b", "vic1ext",
                                "vic1exta", "aoj11", "dst1b", "iarea1", "iarea2",
                                "iarea3", "iarea4", "iarea6", "iarea7", "sd2new2",
                                "sd3new2", "sd6new2", "ccq1", "ccq2", "ccq3",
                                "ccq4", "q14", "www1", "r3", "r4", "r4a", "r5",
                                "r6", "r7", "r8", "r12", "r14", "r15", "r18",
                                "r1", "r16")]
```

▼ Code

```
venezuela_subset <- venezuela_16_17[, c("pais", "uniq_id",
                                       "ls3",
                                       "q10g", "q10new", "q10d", "q10e",
                                       "q2y", "q2", "q1", "sexi", "inf1",
                                       "etid", "colorr", "colori", "vb1", "vb2",
                                       "ed", "ed2", "conocim", "ocup1a",
                                       "ocup4a",
                                       "np1", "cp7", "cp8", "cp13", "cp20",
                                       "prot3", "cp6", "q5a",
                                       "it1", "idio2", "w14a", "e5", "d1",
                                       "d2", "d3", "d4", "d5", "d6", "lib1",
                                       "lib2b", "lib2c", "lib4", "exc2", "exc6",
                                       "exc20", "exc11", "exc13", "exc14",
                                       "exc15", "exc16", "exc18", "exc7new",
                                       "aoj12", "b1", "b2", "b3", "b4", "b6",
```

```
                                    "b43", "b12", "b13", "b18", "b21",
                                    "b21a", "b32", "b37", "b47a", "infrax",
                                    "infra3", "mil10a", "mil10e", "mil10oas",
                                    "mil10un", "gi0", "l1", "jc10", "jc13",
                                    "jc15a", "m1", "m2", "ros1", "ros4",
                                    "ing4", "eff1", "eff2", "aoj22new",
                                    "media3", "media4", "vb3n", "vb10",
                                    "vb11", "pol1", "vb20", "q5b", "q3c",
                                    "q11n", "q12", "q12m", "q12f", "q12c",
                                    "q12bn",
                                    "a4", "soct2", "wf1", "cct1b", "pn4",
                                    "for5", "drk1", "env1c", "env2b",
                                    "vic1ext", "vic1exta", "aoj11", "dst1b",
                                    "iarea1", "iarea2", "iarea3", "iarea4",
                                    "iarea6", "iarea7", "sd2new2", "sd3new2",
                                    "sd6new2", "q14", "www1", "r3", "r4",
                                    "r4a", "r5", "r6", "r7", "r8", "r12",
                                    "r14", "r15", "r18", "r1", "r16")]
```

▼ Code

```
argentina_subset <- argentina_17[, c("pais", "uniq_id",
                                    "ls3",
                                    "q10g", "q10new", "q10d", "q10e",
                                    "q2", "q1", "sexi", "inf1", "etid",
                                   "colorr", "colori", "vb1", "vb2",
                                    "ed", "ed2", "conocim", "vac1",
                                   "ocup1a", "ocup4a",
                                    "cp7", "cp8", "cp13", "cp20", "prot3",
                                   "cp6", "q5a",
                                    "it1", "idio2", "w14a", "e5", "d1",
                                   "d2", "d3", "d4", "d5", "d6", "lib1",
                                   "lib2b", "lib2c", "lib4", "exc2", "exc6",
                                   "exc20", "exc11", "exc13", "exc14", "exc15",
                                   "exc16", "exc18", "exc7new", "aoj12", "b1",
                                   "b2", "b3", "b4", "b6", "b43", "b12", "b13",
                                   "b18", "b21", "b21a", "b32", "b37", "b47a",
                                   "infrax", "infra3", "mil10a", "mil10e",
                                   "mil10oas", "mil10un", "gi0", "l1", "jc10",
                                   "jc13", "jc15a", "m1", "m2", "ros1", "ros4",
                                   "ing4", "eff1", "eff2", "media3", "media4",
                                   "vb3n", "vb10", "vb11", "pol1", "vb20",
                                   "q5b", "q3c",
                                    "q11n", "q12", "q12c", "q12bn",
                                    "a4", "soct2", "wf1", "cct1b", "pn4",
                                   "for5", "drk1", "env1c", "env2b", "vic1ext",
                                   "vic1exta", "aoj11", "dst1b", "iarea1",
                                   "iarea2", "iarea3", "iarea4", "iarea6",
                                   "iarea7", "sd2new2", "sd3new2", "sd6new2",
                                   "ccq1", "ccq2", "ccq3", "ccq4", "q14",
                                   "www1", "r3", "r4", "r4a", "r5", "r6", "r7",
                                   "r8", "r12", "r14", "r15", "r18", "r1",
                                   "r16")]
```

Appending Income Columns

The following codes first add the local currency per country, replacing the current dataset range from 0 to 16 with specific currency values. Each dataset has its own individual questionnaire where the local currencies are displayed. Next, the exchange rate is applied to convert the local currency into USD, creating the column income_usd. Finally, the purchasing power parity (PPP) is used to standardize all currencies, resulting in the column income_ppp. A conversion rate will be utilized for this purpose.

The data is sourced from the World Bank, using the 2017 exchange rate to US dollars and purchasing power in market value. Venezuela will be the exception, using the exchange rate data available up to 2014 and PPP data available up to 2011.

▼ Code

```
# Define income ranges in Colombian Peso (COP)
income_ranges_colombia <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 225000, 275000, 375000, 485000, 582500, 640000, 680000,
                     725000, 795000, 910000, 1090000, 1250000, 1450000, 1800000,
                     2625000, 3250000)
)

# Define exchange rate for Colombia in 2017 (1 COP to 0.00026 USD)
exchange_rate_colombia_2017 <- 1 / 2951.49

# Define 2017 PPP conversion factor for Colombia (to market value)
ppp_conversion_factor_colombia_2017 <- 0.4

# Convert local currency to USD using the exchange rate
income_ranges_colombia$income_usd <- income_ranges_colombia$local_currency *
  exchange_rate_colombia_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_colombia$income_ppp <- income_ranges_colombia$income_usd *
  ppp_conversion_factor_colombia_2017

# Add the converted income data to the colombia_subset dataset
colombia_income <- colombia_subset %>%
  left_join(income_ranges_colombia, by = "q10g")
```

▼ Code

```
# Define income ranges in Ecuadorian Dollar (USD)
income_ranges_ecuador <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 135, 167, 225, 272, 305, 325, 347, 377, 402, 425, 477,
                     537, 615, 720, 890, 1000)
)

# Define exchange rate for Ecuador (1 USD to 1 USD)
exchange_rate_ecuador_2017 <- 1
```

```
# Define 2017 PPP conversion factor for Ecuador (to market value)
ppp_conversion_factor_ecuador_2017 <- 0.5

# Convert local currency to USD using the exchange rate
income_ranges_ecuador$income_usd <- income_ranges_ecuador$local_currency *
  exchange_rate_ecuador_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_ecuador$income_ppp <- income_ranges_ecuador$income_usd *
  ppp_conversion_factor_ecuador_2017

# Add the converted income data to the ecuador_subset dataset
ecuador_income <- ecuador_subset %>%
  left_join(income_ranges_ecuador, by = "q10g")
```

▼ Code

```
# Define income ranges in Peruvian Sol (PEN)
income_ranges_peru <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 250, 300, 425, 550, 650, 775, 900, 1000, 1075, 1150,
                     1250, 1400, 1650, 1975, 2425, 2700)
)

# Define 2017 exchange rate for Peru (1 PEN to USD)
exchange_rate_peru_2017 <- 1 / 3.26

# Define 2017 PPP conversion factor for Peru (to market value)
ppp_conversion_factor_peru_2017 <- 0.5

# Convert local currency to USD using the 2017 exchange rate
income_ranges_peru$income_usd <- income_ranges_peru$local_currency *
  exchange_rate_peru_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_peru$income_ppp <- income_ranges_peru$income_usd *
  ppp_conversion_factor_peru_2017

# Add the converted income data to the peru_subset dataset
peru_income <- peru_subset %>%
  left_join(income_ranges_peru, by = "q10g")
```

▼ Code

```
# Define income ranges in Bolivian Boliviano (Bs.)
income_ranges_bolivia <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 750, 875, 1125, 1375, 1600, 1825, 2025, 2225, 2500,
                     2825, 3175, 3475, 4025, 4900, 6040, 6850)
)

# Define 2017 exchange rate for Bolivia (1 BOB to USD)
exchange_rate_bolivia_2017 <- 1 / 6.91
```

```
# Define 2017 PPP conversion factor for Bolivia (to market value)
ppp_conversion_factor_bolivia_2017 <- 0.4

# Convert local currency to USD using the 2017 exchange rate
income_ranges_bolivia$income_usd <- income_ranges_bolivia$local_currency *
  exchange_rate_bolivia_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_bolivia$income_ppp <- income_ranges_bolivia$income_usd *
  ppp_conversion_factor_bolivia_2017

# Add the converted income data to the bolivia_subset dataset
bolivia_income <- bolivia_subset %>%
  left_join(income_ranges_bolivia, by = "q10g")
```

▼ Code

```
# Define income ranges in Paraguayan Guaraní (PYG)
income_ranges_paraguay <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 320465, 462203, 758010, 1026565, 1232083, 1477348,
                     1710873, 1811680, 1995370, 2184856, 2406856, 2671786,
                     3058063, 3577318, 4639026, 6000000)
)

# Define 2017 exchange rate for Paraguay (1 PYG to USD)
exchange_rate_paraguay_2017 <- 1 / 5618.93

# Define 2017 PPP conversion factor for Paraguay (to market value)
ppp_conversion_factor_paraguay_2017 <- 0.5

# Convert local currency to USD using the 2017 exchange rate
income_ranges_paraguay$income_usd <- income_ranges_paraguay$local_currency *
  exchange_rate_paraguay_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_paraguay$income_ppp <- income_ranges_paraguay$income_usd *
  ppp_conversion_factor_paraguay_2017

# Add the converted income data to the paraguay_subset dataset
paraguay_income <- paraguay_subset %>%
  left_join(income_ranges_paraguay, by = "q10g")
```

▼ Code

```
# Define income ranges in Chilean Peso (CLP)
income_ranges_chile <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 142600, 167700, 207400, 235000, 268500, 315000, 360000,
                     400500, 439500, 485500, 564000, 612500, 701500, 835000,
                     1018000, 1118000)
)

# Define 2017 exchange rate for Chile (1 CLP to USD)
```

```r
exchange_rate_chile_2017 <- 1 / 649.33

# Define 2017 PPP conversion factor for Chile (to market value)
ppp_conversion_factor_chile_2017 <- 0.6

# Convert local currency to USD using the 2017 exchange rate
income_ranges_chile$income_usd <- income_ranges_chile$local_currency *
  exchange_rate_chile_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_chile$income_ppp <- income_ranges_chile$income_usd *
  ppp_conversion_factor_chile_2017

# Add the converted income data to the chile_subset dataset
chile_income <- chile_subset %>%
  left_join(income_ranges_chile, by = "q10g")
```

▼ Code

```r
# Define income ranges in Uruguayan Pesos (UYU)
income_ranges_uruguay <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 8000, 9500, 12250, 14975, 17575, 20100, 22650, 25250,
                     28350, 32000, 36250, 40750, 48000, 63000, 86500, 100000)
)

# Define 2017 exchange rate for Uruguay (1 UYU to USD)
exchange_rate_uruguay_2017 <- 1 / 28.68

# Define 2017 PPP conversion factor for Uruguay (to market value)
ppp_conversion_factor_uruguay_2017 <- 0.8

# Convert local currency to USD using the 2017 exchange rate
income_ranges_uruguay$income_usd <- income_ranges_uruguay$local_currency *
  exchange_rate_uruguay_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_uruguay$income_ppp <- income_ranges_uruguay$income_usd *
  ppp_conversion_factor_uruguay_2017

# Add the converted income data to the uruguay_subset dataset
uruguay_income <- uruguay_subset %>%
  left_join(income_ranges_uruguay, by = "q10g")
```

▼ Code

```r
# Define income ranges and their midpoints in Brazilian Real (BRL)
income_ranges_brazil <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 700, 825, 1000, 1125, 1275, 1425, 1625, 1850, 2050,
                     2250, 2450, 2850, 3475, 4375, 5825, 6700)
)

# Define exchange rate for Brazil in 2017 (1 BRL to 1/3.19 USD)
```

```r
exchange_rate_brazil_2017 <- 1 / 3.19

# Define 2017 PPP conversion factor for Brazil (0.7)
ppp_conversion_factor_brazil_2017 <- 0.7

# Convert local currency to USD using the exchange rate
income_ranges_brazil$income_usd <- income_ranges_brazil$local_currency *
  exchange_rate_brazil_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_brazil$income_ppp <- income_ranges_brazil$income_usd *
  ppp_conversion_factor_brazil_2017

# Add the converted income data to the brazil_subset dataset
brazil_income <- brazil_subset %>%
  left_join(income_ranges_brazil, by = "q10g")
```

▼ Code

```r
# Define income ranges in Venezuelan Bolívares (1 VES to USD)
income_ranges_venezuela <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 22100, 23225, 25500, 28150, 31100, 33975, 36750, 40125,
                     43700, 47650, 51050, 55500, 61300, 72575, 87125, 96600)
)

# Define exchange rate for Venezuela in 2014 (1 VES to 1/6.28 USD)
exchange_rate_venezuela_2014 <- 1 / 6.28

# Define 2011 PPP conversion factor for Venezuela (0.6)
ppp_conversion_factor_venezuela_2011 <- 0.6

# Convert local currency to USD using the exchange rate
income_ranges_venezuela$income_usd <- income_ranges_venezuela$local_currency *
  exchange_rate_venezuela_2014

# Convert local currency to USD using the 2011 PPP conversion factor
income_ranges_venezuela$income_ppp <- income_ranges_venezuela$income_usd *
  ppp_conversion_factor_venezuela_2011

# Add the converted income data to the venezuela_subset dataset
venezuela_income <- venezuela_subset %>%
  left_join(income_ranges_venezuela, by = "q10g")
```

▼ Code

```r
# Define income ranges and their midpoints in Argentine Pesos (ARS)
income_ranges_argentina <- data.frame(
  q10g = 0:16,
  local_currency = c(0, 6000, 7000, 9000, 11000, 13000, 15000, 17000, 19000,
                     21000,
                     23000, 25000, 27000, 29000, 32000, 37000, 40000)
)
```

```
# Define exchange rate for Argentina in 2017 (1 ARS to 1/16.56 USD)
exchange_rate_argentina_2017 <- 1 / 16.56

# Define 2017 PPP conversion factor for Argentina (0.6)
ppp_conversion_factor_argentina_2017 <- 0.6

# Convert local currency to USD using the exchange rate
income_ranges_argentina$income_usd <- income_ranges_argentina$local_currency *
  exchange_rate_argentina_2017

# Convert local currency to USD using the 2017 PPP conversion factor
income_ranges_argentina$income_ppp <- income_ranges_argentina$income_usd *
  ppp_conversion_factor_argentina_2017

# Add the converted income data to the argentina_subset dataset
argentina_income <- argentina_subset %>%
  left_join(income_ranges_argentina, by = "q10g")
```

Factor Creation Similar Variables

This section consolidates variables that are similar into single factors. For instance, the core questionnaire contains multiple questions with similar codes that address the same subject. These variables will be combined into one column to reduce the total number of variables, thereby improving the efficiency and runtime of the models used in the analyses.

Hidden in the HTML (First, a list of all variables to be combined is created. The code is presented in comment form but still added in the R Markdown to identify which variables/columns are missing from each country. Just like the missing columns code in the 'Subset Columns' section.)

Below is the code for creating factors per country, customized to fit the specific columns available in each country's dataset.

▼ Code

```
# Rename income file so it stays intact
colombia_factor <- colombia_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b12", "b13", "b18", "b21", "b21a", "b32",
```

```
        "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2", "aoj22new"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  colombia_factor <- create_composite(colombia_factor, factor,
                                      composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
ecuador_factor <- ecuador_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc13", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2", "aoj22new"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)
```

```
# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  ecuador_factor <- create_composite(ecuador_factor, factor,
                                     composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
bolivia_factor <- bolivia_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc13", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2", "aoj22new"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  bolivia_factor <- create_composite(bolivia_factor, factor,
                                     composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
peru_factor <- peru_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
```

```
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e"),
  JC = c("jc10", "jc13", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2", "aoj22new"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  peru_factor <- create_composite(peru_factor, factor,
                                  composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
paraguay_factor <- paraguay_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
```

```
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc13", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2", "aoj22new"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  paraguay_factor <- create_composite(paraguay_factor, factor,
                                      composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
chile_factor <- chile_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc13", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
```

```
  chile_factor <- create_composite(chile_factor, factor,
                                   composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
uruguay_factor <- uruguay_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc13", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  uruguay_factor <- create_composite(uruguay_factor, factor,
                                     composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
brazil_factor <- brazil_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
```

```
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc13", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2", "aoj22new"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  brazil_factor <- create_composite(brazil_factor, factor,
                                    composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
venezuela_factor <- venezuela_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc13", "jc15a"),
```

```
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2", "aoj22new"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  venezuela_factor <- create_composite(venezuela_factor, factor,
                                       composite_factors[[factor]])
}
```

▼ Code

```
# Rename income file so it stays intact
argentina_factor <- argentina_income

# Define the function using vectorized operations
create_composite <- function(data, new_col, cols) {
  data %>%
    mutate(!!sym(new_col) := round(rowMeans(dplyr::select(., all_of(cols)),
                                            na.rm = TRUE), 2))
}

# List of columns to be combined for each composite factor
composite_factors <- list(
  CP = c("cp7", "cp8", "cp13"),
  D = c("d1", "d2", "d3", "d4", "d5", "d6"),
  LIB = c("lib1", "lib2b", "lib2c", "lib4"),
  EXC = c("exc2", "exc6", "exc20", "exc11", "exc13", "exc14", "exc15", "exc16",
          "exc18"),
  B = c("b1", "b2", "b3", "b4", "b6", "b43", "b12", "b13", "b18", "b21", "b21a",
        "b32", "b37", "b47a"),
  INFRA = c("infra3", "infrax"),
  MIL10 = c("mil10a", "mil10e", "mil10oas", "mil10un"),
  JC = c("jc10", "jc13", "jc15a"),
  M = c("m1", "m2"),
  ROS = c("ros1", "ros4"),
  POL = c("ing4", "eff1", "eff2"),
  MEDIA = c("media3", "media4"),
  IAREA = c("iarea1", "iarea2", "iarea3", "iarea4", "iarea6", "iarea7"),
  SDNEW2 = c("sd2new2", "sd3new2", "sd6new2"),
  R = c("r3", "r4", "r4a", "r6", "r7", "r8", "r12", "r14", "r15", "r18", "r1",
        "r16")
)

# Apply the function for each composite factor
for (factor in names(composite_factors)) {
  argentina_factor <- create_composite(argentina_factor, factor,
```

```
                                        composite_factors[[factor]])
}
```

Combine Country Datasets

Before merging all datasets into one, missing columns from some countries are reintroduced with 'Na' as the input. This is done to prevent an error message from the code when combining all datasets, as these columns/variables do not exist in those specific countries but are present in other South American countries.

▼ Code

```
all_columns <- c("pais", "uniq_id",
                 "ls3",
                 "q10g", "q10new", "q10d", "q10e", "local_currency",
                 "income_usd",
                 "income_ppp",
                 "q2", "q1", "sexi", "inf1", "etid", "vb1", "vb2",
                 "ed", "ed2", "conocim", "ocup1a", "ocup4a",
                 "np1", "CP", "cp20", "cp6", "q5a",
                 "idio2", "w14a", "e5", "D", "LIB", "EXC", "exc7new", "it1",
                 "aoj12", "B", "INFRA", "MIL10", "gi0", "l1", "JC", "M", "ROS",
                 "POL", "aoj22new", "MEDIA", "vb3n", "vb10", "vb11", "pol1",
                 "vb20", "q5b", "q3c",
                 "q11n", "q12", "q12m", "q12f", "q12c", "q12bn",
                 "a4", "soct2", "wf1", "cct1b", "pn4", "for5", "drk1", "env1c",
                 "env2b", "vic1ext", "vic1exta", "aoj11", "dst1b", "IAREA",
                 "SDNEW2", "ccq1", "ccq2", "ccq3", "ccq4", "q14", "www1", "R")

add_missing_columns <- function(data, all_columns) {
  missing_columns <- setdiff(all_columns, names(data))
  for (col in missing_columns) {
    data[[col]] <- NA
  }
  return(data)
}

colombia_factor_2 <- add_missing_columns(colombia_factor, all_columns)
ecuador_factor_2 <- add_missing_columns(ecuador_factor, all_columns)
peru_factor_2 <- add_missing_columns(peru_factor, all_columns)
bolivia_factor_2 <- add_missing_columns(bolivia_factor, all_columns)
paraguay_factor_2 <- add_missing_columns(paraguay_factor, all_columns)
chile_factor_2 <- add_missing_columns(chile_factor, all_columns)
uruguay_factor_2 <- add_missing_columns(uruguay_factor, all_columns)
brazil_factor_2 <- add_missing_columns(brazil_factor, all_columns)
venezuela_factor_2 <- add_missing_columns(venezuela_factor, all_columns)
argentina_factor_2 <- add_missing_columns(argentina_factor, all_columns)
```

The following code will concatenate all rows to create a unified South American dataset. Prior steps ensured that all necessary columns are present in each individual dataset, thereby preventing errors during the row concatenation process.

▼ Code

```
south_america_dataset <- bind_rows(
  colombia_factor_2,
  ecuador_factor_2,
  peru_factor_2,
  bolivia_factor_2,
  paraguay_factor_2,
  chile_factor_2,
  uruguay_factor_2,
  brazil_factor_2,
  venezuela_factor_2,
  argentina_factor_2
)
```

Checking Missing Values

Before commencing the analyses, a check for missing values will be conducted on the combined dataset. This step is essential to optimize performance in the machine learning models.

▼ Code

```
# Check the number of missing values per column in the combined dataset
missing_values_per_column <- sapply(south_america_dataset,
                                    function(x) sum(is.na(x)))

# Convert the missing values information into a data frame
missing_values_df <- data.frame(
  Column = names(missing_values_per_column),
  MissingValues = missing_values_per_column
)

# Print the data frame
print(missing_values_df)
```

```
                     Column MissingValues
pais                   pais             0
uniq_id             uniq_id             0
ls3                     ls3           412
q10g                   q10g          7286
q10new               q10new          2102
q10d                   q10d           383
q10e                   q10e           217
q2y                     q2y         10544
q2                       q2             9
q1                       q1             1
sexi                   sexi            24
inf1                   inf1          2654
etid                   etid           856
colorr               colorr            24
colori               colori            24
vb2                     vb2           259
ed                       ed           216
```

ed2	ed2	2393
conocim	conocim	24
ocup1a	ocup1a	8021
ocup4a	ocup4a	77
np1	np1	6432
cp7	cp7	240
cp8	cp8	153
cp13	cp13	159
cp20	cp20	8380
prot3	prot3	25
cp6	cp6	174
q5a	q5a	124
it1	it1	428
idio2	idio2	139
w14a	w14a	795
d1	d1	483
d2	d2	376
d3	d3	488
d4	d4	443
d5	d5	406
d6	d6	352
lib1	lib1	538
lib2b	lib2b	394
lib2c	lib2c	416
lib4	lib4	408
exc2	exc2	46
exc6	exc6	50
exc20	exc20	50
exc11	exc11	11555
exc13	exc13	7118
exc14	exc14	14650
exc15	exc15	7562
exc16	exc16	9949
exc18	exc18	232
exc7new	exc7new	1383
aoj12	aoj12	333
b1	b1	393
b2	b2	400
b3	b3	413
b4	b4	351
b6	b6	491
b12	b12	379
b13	b13	514
b18	b18	157
b21	b21	317
b21a	b21a	221
b32	b32	213
b37	b37	207
b47a	b47a	302
infrax	infrax	1240
infra3	infra3	1001
mil10a	mil10a	8684
mil10e	mil10e	5306
mil10oas	mil10oas	9034

mil10un	mil10un	7859
gi0	gi0	74
l1	l1	1764
jc10	jc10	7995
jc15a	jc15a	1254
m1	m1	263
m2	m2	565
ros1	ros1	530
ros4	ros4	336
ing4	ing4	554
eff1	eff1	344
eff2	eff2	402
aoj22new	aoj22new	4796
media3	media3	366
media4	media4	1522
vb3n	vb3n	6615
vb10	vb10	157
vb11	vb11	12977
pol1	pol1	104
vb20	vb20	1343
q5b	q5b	159
q3c	q3c	510
q11n	q11n	27
q12	q12	37
q12m	q12m	9184
q12f	q12f	9182
q12c	q12c	41
q12bn	q12bn	44
a4	a4	612
soct2	soct2	296
wf1	wf1	72
cct1b	cct1b	1824
pn4	pn4	597
for5	for5	1634
drk1	drk1	441
env1c	env1c	189
env2b	env2b	262
vic1ext	vic1ext	33
vic1exta	vic1exta	12125
aoj11	aoj11	282
dst1b	dst1b	255
iarea1	iarea1	24
iarea2	iarea2	24
iarea3	iarea3	24
iarea4	iarea4	24
iarea6	iarea6	24
iarea7	iarea7	24
sd2new2	sd2new2	240
sd3new2	sd3new2	859
sd6new2	sd6new2	535
q14	q14	173
www1	www1	72
r3	r3	57
r4	r4	58

r4a	r4a	48
r5	r5	84
r6	r6	59
r7	r7	72
r8	r8	60
r12	r12	44
r14	r14	32
r15	r15	64
r18	r18	64
r1	r1	44
r16	r16	812
local_currency	local_currency	7286
income_usd	income_usd	7286
income_ppp	income_ppp	7286
CP	CP	15
D	D	59
LIB	LIB	108
EXC	EXC	11
B	B	12
INFRA	INFRA	103
MIL10	MIL10	3047
JC	JC	302
M	M	133
ROS	ROS	173
POL	POL	56
MEDIA	MEDIA	246
IAREA	IAREA	24
SDNEW2	SDNEW2	53
R	R	15
vb1	vb1	3255
e5	e5	1805
ccq1	ccq1	7254
ccq2	ccq2	9938
ccq3	ccq3	7968
ccq4	ccq4	11784
b43	b43	1653
jc13	jc13	9643
vac1	vac1	10081

For the first analysis, no changes will be made to the combined dataset to avoid introducing bias into the analysis outcomes. The income variable will only be used to correlate with the outcome variable, life satisfaction. It appears that there are sufficient filled values to proceed with the analysis. If necessary, a code will be implemented later to remove all missing values specifically for this analysis, but this is not necessary during the pre-processing stage.

For the second analysis, an imputation method will be employed to ensure the proper functioning of machine learning models and to handle any missing variables. This code will be incorporated into the analysis itself. The goal is to use the optimal imputation method to minimize bias.

Subsetting Combined Dataset

The combined dataset will be subsetted to include only the variables intended for the analyses. This step is necessary to exclude older variables from which factors were derived, which are still present in the combined dataset.

▼ Code

```
south_america_dataset_subset <- south_america_dataset[, c("pais", "uniq_id",
                "ls3",
                "q10g", "q10new", "q10d", "q10e", "local_currency",
                "income_usd",
                "income_ppp",
                "q2", "q1", "sexi", "inf1", "etid", "vb1", "vb2",
                "ed", "ed2", "conocim", "ocup1a", "ocup4a",
                "np1", "CP", "cp20", "cp6", "q5a",
                "idio2", "w14a", "e5", "D", "LIB", "EXC", "exc7new", "it1",
                "aoj12", "B", "INFRA", "MIL10", "gi0", "l1", "JC", "M", "ROS",
                "POL", "aoj22new", "MEDIA", "vb3n", "vb10", "vb11", "pol1",
                "vb20", "q5b", "q3c",
                "q11n", "q12", "q12m", "q12f", "q12c", "q12bn",
                "a4", "soct2", "wf1", "cct1b", "pn4", "for5", "drk1", "env1c",
                "env2b", "vic1ext", "vic1exta", "aoj11", "dst1b", "IAREA",
                "SDNEW2", "ccq1", "ccq2", "ccq3", "ccq4", "q14", "www1", "R")]
```

Analysis 1: Examining the Influence of Income on Life Satisfaction

The first analysis focuses on the income variables derived from the q10g question in the questionnaire, which asks:

“Can you tell me into which of the following ranges fits the income you personally earn each month in your work or retirement or pension, without taking into account the income of other members of the home?”

The three factors calculated from the q10g question are local currency, income in USD, and income in PPP (USD).

The objective of this analysis is to determine if income significantly influences the outcome variable, life satisfaction. Additionally, it will assess whether a higher income correlates with increased overall life satisfaction for individuals per country or if the influence is minimal.

Calculating Means

First, the mean values of the three variables (ls3, income_usd, income_ppp) to be used in this analysis will be calculated for each country. This allows for a comparison of the mean values across the ten countries.

The income datasets specific to each country (e.g., colombia_income) will be used for these calculations. These datasets have undergone the least amount of pre-processing, making the code execution more robust and less prone to errors.

▼ Code

```r
# List of data files with their corresponding country names
data_files <- list(
  Colombia = colombia_income,
  Ecuador = ecuador_income,
  Peru = peru_income,
  Bolivia = bolivia_income,
  Paraguay = paraguay_income,
  Chile = chile_income,
  Uruguay = uruguay_income,
  Brazil = brazil_income,
  Venezuela = venezuela_income,
  Argentina = argentina_income
)

# Calculate means for each data file and store in a list
mean_values <- lapply(data_files, function(df) {
  df %>%
    summarise(
      mean_ls3 = mean(ls3, na.rm = TRUE),
      mean_income_usd = mean(income_usd, na.rm = TRUE),
      mean_income_ppp = mean(income_ppp, na.rm = TRUE)
    )
})

# Combine the results into a single data frame with a 'Country' column
mean_values_df <- bind_rows(mean_values, .id = "Country")
```

K-Means Clustering

K-means clustering is employed here to identify patterns and groupings based on life satisfaction and income variables across the analyzed countries. This approach allows for identifying similarities among countries and determining which income variable—USD or PPP—is more effective for clustering. The selected variable will be used in the subsequent analyses.

Clusters are formed based on similarities in life satisfaction and income, with different values of K used initially (starting with 3 clusters and then moving to 5 clusters). After generating both sets of clusters, the optimal K will be chosen for further analysis. It is important to note that cluster numbers are assigned randomly by default. Therefore, to ensure reproducibility and consistency in the results, seed codes are set at the beginning of the analysis.

K-Means with 3 Clusters

The following code prepares the 3 clusters for income_usd and income_ppp for use in a plot with the number of clusters (K) set to 3. The cluster assignments are then added to the data frame for further analysis and visualization.

▼ Code

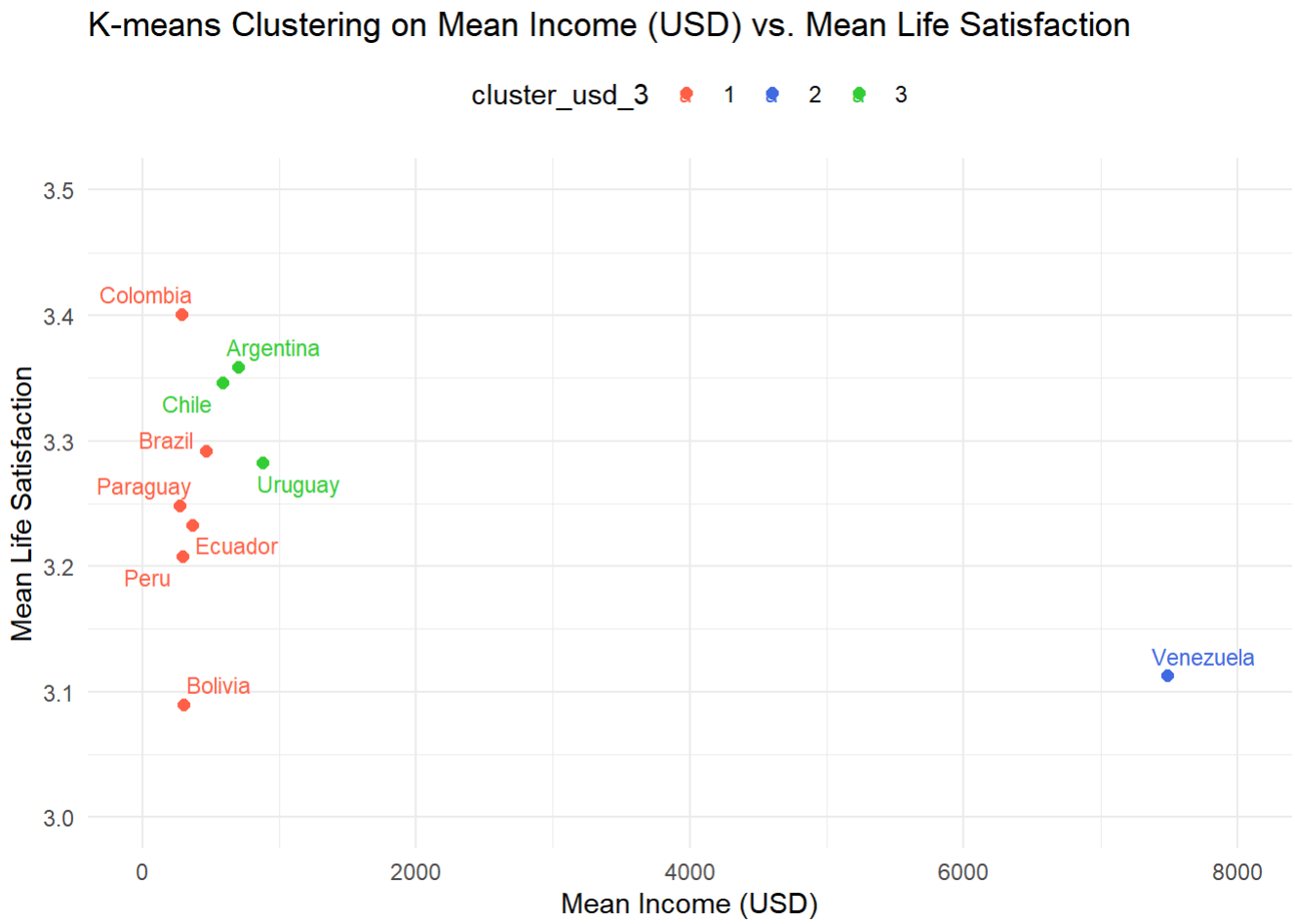
```r
set.seed(123)

# Perform k-means clustering on ls3 and income_usd
```

```
kmeans_usd_3 <- kmeans(mean_values_df[, c("mean_ls3", "mean_income_usd")],
                       centers = 3)

# Add cluster assignments to the data frame
mean_values_df <- mean_values_df %>%
  mutate(cluster_usd_3 = factor(kmeans_usd_3$cluster))

# Perform k-means clustering on ls3 and income_ppp
kmeans_ppp_3 <- kmeans(mean_values_df[, c("mean_ls3", "mean_income_ppp")],
                       centers = 3)

# Add cluster assignments to the data frame
mean_values_df <- mean_values_df %>%
  mutate(cluster_ppp_3 = factor(kmeans_ppp_3$cluster))
```

The code below generates a plot of the 3 clusters derived from the k-means algorithm using ggplot. The plot is adjusted with x/y limits to ensure all countries fit within the frame and the text repel function is employed to prevent overlapping country names.

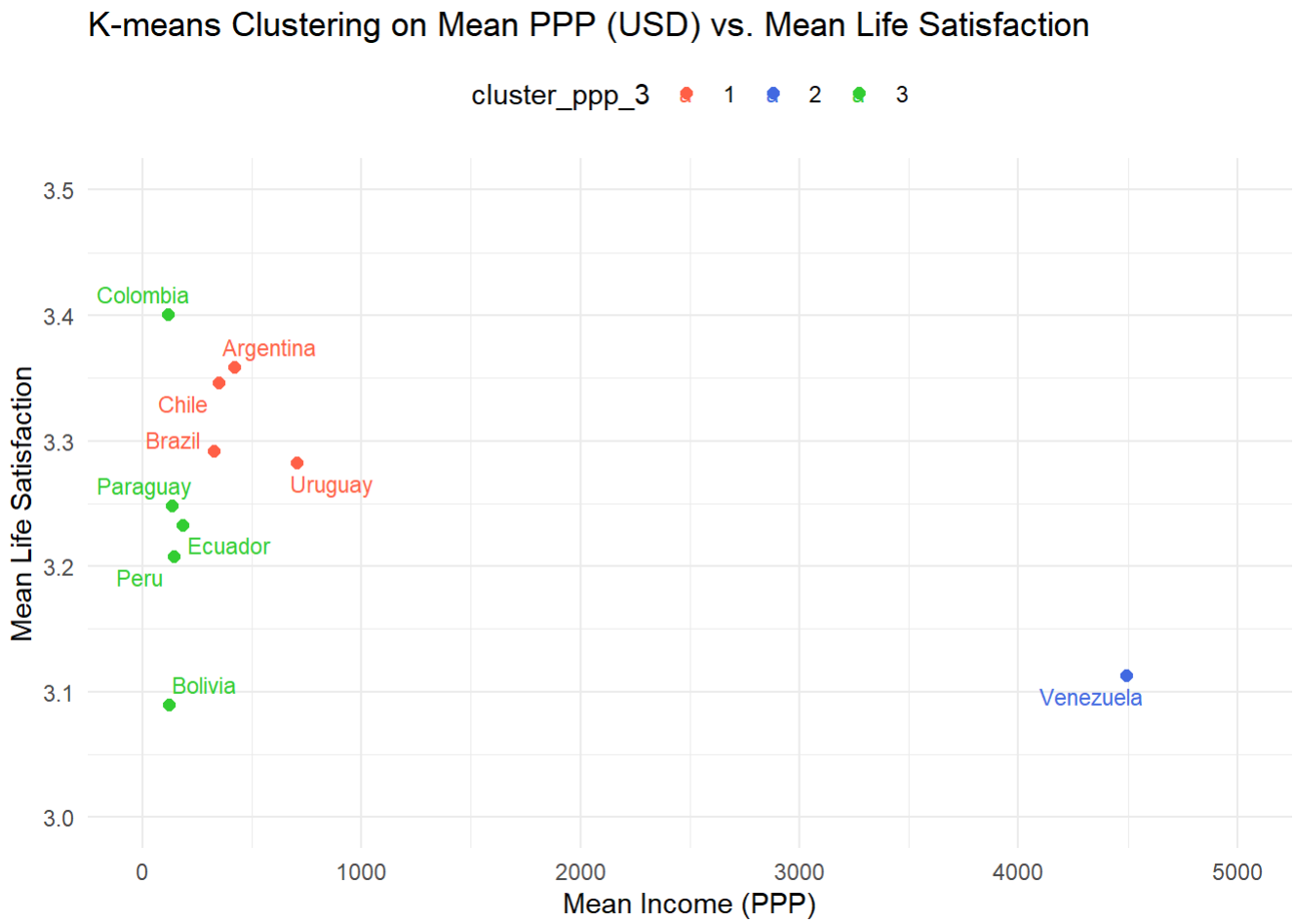
▼ Code

```
# Plot the clusters for USD with k = 3
plot_6 <- ggplot(mean_values_df, aes(x = mean_income_usd, y = mean_ls3,
                                     label = Country, color = cluster_usd_3)) +
  geom_point(size = 2) +
  geom_text_repel(size = 3) +  # Adjust the size parameter as needed
  scale_color_manual(values = c("tomato", "royalblue", "limegreen")) +
  labs(
    title = "K-means Clustering on Mean Income (USD) vs. Mean Life Satisfaction",
    x = "Mean Income (USD)",
    y = "Mean Life Satisfaction"
  ) +
  xlim(0, 8000) +  # Extend x-axis range
  ylim(3, 3.5) +  # Extend y-axis range
  theme_minimal() +
  theme(legend.position = "top")

print(plot_6)
```



▼ Code

```
# Plot the clusters for PPP with k =3
plot_7 <- ggplot(mean_values_df, aes(x = mean_income_ppp, y = mean_ls3,
                                     label = Country, color = cluster_ppp_3)) +
  geom_point(size = 2) +
  geom_text_repel(size = 3) +  # Adjust the size parameter as needed
  scale_color_manual(values = c("tomato", "royalblue", "limegreen")) +
  labs(
    title = "K-means Clustering on Mean PPP (USD) vs. Mean Life Satisfaction",
    x = "Mean Income (PPP)",
    y = "Mean Life Satisfaction"
  ) +
  xlim(0, 5000) +  # Extend x-axis range
  ylim(3, 3.5) +  # Extend y-axis range
  theme_minimal() +
  theme(legend.position = "top")

print(plot_7)
```



The clusters for income in USD and life satisfaction are as follows:

- Cluster 1: Bolivia, Peru, Ecuador, Paraguay, Brazil, Colombia
- Cluster 2: Venezuela
- Cluster 3: Uruguay, Chile, Argentina

The clusters for PPP in USD and life satisfaction are as follows:

- Cluster 1: Uruguay, Brazil, Chile, Argentina
- Cluster 2: Venezuela
- Cluster 3: Bolivia, Peru, Ecuador, Paraguay, Colombia

Two observations can be made from these results. Firstly, Venezuela stands out as a clear outlier. Secondly, the only difference between the two clustering methods is that Brazil shifts to a different cluster when using PPP instead of the straightforward conversion from local currency to income in USD.

K-means with 5 Clusters

The code below prepares the clusters for income_usd and income_ppp for plotting, with the number of clusters (K) set to 5. The cluster assignments are added to the data frame to facilitate further analysis and visualization.

▼ Code

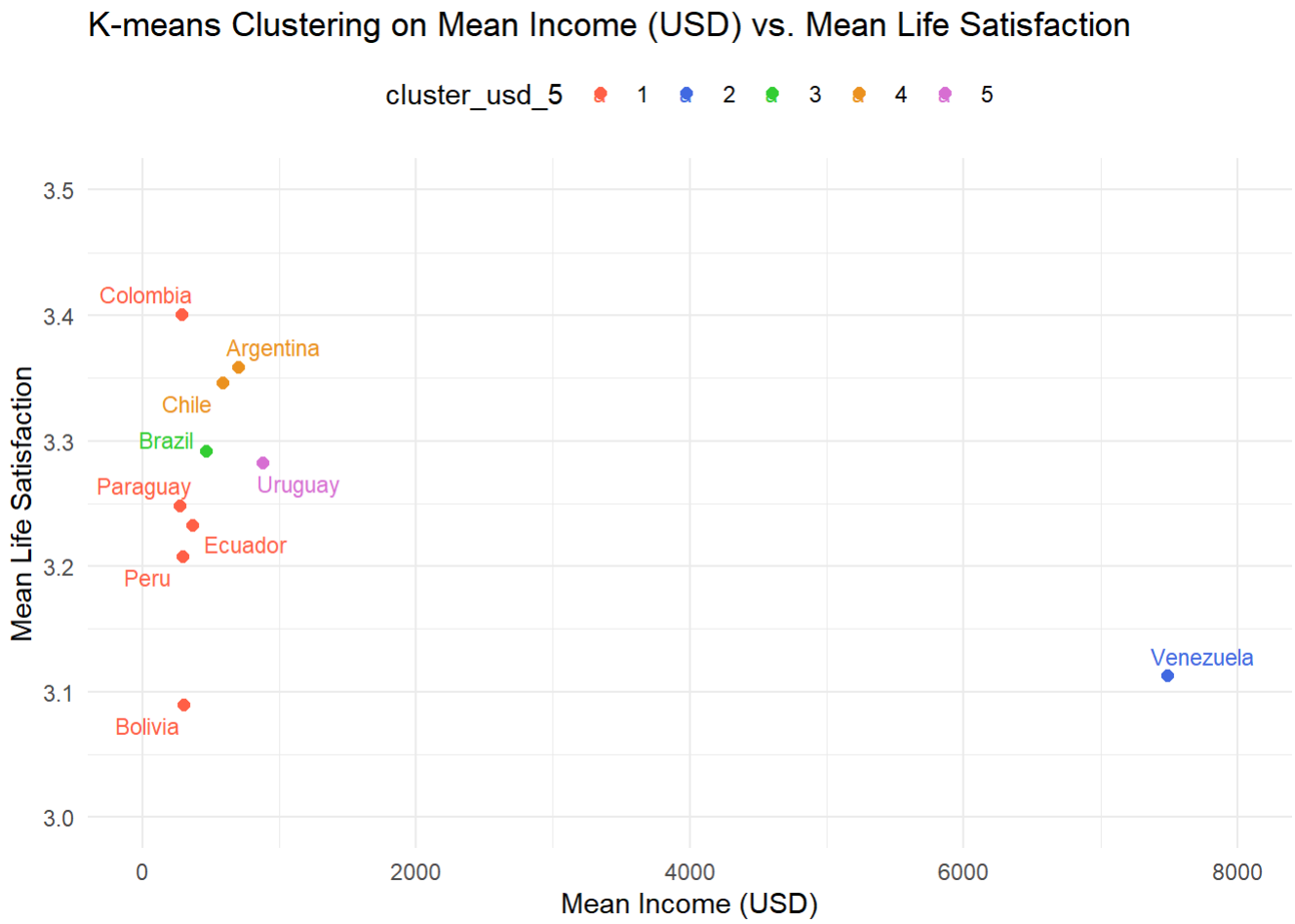
```
set.seed(123)

# Perform k-means clustering on ls3 and income_usd
```

```
kmeans_usd_5 <- kmeans(mean_values_df[, c("mean_ls3", "mean_income_usd")],
                       centers = 5)

# Add cluster assignments to the data frame
mean_values_df <- mean_values_df %>%
  mutate(cluster_usd_5 = factor(kmeans_usd_5$cluster))

# Perform k-means clustering on ls3 and income_ppp
kmeans_ppp_5 <- kmeans(mean_values_df[, c("mean_ls3", "mean_income_ppp")],
                       centers = 5)

# Add cluster assignments to the data frame
mean_values_df <- mean_values_df %>%
  mutate(cluster_ppp_5 = factor(kmeans_ppp_5$cluster))
```

The code below will plot the 5 clusters from the k-means analysis. It ensures that the labels do not overlap and that all countries fit within the plot using adjusted x and y limits.

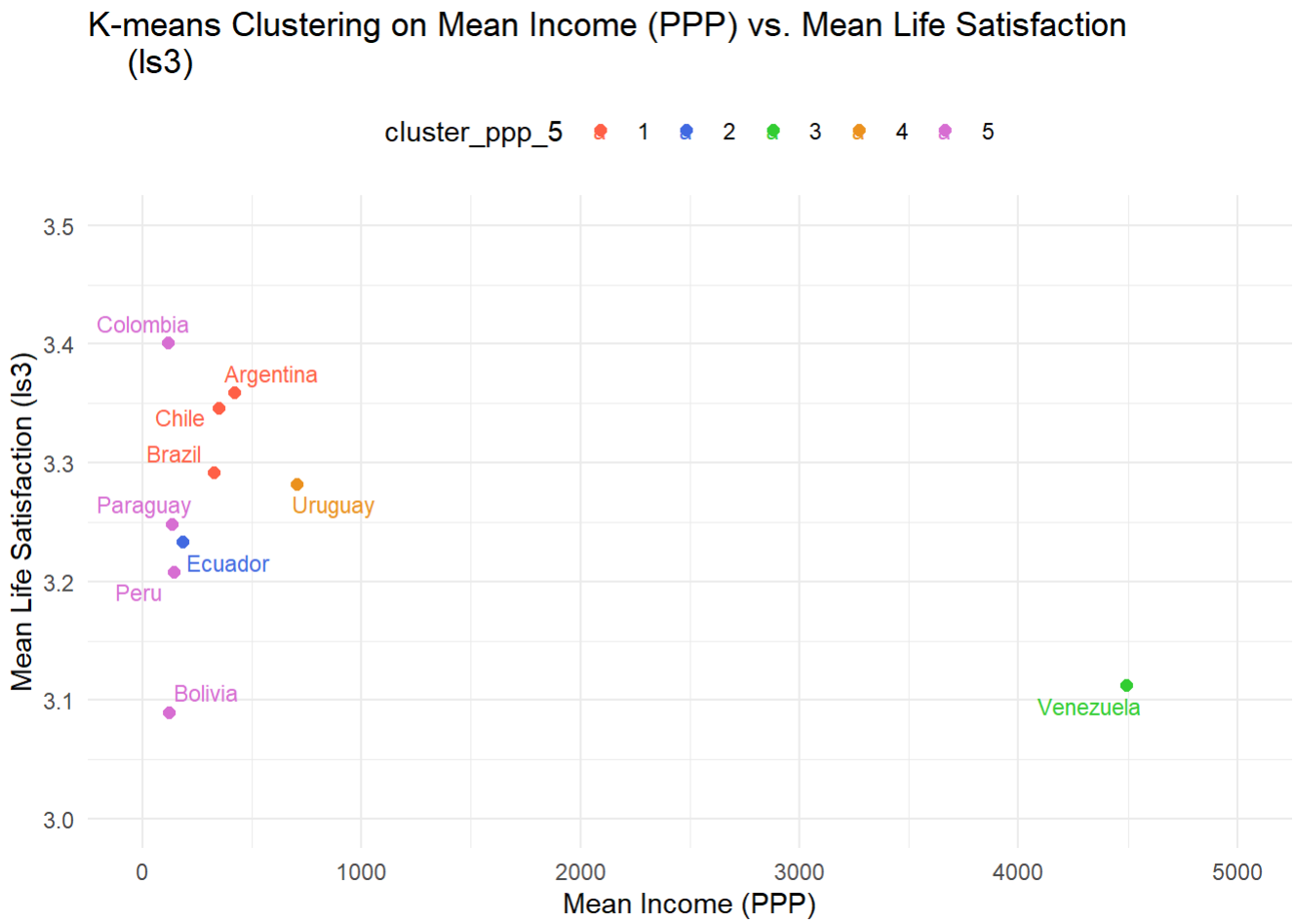
▼ Code

```
# Plot the clusters for USD with k = 5
plot_8 <- ggplot(mean_values_df, aes(x = mean_income_usd, y = mean_ls3,
                                     label = Country, color = cluster_usd_5)) +
  geom_point(size = 2) +
  geom_text_repel(size = 3) +  # Adjust the size parameter as needed
  scale_color_manual(values = c("tomato", "royalblue", "limegreen", "#ED9121",
                                "orchid")) +
  labs(
    title = "K-means Clustering on Mean Income (USD) vs. Mean Life Satisfaction",
    x = "Mean Income (USD)",
    y = "Mean Life Satisfaction"
  ) +
  xlim(0, 8000) +  # Extend x-axis range
  ylim(3, 3.5) +  # Extend y-axis range
  theme_minimal() +
  theme(legend.position = "top")

print(plot_8)
```



▼ Code

```
# Plot the clusters for PPP with k =5
plot_9 <- ggplot(mean_values_df, aes(x = mean_income_ppp, y = mean_ls3,
                                     label = Country, color = cluster_ppp_5)) +
  geom_point(size = 2) +
  geom_text_repel(size = 3) +  # Adjust the size parameter as needed
  scale_color_manual(values = c("tomato", "royalblue", "limegreen", "#ED9121",
                                "orchid")) +
  labs(
    title = "K-means Clustering on Mean Income (PPP) vs. Mean Life Satisfaction
    (ls3)",
    x = "Mean Income (PPP)",
    y = "Mean Life Satisfaction (ls3)"
  ) +
  xlim(0, 5000) +  # Extend x-axis range
  ylim(3, 3.5) +  # Extend y-axis range
  theme_minimal() +
  theme(legend.position = "top")

print(plot_9)
```



For the clusters of life satisfaction with income in USD for k = 5, the clustering is as follows:

- Cluster 1: Bolivia, Peru, Ecuador, Paraguay, Colombia
- Cluster 2: Venezuela
- Cluster 3: Brazil
- Cluster 4: Chile, Argentina
- Cluster 5: Uruguay

For the clusters of life satisfaction with PPP in USD for k = 5, the clustering is as follows:

- Cluster 1: Brazil, Chile, Argentina
- Cluster 2: Ecuador
- Cluster 3: Venezuela
- Cluster 4: Uruguay
- Cluster 5: Bolivia, Peru, Paraguay, Colombia

In this clustering with k = 5, there are more significant changes compared to the clustering with k = 3. Firstly, Venezuela remains a clear outlier. Secondly, only Venezuela and Uruguay stay in the same clusters as before. Brazil moves to the cluster with Chile and Argentina, while Ecuador shifts to its own cluster, leaving the largest cluster of Bolivia, Peru, Paraguay, and Colombia.

K-means Clustering Excluding Outlier

To achieve clearer visualization and trends in the clustering analysis, the outlier Venezuela will be excluded. This will be performed using k = 3 clustering for simplification, considering there are only ten points in the plot.

First, the means will be recalculated for the remaining countries.

▼ Code

```
# Remove Venezuela from the mean data
data_files_no_venezuela <- data_files[names(data_files) != "Venezuela"]

# Calculate means for each data file without Venezuela
mean_values_no_venezuela <- lapply(data_files_no_venezuela, function(df) {
  df %>%
    summarise(
      mean_ls3_2 = mean(ls3, na.rm = TRUE),
      mean_income_usd_2 = mean(income_usd, na.rm = TRUE),
      mean_income_ppp_2 = mean(income_ppp, na.rm = TRUE)
    )
})

# Combine the results into a single data frame with a 'Country' column
mean_values_df_no_venezuela <-
  bind_rows(mean_values_no_venezuela, .id = "Country")
```

Next, the k-means clustering preparation will proceed for plotting, ensuring the data names are distinct from previous k-means analyses to avoid duplicating results from the k=3 clusters.

▼ Code

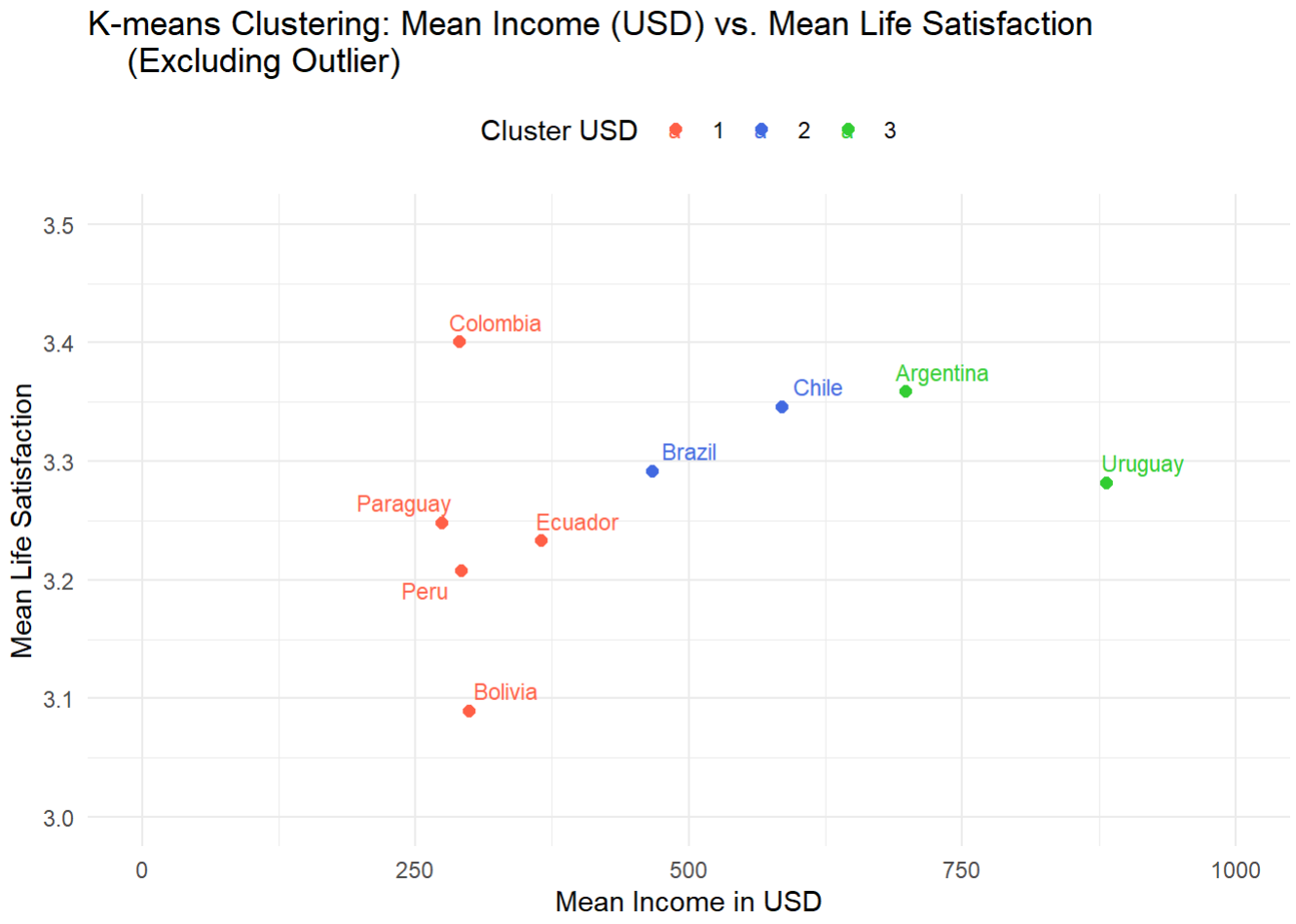
```
set.seed(123)

# Perform k-means clustering on ls3 and income_usd without Venezuela
kmeans_usd_3_no_venezuela <-
  kmeans(mean_values_df_no_venezuela[, c("mean_ls3_2", "mean_income_usd_2")],
                                     centers = 3)

# Add cluster assignments to the data frame
mean_values_df_no_venezuela <- mean_values_df_no_venezuela %>%
  mutate(cluster_usd_3_no_venezuela = factor(kmeans_usd_3_no_venezuela$cluster))

# Perform k-means clustering on ls3 and income_ppp
kmeans_ppp_3_no_venezuela <-
  kmeans(mean_values_df_no_venezuela[, c("mean_ls3_2", "mean_income_ppp_2")],
                                     centers = 3)

# Add cluster assignments to the data frame
mean_values_df_no_venezuela <- mean_values_df_no_venezuela %>%
  mutate(cluster_ppp_3_no_venezuela = factor(kmeans_ppp_3_no_venezuela$cluster))
```

The last step is ensuring the plot can be generated accurately using the updated data frame without Venezuela and the new variable names.

▼ Code

```
# Plot the clusters for USD with k = 3 without Venezuela
plot_1 <- ggplot(mean_values_df_no_venezuela,
```

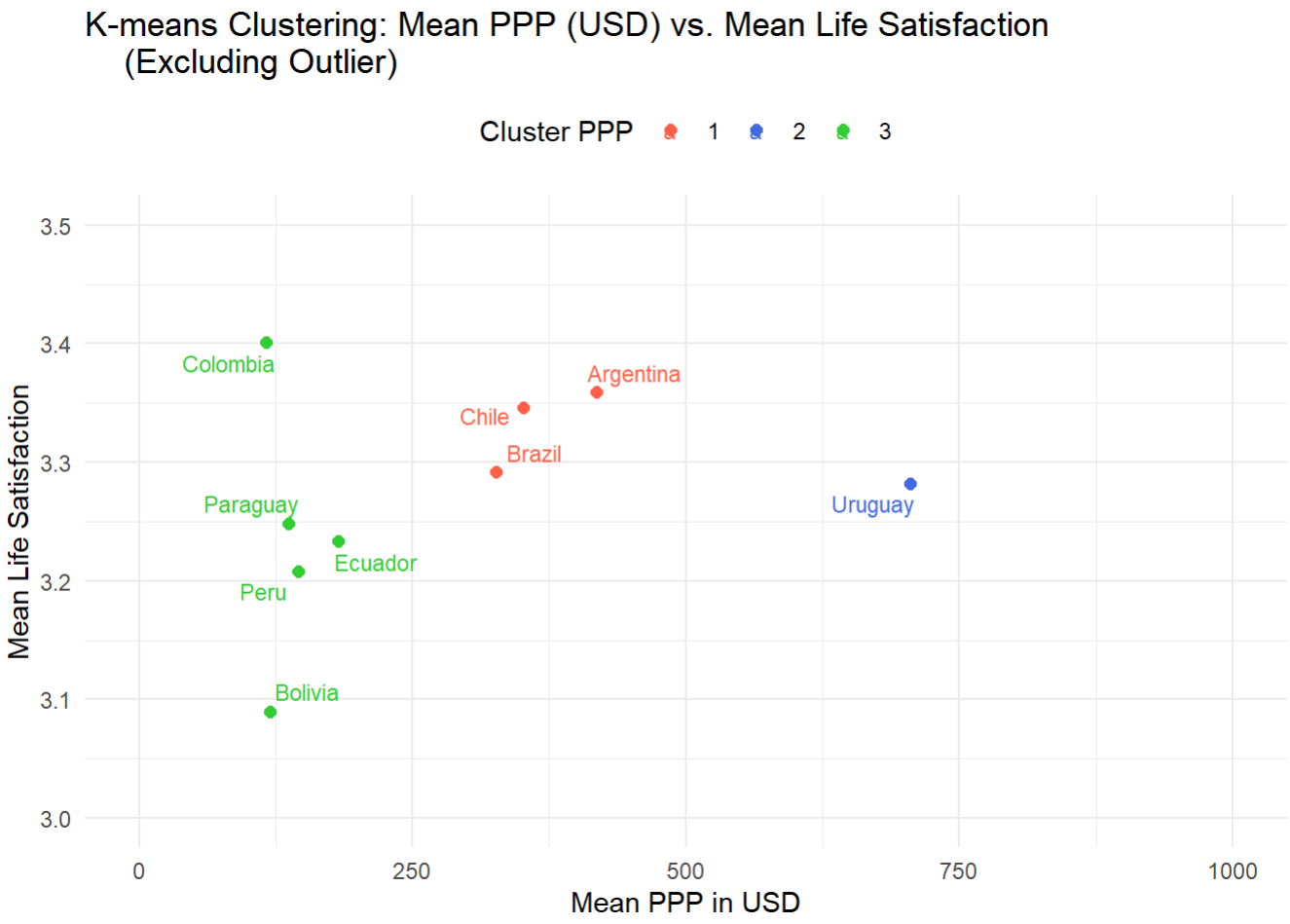
```
                aes(x = mean_income_usd_2,
                    y = mean_ls3_2, label = Country,
                    color = cluster_usd_3_no_venezuela)) +
  geom_point(size = 2) +
  geom_text_repel(size = 3) +  # Adjust the size parameter as needed
  scale_color_manual(values = c("tomato", "royalblue", "limegreen"),
                     name = "Cluster USD") +
  labs(
    title = "K-means Clustering: Mean Income (USD) vs. Mean Life Satisfaction
    (Excluding Outlier)",
    x = "Mean Income in USD",
    y = "Mean Life Satisfaction"
  ) +
  xlim(0, 1000) +  # Extend x-axis range
  ylim(3, 3.5) +  # Extend y-axis range
  theme_minimal() +
  theme(legend.position = "top")

print(plot_1)
```



▼ Code

```
# Plot the clusters for PPP with k = 3 without Venezuela
plot_2 <- ggplot(mean_values_df_no_venezuela,
                aes(x = mean_income_ppp_2,
                    y = mean_ls3_2, label = Country,
                    color = cluster_ppp_3_no_venezuela)) +
  geom_point(size = 2) +
```

```
  geom_text_repel(size = 3) +  # Adjust the size parameter as needed
  scale_color_manual(values = c("tomato", "royalblue", "limegreen"),
                     name = "Cluster PPP") +
  labs(
    title = "K-means Clustering: Mean PPP (USD) vs. Mean Life Satisfaction
    (Excluding Outlier)",
    x = "Mean PPP in USD",
    y = "Mean Life Satisfaction"
  ) +
  xlim(0, 1000) +  # Extend x-axis range
  ylim(3, 3.5) +  # Extend y-axis range
  theme_minimal() +
  theme(legend.position = "top")

print(plot_2)
```



The clusters now exhibit a clearer trend compared to when the outlier (Venezuela) was included in the analysis. It is evident that higher income in USD correlates positively with higher life satisfaction. This relationship holds true for the PPP column as well, despite the overall income levels being lower, the trend remains consistent.

For the income in USD cluster:

- Cluster 1: Bolivia, Peru, Ecuador, Paraguay, Colombia
- Cluster 2: Brazil, Chile
- Cluster 3: Uruguay, Argentina

In the PPP cluster:

- Cluster 1: Brazil, Chile, Argentina
- Cluster 2: Uruguay
- Cluster 3: Bolivia, Peru, Ecuador, Paraguay, Colombia

The notable difference is that Argentina has moved to the Brazil-Chile cluster. Uruguay appears to have the highest income overall. However, the largest cluster with the lost income shows significant variation in life satisfaction, ranging from approximately 3.1 to 3.4 on a scale of 1 to 4. While this difference may seem minor numerically, it is visually significant in the plot.

Trend K-means Clustering Excluding Outlier

This section focuses on analyzing the correlation between life satisfaction and income variables across all countries in the cluster without the outlier. By excluding the outlier, aim to obtain a clearer trend in how income impacts life satisfaction among South American countries. This can be seen as a between-country comparison.

This approach allows for examining the overall relationship between income and life satisfaction, providing insights into the collective impact across the region.

▼ Code

```
# Calculate correlation for USD without Venezuela
correlation_usd <- cor(mean_values_df_no_venezuela$mean_income_usd_2,
                       mean_values_df_no_venezuela$mean_ls3_2)

# Calculate correlation for PPP without Venezuela
correlation_ppp <- cor(mean_values_df_no_venezuela$mean_income_ppp_2,
                       mean_values_df_no_venezuela$mean_ls3_2)

# Create a dataframe with the correlation results of the clustering
correlation_df_cluster <- data.frame(
  Measure = c("Mean Income (USD)", "Mean Income (PPP)"),
  Correlation = c(round(correlation_usd, 3), round(correlation_ppp, 3))
)

# Use kable for a nicely formatted HTML table and adjust column widths
correlation_table_trend <- correlation_df_cluster %>%
  kable(format = "html", escape = FALSE, col.names = c("Measure", "Trend")) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed",
                                      "responsive"),
                full_width = FALSE) %>%
  column_spec(1, width = "10em") %>%
  column_spec(2, width = "10em")

# Print the table for display in the output
correlation_table_trend
```

Measure	Trend
Mean Income (USD)	0.374
Mean Income (PPP)	0.317

The table above displays the correlations between mean income (USD) and mean life satisfaction, as well as mean income (PPP) and mean life satisfaction, resulting from clustering analysis across South American countries, excluding Venezuela. The correlation coefficient between mean income (USD) and mean life satisfaction is 0.374, while the correlation between mean income (PPP) and mean life satisfaction is 0.317. The difference between these correlations is not significant enough to decisively determine which variable will be prioritized in subsequent analyses. Further insights will be gained from the assumptions underlying the k-means clustering.

These correlations illustrate a consistent trend observed across the countries in the cluster, indicating a moderate positive relationship between higher income and higher life satisfaction. This suggests that, generally, as income increases in both USD and PPP terms, life satisfaction tends to rise as well.

Summary Statistics Table with Correlation

Now that the overall trend between countries has been identified, the next step is to examine the correlation of income and life satisfaction within each South American country individually. This within-country analysis aims to provide a more detailed understanding of how income impacts life satisfaction on a country-specific level.

For this analysis, one income variable will be selected, specifically using Purchasing Power Parity (PPP) where available. PPP-adjusted income provides a more comparable measure across countries by equalizing the purchasing power of different currencies.

The output will be presented in the form of a summary statistics table, which will include correlation coefficients between income (measured in PPP) and life satisfaction for each South American country. This approach allows for a focused examination of the relationship between income and life satisfaction within the context of each country, providing insights into variations that may not be apparent in the broader regional analysis.

▼ Code

```
# Define the function to calculate summary statistics and correlations
summary_stats <- function(data) {
  data %>%
    mutate(
      country = case_when(
        pais == 8 ~ "Colombia",
        pais == 9 ~ "Ecuador",
        pais == 10 ~ "Bolivia",
        pais == 11 ~ "Peru",
        pais == 12 ~ "Paraguay",
        pais == 13 ~ "Chile",
        pais == 14 ~ "Uruguay",
        pais == 15 ~ "Brazil",
        pais == 16 ~ "Venezuela",
        pais == 17 ~ "Argentina",
        TRUE ~ as.character(pais)
      )
    ) %>%
    group_by(country) %>%
```

```
    summarise(
      mean_ls3 = mean(ls3, na.rm = TRUE),
      se_ls3 = ifelse(n() > 1, sd(ls3, na.rm = TRUE) / sqrt(n()), NA),
      mean_income_ppp = mean(income_ppp, na.rm = TRUE),
      se_income_ppp = ifelse(n() > 1, sd(income_ppp, na.rm = TRUE) / sqrt(n()),
                             NA),
      corr_income_ppp = cor(ls3, income_ppp, use = "complete.obs")
    )
}
```

The function summary_stats is designed to compute summary statistics and correlations for a given dataset. It first assigns meaningful country names by replacing numeric codes with corresponding country names. Then, it calculates summary statistics including mean life satisfaction (mean_ls3) and mean income adjusted by Purchasing Power Parity (mean_income_ppp). Additionally, it calculates the standard error (se_ls3 and se_income_ppp) where applicable. These standard errors provide insights into the variability and precision of the mean estimates for life satisfaction and income across different countries.

Below, the summary_stats function is applied here to the dataset south_america_dataset_subset, which consolidates data from various South American countries.

▼ Code

```
summary_table <- summary_stats(south_america_dataset_subset)
```

The last code will display the summary statisctics table generated in the previous codes in a formatted HTML table. The ls3 and income_ppp will be added as numeric values with two decimals. Furthermore, the standard error (SE) and the correlation of the PPP in USD with the outcome variable will be added to give a detailed within country analysis of the income variable.

The table is displayed in a nicely formatted HTML format using kable for better readability and presentation.

▼ Code

```
sum_stats_table <- summary_table %>%
  mutate(
    mean_ls3 = sprintf("%.2f", mean_ls3),
    se_ls3 = ifelse(!is.na(se_ls3), sprintf("(%.2f)", se_ls3), "(NA)"),
    mean_income_ppp = sprintf("%.2f", mean_income_ppp),
    se_income_ppp = ifelse(!is.na(se_income_ppp), sprintf("(%.2f)",
                                                          se_income_ppp),
                           "(NA)"),
    corr_income_ppp = sprintf("%.3f", corr_income_ppp)
  ) %>%
  dplyr::select(country, mean_ls3, se_ls3, mean_income_ppp, se_income_ppp,
                corr_income_ppp) %>%
  kable(format = "html", escape = FALSE, col.names = c("Country", "Mean LS",
                                                       "SE", "Mean PPP (USD)",
                                                       "SE", "Corr."),
        align = c("l", "c", "c", "c", "c", "c")) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed")) %>%
  column_spec(1, width = "3em") %>%  # Adjust width for the first column
```

```
  column_spec(1, bold = FALSE) %>%
  column_spec(2:6, bold = FALSE) %>%
  row_spec(0, bold = TRUE)  # Make the header row bold

# Print the table for display in the output
sum_stats_table
```

Country	Mean LS	SE	Mean PPP (USD)	SE	Corr.
Argentina	3.36	(0.02)	418.69	(6.98)	0.130
Bolivia	3.09	(0.02)	119.65	(2.08)	0.119
Brazil	3.29	(0.02)	326.41	(6.49)	0.085
Chile	3.35	(0.02)	351.13	(5.76)	0.097
Colombia	3.40	(0.02)	115.93	(2.14)	0.074
Ecuador	3.23	(0.02)	182.43	(3.06)	0.091
Paraguay	3.25	(0.02)	136.82	(2.68)	0.182
Peru	3.21	(0.02)	145.69	(1.91)	0.104
Uruguay	3.28	(0.02)	705.56	(13.62)	0.202
Venezuela	3.11	(0.03)	4494.41	(65.78)	0.094

The correlations between income (PPP) and life satisfaction (LS) within each country range from 0.074 to 0.202. This indicates that the relationship between income and life satisfaction within individual countries is relatively modest.

- The highest correlation is observed in Uruguay (0.202), suggesting a stronger association between higher income and greater life satisfaction compared to other countries in the region.
- On the lower end, Colombia shows the weakest correlation (0.074), implying a less pronounced link between income and life satisfaction.

This suggests that while income differences between countries may have a more significant impact on life satisfaction, within each country, other factors might play a more substantial role in determining individual life satisfaction levels.

Assumptions Analysis 1

To analyze the assumptions, the focus will be on k-means clustering and the correlations in the summary statistics table.

For clustering, not all assumptions will be examined due to the small number of data points (only ten countries). The primary assumption considered will be the average silhouette width, which will be displayed using a silhouette plot.

In the summary statistics table, the key assumptions involve the classification of the outcome variable used for correlation analysis. This outcome variable is a multiple categorical factor in ordinal order.

Assumptions for K-Means Clustering

The analysis begins with the silhouette score to evaluate the clustering assumptions. The silhouette score interpretation is as follows: a score of 0 or below indicates poor clustering, a score between 0 and 0.5 is acceptable, and a score above 0.5 is considered good.

The clustering analysis adheres to the same sequence: first with k=3, then k=5, and finally k=3 without the outlier.

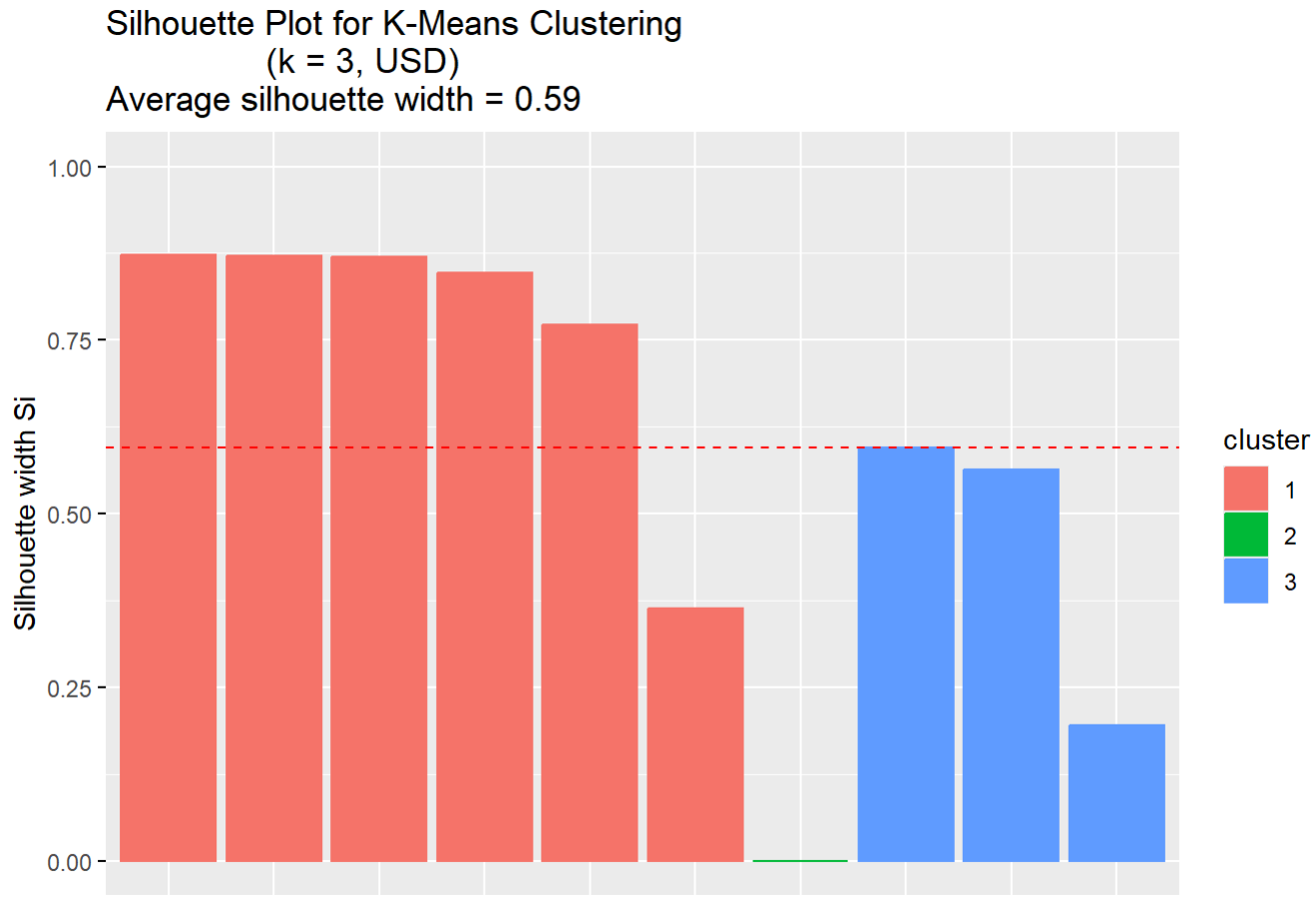
▼ Code

```
# Calculate silhouette scores and average silhouette width
# for k = 3 clusters with USD
sil_usd_3 <-
  silhouette(kmeans_usd_3$cluster, dist(mean_values_df[, c("mean_ls3",
                                                           "mean_income_usd")]))
avg_sil_usd_3 <- mean(sil_usd_3[, 3])
sil_plot_usd_2 <-
  fviz_silhouette(sil_usd_3) +
  ggtitle(paste("Silhouette Plot for K-Means Clustering
                (k = 3, USD)\nAverage silhouette width =",
                round(avg_sil_usd_3, 2)))
```

```
  cluster size ave.sil.width
1       1    6          0.77
2       2    1          0.00
3       3    3          0.45
```

▼ Code

```
print(sil_plot_usd_2)
```



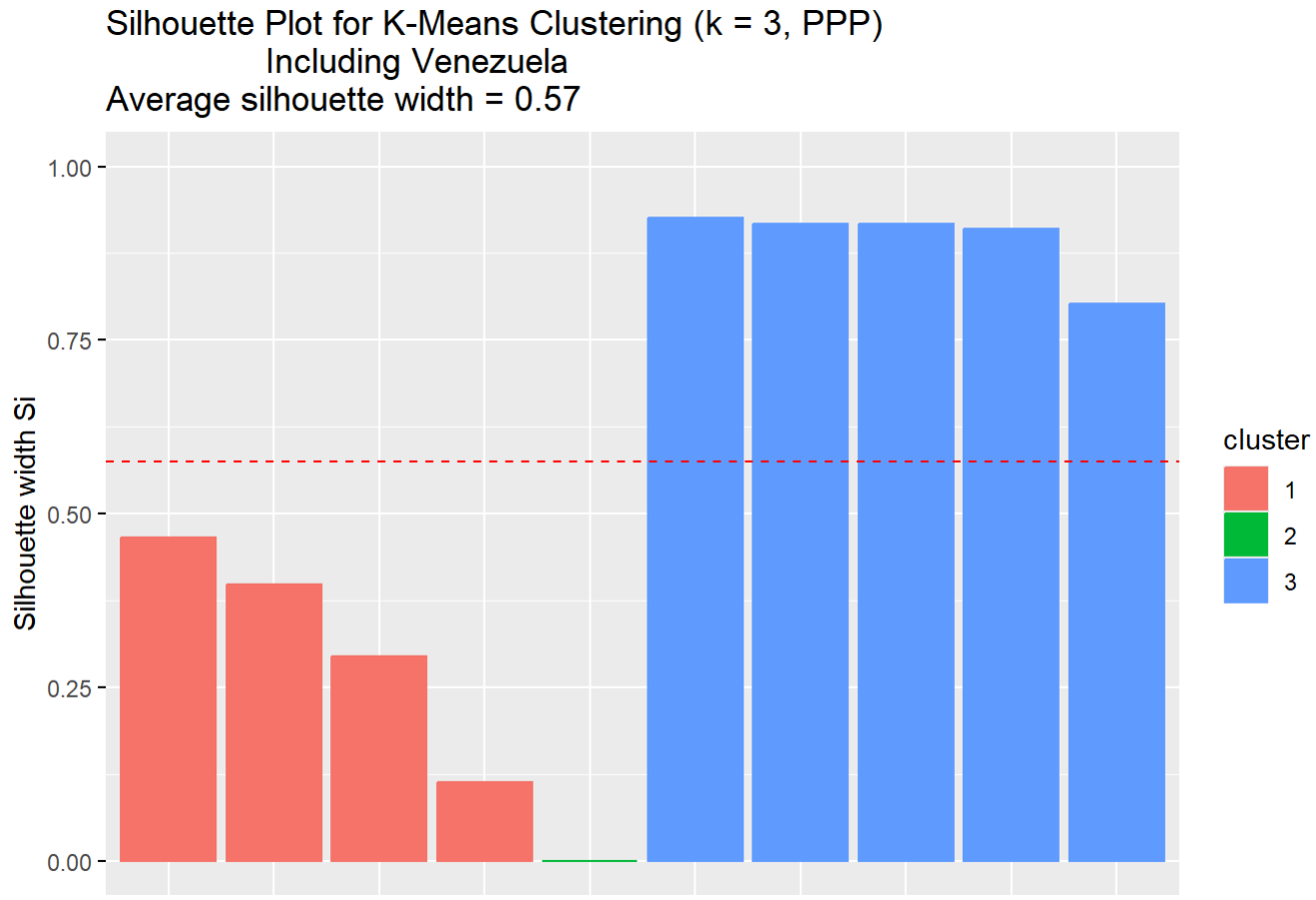
▼ Code

```
# Calculate silhouette scores and average silhouette width
# for k = 3 clusters with PPP
sil_ppp_3 <-
  silhouette(kmeans_ppp_3$cluster,
             dist(mean_values_df[, c("mean_ls3", "mean_income_ppp")]))
avg_sil_ppp_3 <- mean(sil_ppp_3[, 3])
sil_plot_ppp_2 <-
  fviz_silhouette(sil_ppp_3) +
  ggtitle(paste("Silhouette Plot for K-Means Clustering (k = 3, PPP)
                Including Venezuela\nAverage silhouette width =",
                round(avg_sil_ppp_3, 2)))
```

```
  cluster size ave.sil.width
1       1    4          0.32
2       2    1          0.00
3       3    5          0.89
```

▼ Code

```
print(sil_plot_ppp_2)
```



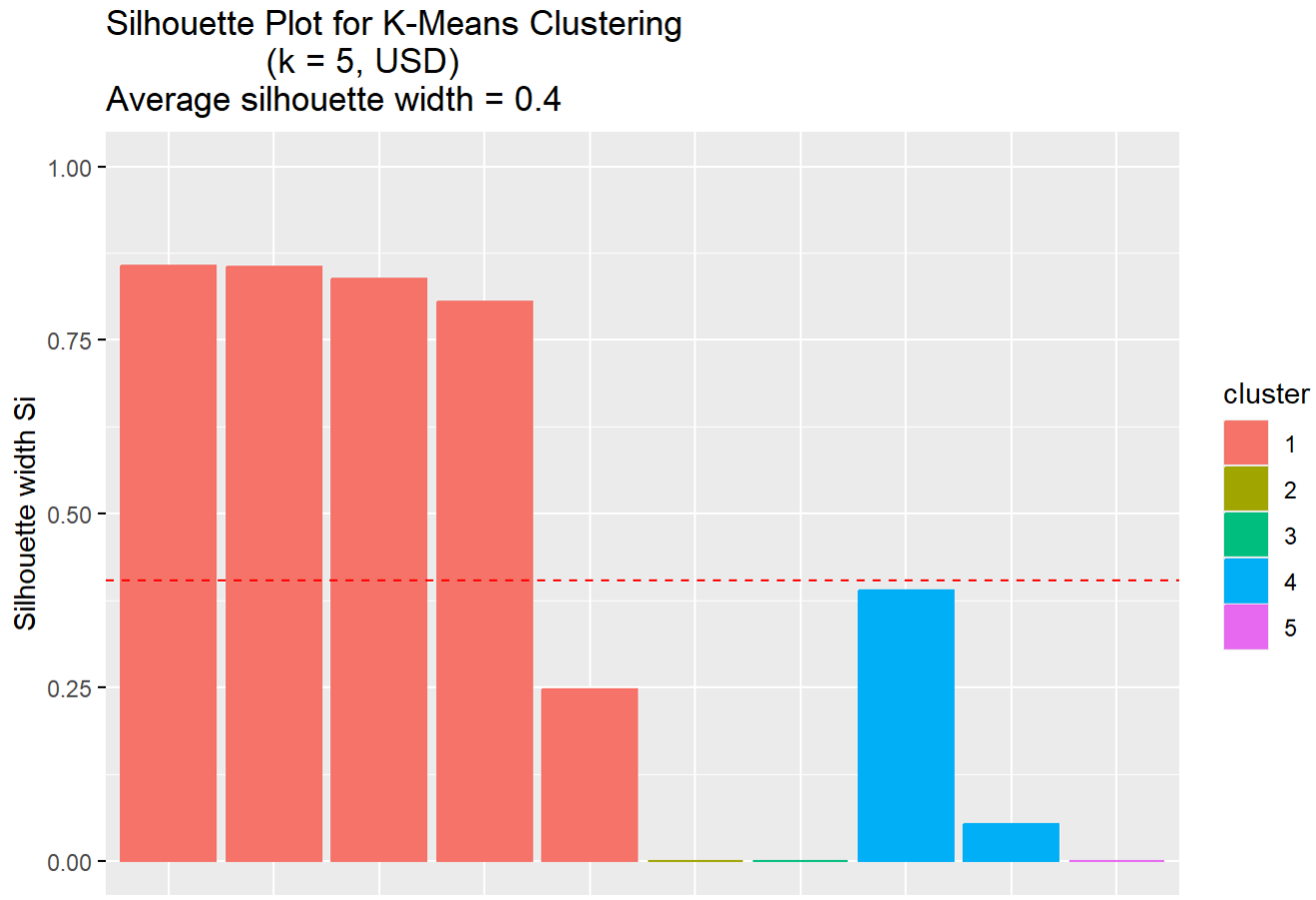
▼ Code

```
# Calculate silhouette scores and average silhouette width
# for k = 5 clusters with USD
sil_usd_5 <-
  silhouette(kmeans_usd_5$cluster,
             dist(mean_values_df[, c("mean_ls3", "mean_income_usd")]))
avg_sil_usd_5 <- mean(sil_usd_5[, 3])
sil_plot_usd_3 <- fviz_silhouette(sil_usd_5) +
  ggtitle(paste("Silhouette Plot for K-Means Clustering
                (k = 5, USD)\nAverage silhouette width =",
                round(avg_sil_usd_5, 2)))
```

```
  cluster size ave.sil.width
1       1    5          0.72
2       2    1          0.00
3       3    1          0.00
4       4    2          0.22
5       5    1          0.00
```

▼ Code

```
print(sil_plot_usd_3)
```



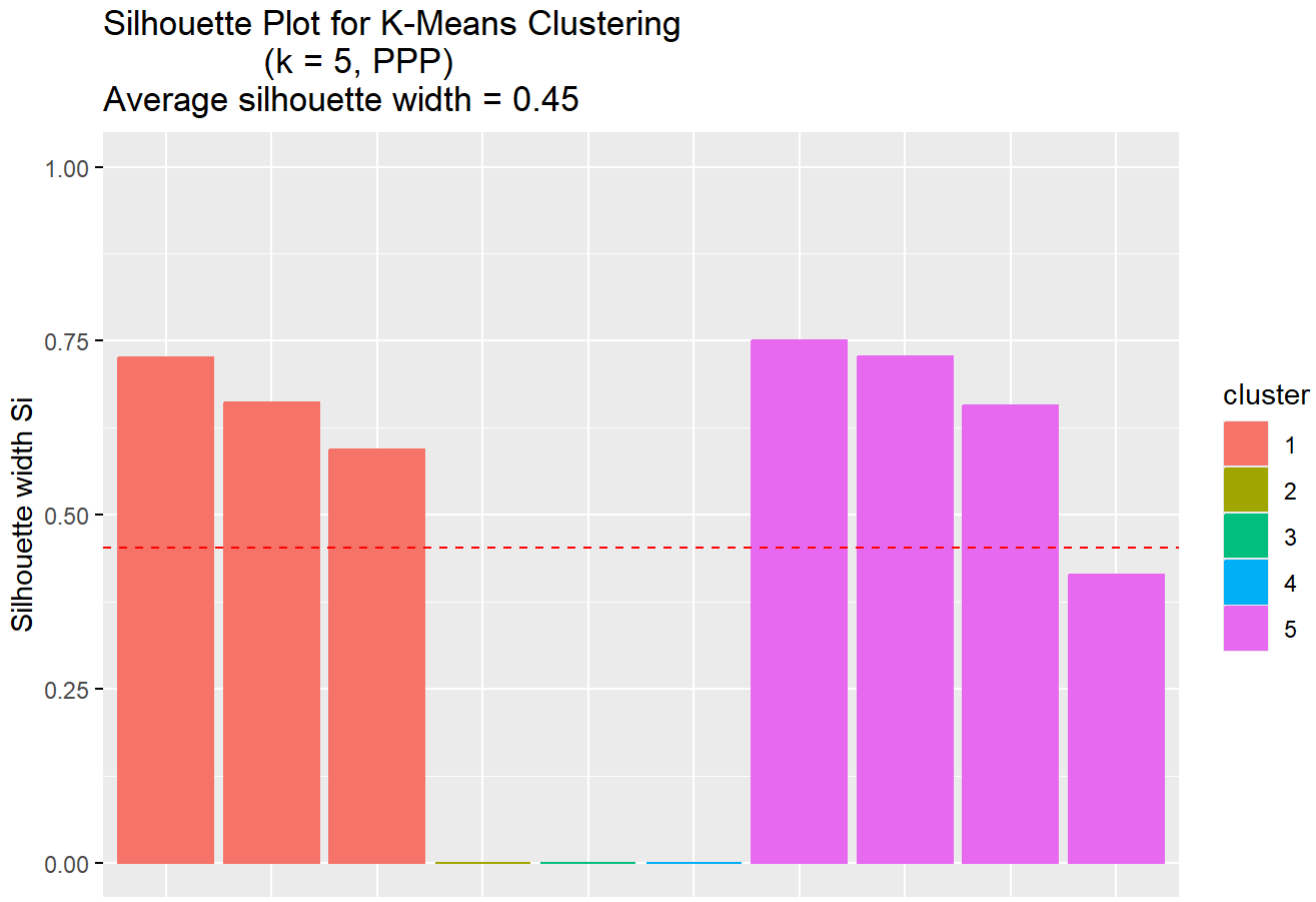
▼ Code

```
# Calculate silhouette scores and average silhouette width
# for k = 5 clusters with PPP
sil_ppp_5 <-
  silhouette(kmeans_ppp_5$cluster,
             dist(mean_values_df[, c("mean_ls3", "mean_income_ppp")]))
avg_sil_ppp_5 <- mean(sil_ppp_5[, 3])
sil_plot_ppp_3 <- fviz_silhouette(sil_ppp_5) +
  ggtitle(paste("Silhouette Plot for K-Means Clustering
                (k = 5, PPP)\nAverage silhouette width =",
                round(avg_sil_ppp_5, 2)))
```

```
  cluster size ave.sil.width
1       1    3          0.66
2       2    1          0.00
3       3    1          0.00
4       4    1          0.00
5       5    4          0.64
```

▼ Code

```
print(sil_plot_ppp_3)
```



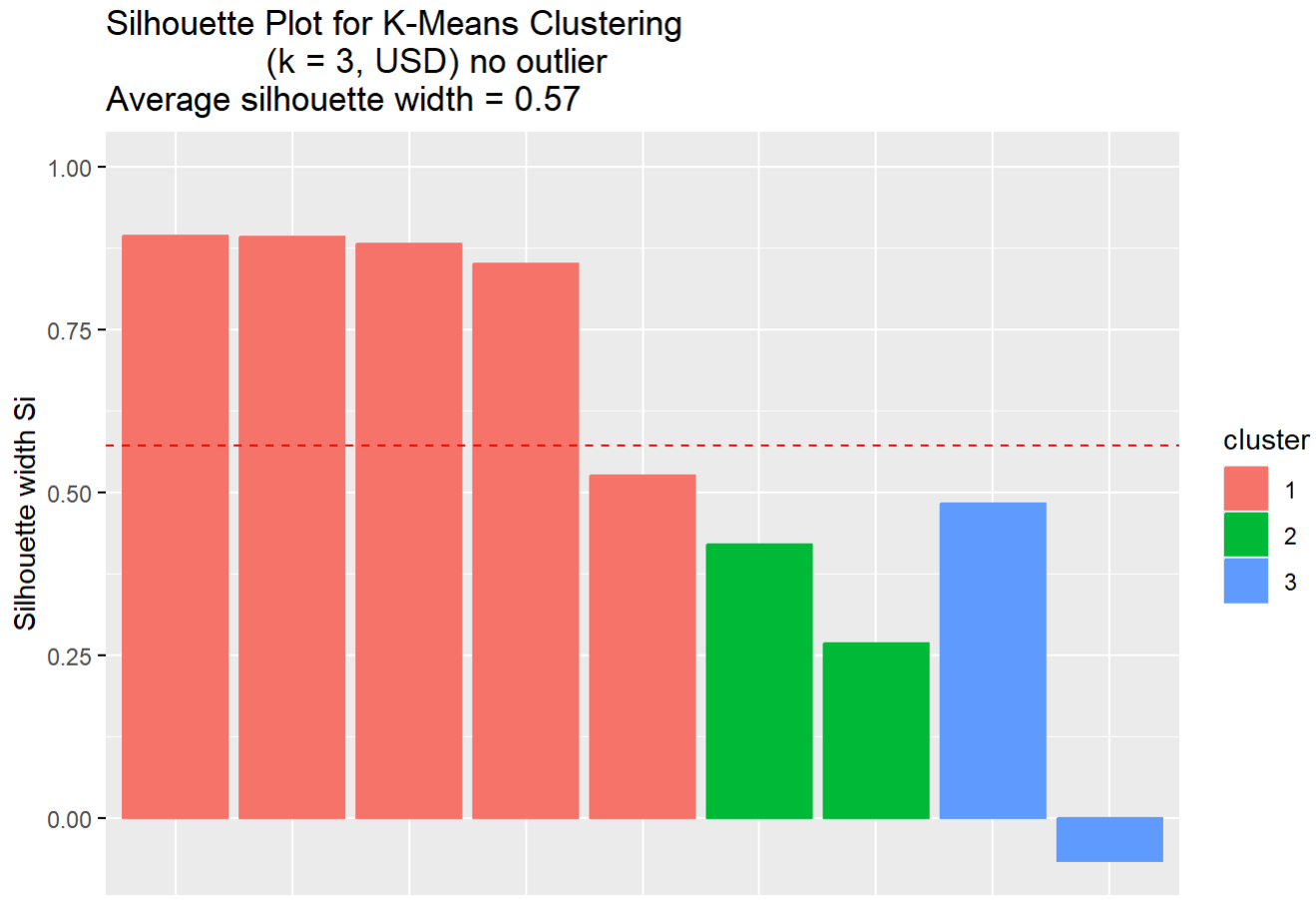
▼ Code

```
# Calculate silhouette scores and average silhouette width
# for k = 3 clusters with USD without Venezuela
sil_usd_3_no_venezuela <-
  silhouette(kmeans_usd_3_no_venezuela$cluster,
             dist(mean_values_df_no_venezuela[, c("mean_ls3_2",
                                                  "mean_income_usd_2")]))
avg_sil_usd_3_no_venezuela <- mean(sil_usd_3_no_venezuela[, 3])
sil_plot_usd <- fviz_silhouette(sil_usd_3_no_venezuela) +
  ggtitle(paste("Silhouette Plot for K-Means Clustering
                (k = 3, USD) no outlier\nAverage silhouette width =",
                round(avg_sil_usd_3_no_venezuela, 2)))
```

```
  cluster size ave.sil.width
1       1    5          0.81
2       2    2          0.34
3       3    2          0.21
```

▼ Code

```
print(sil_plot_usd)
```



▼ Code

```
# Calculate silhouette scores and average silhouette width
# for k = 3 clusters with PPP without Venezuela
sil_ppp_3_no_venezuela <-
  silhouette(kmeans_ppp_3_no_venezuela$cluster,
             dist(mean_values_df_no_venezuela[, c("mean_ls3_2",
                                                  "mean_income_ppp_2")]))
avg_sil_ppp_3_no_venezuela <- mean(sil_ppp_3_no_venezuela[, 3])
sil_plot_ppp <- fviz_silhouette(sil_ppp_3_no_venezuela) +
  ggtitle(paste("Silhouette Plot for K-Means Clustering
                (k = 3, PPP) no outlier\nAverage silhouette width =",
                round(avg_sil_ppp_3_no_venezuela, 2)))
```

```
  cluster size ave.sil.width
1       1    3          0.73
2       2    1          0.00
3       3    5          0.85
```

▼ Code

```
print(sil_plot_ppp)
```

The optimal average silhouette width (ASW) is achieved in the plot without the outlier, yielding scores of approximately 0.57 for USD and 0.72 for PPP. The clustering with k=5 produces the lost scores, around 0.4 for USD and 0.45 for PPP, primarily due to the presence of single points in most clusters, which hinders proper ASW calculation. Consequently, for datasets with fewer points, it is more effective to use a lower number of clusters.

The intermediate result is obtained with k=3, including the outlier, resulting in ASW scores of approximately 0.59 for USD and 0.57 for PPP. While these results are acceptable, the k=3 clustering without the outlier demonstrates superior performance.

In conclusion, k-means clustering with fewer clusters is more appropriate for plots with limited data points. Additionally, removing outliers enhances performance. On average, the PPP variable outperforms the income USD variable, validating its selection for the summary statistics table. Since the ASW performs above 0.5, it can be concluded that the assumptions for clustering are satisfactorily met.

Assumptions for Summary Statistics Table

1. Level of Measurement:

- Ordinal Variables: Ensure the variables are truly ordinal, meaning they have a clear, ordered relationship among categories (e.g., Very Dissatisfied, Dissatisfied, Satisfied, Very Satisfied).

2. Appropriate Measures:

- Summary Statistics: Utilize measures suitable for ordinal data.
- Statistical Test: Spearman's Rank Correlation is appropriate for examining relationships between ordinal variables.

3. Independence:

- Independence of Observations: Each observation within a category should be independent of others. The occurrence of one response should not influence another to avoid biased results.

4. Sample Size:

- Adequate Sample Size: Ensure there are enough observations in each category to support meaningful analysis. Small sample sizes in ordinal categories can lead to unreliable and unstable estimates.

By adhering to these assumptions, the summary statistics table will provide a more accurate and meaningful analysis.

- *1. Level of Measurement*

▼ Code

```
# Create a backup of the original dataset before converting columns
south_america_dataset_backup <- south_america_dataset_subset

# Reverting to factor and ordered factor
south_america_dataset_backup$ls3 <- as.factor(south_america_dataset_backup$ls3)
south_america_dataset_backup$income_usd <-
  as.factor(south_america_dataset_backup$income_usd)
south_america_dataset_backup$income_ppp <-
  as.factor(south_america_dataset_backup$income_ppp)

# Check the structure of the ls3 variable
str(south_america_dataset_backup$ls3)
```

```
Factor w/ 4 levels "1","2","3","4": 2 4 4 4 3 2 4 3 4 3 ...
```

▼ Code

```
# Check the structure of the income_usd variable
str(south_america_dataset_backup$income_usd)
```

```
Factor w/ 161 levels "0","57.0331006081229",..: 10 NA NA 43 NA 3 1 NA NA NA ...
```

▼ Code

```
# Check the structure of the income_ppp variable
str(south_america_dataset_backup$income_ppp)
```

```
Factor w/ 161 levels "0","28.5165503040615",..: 10 NA NA 32 NA 3 1 NA NA NA ...
```

All variables exhibit ordinal levels, confirming that this assumption is met.

- *2. Appropriate Measures*

▼ Code

```
# Convert columns to numeric if they are not already in the backup dataset
south_america_dataset_backup$ls3 <-
  as.numeric(as.character(south_america_dataset_backup$ls3))
south_america_dataset_backup$income_usd <-
  as.numeric(as.character(south_america_dataset_backup$income_usd))
south_america_dataset_backup$income_ppp <-
  as.numeric(as.character(south_america_dataset_backup$income_ppp))

# Compute Spearman's Rank Correlation on the numeric columns in backup dataset
spearman_corr_ls3_income_usd <- cor(south_america_dataset_backup$ls3,
                                    south_america_dataset_backup$income_usd,
                                    method = "spearman", use = "complete.obs")
spearman_corr_ls3_income_ppp <- cor(south_america_dataset_backup$ls3,
                                    south_america_dataset_backup$income_ppp,
                                    method = "spearman", use = "complete.obs")
spearman_corr_income_usd_income_ppp <-
  cor(south_america_dataset_backup$income_usd,
      south_america_dataset_backup$income_ppp,
      method = "spearman", use = "complete.obs")

print(paste("Spearman's rank correlation between ls3 and income_usd:",
            spearman_corr_ls3_income_usd))
```

```
[1] "Spearman's rank correlation between ls3 and income_usd: 0.106368321137296"
```

▼ Code

```
print(paste("Spearman's rank correlation between ls3 and income_ppp:",
            spearman_corr_ls3_income_ppp))
```

```
[1] "Spearman's rank correlation between ls3 and income_ppp: 0.104715210070918"
```

▼ Code

```
print(paste("Spearman's rank correlation between income_usd and income_ppp:",
            spearman_corr_income_usd_income_ppp))
```

```
[1] "Spearman's rank correlation between income_usd and income_ppp: 0.970979197987847"
```

Spearman's rank correlation is a non-parametric method used to assess the strength and direction of association between ranked variables. It is particularly suited for ordinal variables or continuous variables that do not meet the assumptions of parametric tests. The Spearman's rank correlation coefficient (rho) ranges from -1 to 1:

- 1 indicates a perfect positive correlation,
- -1 indicates a perfect negative correlation,
- 0 indicates no correlation.

Interpretation of Results

1. Spearman's rank correlation between ls3 and income_usd: 0.106

Interpretation: There is a weak positive correlation between life satisfaction (ls3) and personal income (income_usd). This suggests that as personal income increases, there is a slight tendency for life satisfaction to also increase. However, the correlation is weak, indicating that income is not a strong predictor of life satisfaction in this dataset.

2. Spearman's rank correlation between ls3 and income_ppp: 0.105

Interpretation: Similarly, there is a weak positive correlation between life satisfaction (ls3) and income adjusted for purchasing power parity (income_ppp). This also suggests a slight tendency for higher life satisfaction with increased income, but the relationship remains weak.

3. Spearman's rank correlation between income_usd and income_ppp: 0.971

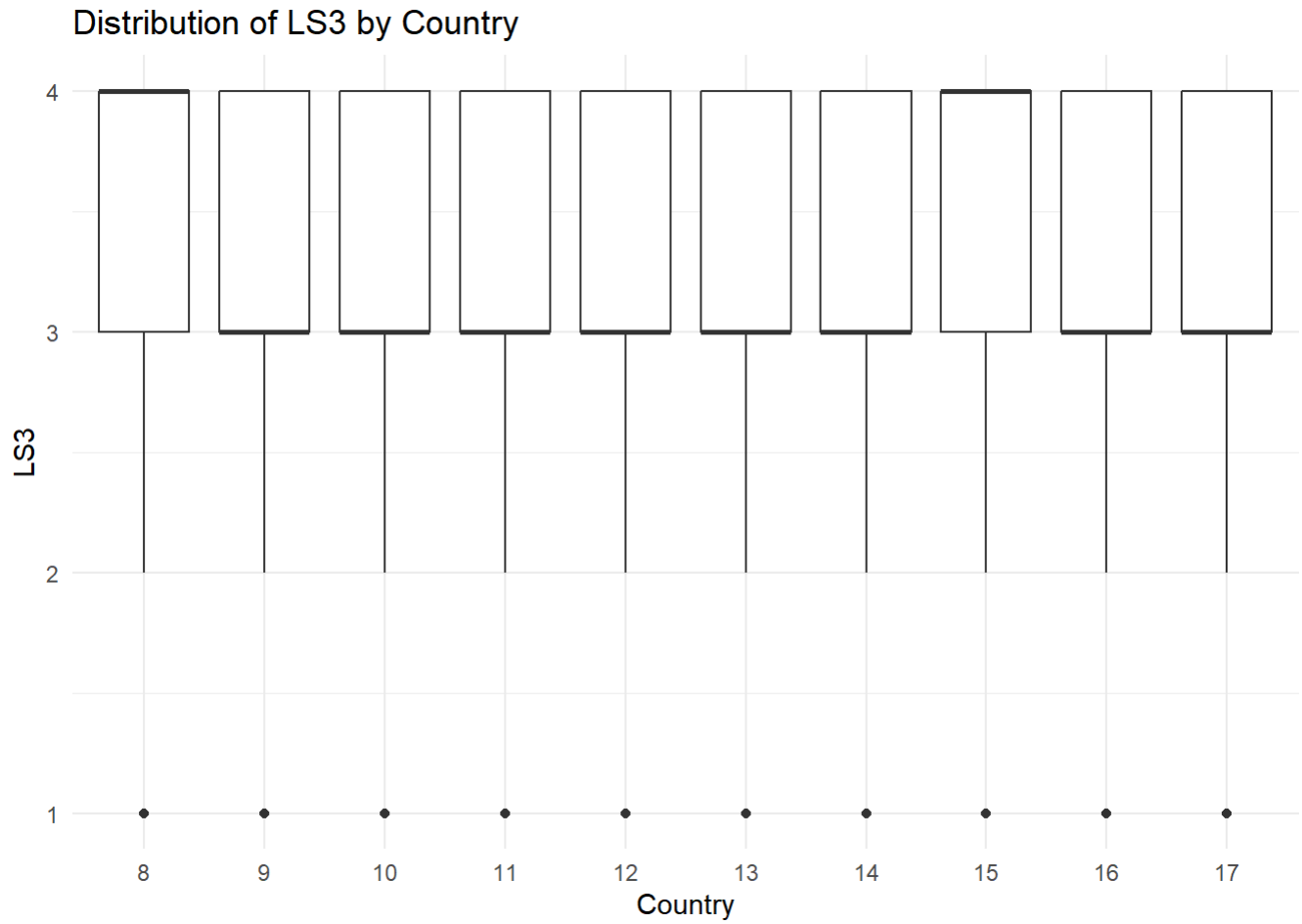
Interpretation: There is a very strong positive correlation between personal income (income_usd) and income adjusted for purchasing power parity (income_ppp). This high correlation indicates that these two income measures are almost perfectly correlated, as they represent the same underlying concept of income adjusted differently.

The weak correlations between life satisfaction and income suggest that factors other than income may play a more significant role in determining life satisfaction. Meanwhile, the strong correlation between income measures validates their expected relationship. Therefore, it is advisable to retain only one income variable in the analysis, thus income_usd can be removed.

- *3. Independence*

▼ Code

```
# Ensure pais is a factor
south_america_dataset_backup$pais <-
  as.factor(south_america_dataset_backup$pais)

# Remove rows with missing or non-finite values
cleaned_data_assumption <-
  south_america_dataset_backup %>%
  filter(is.finite(ls3) & is.finite(income_usd) & is.finite(income_ppp))

# Recreate the boxplot with cleaned data
plot_10 <- ggplot(cleaned_data_assumption, aes(x = pais, y = ls3)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Distribution of LS3 by Country", x = "Country", y = "LS3")

print(plot_10)
```



Each observation within the dataset is assumed to be independent. This assumption was evaluated by analyzing the distribution of the ls3 variable across countries using boxplots.

The boxplots revealed no significant patterns or dependencies across countries. The distributions appeared similar, with minor shifts in central tendencies. This provides supporting evidence for the assumption of independence in the analysis.

- *4. Sample Size*

To facilitate measurements, the original income variable q10g will be utilized. This decision is based on q10g having only 17 categories (ranging from 0 to 16). The converted variables such as income_usd and income_ppp result in at least 161 data points (like seen in the level of measurement). While this constitutes a substantial dataset, it does not affect the distribution of sample sizes when using the original q10g variable for this assumption.

▼ Code

```
# Count observations for LS3
ls3_counts <- table(south_america_dataset_subset$ls3)
print(ls3_counts)
```

```
   1    2    3    4 
 724 1674 6660 7261 
```

▼ Code

```
# Count observations for q10g
q10g_counts <- table(south_america_dataset_subset$q10g)
print(q10g_counts)
```

```
   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
 191 1362 1149  716  663  570  640  569  539  481  485  366  409  355  367  277
  16
 306
```

▼ Code

```
# Count observations for each country (pais)
country_counts <- table(south_america_dataset_subset$pais)
print(country_counts)
```

```
   8    9   10   11   12   13   14   15   16   17
1563 1545 1691 2647 1528 1625 1514 1532 1558 1528
```

LS3 Variable: Each category (1 to 4) shows a substantial number of observations, with counts of 724, 1674, 6660, and 7261 respectively. This ensures robust representation across all levels of life satisfaction.

Q10G Variable: The income variable exhibits varying counts across its 17 categories. Categories 0, 13, 14, 15, and 16 have fewer observations compared to others, but overall, there are sufficient data points (ranging from 191 to 7261) to conduct reliable statistical analysis across income levels.

Countries: The count of observations for each country (pais) ranges from 1528 to 1691, well above the minimum threshold of 1500 as assumed by LAPOP's technical guidelines. This ensures a solid foundation for making valid comparisons between countries.

Assumptions Statement:

The distribution of observations across ordinal categories (ls3 and q10g) confirms a sufficient sample size for meaningful statistical analysis. Each category within these variables contains an adequate number of observations, supporting reliable assessments across their respective ranges. Additionally, the observation counts for each country meet or exceed the minimum requirement of 1500 as stipulated by LAPOP, ensuring robust comparisons between countries. These findings validate that the sample size assumption is appropriately met for this analysis.

Conclusion Analysis 1

The conclusion draws upon an in-depth analysis of the relationship between income and life satisfaction across South American countries. By examining both between-countries and within-country correlations, this analysis illuminates how income impacts life satisfaction in diverse socio-economic contexts. This section explores the findings in detail, highlighting key differences observed at both macro and micro levels of analysis.

Between-Countries vs. Within-Country Correlations

- Between-Countries Correlation: When calculating the correlation for the entire cluster of countries, the analysis captures a broader dataset with more variability. This approach highlights overarching trends that may not be evident within smaller subsets.

The between-countries correlation (0.374 for mean income in USD and 0.317 for mean PPP) suggests a moderate positive relationship between income and life satisfaction when considering all countries together. This overall trend indicates that, generally, higher income is associated with higher life satisfaction across South America.

- Within-Country Correlations: For individual countries, the dataset is smaller and potentially more homogenous, which can lead to weaker correlations. There is less variability to capture a strong trend, and country-specific factors become more pronounced.

The weaker correlation can be attributed to various country-specific factors, including cultural differences, social safety nets, cost of living, economic stability, and other non-income-related factors affecting life satisfaction.

Factors Contributing to Lower Individual Correlations

- Country-Specific Variations: Each country has unique economic conditions, social structures, and cultural factors influencing the relationship between income and life satisfaction. These factors can dilute the impact of income on life satisfaction within individual countries.
- Sample Size: Smaller sample sizes within each country can lead to less reliable estimates of correlation, often resulting in weaker observed relationships.
- Income Distribution: The distribution of income within each country might vary significantly, affecting how income impacts life satisfaction. In countries with more equitable income distribution, the correlation might be weaker compared to countries with greater income inequality.
- Non-Linear Relationships: The relationship between income and life satisfaction might not be linear within countries. For instance, after a certain income threshold, increases in income might not lead to proportional increases in life satisfaction.
- Other Influencing Factors: Factors such as healthcare, education, social support, and political stability also play crucial roles in determining life satisfaction and might overshadow the influence of income within individual countries.

Understanding the Differences

The observed differences in correlations highlight that while income is an important factor in life satisfaction, its impact can vary significantly based onthe context and specific conditions within each country. The between-countries analysis captures a broader trend that might not be as pronounced when looking at smaller, more homogenous groups. This understanding underscores the importance of considering both macro (between countries) and micro (within countries) perspectives in such analyses.

Assumptions Met

The assumptions for the clustering and the summary statistics table were thoroughly evaluated and met:

- The silhouette scores indicate good clustering quality, particularly with k = 3 clusters without the outlier.
- The ordinal nature of ls3 and income variables is appropriately handled.
- The independence assumption is supported by the distribution analysis of ls3 across countries.
- Adequate sample sizes across all categories and countries confirm the reliability of the analysis.

Summary Conclusion Analysis 1

In conclusion, the analysis demonstrates that income does have a moderate correlation with life satisfaction when looking at the broader, aggregated data across South American countries. However, within individual countries the correlation is weaker, suggesting that other factors play a substantial role in determining life satisfaction. This nuanced view is crucial for policymakers and researchers aiming to improve life satisfaction through economic measures. The thorough checking and meeting of assumptions further validate the robustness of these findings.

In Analysis 2, there will be a deeper dive into other variables that could have a significant impact on life satisfaction. This contextualizes the conclusion of Analysis 1 with additional influential factors that are crucial for a comprehensive understanding of life satisfaction dynamics.

Analysis 2: Identifying the Key Predictors of Life Satisfaction

In this second analysis, the aim is to identify the most significant predictors of life satisfaction using advanced statistical and machine learning techniques. The process will include the following steps:

- Multiple Imputation: Multiple imputation methods will be employed to handle missing data, ensuring a more robust analysis compared to single imputation methods.
- Correlation Matrix: A correlation matrix will be constructed to identify the top predictors in a bivariate context. Training and Test Sets: The dataset will be split into training and test sets to facilitate the evaluation of machine learning models.
- Machine Learning Models: Four machine learning models will be utilized: multinomial logistic regression, decision tree, random forest, and gradient boosting (xgboost).
- Feature Importance: The best-performing model among the four will be used to determine feature importance, highlighting the most influential predictors.
- Assumptions: The primary assumption considered will be multicollinearity, ensuring that predictors are not too highly correlated with each other, which is crucial for reliable results.

Through this comprehensive approach, the analysis aims to uncover the key factors influencing life satisfaction and provide insights that can inform policy and decision-making. This structured methodology allows for meaningful conclusions and ensures that the analysis is both thorough and accurate.

Preparing the Data After the Assumptions

Based on the initial assumptions, the dataset was modified to ensure the most optimal and reliable results for the second analysis. This involved removing columns that exhibited multicollinearity or aliasing issues. Specifically, the columns “q12f” and “q12m” were removed due to perfect multicollinearity. Additionally, only one income column was retained (“income_ppp”), while other related columns were excluded to prevent redundancy and potential bias.

▼ Code

```
# Backup the original dataset
south_america_dataset_ML <- south_america_dataset_subset

# Define columns to be removed due to multicollinearity or aliasing issues
columns_to_remove <- c("q12f", "q12m", "q10g", "local_currency", "income_usd")

# Remove the specified columns from the dataset
south_america_dataset_ML <-
  south_america_dataset_ML[, !(names(south_america_dataset_ML) %in%
                                 columns_to_remove)]
```

By addressing these issues upfront, the dataset is now better suited for the subsequent steps in the analysis, including multiple imputation, correlation matrix construction, and machine learning model training. This preparation ensures that the analysis is both thorough and free from the confounding effects of multicollinearity, allowing for more accurate identification of the key predictors of life satisfaction.

Multiple Imputation

Several imputation methods were considered to handle missing data, particularly within the single imputation category. Initially, listwise deletion was attempted, but this approach resulted in only 4 complete rows out of 16,731 observations, making it impractical. Consequently, imputing the missing values became necessary, despite econometricians’ preference for complete data.

Among the single imputation methods considered were mean imputation, regression imputation, and stochastic imputation. However, it was determined that multiple imputation offers the best approach with the least error. Multiple imputation involves creating several complete datasets, each with different plausible values for the missing data points, generated from the distribution of the observed data.

Each of these complete datasets is analyzed separately, and the results are then pooled to produce overall estimates and standard errors. This method reduces bias, increases efficiency, and better reflects the uncertainty due to missing data, providing a more robust solution compared to single imputation methods.

▼ Code

```
# Define columns to be used for the model, including 'ls3'
columns_used <- c("ls3", setdiff(names(south_america_dataset_ML),
                                 c("pais", "uniq_id")))

# Select the relevant data
impute_data <- south_america_dataset_ML %>%
```

```
  dplyr::select(all_of(columns_used))

# Convert all columns to appropriate types (numeric) for imputation
# Ensure that 'ls3' is treated as a factor
impute_data <- impute_data %>%
  mutate(across(where(is.numeric), as.numeric)) %>%
  mutate(ls3 = as.factor(ls3))

# Perform imputation using mice with the pmm method
imputed_data <- mice(impute_data, m = 1, method = 'pmm', seed = 123)
```

```
 iter imp variable
  1   1  ls3  q10new  q10d  q10e  income_ppp  q2  q1  sexi  inf1  etid  vb1  vb2  ed  ed2
conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a  e5  D  LIB  EXC  exc7new  it1
aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new  MEDIA  vb3n  vb10  vb11  pol1
vb20  q5b  q3c  q11n  q12  q12c  q12bn  a4  soct2  wf1  cct1b  pn4  for5  drk1  env1c  env2b
vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2  ccq1  ccq2  ccq3  ccq4  q14  www1  R
  2   1  ls3  q10new  q10d  q10e  income_ppp  q2  q1  sexi  inf1  etid  vb1  vb2  ed  ed2
conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a  e5  D  LIB  EXC  exc7new  it1
aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new  MEDIA  vb3n  vb10  vb11  pol1
vb20  q5b  q3c  q11n  q12  q12c  q12bn  a4  soct2  wf1  cct1b  pn4  for5  drk1  env1c  env2b
vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2  ccq1  ccq2  ccq3  ccq4  q14  www1  R
  3   1  ls3  q10new  q10d  q10e  income_ppp  q2  q1  sexi  inf1  etid  vb1  vb2  ed  ed2
conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a  e5  D  LIB  EXC  exc7new  it1
aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new  MEDIA  vb3n  vb10  vb11  pol1
vb20  q5b  q3c  q11n  q12  q12c  q12bn  a4  soct2  wf1  cct1b  pn4  for5  drk1  env1c  env2b
vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2  ccq1  ccq2  ccq3  ccq4  q14  www1  R
  4   1  ls3  q10new  q10d  q10e  income_ppp  q2  q1  sexi  inf1  etid  vb1  vb2  ed  ed2
conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a  e5  D  LIB  EXC  exc7new  it1
aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new  MEDIA  vb3n  vb10  vb11  pol1
vb20  q5b  q3c  q11n  q12  q12c  q12bn  a4  soct2  wf1  cct1b  pn4  for5  drk1  env1c  env2b
vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2  ccq1  ccq2  ccq3  ccq4  q14  www1  R
  5   1  ls3  q10new  q10d  q10e  income_ppp  q2  q1  sexi  inf1  etid  vb1  vb2  ed  ed2
conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a  e5  D  LIB  EXC  exc7new  it1
aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new  MEDIA  vb3n  vb10  vb11  pol1
vb20  q5b  q3c  q11n  q12  q12c  q12bn  a4  soct2  wf1  cct1b  pn4  for5  drk1  env1c  env2b
vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2  ccq1  ccq2  ccq3  ccq4  q14  www1  R
```

▼ Code

```
# Extract the complete data after imputation
south_america_complete_data <- complete(imputed_data)

# Print the first few rows of the complete data to verify
print(head(south_america_complete_data))
```

```
  ls3 q10new q10d q10e income_ppp q2 q1 sexi inf1 etid vb1 vb2 ed ed2 conocim
1   2      5    2    2   50.82179 68  2    2    1    2   1   1  5   1       4
2   4      4    3    2  560.66946 51  2    1    1    1   1   2 11   0       5
3   4      1    3    1   79.59479 37  2    1    1    2   1   1  5   1       5
4   4     11    1    1  123.32754 25  1    1    1    1   1   1  9   1       2
5   3      1    3    2   41.12909 36  2    2    1    3   1   2  5   0       3
```

```
6   2      1    4    3   30.49307 64  2    2    1    2   1   1  3   0       4
  ocup1a ocup4a np1   CP cp20 cp6 q5a idio2 w14a e5    D LIB EXC exc7new it1
1      4      1   2 4.00    4   4   2     2    2  1 3.17 1.0   0       5   3
2      4      3   2 4.00    4   4   4     3    1  1 2.67 1.5   0       5   4
3      5      5   2 3.33    4   2   1     3    1  6 1.67 1.0   0       5   3
4      4      1   2 2.67    4   2   2     1    2  5 3.33 1.0   0       4   1
5      4      5   2 4.00    4   1   2     2    2  1 1.00 1.5   0       2   1
6      4      1   2 2.00    4   3   3     3    2  9 7.00 2.0   0       4   2
  aoj12    B INFRA MIL10 gi0 l1  JC   M ROS  POL aoj22new MEDIA vb3n vb10 vb11
1     2 3.31     5  3.00   2  5 2.0 3.0 4.0 4.00        5   4.0  804    2  913
2     1 2.69     5  4.00   2  1 1.5 4.5 7.0 4.50        7   1.0 1402    2 1304
3     1 3.08     5  4.00   5  5 1.0 3.0 6.5 5.33        5   6.0  804    2  805
4     3 3.31     6  2.25   1 10 2.0 3.0 5.0 5.50        7   3.5  804    1  801
5     2 4.38     5  1.00   2  6 2.0 2.0 5.0 5.00        7   4.5 1102    2  909
6     1 4.17     2  3.00   1 10 2.0 3.0 6.5 6.67        7   5.0  877    2 1001
  pol1 vb20 q5b q3c q11n q12 q12c q12bn a4 soct2 wf1 cct1b pn4 for5 drk1 env1c
1    4    1   2   1    1   3    6     0 30     3   2     2   2    1    3     7
2    4    1   1   1    6   3    1     0 59     3   2     2   4    1    4     4
3    4    1   1   5    2   3    4     1 30     3   2     1   4    4    3     6
4    3    2   1   5    3   0    4     2  1     1   2     1   3    2    3     5
5    4    1   1   1    3   4    8     2 18     3   2     2   2    1    1     7
6    3    4   2   1    6   5    1     0  1     3   1     1   3   10    4     1
  env2b vic1ext vic1exta aoj11 dst1b IAREA SDNEW2 ccq1 ccq2 ccq3 ccq4 q14 www1
1     1       2        3     2     5  1.83   3.00    1    3    2    3   2    5
2     1       2        1     2     7  2.00   2.33    1    3    2    2   2    1
3     1       1        4     4     7  1.17   3.33    2    3    2    3   1    5
4     2       2        1     1     6  1.67   3.00    1    4    2    4   2    3
5     4       2        1     2     5  2.00   4.00    2    3    1    1   2    5
6     3       2        1     2     7  1.83   3.00    2    4    2    4   2    5
     R
1 0.00
2 0.00
3 0.09
4 0.09
5 0.09
6 0.09
```

With the removal of the income variables (q10g, local_currency, income_usd) and the perfectly collinear variables (q12f and q12m), the data is now more suitable for analysis. The income variables were highly correlated because they were all derived from the q10g variable, leading to redundancy. By eliminating these, the issue of multiple imputation introducing randomness and breaking their inherent relationship is resolved.

Additionally, the perfect collinearity between q12f and q12m (children by gender) and q12 (total children) posed a problem. Initially, the imputation method assigned random values to q12f and q12m, inadvertently breaking the perfect relationship with q12. Removing q12f and q12m eliminates this issue, ensuring the imputation method does not artificially alter the data structure.

By addressing these multicollinearities, there is no longer a concern about the imputation method assigning random values that disrupt variable relationships. The remaining variables are not bound to each other or excessively correlated, making the dataset more robust for further analysis.

Training & Test Set

To train the machine learning models effectively, separate training and test sets are essential. The training set will be used to develop the models, while the test set serves to evaluate their performance.

▼ Code

```
# Split the data into training and testing sets
# (e.g., 80% training and 20% testing)
set.seed(123) # for reproducibility
train_indices <- sample(1:nrow(south_america_complete_data), 0.8 *
                          nrow(south_america_complete_data))
train_data <- south_america_complete_data[train_indices, ]
test_data <- south_america_complete_data[-train_indices, ]

dim(train_data)
```

```
[1] 13384    75
```

▼ Code

```
dim(test_data)
```

```
[1] 3347   75
```

Using the dim() function, the dimensions of both the training and test sets are checked to ensure the split is accurate and to confirm the number of variables in each.

Exploration of Variable Correlation

With the pre-processing complete, a correlation matrix can be created to identify the top 10 variables to be used in the model. It is important to note that this process is influenced by the imputation method and the creation of the training/test sets, making it inherently random. To ensure consistency, a seed is set in every piece of code involving randomness.

▼ Code

```
# Convert the outcome variable to numeric for correlation purposes
south_america_complete_data$ls3 <-
  as.numeric(as.character(south_america_complete_data$ls3))

# Function to calculate correlation or association
calculate_correlation <- function(column, outcome) {
  if (is.numeric(column)) {
    # Use Spearman correlation for numeric columns
    return(cor(column, outcome, use = "complete.obs", method = "spearman"))
  } else if (is.factor(column) || is.character(column)) {
    # Use Cramér's V for categorical columns
    table_data <- table(column, outcome)
    return(CramerV(table_data))
```

```
  } else {
    return(NA)  # Return NA for unsupported column types
  }
}

# Apply the function to each column
results <- sapply(south_america_complete_data, calculate_correlation,
                  outcome = south_america_complete_data$ls3)

# Create a data frame from the results
results_df <- data.frame(variable = names(results), correlation = results)

# Remove NA values
results_df <- results_df[!is.na(results_df$correlation), ]

# Sort the dataframe by correlation in descending order
results_df <- results_df %>%
  arrange(desc(correlation))

# Format the correlation values to avoid scientific notation
results_df$correlation <- format(results_df$correlation, scientific = FALSE)

# Print the sorted results
print(results_df)
```

```
             variable  correlation
ls3               ls3  1.000000000
R                   R  0.135380467
q10new          q10new  0.118427888
ed2                ed2  0.097903607
B                   B  0.094075752
LIB               LIB  0.074167591
vic1ext       vic1ext  0.063079013
ed                 ed  0.061540361
a4                 a4  0.060043033
q14               q14  0.059092003
income_ppp income_ppp  0.057077448
POL               POL  0.046642113
vb2               vb2  0.037889595
D                   D  0.030567946
CP                 CP  0.028491789
JC                 JC  0.028365636
vb1               vb1  0.025323810
vb3n             vb3n  0.024238661
sexi             sexi  0.023644346
aoj22new     aoj22new  0.023010670
vb11             vb11  0.018854720
MEDIA           MEDIA  0.018689382
l1                 l1  0.014433772
ROS               ROS  0.011514403
cct1b           cct1b  0.011423329
np1               np1  0.011073591
e5                 e5  0.007231637
wf1               wf1  0.005751132
```

cp20	cp20	0.002962637
dst1b	dst1b	0.002829761
q3c	q3c	0.002681801
vb10	vb10	0.001333658
for5	for5	-0.004487035
w14a	w14a	-0.005101704
q12bn	q12bn	-0.005245479
q12c	q12c	-0.006940445
inf1	inf1	-0.008481592
pol1	pol1	-0.008643956
cp6	cp6	-0.014168404
env1c	env1c	-0.020528915
q5a	q5a	-0.022046512
gi0	gi0	-0.022716522
q5b	q5b	-0.023627958
q1	q1	-0.028351163
IAREA	IAREA	-0.028641361
exc7new	exc7new	-0.028657820
ocup4a	ocup4a	-0.028864697
vb20	vb20	-0.028941642
env2b	env2b	-0.029460484
q11n	q11n	-0.032198196
aoj12	aoj12	-0.038532703
ccq2	ccq2	-0.039125522
ccq4	ccq4	-0.039393596
drk1	drk1	-0.042155213
ccq3	ccq3	-0.045365816
etid	etid	-0.047685229
ccq1	ccq1	-0.047804243
conocim	conocim	-0.048341168
M	M	-0.049511329
MIL10	MIL10	-0.050465057
vic1exta	vic1exta	-0.063138673
q12	q12	-0.068567755
EXC	EXC	-0.068589673
soct2	soct2	-0.070680535
ocup1a	ocup1a	-0.074687352
q2	q2	-0.083457472
pn4	pn4	-0.089188088
INFRA	INFRA	-0.100976535
www1	www1	-0.113225387
it1	it1	-0.118122071
SDNEW2	SDNEW2	-0.125643748
aoj11	aoj11	-0.131834487
q10e	q10e	-0.137599929
q10d	q10d	-0.194151462
idio2	idio2	-0.208803781

The use of a top 10 variable selection is now more reliable due to the improved multiple imputation method and the removal of income and q12 variables, eliminating the need to use a wider range of top 15 selection. This adjustment ensures that the selection of variables is less influenced by random seeds and the assumptions of multicollinearity, resulting in a more stable and dependable top 10 variable list.

Machine Learning Models

Four machine learning models will be utilized in this analysis: multinomial logistic regression, decision tree, random forest, and gradient boosting. The selection of these models is based on their suitability for an ordinal level categorical variable.

- Multinomial Logistic Regression: Unlike binary logistic regression, multinomial logistic regression is capable of handling multiple categories in the outcome variable, making it appropriate for this analysis.
- Decision Tree: This model is straightforward and interpretable, suitable for both continuous and categorical outcomes.
- Random Forest: Building on the decision tree, random forest combines multiple trees to enhance prediction accuracy and robustness.
- Gradient Boosting (XGBoost): This is the most advanced model among the four, offering improved performance by sequentially building trees and correcting errors from previous iterations.

These models are arranged from the least to the most technically sophisticated, allowing for a progressive assessment of their predictive power. The performance of each model will be evaluated using accuracy, derived from a confusion matrix.

Initially, the top 10 variables, identified through the correlation analysis, will be used to assess accuracy. Subsequently, all variables will be considered, totaling 74. It is important to note that some variables have been aggregated into single factors, so the actual number of variables is closer to 130-140. This comprehensive approach mirrors the initial subset dataset per country, which had a similar variable count.

Subsetting the Top 10 Variables

Start with subsetting the top 10 variables in the training and test set so that can be used for machine learning models at first.

▼ Code

```
# Define the selected features based on the given list
selected_features <- c("R", "q10new", "ed2", "B", "LIB", "vic1ext", "ed", "a4",
                       "q14", "income_ppp")


train_data_subset <- train_data[, c("ls3", selected_features)]
test_data_subset <- test_data[, c("ls3", selected_features)]

dim(train_data_subset)
```

```
[1] 13384    11
```

▼ Code

```
dim(test_data_subset)
```

```
[1] 3347   11
```

The dimensions confirm the correct subset, with 10 predictor variables and 1 outcome variable, maintaining the same split as the original datasets.

Multinomial Logistic Regression (Top 10)

This code sets up a multinomial logistic regression model using the top 10 selected variables. The model is trained with cross-validation to ensure robustness. Predictions are then made on the test data subset, and the accuracy is calculated using a confusion matrix, providing a measure of the model's performance.

▼ Code

```
set.seed(42)

# Multinomial logistic regression
multinom_model <- multinom(ls3 ~ ., data = train_data_subset)

# Cross-validation
ctrl_log <- trainControl(method = "cv", number = 5)
multinom_cv <- train(ls3 ~ ., data = train_data_subset, method = "multinom",
                     trControl = ctrl_log)

# Performance evaluation
multinom_pred <- predict(multinom_cv, newdata = test_data_subset)
multinom_accuracy <- confusionMatrix(multinom_pred,
                                     test_data_subset$ls3)$overall["Accuracy"]
```

Decision Tree (Top 10)

This section focuses on creating a decision tree model using the top 10 selected variables. It aims to understand how these variables collectively influence the prediction of ls3. This approach provides insights into how well the decision tree method captures the relationship between the selected variables and ls3, offering a clear assessment of its predictive accuracy.

▼ Code

```
set.seed(42)

# Decision tree
decision_tree_model <- rpart(ls3 ~ .,
                             data = train_data_subset, method = "class")

# Cross-validation
ctrl_dt <- trainControl(method = "cv", number = 5)
decision_tree_cv <- train(ls3 ~ ., data = train_data_subset, method = "rpart",
                          trControl = ctrl_dt)

# Performance evaluation
decision_tree_pred <- predict(decision_tree_cv, newdata = test_data_subset)
decision_tree_accuracy <-
```

```
  confusionMatrix(decision_tree_pred,
                  test_data_subset$ls3)$overall["Accuracy"]
```

Random Forest (Top 10)

This segment focuses on constructing a random forest model using the top 10 selected variables. It aims to assess how these variables collectively influence the prediction of ls3, the target variable. By utilizing ensemble learning techniques, such as bagging and feature randomization, the random forest model enhances predictive accuracy and robustness compared to individual decision trees. Performance evaluation involves analyzing predictions against actual ls3 values in the test dataset, providing insights into the model's effectiveness in capturing complex relationships among the selected variables and ls3.

▼ Code

```
set.seed(42)

# Random Forest
rf_model <- randomForest(ls3 ~ ., data = train_data_subset)

# Cross-validation
ctrl_rf <- trainControl(method = "cv", number = 5)
rf_cv <- train(ls3 ~ ., data = train_data_subset, method = "rf",
               trControl = ctrl_rf)

# Performance evaluation
rf_pred <- predict(rf_cv, newdata = test_data_subset)
rf_accuracy <- confusionMatrix(rf_pred,
                               test_data_subset$ls3)$overall["Accuracy"]
```

Gradient Boosting (Top 10)

Gradient boosting, being a more advanced model, requires meticulous preparation to ensure optimal performance with ordinal categorical variables like ls3. The initial step involves data preparation, where the ls3 variable is treated as a factor and converted to numeric values suitable for training with XGBoost. This ensures compatibility and accuracy in modeling the ordinal nature of ls3.

▼ Code

```
# Create backups of the training and testing datasets
train_data_backup <- train_data_subset
test_data_backup <- test_data_subset

set.seed(42)

# Ensure ls3 is a factor
train_data_backup$ls3 <- factor(train_data_backup$ls3)
test_data_backup$ls3 <- factor(test_data_backup$ls3,
                               levels = levels(train_data_backup$ls3))

# Convert factors to numeric for XGBoost
train_labels <- as.numeric(train_data_backup$ls3) - 1
test_labels <- as.numeric(test_data_backup$ls3) - 1
```

```
# Create DMatrix
train_matrix <- xgb.DMatrix(data = as.matrix(train_data_backup[, -1]),
                            label = train_labels)
test_matrix <- xgb.DMatrix(data = as.matrix(test_data_backup[, -1]),
                           label = test_labels)
```

Next, cross-validation is employed to determine the optimal number of boosting rounds, crucial for preventing overfitting and achieving balanced model performance. By iterating through multiple rounds and evaluating performance metrics like multi-class log loss, the process identifies the point where further boosting rounds no longer improve model accuracy.

▼ Code

```
set.seed(42)

# Define parameters
xgb_params <- list(
  objective = "multi:softprob",
  num_class = length(unique(train_labels)),
  eval_metric = "mlogloss",
  max_depth = 5,
  eta = 0.1
)

# Perform cross-validation
cv_results <- xgb.cv(
  params = xgb_params,
  data = train_matrix,
  nrounds = 100,
  nfold = 5,
  verbose = TRUE,
  early_stopping_rounds = 10
)

# Find the optimal number of rounds
optimal_nrounds <- cv_results$best_iteration
cat("Optimal number of rounds:", optimal_nrounds, "\n")
```

Following cross-validation, the XGBoost model is trained using the determined optimal number of rounds. This trained model utilizes gradient boosting techniques to iteratively improve predictive accuracy by focusing on areas where previous models may have struggled, thereby enhancing the robustness of predictions for ls3 based on the selected top 10 variables.

▼ Code

```
# Train XGBoost model
xgb_model <- xgboost(
  data = train_matrix,
  params = xgb_params,
  nrounds = optimal_nrounds,
  verbose = 0
)
```

```
# Make predictions
xgb_pred <- predict(xgb_model, newdata = test_matrix)
xgb_pred_labels <-
  max.col(matrix(xgb_pred, ncol = length(unique(train_labels)))) - 1

# Ensure labels are factors with the correct levels
xgb_pred_labels_factor <- factor(xgb_pred_labels,
                                 levels = 0:(length(unique(train_labels))-1))
actual_labels_factor <- factor(test_labels,
                               levels = 0:(length(unique(test_labels))-1))

# Calculate accuracy
xgb_accuracy <- confusionMatrix(xgb_pred_labels_factor,
                                actual_labels_factor)$overall["Accuracy"]
```

Accuracy of Machine Learning Models (Top 10)

After executing the machine learning models using the top 10 variables, the table below displays their respective accuracies for comparison.

▼ Code

```
# Model accuracies for top 10 variables
model_accuracies_top15_df <- data.frame(
  Model = c("Multinomial Logistic Regression", "Decision Tree", "Random Forest",
            "XGBoost"),
  Accuracy = c(multinom_accuracy, decision_tree_accuracy,
               rf_accuracy, xgb_accuracy)
)

# Generate the HTML table
acc_top_10 <- kable(model_accuracies_top15_df, format = "html", digits = 3,
                    col.names = c("Model", "Accuracy")) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed",
                                      "responsive"),
                full_width = F) %>%
  column_spec(2, bold = TRUE)

acc_top_10
```

Model	Accuracy
Multinomial Logistic Regression	**0.480**
Decision Tree	**0.469**
Random Forest	**0.466**
XGBoost	**0.264**

Multinomial Logistic Regression 0.480 Decision Tree 0.469 Random Forest 0.466 XGBoost 0.264

From the results, the multinomial logistic regression model shows the highest accuracy among the top 10 predictors according to the confusion matrix. It outperforms both the decision tree and random forest models by a slight margin, while XGBoost exhibits the lost accuracy among the models tested.

The current findings suggest that the multinomial logistic regression model is promising for predicting ls3 based on the selected top 10 variables. However, further investigation with all variables considered may alter these results, potentially revealing different strengths among the models.

Multinomial Logistic Regression (All Variables)

The multinomial logistic regression model is now applied to the full dataset, comprising 74 variables, to observe if there is an improvement in predictive accuracy compared to the model using only the top 10 variables.

▼ Code

```
set.seed(42)

# Multinomial logistic regression
multinom_model_2 <- multinom(ls3 ~ ., data = train_data)

# Cross-validation
ctrl_log_2 <- trainControl(method = "cv", number = 5)
multinom_cv_2 <- train(ls3 ~ ., data = train_data, method = "multinom",
                       trControl = ctrl_log_2)

# Performance evaluation
multinom_pred_2 <- predict(multinom_cv_2, newdata = test_data)
multinom_accuracy_2 <- confusionMatrix(multinom_pred_2,
                                       test_data$ls3)$overall["Accuracy"]
```

Decision Tree (All Variables)

The decision tree model is then applied to the full dataset, using all 74 variables. This will determine if there is an improvement in the model's performance compared to using only the top 10 variables.

▼ Code

```
set.seed(42)

# Decision tree
decision_tree_model_2 <- rpart(ls3 ~ ., data = train_data, method = "class")

# Cross-validation
ctrl_dt_2 <- trainControl(method = "cv", number = 5)
decision_tree_cv_2 <- train(ls3 ~ ., data = train_data, method = "rpart",
                            trControl = ctrl_dt_2)

# Performance evaluation
decision_tree_pred_2 <- predict(decision_tree_cv_2, newdata = test_data)
decision_tree_accuracy_2 <- confusionMatrix(decision_tree_pred_2,
                                            test_data$ls3)$overall["Accuracy"]
```

Random Forest (All Variables)

Next, the random forest model is used on the complete set of variables. This will show if incorporating all variables leads to better predictive accuracy compared to the top 10 variable subset.

▼ Code

```
set.seed(42)

# Random Forest
rf_model_2 <- randomForest(ls3 ~ ., data = train_data)

# Cross-validation
ctrl_rf_2 <- trainControl(method = "cv", number = 5)
rf_cv_2 <- train(ls3 ~ ., data = train_data, method = "rf",
                 trControl = ctrl_rf_2)

# Performance evaluation
rf_pred_2 <- predict(rf_cv_2, newdata = test_data)
rf_accuracy_2 <- confusionMatrix(rf_pred_2, test_data$ls3)$overall["Accuracy"]
```

Gradient Boosting

The final model examines the performance of gradient boosting using all available variables. By applying the gradient boosting model to all variables, it can be determined if including the entire set of features enhances the model's accuracy compared to using only the top 10 predictors.

- Data Preparation of XGBoost

▼ Code

```
# Create backups of the training and testing datasets
train_data_backup_2 <- train_data
test_data_backup_2 <- test_data

set.seed(42)

# Ensure ls3 is a factor
train_data_backup_2$ls3 <- factor(train_data_backup_2$ls3)
test_data_backup_2$ls3 <- factor(test_data_backup_2$ls3,
                                 levels = levels(train_data_backup_2$ls3))

# Convert factors to numeric for XGBoost
train_labels_2 <- as.numeric(train_data_backup_2$ls3) - 1
test_labels_2 <- as.numeric(test_data_backup_2$ls3) - 1

# Create DMatrix
train_matrix_2 <- xgb.DMatrix(data = as.matrix(train_data_backup_2[, -1]),
                              label = train_labels_2)
test_matrix_2 <- xgb.DMatrix(data = as.matrix(test_data_backup_2[, -1]),
                             label = test_labels_2)
```

- Cross-Validation to Find Optimal Number of Rounds

▼ Code

```
set.seed(42)

# Define parameters
xgb_params_2 <- list(
  objective = "multi:softprob",
  num_class = length(unique(train_labels_2)),
  eval_metric = "mlogloss",
  max_depth = 5,
  eta = 0.1
)

# Perform cross-validation
cv_results_2 <- xgb.cv(
  params = xgb_params_2,
  data = train_matrix_2,
  nrounds = 100,
  nfold = 5,
  verbose = TRUE,
  early_stopping_rounds = 10
)

# Find the optimal number of rounds
optimal_nrounds_2 <- cv_results_2$best_iteration
cat("Optimal number of rounds:", optimal_nrounds_2, "\n")
```

- Train XGBoost Model Using Optimal Number of Rounds

▼ Code

```
# Train XGBoost model
xgb_model_2 <- xgboost(
  data = train_matrix_2,
  params = xgb_params_2,
  nrounds = optimal_nrounds_2,
  verbose = 0
)

# Make predictions
xgb_pred_2 <- predict(xgb_model_2, newdata = test_matrix_2)
xgb_pred_labels_2 <- max.col(matrix(xgb_pred_2,
                                    ncol = length(unique(train_labels_2)))) - 1

# Ensure labels are factors with the correct levels
xgb_pred_labels_factor_2 <-
  factor(xgb_pred_labels_2,
         levels = 0:(length(unique(train_labels_2))-1))
actual_labels_factor_2 <- factor(test_labels_2,
                                 levels = 0:(length(unique(test_labels_2))-1))

# Calculate accuracy
```

```
xgb_accuracy_2 <- confusionMatrix(xgb_pred_labels_factor_2,
                                  actual_labels_factor_2)$overall["Accuracy"]
```

Accuracy Machine Learning Models

Let's find out if the accuracy improved when including all variables.

▼ Code

```
# Model accuracies
model_accuracies_df_2 <- data.frame(
  Model = c("Multinomial Logistic Regression", "Decision Tree", "Random Forest",
            "XGBoost"),
  Accuracy = c(multinom_accuracy_2, decision_tree_accuracy_2, rf_accuracy_2,
               xgb_accuracy_2)
)

# Print the table using kable with three decimals
acc_all_var <- kable(model_accuracies_df_2, format = "html", digits = 3,
                     col.names = c("Model", "Accuracy")) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed",
                                      "responsive"),
                full_width = F) %>%
  column_spec(2, bold = TRUE)

acc_all_var
```

Model	Accuracy
Multinomial Logistic Regression	**0.489**
Decision Tree	**0.475**
Random Forest	**0.531**
XGBoost	**0.264**

Multinomial Logistic Regression 0.489 Decision Tree 0.475 Random Forest 0.531 XGBoost 0.264

The accuracies increased slightly, particularly for the random forest model. To gain a clearer understanding of how each model's performance changed, it is helpful to compare the accuracies with those achieved using the top 10 variables. This comparison will reveal whether the accuracies increased or decreased when using all variables.

▼ Code

```
# Calculate the differences in accuracy between
# top 10 variables and all variables
difference_multinom <- (multinom_accuracy_2 - multinom_accuracy) * 100
difference_decision_tree <-
  (decision_tree_accuracy_2 - decision_tree_accuracy) * 100
difference_rf <- (rf_accuracy_2 - rf_accuracy) * 100
```

```
difference_xgb <- (xgb_accuracy_2 - xgb_accuracy) * 100

# Create a data frame with the differences
accuracy_diff_df <- data.frame(
  Model = c("Multinomial Logistic Regression", "Decision Tree", "Random Forest",
            "XGBoost"),
  Difference = c(difference_multinom, difference_decision_tree, difference_rf,
                 difference_xgb)
)

# Print the table using kable
diff_acc_var <- kable(accuracy_diff_df, format = "html", digits = 2,
                      col.names = c("Model", "Accuracy Difference (%)")) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed",
                                      "responsive"),
                full_width = F) %>%
  column_spec(2, bold = TRUE)

diff_acc_var
```

Model	Accuracy Difference (%)
Multinomial Logistic Regression	**0.90**
Decision Tree	**0.60**
Random Forest	**6.42**
XGBoost	**0.09**

Please note that the differences are multiplied by 100 to be displayed as percentages. The original accuracies for the top 10 variables and all variables are shown in their original decimal form.

Accuracy Difference (%) Multinomial Logistic Regression 0.90 Decision Tree 0.60 Random Forest 6.42 XGBoost 0.09

Upon examining the improvements of all four models, it becomes evident that no model improved significantly enough to make a substantial difference, except for the random forest model, which saw an increase of over 6%. This indicates that the random forest model is the most suitable for this analysis.

The multinomial model showed a slight improvement over the top 10 model but did not enhance significantly with the inclusion of all features. Therefore, the random forest model will be utilized for further analysis.

Random Forest is the best overall model in this case, both in terms of improvement and accuracy with all features. This suggests that it is particularly effective at utilizing the full feature set to make accurate predictions.

Feature Importance in the Best Model

It is useful to investigate whether the significant features for predicting life satisfaction identified by the random forest model differ from those identified by the correlation matrix. Differences may arise

because the correlation matrix analyzes individual relationships between variables and the target, while the random forest model accounts for complex interactions among multiple variables. This comparison can help determine the added value of considering multivariate interactions in feature importance.

▼ Code

```
# Calculate feature importance
importance <- importance(rf_model_2)
importance_df <- data.frame(Feature = rownames(importance),
                            Importance = importance[, 1])
importance_df <- importance_df[order(-importance_df$Importance), ]

# Print feature importance
print(importance_df)
```

```
              Feature Importance
income_ppp income_ppp 259.026323
B                   B 258.343796
q2                 q2 247.465076
D                   D 236.446577
vb11             vb11 220.266757
POL               POL 212.260512
vb3n             vb3n 203.334601
q10new         q10new 191.560808
ed                 ed 185.023628
MIL10           MIL10 178.278916
ROS               ROS 176.548768
R                   R 176.033768
SDNEW2         SDNEW2 170.822935
IAREA           IAREA 170.524140
MEDIA           MEDIA 169.396221
a4                 a4 169.356094
l1                 l1 156.595552
q12c             q12c 152.175228
INFRA           INFRA 150.252008
e5                 e5 146.727245
for5             for5 144.059834
LIB               LIB 143.690175
M                   M 143.072749
q12               q12 132.084337
env1c           env1c 126.445486
ed2               ed2 126.327154
it1               it1 124.410099
aoj11           aoj11 123.741236
dst1b           dst1b 118.484162
CP                 CP 117.889835
etid             etid 115.788831
EXC               EXC 112.893596
q10d             q10d 111.755840
idio2           idio2 111.306642
ccq4             ccq4 110.340669
ccq2             ccq2 110.084864
q5a               q5a 106.107918
```

aoj22new	aoj22new	105.341626
q11n	q11n	103.741188
vic1exta	vic1exta	102.277739
ocup4a	ocup4a	101.294580
conocim	conocim	99.644759
q12bn	q12bn	96.852612
exc7new	exc7new	95.675072
drk1	drk1	93.304192
pol1	pol1	91.231141
pn4	pn4	88.777194
vb20	vb20	88.343151
q3c	q3c	87.613814
aoj12	aoj12	86.337505
cp6	cp6	84.124265
www1	www1	84.091958
ocup1a	ocup1a	82.340850
q5b	q5b	78.792447
q10e	q10e	75.893957
gi0	gi0	72.660132
JC	JC	69.496488
env2b	env2b	61.659383
soct2	soct2	56.027159
cp20	cp20	54.469814
vic1ext	vic1ext	42.250335
sexi	sexi	42.238156
q1	q1	42.103355
w14a	w14a	41.585692
ccq3	ccq3	41.559878
q14	q14	39.993353
ccq1	ccq1	38.017688
vb10	vb10	36.545572
vb2	vb2	35.296890
cct1b	cct1b	33.304019
np1	np1	30.638993
wf1	wf1	26.240238
vb1	vb1	21.856686
inf1	inf1	4.179449

Top 10 importance features: 1. income_ppp 2. B 3. q2 4. D 5. vb11 6. POL 7. vb3n 8. q10new 9. ed 10. MIL10

Fill out with top 16: 11. ROS 12. R 13. SDNEW2 14. IAREA 15. MEDIA 16. a4

It is important to note that the ranking of features may vary slightly with different random seeds. However, the top 16 features generally remain consistent across runs. Notably, there is a significant drop in importance after the top 9 features, which range from 185 to 178 in importance. Interestingly, the difference in importance between the 9th and 10th ranked features is almost equivalent to the difference between the 10th and 16th ranked features.

Top 10 Feature Importance: 1. Income: income_ppp, q10new 2. Personal Characteristics (Age): q2 3. Socially Developed Characteristics (Education): ed 5. Attitude and Beliefs Towards; Self, Others, Life (Attitude Towards our Circumstances): D 5. Attitude and Beliefs Towards; Self, Others, Life (Social/Institutional Trust): B, POL, MIL10 5. Attitude and Beliefs Towards; Self, Others, Life (Political Persuasion): vb3n, vb11

▼ Code

```
# Create a dataframe for the top 10 feature importance
feature_importance_poster <- data.frame(
  Category = c(
    "Income",
    "Age",
    "Education",
    "Attitude Towards our Circumstances",
    "Social/Institutional Trust",
    "Political Persuasion"
  ),
  Features = c(
    "income_ppp, q10new",
    "q2",
    "ed",
    "D",
    "B, POL, MIL10",
    "vb3n, vb11"
  )
)

# Generate the HTML table
feature_importance_table_poster <- kable(
  feature_importance_poster,
  format = "html",
  col.names = c("Category", "Features"),
  align = c("l", "l") # Left align Category and Features
) %>%
  kable_styling(
    bootstrap_options = c("striped", "hover", "condensed", "responsive"),
    full_width = F
  ) %>%
  row_spec(0, bold = TRUE) # Make the header row bold

# Print the table
feature_importance_table_poster
```

Category	Features
Income	income_ppp, q10new
Age	q2
Education	ed
Attitude Towards our Circumstances	D
Social/Institutional Trust	B, POL, MIL10
Political Persuasion	vb3n, vb11

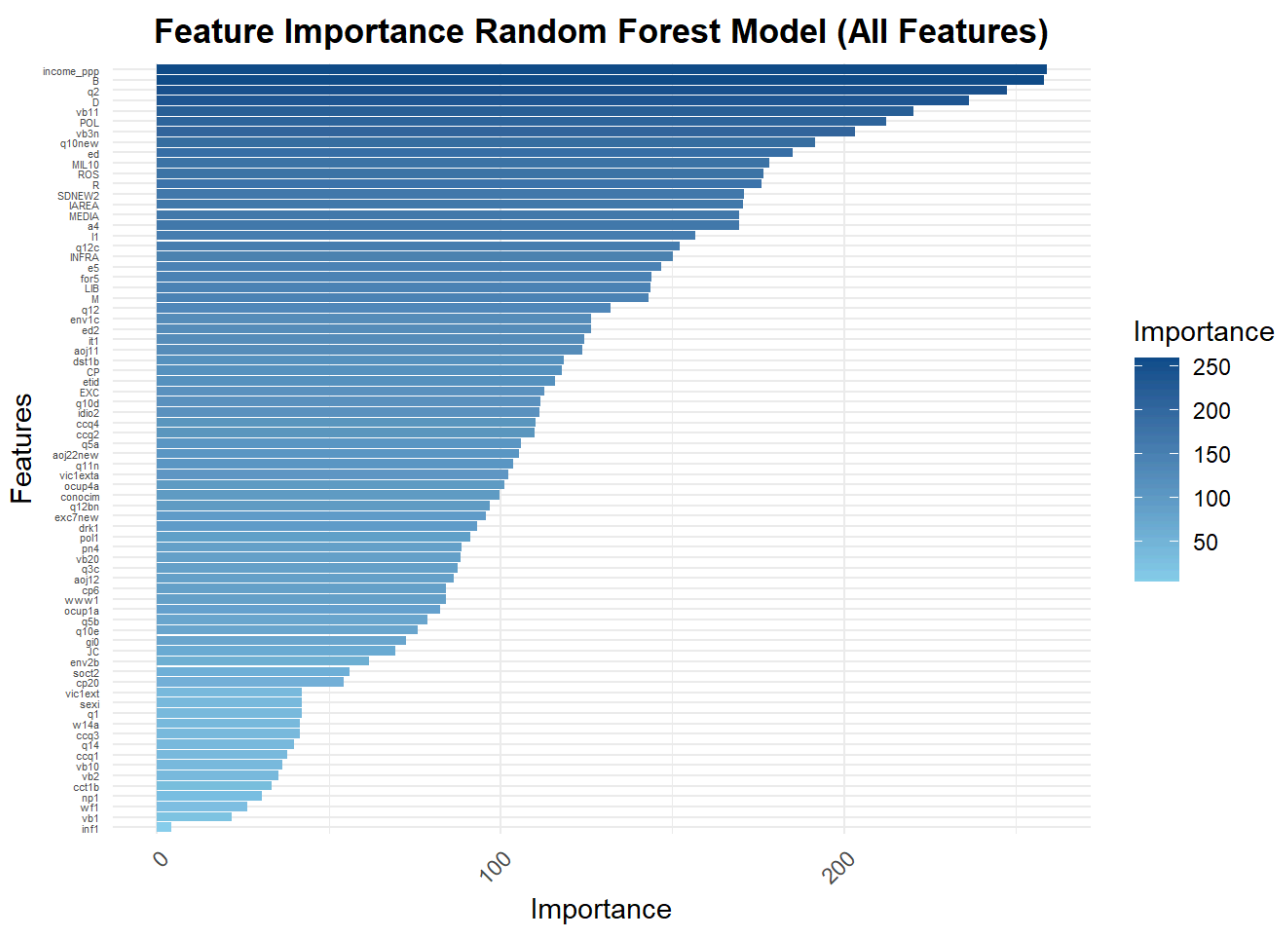
Top 11-16: 5. political persuasion: ROS, MEDIA 7. (entire category): a4 7. safety of the area: IAREA 7. urbanization: SDNEW2 7. housing condition: R

The above table categorizes the variables based on their corresponding question codes within the subcategories of the 7 main categories outlined in the beginning of the report. This classification allows for the identification of the real-life factors that are most influential in predicting life satisfaction.

To enhance visibility and comprehension, the feature importance will now be visualized through a plot, showcasing the relative importance of each feature in the random forest model.

▼ Code

```
# Plot feature importance with colors and adjusted text
plot_3 <- ggplot(importance_df, aes(x = reorder(Feature, Importance),
                                    y = Importance, fill = Importance)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  xlab("Features") +
  ylab("Importance") +
  ggtitle("Feature Importance Random Forest Model (All Features)") +
  scale_fill_gradient(low = "skyblue", high = "dodgerblue4") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.text.y = element_text(size = 4, hjust = 1),  # Adjust text size
        plot.title = element_text(hjust = 0.5, face = "bold"))

print(plot_3)
```



Assumptions Analysis 2

One of the critical assumptions underlying machine learning models is multicollinearity. This refers to the condition where variables are highly correlated with each other, potentially biasing the model's results. Therefore the correlation between variables will be examined in this section.

- No Perfect Multicollinearity: Ensuring no aliased variables.
- Multicollinearity: Checking for high variance inflation factors (VIF) among variables.

These checks are essential to maintain the integrity and reliability of the machine learning models used in this analysis.

Making Backup for Assumptions

Because the machine learning models may need to be run again after performing the assumptions, a second backup of the dataset will be created. This ensures the output of the assumptions remains unchanged while effectively handling any necessary adjustments.

▼ Code

```
# Backup the original dataset
south_america_dataset_ML_2 <- south_america_dataset_subset

# Define columns to be used for the model, including 'ls3'
columns_used_2 <- c("ls3", setdiff(names(south_america_dataset_ML_2),
                                   c("pais", "uniq_id")))

# Select the relevant data
impute_data_2 <- south_america_dataset_ML_2 %>%
  dplyr::select(all_of(columns_used_2))

# Convert all columns to appropriate types (numeric) for imputation
# Ensure that 'ls3' is treated as a factor
impute_data_2 <- impute_data_2 %>%
  mutate(across(where(is.numeric), as.numeric)) %>%
  mutate(ls3 = as.factor(ls3))

# Perform imputation using mice with the pmm method
imputed_data_2 <- mice(impute_data_2, m = 1, method = 'pmm', seed = 123)
```

```
 iter imp variable
  1   1  ls3  q10g  q10new  q10d  q10e  local_currency  income_usd  income_ppp  q2  q1  sexi
inf1  etid  vb1  vb2  ed  ed2  conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a
e5  D  LIB  EXC  exc7new  it1  aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new
MEDIA  vb3n  vb10  vb11  pol1  vb20  q5b  q3c  q11n  q12  q12m  q12f  q12c  q12bn  a4  soct2
wf1  cct1b  pn4  for5  drk1  env1c  env2b  vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2
ccq1  ccq2  ccq3  ccq4  q14  www1  R
  2   1  ls3  q10g  q10new  q10d  q10e  local_currency  income_usd  income_ppp  q2  q1  sexi
inf1  etid  vb1  vb2  ed  ed2  conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a
e5  D  LIB  EXC  exc7new  it1  aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new
MEDIA  vb3n  vb10  vb11  pol1  vb20  q5b  q3c  q11n  q12  q12m  q12f  q12c  q12bn  a4  soct2
```

```
wf1  cct1b  pn4  for5  drk1  env1c  env2b  vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2
ccq1  ccq2  ccq3  ccq4  q14  www1  R
  3   1  ls3  q10g  q10new  q10d  q10e  local_currency  income_usd  income_ppp  q2  q1  sexi
inf1  etid  vb1  vb2  ed  ed2  conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a
e5  D  LIB  EXC  exc7new  it1  aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new
MEDIA  vb3n  vb10  vb11  pol1  vb20  q5b  q3c  q11n  q12  q12m  q12f  q12c  q12bn  a4  soct2
wf1  cct1b  pn4  for5  drk1  env1c  env2b  vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2
ccq1  ccq2  ccq3  ccq4  q14  www1  R
  4   1  ls3  q10g  q10new  q10d  q10e  local_currency  income_usd  income_ppp  q2  q1  sexi
inf1  etid  vb1  vb2  ed  ed2  conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a
e5  D  LIB  EXC  exc7new  it1  aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new
MEDIA  vb3n  vb10  vb11  pol1  vb20  q5b  q3c  q11n  q12  q12m  q12f  q12c  q12bn  a4  soct2
wf1  cct1b  pn4  for5  drk1  env1c  env2b  vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2
ccq1  ccq2  ccq3  ccq4  q14  www1  R
  5   1  ls3  q10g  q10new  q10d  q10e  local_currency  income_usd  income_ppp  q2  q1  sexi
inf1  etid  vb1  vb2  ed  ed2  conocim  ocup1a  ocup4a  np1  CP  cp20  cp6  q5a  idio2  w14a
e5  D  LIB  EXC  exc7new  it1  aoj12  B  INFRA  MIL10  gi0  l1  JC  M  ROS  POL  aoj22new
MEDIA  vb3n  vb10  vb11  pol1  vb20  q5b  q3c  q11n  q12  q12m  q12f  q12c  q12bn  a4  soct2
wf1  cct1b  pn4  for5  drk1  env1c  env2b  vic1ext  vic1exta  aoj11  dst1b  IAREA  SDNEW2
ccq1  ccq2  ccq3  ccq4  q14  www1  R
```

▼ Code

```
# Extract the complete data after imputation
south_america_complete_data_2 <- complete(imputed_data_2)

# Print the first few rows of the complete data to verify
print(head(south_america_complete_data_2))
```

```
  ls3 q10g q10new q10d q10e local_currency income_usd income_ppp q2 q1 sexi
1   2    3      5    2    2         375000  127.05447   50.82179 68  2    2
2   4    4      4    3    2            135  245.63864  318.32733 51  2    1
3   4    1      1    3    1           6000  313.47962   30.49307 37  2    1
4   4   10     11    1    1         910000  308.31885  123.32754 25  1    1
5   3    1      1    3    2           1000  219.43574  279.78056 36  2    2
6   2    1      0    4    3         225000   76.23268   30.49307 64  2    2
  inf1 etid vb1 vb2 ed ed2 conocim ocup1a ocup4a np1   CP cp20 cp6 q5a idio2
1    1    2   2   1  5   1       4      4      1   2 4.00    4   4   2     2
2    1    1   2   2 11   4       5      4      3   2 4.00    4   4   4     3
3    1    2   1   1  5   1       5      4      5   2 3.33    4   2   1     3
4    1    1   1   1  9   1       2      4      1   2 2.67    4   2   2     1
5    1    3   2   2  5   0       3      4      5   2 4.00    4   1   2     2
6    1    2   1   1  3   0       4      4      1   2 2.00    4   3   3     3
  w14a e5    D LIB EXC exc7new it1 aoj12    B INFRA MIL10 gi0 l1  JC   M ROS
1    2 10 3.17 1.0   0       5   3     2 3.31     5  3.50   2  5 2.0 3.0 4.0
2    1 10 2.67 1.5   0       5   4     1 2.69     5  4.00   2  1 1.5 4.5 7.0
3    1  1 1.67 1.0   0       2   3     1 3.08     5  4.00   5  3 1.0 3.0 6.5
4    2  5 3.33 1.0   0       5   1     3 3.31     6  2.25   1 10 2.0 3.0 5.0
5    2  1 1.00 1.5   0       4   1     2 4.38     5  1.50   2 10 2.0 2.0 5.0
6    2 10 7.00 2.0   0       5   2     1 4.17     2  3.00   1 10 2.0 3.0 6.5
   POL aoj22new MEDIA vb3n vb10 vb11 pol1 vb20 q5b q3c q11n q12 q12m q12f q12c
1 4.00        5   4.0  804    2  801    4    1   2   1    1   3    2    1    6
2 4.50        7   1.0  804    2 1104    4    1   1   1    6   3    0    3    1
3 5.33        5   6.0  804    2 1401    4    1   1   5    2   3    1    2    4
```

```
4 5.50        7   3.5  804    1  801    3    2   1   5    3   0    0    0    4
5 5.00        7   4.5 1101    2 1002    4    1   1   1    3   4    2    2    8
6 6.67        7   5.0  877    2 1102    3    4   2   1    6   5    3    2    1
  q12bn a4 soct2 wf1 cct1b pn4 for5 drk1 env1c env2b vic1ext vic1exta aoj11
1     0 30     3   2     2   2    1    3     7     1       2        1     2
2     0 59     3   2     2   4   12    4     4     1       2        3     2
3     1 30     3   2     1   4    4    3     6     1       1        4     4
4     2  1     1   2     1   3    2    3     5     2       2        1     1
5     2 18     3   2     2   2    3    1     4     3       2        1     2
6     0  1     3   1     1   3    2    4     1     1       2        1     3
  dst1b IAREA SDNEW2 ccq1 ccq2 ccq3 ccq4 q14 www1    R
1     5  1.83   3.00    2    1    2    2   2    5 0.00
2     7  2.00   2.33    1    4    2    5   2    1 0.00
3     7  1.17   3.33    2    4    2    4   1    5 0.09
4     6  1.67   3.00    1    4    1    3   2    3 0.09
5     3  2.00   4.00    2    3    2    2   2    5 0.09
6     7  1.83   3.00    2    3    2    4   2    5 0.09
```

No Perfect Multicollinearity

For the first part of the assumption for machine learning models, a code will be executed to identify variables exhibiting perfect multicollinearity. If such multicollinearity exists, adjustments will be necessary in the model code to mitigate its effects.

▼ Code

```r
# Make backup of data that is going to be used and changed during the assumption
south_america_complete_data_backup <- south_america_complete_data_2

# Fit a linear model to detect aliased coefficients using the complete data
fit_model <- lm(ls3 ~ ., data = south_america_complete_data_backup)

# Check for aliased coefficients
aliased <- alias(fit_model)$Complete

# If there are aliased coefficients, print them
if(length(aliased) > 0) {
  cat("Aliased coefficients detected:\n")
  print(aliased)
}
```

```
Aliased coefficients detected:
     (Intercept) q10g q10new q10d q10e local_currency income_usd income_ppp q2
q12f  0           0    0      0    0    0              0          0          0
     q1 sexi inf1 etid vb1 vb2 ed ed2 conocim ocup1a ocup4a np1 CP cp20 cp6 q5a
q12f  0  0    0    0    0   0   0  0   0       0      0      0   0  0    0   0
     idio2 w14a e5 D  LIB EXC exc7new it1 aoj12 B  INFRA MIL10 gi0 l1 JC M  ROS
q12f  0     0    0  0  0   0   0       0   0     0  0     0     0   0  0  0  0  0
     POL aoj22new MEDIA vb3n vb10 vb11 pol1 vb20 q5b q3c q11n q12 q12m q12c
q12f  0   0        0     0    0    0    0    0    0   0   0    1  -1    0
     q12bn a4 soct2 wf1 cct1b pn4 for5 drk1 env1c env2b vic1ext vic1exta aoj11
q12f  0     0  0     0   0     0   0    0    0     0     0       0        0
```

```
      dst1b IAREA SDNEW2 ccq1 ccq2 ccq3 ccq4 q14 www1 R
q12f  0     0     0      0    0    0    0    0   0    0
```

It is evident that q12f is perfectly correlated with q12m (correlation of -1), and each is perfectly correlated with q12. This correlation arises because q12f (number of daughters) and q12m (number of sons) together perfectly predict q12 (total number of children), aligning with the survey questions in the LAPOP dataset. Consideration should be given to removing q12f and q12m when evaluating model performance. This poses challenges for model estimation and interpretation:

- Inflates Standard Errors: Coefficient estimation becomes unstable due to inflated standard errors, complicating the assessment of predictor significance.
- Impairs Model Interpretation: The model struggles to distinguish between the effects of q12f and q12m individually, leading to ambiguous coefficient interpretations.

These issues underscore the importance of addressing perfect multicollinearity to ensure reliable and interpretable results from machine learning models.

VIF Check

Following the removal of q12f and q12m, the Variance Inflation Factor (VIF) check is conducted to assess correlations among remaining variables.

▼ Code

```r
# Remove specific columns that are known to produce aliased coefficients.
columns_to_remove <- c("q12f", "q12m")
data_without_multicollinearity <-
  south_america_complete_data_backup[, !(names(south_america_complete_data_backup)
                                         %in% columns_to_remove)]

# Convert factor variables to numeric dummy variables
data_without_multicollinearity <- data_without_multicollinearity %>%
  mutate(across(where(is.factor), as.numeric))

# Fit a linear model to calculate VIF using the adjusted data
fit_model_vif <- lm(ls3 ~ ., data = data_without_multicollinearity)

# Calculate VIF values
vif_values <- vif(fit_model_vif)

# Create a data frame of the VIF values
vif_df <- data.frame(
  Variable = names(vif_values),
  VIF = round(vif_values, 2)
)

# Print the VIF dataframe
print(vif_df)
```

```
                  Variable  VIF
q10g                   q10g 3.37
q10new               q10new 2.80
q10d                   q10d 1.57
```

```
q10e                     q10e 1.30
local_currency local_currency 1.18
income_usd         income_usd 7.14
income_ppp         income_ppp 7.08
q2                         q2 2.74
q1                         q1 1.32
sexi                     sexi 1.03
inf1                     inf1 1.05
etid                     etid 1.07
vb1                       vb1 1.37
vb2                       vb2 1.48
ed                         ed 1.98
ed2                       ed2 1.51
conocim                conocim 1.41
ocup1a                  ocup1a 1.50
ocup4a                  ocup4a 1.52
np1                        np1 1.13
CP                          CP 1.36
cp20                     cp20 1.17
cp6                       cp6 2.13
q5a                       q5a 2.37
idio2                   idio2 1.53
w14a                     w14a 1.09
e5                         e5 1.31
D                           D 1.45
LIB                       LIB 1.20
EXC                       EXC 1.20
exc7new                exc7new 1.31
it1                        it1 1.19
aoj12                    aoj12 1.28
B                            B 2.27
INFRA                    INFRA 1.27
MIL10                    MIL10 1.19
gi0                        gi0 1.14
l1                          l1 1.07
JC                          JC 1.11
M                            M 1.78
ROS                        ROS 1.22
POL                        POL 2.03
aoj22new              aoj22new 1.53
MEDIA                    MEDIA 1.25
vb3n                      vb3n 2.47
vb10                      vb10 1.23
vb11                      vb11 3.63
pol1                      pol1 1.40
vb20                      vb20 1.19
q5b                        q5b 1.43
q3c                        q3c 1.02
q11n                      q11n 1.25
q12                        q12 1.90
q12c                      q12c 2.02
q12bn                    q12bn 2.11
a4                           a4 1.52
soct2                    soct2 1.42
```

```
wf1                   wf1 1.18
cct1b                 cct1b 1.28
pn4                     pn4 1.35
for5                   for5 1.11
drk1                   drk1 1.09
env1c                 env1c 1.12
env2b                 env2b 1.12
vic1ext             vic1ext 1.15
vic1exta           vic1exta 1.07
aoj11                 aoj11 1.23
dst1b                 dst1b 1.13
IAREA                 IAREA 1.09
SDNEW2               SDNEW2 1.40
ccq1                   ccq1 1.38
ccq2                   ccq2 1.62
ccq3                   ccq3 1.36
ccq4                   ccq4 1.59
q14                     q14 1.20
www1                   www1 2.21
R                         R 2.11
```

Next, apply thresholds to identify variables that may still exhibit multicollinearity.

▼ Code

```
# Categorize VIF values
high_vif <- vif_values[vif_values > 10]
high_vif_2 <- vif_values[vif_values > 5 & vif_values <= 10]
high_vif_3 <- vif_values[vif_values > 2.5 & vif_values <= 5]

# Create a data frame for high VIF values
high_vif_df <- data.frame(
  Threshold = c("VIF > 10", "VIF > 5", "VIF > 2.5"),
  Variables = c(paste(names(high_vif), collapse = ", "),
                paste(names(high_vif_2), collapse = ", "),
                paste(names(high_vif_3), collapse = ", "))
)

# Print as a formatted HTML table
kable(high_vif_df, format = "html", col.names = c("Threshold", "Variables")) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed",
                                      "responsive"),
                full_width = F) %>%
  column_spec(2, width = "50%")
```

Threshold	Variables
VIF > 10	
VIF > 5	income_usd, income_ppp
VIF > 2.5	q10g, q10new, q2, vb11

- Interpretation of VIF Thresholds:

VIF > 10: Indicates significant multicollinearity, suggesting variables highly correlated with each other. VIF > 5: Cause for concern, suggesting potential multicollinearity issues warranting attention. VIF > 2.5: Indicates considerable collinearity, which may impact model performance.

Upon reviewing the VIF results, no variables exhibit VIF values above 10, indicating no severe multicollinearity. Two variables show VIF values above 5, which is understandable given their similar nature. These variables will be handled by retaining only one in the machine learning models to mitigate multicollinearity challenges.

The multicollinearity assumption can be considered met after removing the same income variables and the q12f and q12m columns. This ensures that the analysis of machine learning models proceeds without issues during accuracy measures and feature importance evaluations. Adjustments to the model code will be made accordingly to maintain adherence to these assumptions and ensure reliable results.

Conclusion Analysis 2

Predicting an individual's well-being using 74 variables is inherently challenging. Achieving an accuracy of around 50% is quite high and can be considered acceptable in this context.

Insights from Analysis 1 on Analysis 2

Expanding from the initial analysis 1, additional variables were explored to evaluate their influence on life satisfaction. This exploration pinpointed a set of top 10 influential variables. While the top 10 variables showed a somewhat similar accuracy, the model's accuracy improved marginally by 0-6% when utilizing all 74 variables. This indicates that while these top 10 variables are influential, the remaining 64 variables also contribute significantly.

Categorization and Validation

Variables were categorized into seven distinct categories to better understand their impact. This categorization is based on the findings of the four referenced articles. Despite reducing the initial 180+ variables to 74 through factor analysis, certain significant factors like relaxation, recreation, and the vital nexus of nutrition, exercise, and sleep were not included in the dataset. Furthermore, comprehensive health-related questions, both physical and mental, were notably absent.

Recommendations

To enhance the accuracy of predicting subjective well-being (SWB), it is crucial to incorporate additional variables, especially those related to health. Studies, such as those using LISS data in the Netherlands, underscore the critical role of health in SWB. Factors like physical and mental health, as well as issues like addiction, are vital considerations. Health-related variables have the potential to significantly boost the model's accuracy. Incorporating those crucial missing variables could potentially elevate the model's accuracy to 70% or even 80%, signifying substantial performance improvements that are statistically meaningful, although this assertion remains speculative without further validation.

This recommendation is particularly relevant for LAPOP interviewers, emphasizing the importance of gathering comprehensive health-related data to enhance the accuracy of SWB predictions.

Final Thoughts

In conclusion, while the top 10 variables are pivotal, the inclusion of a more comprehensive dataset incorporating health and other missing variables could substantially enhance the predictive power of the model. This approach would lead to a deeper understanding of the factors influencing life satisfaction.

References

(2008 - JofEcoPsy - Do we really know what makes us happy - 1-s2.0-S0167487007000694-main) 1. income 2. personal characteristics (age, gender, ethnicity, personality) 3. socially developed characteristics (education, health, type of work, unemployment) 4. how we spend our time (hours worked, commuting, caring for others, community involvement and volunteering, exercise, religious activities) 5. attitude and beliefs towards self others life (attitude towards our circumstances, trust, political persuasion, religion) 6. relationships (marriage and intimate relationship, having children, seeing family and friends) 7. wider economic -social and political environment (income equality, unemployment rates, inflation, welfare system and public insurance, degree of democracy, climate and the natural environment, safety and deprivation of the area, urbanisation)

(2019 - JofHapSt Pontarollo2019_Article_TheDeterminantsOfSubjectivWel) - individual dimensions: age, sex, income, education, civil status, employment status, socio-economic status - latin-america extra variables: family, internet access, area (R/U), institutional trust, indigenous communities, religion, health insurance, housing conditions

(2019 - SocIndicRes Rogge-Nijverseel2019_Article_QualityOfLifeInTheEuropeanUnio) 8 life domains: financial situation, accommodation, job, commuting time, time we use, recreational and green areas, living environment, personal relationship 1 domain added: quality of life and meaning of life

(The pursuit of human weel-being 978-3-319-39101-4) variables: social, phsycological, environmental, cultural, work, spiritual, access basic needs

https://www.vanderbilt.edu/lapop/core-surveys.php

https://data.worldbank.org/indicator/

Appendix C: Canva Poster

This appendix includes a Canva poster presented in PDF format. The original file was in A1 size, but has been adjusted to fit the page width of this A4-sized article for optimal readability and presentation.

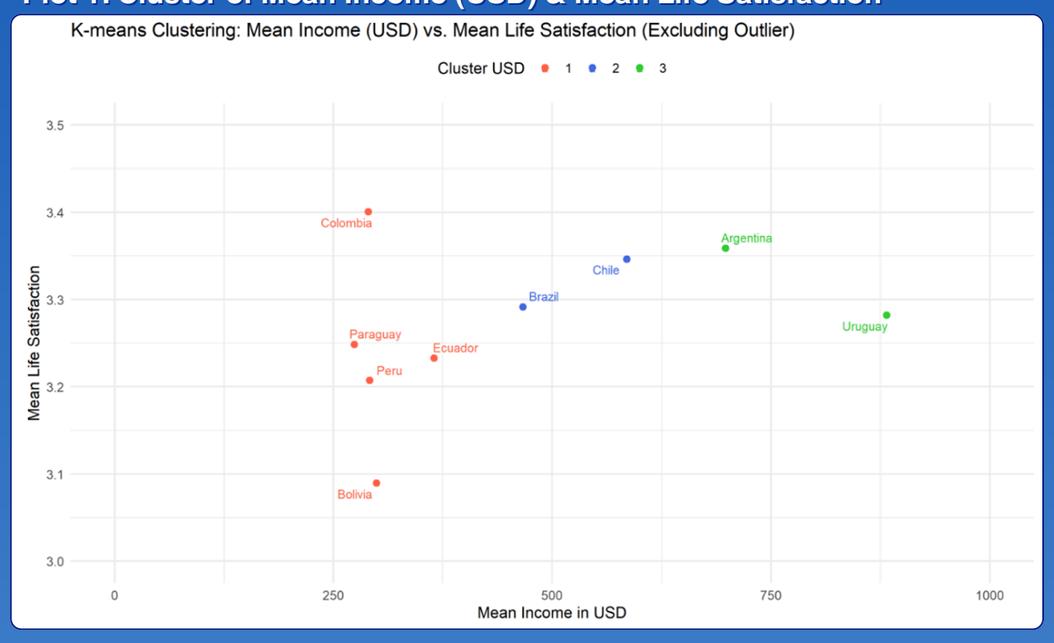
Predictive Analysis of Subjective Well-Being in South America

Author: Rodi Doesburg using Machine Learning Models **Student ID: 1347365**

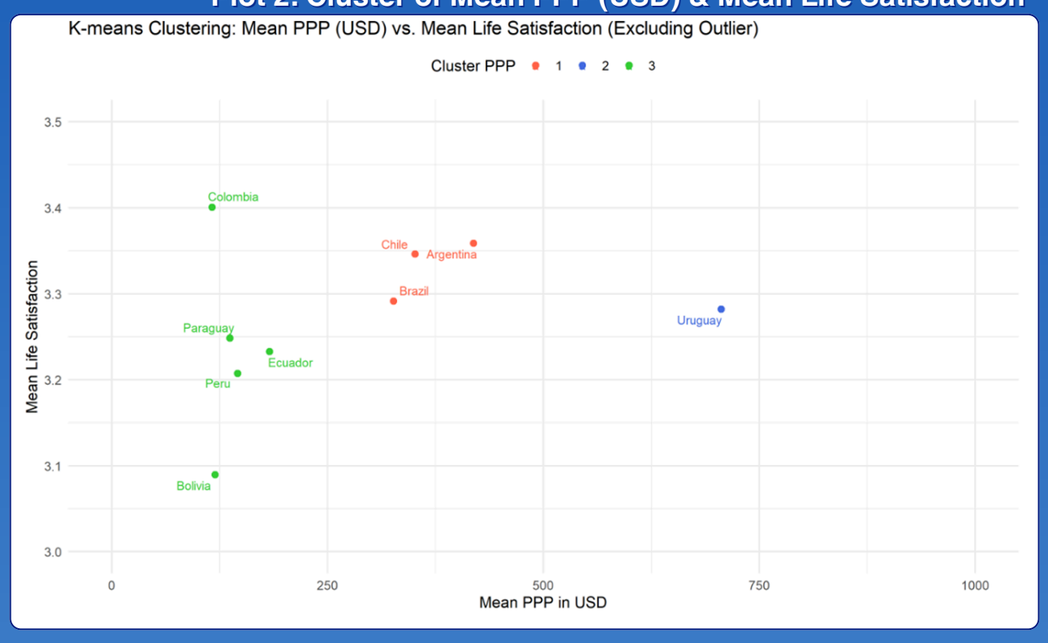
Examining the Influence of Income on Life Satisfaction

Cluster Analysis

Plot 1: Cluster of Mean Income (USD) & Mean Life Satisfaction



Plot 2: Cluster of Mean PPP (USD) & Mean Life Satisfaction



Plot 3: Average Silhouette Width (Income in USD)

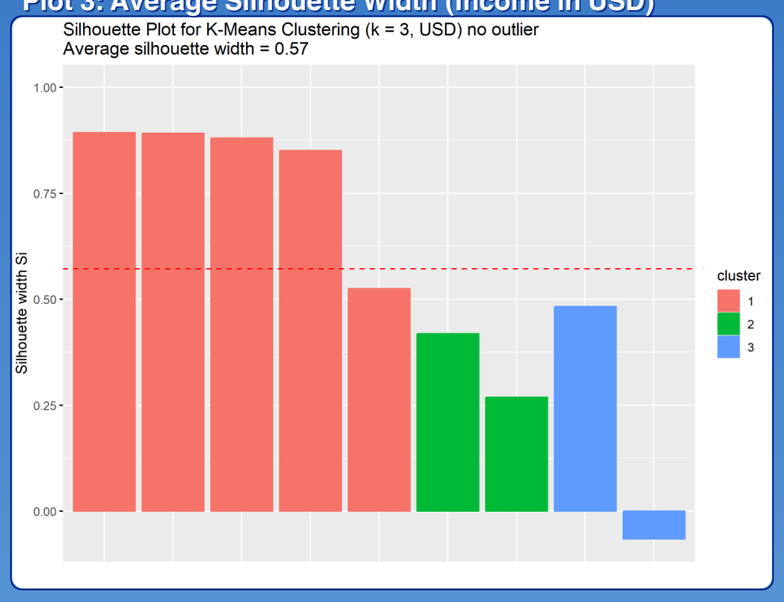
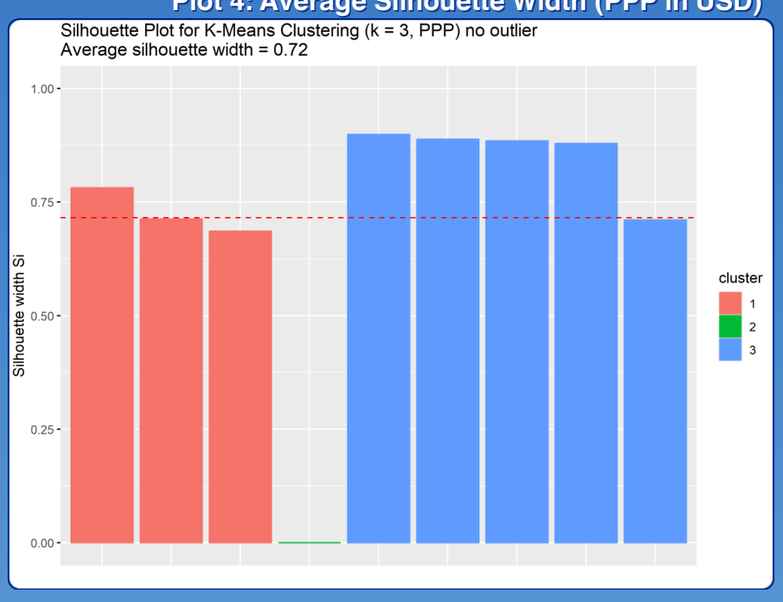


Table 1: Overall Trends (Income on LS)

Measure	Trend
Mean Income (USD)	0.374
Mean Income (PPP)	0.317

- **The overall trends of income in USD and PPP are quite similar, both moderately positive.**
- **Clustering analysis shows PPP has a higher average silhouette width, indicating better cluster quality.**
- **PPP is chosen for further analysis due to its higher silhouette score.**

Plot 4: Average Silhouette Width (PPP in USD)



Between-Country vs. Within-Country

- **Between-Country: The moderate positive trends (0.374 for USD, 0.317 for PPP) indicate that higher income generally corresponds to higher life satisfaction across South America.**
- **Within-Country: Individual country correlations are fairly low, with the highest being 0.202. This suggests that within countries, the impact of income on life satisfaction varies due to factors like cultural differences, social safety nets, and economic stability.**
- **Implications: While income has a moderate influence at a macro level, its impact is variable within individual countries, highlighting the need to consider various factors to improve life satisfaction.**

Table 2: Summary Statistics Table

Country	Mean LS	SE	Mean PPP (USD)	SE	Corr.
Argentina	3.36	(0.02)	418.69	(6.98)	0.130
Bolivia	3.09	(0.02)	119.65	(2.08)	0.119
Brazil	3.29	(0.02)	326.41	(6.49)	0.085
Chile	3.35	(0.02)	351.13	(5.76)	0.097
Colombia	3.40	(0.02)	115.93	(2.14)	0.074
Ecuador	3.23	(0.02)	182.43	(3.06)	0.091
Paraguay	3.25	(0.02)	136.82	(2.68)	0.182
Peru	3.21	(0.02)	145.69	(1.91)	0.104
Uruguay	3.28	(0.02)	705.56	(13.62)	0.202
Venezuela	3.11	(0.03)	4494.41	(65.78)	0.094

Identifying the Key Predictors of Life Satisfaction

Factor Analysis

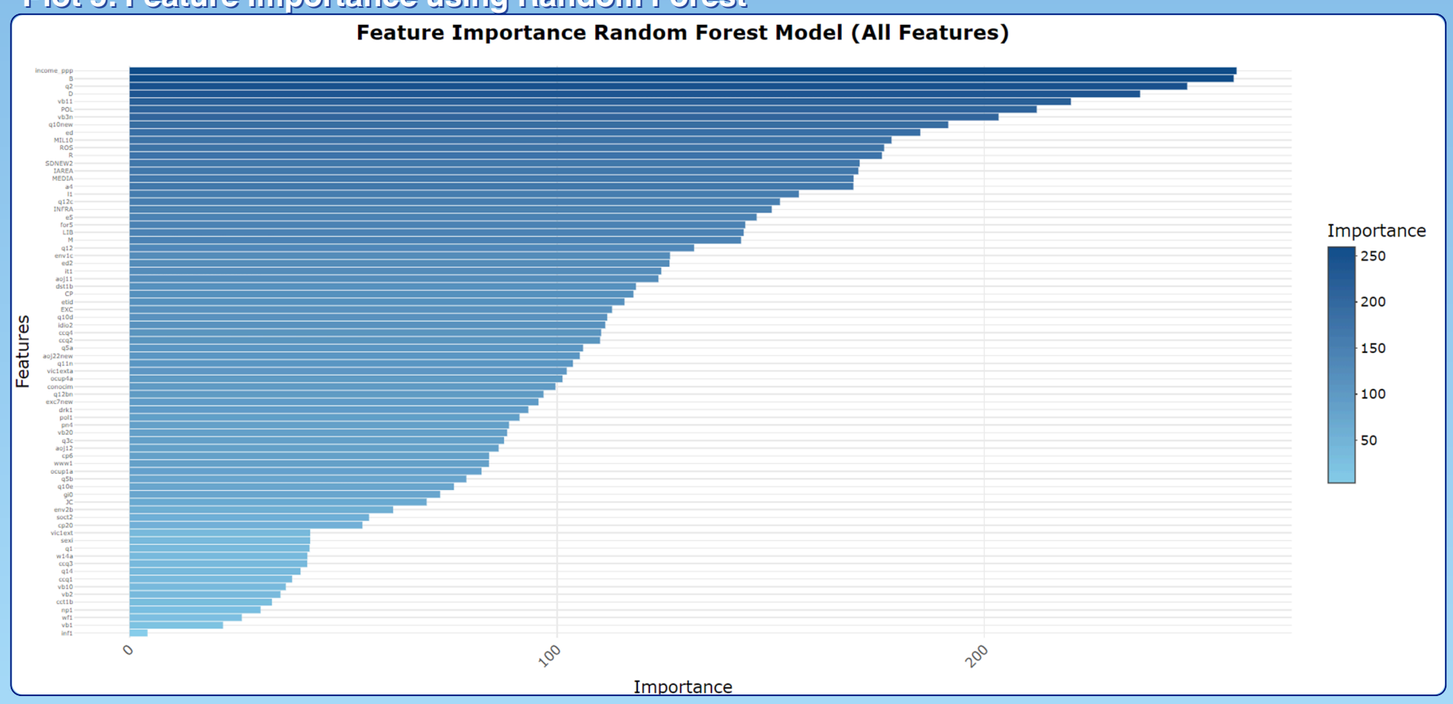
Table 3: Top 10 Feature Importance (RF)

Category	Features
Income	income_ppp, q10new
Age	q2
Education	ed
Attitude Towards our Circumstances	D
Social/Institutional Trust	B, POL, MIL10
Political Persuasion	vb3n, vb11

Table 4: Combined Accuracy Table

Model	Top 10 Accuracy	All Var Accuracy	Accuracy Difference (%)
Multinomial Logistic Regression	0.480	0.489	0.90
Decision Tree	0.469	0.475	0.60
Random Forest	0.466	0.531	6.42
XGBoost	0.264	0.264	0.09

Plot 5: Feature Importance using Random Forest



Overall Conclusion & Follow-up

- **Key Insights: Analysis of additional variables showed that the top 10 influential variables had comparable accuracy. Including all 74 variables only improved the model's accuracy by 0-6%.**
- **Missing Factors: The dataset lacked key factors such as relaxation, recreation, nutrition, exercise, sleep, and detailed questions about physical and mental health.**
- **Recommendation: Including physical and mental health metrics could significantly improve the model's accuracy, as health metrics can be influential for well-being.**
- **Conclusion: While the top 10 variables are important, adding health and other missing factors would increase predictive power and deepen our understanding of the influences on life satisfaction. This approach would advance both research and policy.**