Drought and effect of vegetation on aeolian transport in the Aralkum

Timon Smeets

Universiteit Utrecht

Drought and effect of vegetation on aeolian transport in the Aralkum

MSc Thesis

November-2020

Author: Timon Smeets

Student number: 5559650

E-mail: T.H.Smeets@students.uu.nl

First supervisor: Geert Sterk

Second supervisor: Gerben Ruessink

MSc Programme: Earth Surface and Water

Faculty of Geosciences

Department of Physical Geography

Utrecht University

Abstract

This study focusses on the effects of the desiccation of the Aral Sea, which occurred ~50 years ago. The desiccation of the Aral Sea began a shift in climate and transformed the Aral Sea. The Aral sea that once was the livelihood of many changed into the Aralkum, a dessert which to this day destroys the lives of many. This study focusses on the shift in climate that occurred due to the desiccating of the Aral Sea, the dust storms originating from the Aralkum, and vegetation strategies that can limit the magnitude of these dust storms. This study used precipitation data and satellite imagery to analyse the shift in the meteorological climate and the resulting changes in vegetation. Using the Aerosol optical depth and local wind speed data, the temporal variation of dust storms and high wind events were analysed and linked. The effect of different types of vegetation on reducing wind erosion in the form of suspension and saltation was modelled using a cellular automaton approach. The climate has shifted into a wetter but warmer climate. The regions affected by this are the rangelands in proximity(0-500 km) to the Aral sea, which experienced soil salination and a significant loss in vegetation density. The dust storms occur mainly in spring when high wind events are present. The modelling showed that to stop the erosion entirely, dense vegetation is required. Sparse vegetation will help reduce erosion, yet very sparse vegetation will create higher shear stress on the soil due to the formation of wakes increasing the erosion. Grass patches can protect the soil without creating wakes, making them the best option if possible. These storms have plagued the agriculture sector and massively increased respiratory problems in the general population of Uzbekistan jeopardising public health. Vegetation will play a key role in reducing salt and dust storms in Uzbekistan.

Table of contents

1 introduction

Land degradation is a natural hazard that destroys the soil, reduces the productivity of vegetation and creates an environment susceptible to erosion. Around 50 years ago the Aral Sea started shrinking, which caused vast swaths of land to degrade and is one of the most tragic environmental disasters of the 20th century. During the 1960s the Soviet Union set the goal to be self-sufficient in cotton production. To achieve this goal, an extensive network of irrigation channels and reservoirs was constructed in the arid Aral Sea region. The project doubled the irrigated area from 4 million to 8 million hectares and increased the water abstraction from the inflowing rivers (Micklin, 1988). However, the irrigated area was already increased from 1 to 4 million hectare in the first half of the 20th century, meaning the Aral Sra already was under stress before the Soviet irrigation plan began. The effect was that the inflowing rivers; Amu Darya and Syr Darya, which had an annual flow of 45-55 km^3 into the Aral Sea in 1960 had a yearly total inflow of only < 7 km^3 in 1980. The effect was a substantial decline in volume and surface area of the Aral Sea (Table 1) (Spoor, 2002).

Table 1 Decline in volume and surface area of the Aral sea (Spoor, 2002).

Year	Volume	Surface area	Water level
	km^3	km^2	m
1960	1090	66,900	53.4
1980	602	50,000	45.5
2000	183	24,155	32.4(south) / 41.0(north)

The decrease in water level split the Aral Sea in a southern and a northern part. In 2005 the World Bank, in combination with the Kazakhstani government, constructed a dyke in the strait between the two regions. The dyke stopped excess water from the northern lake being drained to the southern lake and partially restored the water level of the northern lake (Micklin, 2016). This remediated the environmental damage on the Kazakhstani side, yet did not do anything for the Uzbek side and might even worsen the situation (MacKay, 2009). The Uzbek government has meanwhile decided that there will be no attempt to restore the Aral Sea. This is partly because oil and gas have been found beneath the desiccated, former Aral Seabed. Another reason is that all irrigation farming would need to cease to exist. This would be dramatic for the food supply, especially since the population has increased from 8 million to 32 million in the last fifty years. The main focus of the government will be on reducing any adverse effects related to the created desert on the former Aral Seabed (Vinogradov et al., 2001; Novitskiy, 2012).

The desiccation of the Aral Sea has created the Aralkum desert with an area of approximately 57,000 km^2. The desert however, is different from other deserts in central Asia due to its origin. The topsoil of the Aral Sea has a higher concentration of evaporites (8-10%), with a total of about $6*10^9$ metric tonnes of salt (Letolle et al., 2005). The Aral Sea has been a sink for urban, industrial and agricultural untreated wastewater. This practice caused the soil of Aralkum to contain pesticides, heavy metals and lead. The high salt concentration has created puffy solonchak soils, which naturally are light soils that are susceptible to wind erosion and Aeolian dust transport, transforming the region into a source of Aeolian salt and dust storms. The drying up has caused the Aral Sea to change from a sink of $47*10^6$ ton salt per year to a source of $15*10^6$ ton salt and dust (Indoitu et al., 2012).

Due to mainly northern and north-eastern winds, the main transport direction of Aeolian dust is south, affecting Uzbekistan (Figure 1) (Issanova, 2015). Due to the increase in storms and the composition of the transported dust, the health of the populations in Uzbekistan has declined. A significant increase in cancer, tuberculosis and intestinal disorders was found since 1980 (Micklin, 2007). The land degradation due to salinisation partly caused by salt and dust storms in Uzbekistan is estimated to cost $1 billion per year (Sutton et al., 2007). The storms carrying salt and dust can travel up to 500 km, having devastating effects on the health and environment beyond the former shoreline (Saiko and Zonn, 2000).

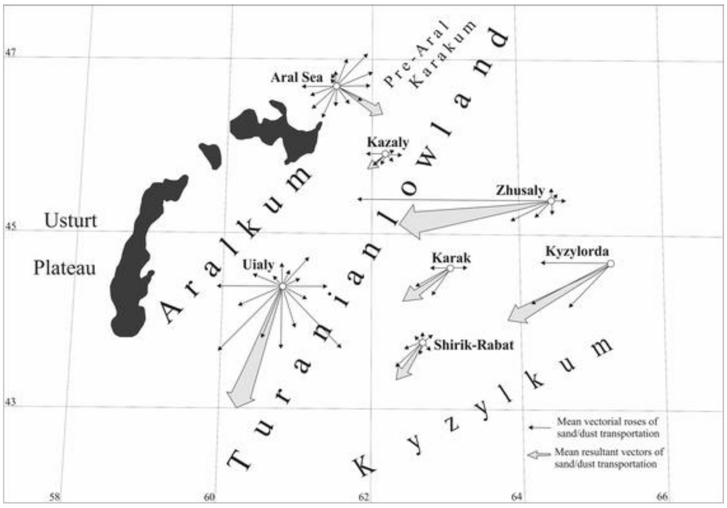


Figure 1 Major directions of aeolian dust transport from the Aralkum desert.

The dust storms are a major problem. In recent years (2000-2014) models have estimated that 18.3-56.5% of the Aeolian dust in Central Asia is due to anthropogenic causes, such as the Aralkum desert. The contribution of anthropogenic dust however, is decreasing, due to a net change in vegetation cover. The frequency and magnitude of dust storms are controlled by four conditions: Surface material susceptible to wind erosion, vegetation cover, strong winds and atmospheric conditions (Xu, 2006). Dust models have shown a dependency of the dust storm frequency on vegetation. Days with dust storms for different vegetation types were compared: desert or bare ground: 60-80 days/year, shrubs: 20-30 days/year, grassland regions: 2-4 days/year (Engelstaedter et al., 2003).

Vegetation plays a vital role in reducing Aeolian dust transport. Saxaul trees are a dominant tree species in the arid regions of central Asia. The trees are highly drought-resistant and able to hold up to 10 tonnes of soil around their roots. Due to their roughness, these trees can play an essential role in reducing airflow. Reduced airflow causes dust entrapment and minimises the chance of re-suspension. Saxaul forests with a medium canopy cover of 40% have been measured to trap 2 g m^{-2} day^{-1} (Heydarnezhad, 2014). The trees have however been on the decline in central Asia, because 1) they are a source of firewood, 2) drought and 3) desertification (Suvdantsetseg, 2008). Still, the Saxaul trees can play an essential role in the stability of the soil and reduce the transport of dust and salt.

Due to the importance of vegetation in reducing erosion, the Uzbek government has shown an interest in planting trees in the area. A plan for building Saxaul plantations throughout the dried former Aral Sea bed has been developed (Botman, 2009). The plan is to plant 2.5 million hectares of

Saxaul forest in Uzbekistan by 2030, of which a significant amount will be in desert areas. This vegetation is expected to play an essential role in stopping further land degradation.

There are multiple plans for planting vegetation: 1) Massive wood plantings in which shrubs, trees and grasses are planted closely together to create a pasture for cattle; 2) Local wood planting in which two strips at 10 m wide with three lines of 3 m each will be planted; 3) Patchy wood planting in which small patches of trees and shrubs are planted that produce new seeds which are transported by the wind up to 1000 m and thus create sparsely vegetated areas. The effects of these plantation strategies on wind erosion are currently unknown, and therefore further research is required to design and test optimal strategies to combat wind erosion (Novitskiy 2012).

In central Asia, there has been a decline in precipitation since 1960 (Qi and Kulmatov, 2008). The steady decline in precipitation is seen as an additional factor of desiccation of the Aral Sea, which has led to a decrease in evapotranspiration, and thus a reduced cooling of the area. Kazakhstan and Uzbekistan have due to the absence of this cooling experienced a warming of 2-4° Celsius (Guo., 2018). The higher temperatures have increased the potential evapotranspiration and thus the water stress of the vegetation. The increased water stress has resulted in a decline in vegetation cover in central Asia between 2000 and 2012 (Xu, 2012). The vegetation plantations planned by the Uzbek government will be hampered by the shift in climate. The water stress can be a considerable constraint on plant growth. When planning the vegetation patterns, the water demand of the vegetation will need to be factored in.

The decline in precipitation has effects on the presence of dust storms. Large amounts of precipitation exceeding 15 mm/day can suppress dust events in arid regions (Amgalan, 2017). Prolonged droughts decrease soil moisture. High soil moisture is one of the factors stabilising the soil and increasing the windspeed needed to erode the soil (Parolari, 2016). The droughts therefore aggravate the wind erosion in two ways, by decreasing soil moisture, and by reducing vegetation cover density.

Most of the research previously done on climate and vegetation change has focussed on central Asia as a whole. In contrast, this study will only focus on the Uzbek part of the Aralkum and Uzbek regions near the Aralkum. The goal of this study is to assess any trends in vegetation cover, the impact such trends will have on Aeolian salt and dust transport and to explore the effects of different vegetation planting strategies on wind erosion.

The following research question has been formulated:

How has the vegetation cover changed from 1980 till 2019, and how will this change impact the Aeolian erosion in the Aralkum?

To answer this question, the following sub-questions have been formulated:

- How has the vegetation cover changed, and what are local trends?
- How have changes in the climate affected the soil and vegetation?
- How will the different vegetation covers impact Aeolian salt and dust transport?

2 Study area

2.1 vegetation

Experimental projects to test the effect of the vegetation on wind-erosion are already ongoing. The tree and shrubs used for these experimental sites are Haloxylon Aphyllum (Saxaul) (Figure 2&3), Calligonum caput-medusae (Figure 4) and Salsola Richteri (Figure 5). These vegetation lines have proven effective at reducing wind erosion at medium-high wind velocities (Table 2) (Novitskiy 2012).

Table 2 Effect on erosion by vegetation (Novitskiy 2012).

Location for sand-blowing Measurements	Wind velocity $m\ s^{-2}$		
	2-5	6-10	11-16
	Amount of blown sand $g\ m^{-2}$		
Dune not covered with plants	226	294	326
Dune with 1-year-old plantation	196	265	289
Sandy valley with 30% grass coverage	156	214	254
Middle zone of 9-year-old wood plantation	38	85	109

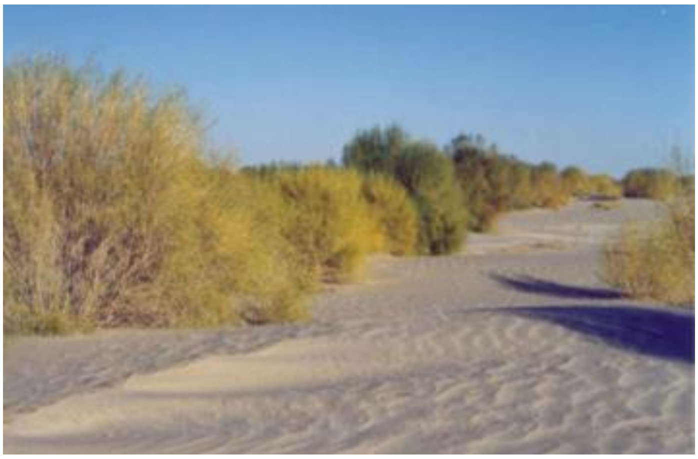


Figure 2 Saxaul (Haloxylon Aphyllum) 6 years old

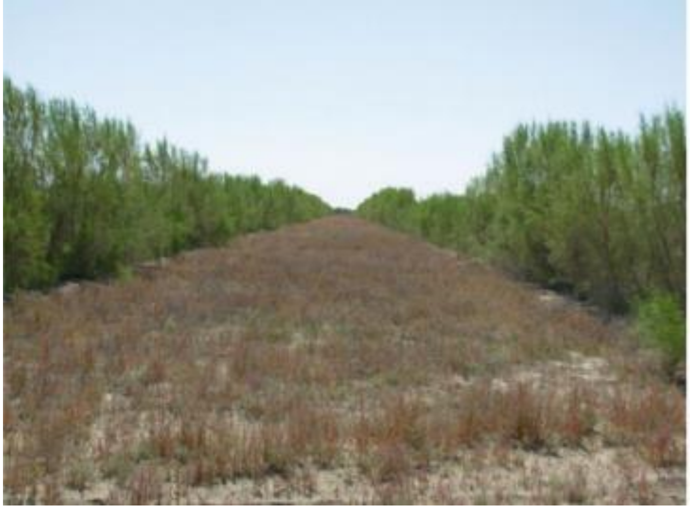


Figure 3 Sazaul (Haloxylon Aphyllum) 12 years old

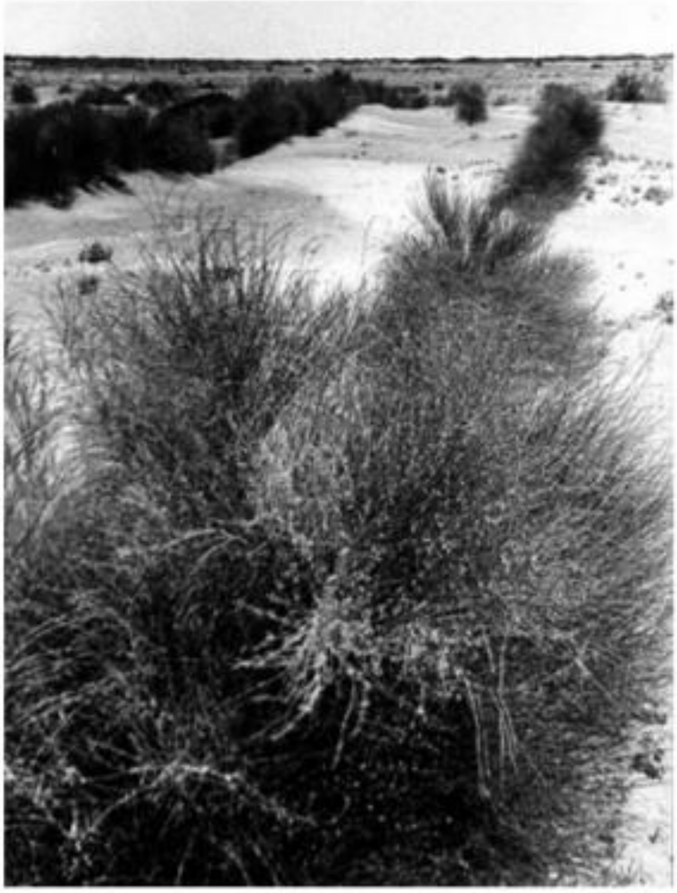


Figure 4 Calligonum caput-medusae 3 years old

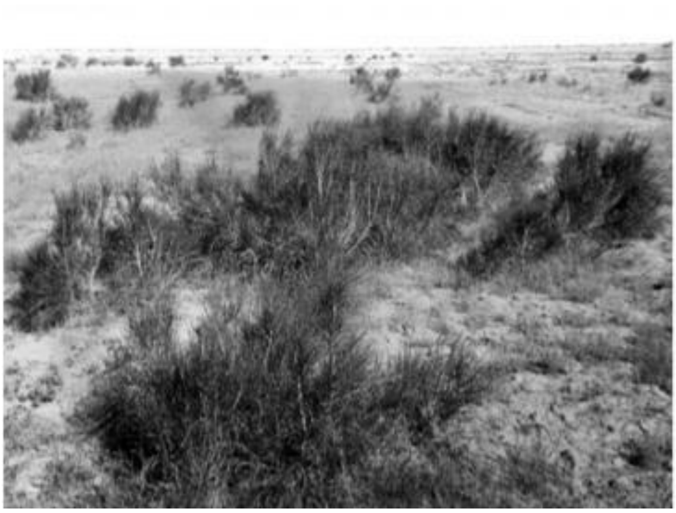


Figure 5 Salsola Richteri (3 years old)

The vegetation reduces wind erosion in three ways. 1)Through the capture of the sand particles in saltation or suspension. This is seen in Figure 6, where small sandy hills are created near the small shrubs as of the vegetation has trapped saltation material. 2) By absorbing momentum from the wind and reducing the shear stress on the soil. 3) By covering and protecting the soil surface (Figure 7) (Wolfe & Nickling, 1993). Vegetation elements extract momentum, produce turbulence in the form of wakes and help reduce large turbulent motions by breaking them up into smaller scale turbulent motions. The result is that single vegetation elements will slightly decrease momentum but increase turbulence resulting in a net increase in erosion. Multiple vegetation elements will however, result in the wakes not fully developing before another vegetation element is encountered, the wakes will then be broken up into smaller turbulent motions making them incapable of eroding the particles from the soil. This will result in greatly decreasing the turbulence and thus (Glendening, 1977).

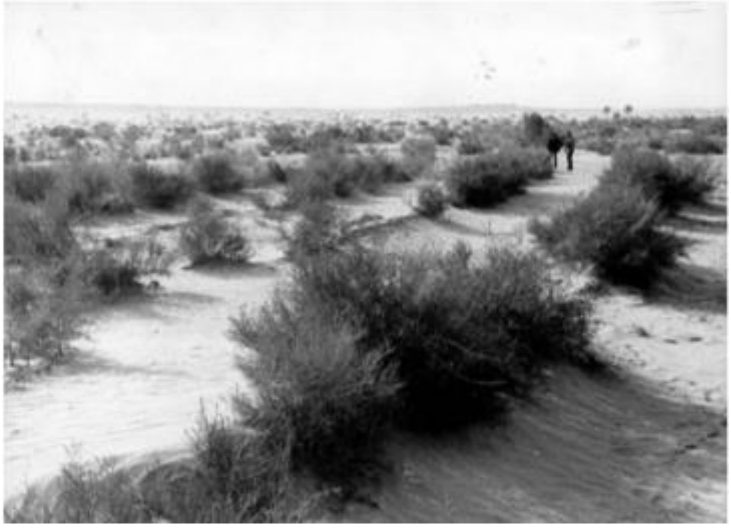


Figure 6 Sand intercepted by vegetation

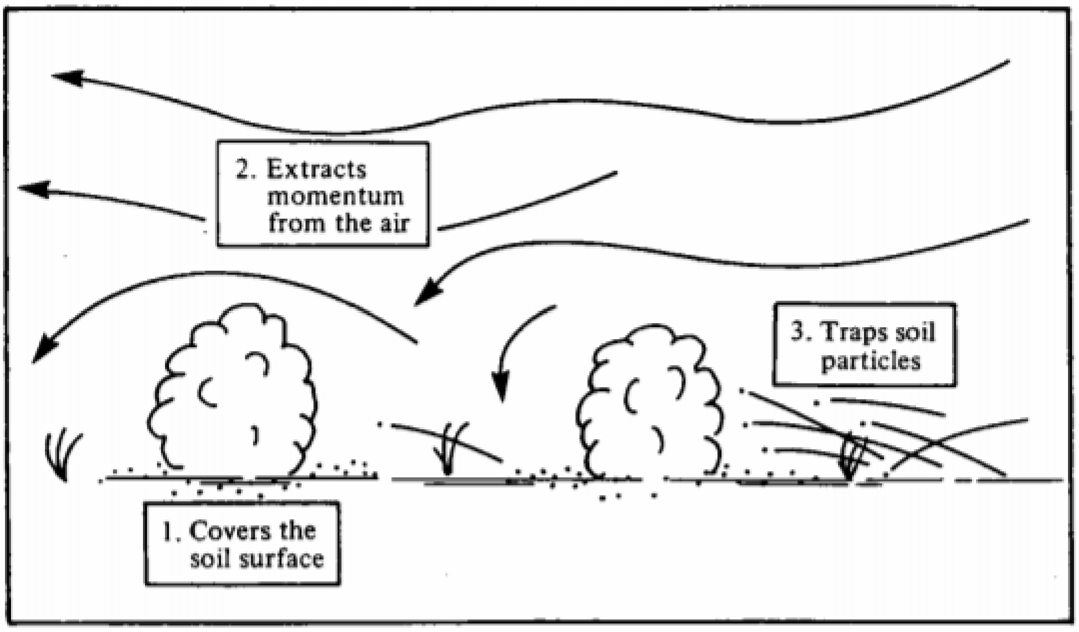


Figure 7 Different ways of vegetation reducing wind erosion

Due to the Northern winds, the main recipient of the dust and salt storms is the province of Karakalpakstan in Uzbekistan (Issanova, 2015) (Figure 8). The study area will therefore comprise of the desiccated Aral Sea bed and Karakalpakstan.

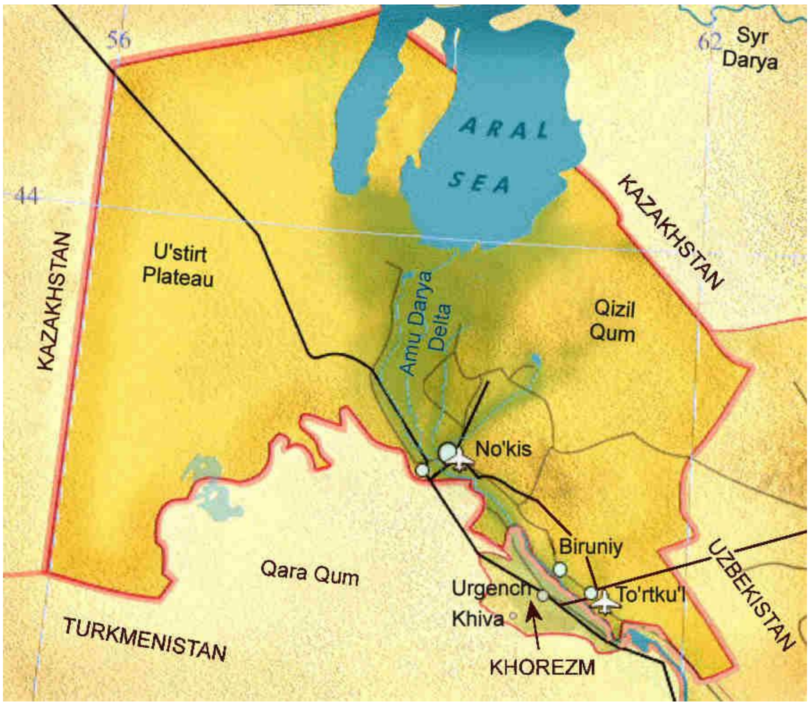


Figure 8 Karakalpakstan province within the red/pink borders.

2.2 climate

The study area has an arid continental climate, with average annual precipitation of 90-140 mm, with the bulk of the precipitation occurring in spring (Figure 9). The average annual potential evapotranspiration is 1100 mm, making the region severely water-limited (Tischbein et al., 2013). The minimum and maximum temperatures range between -40°C and 46 °C, with monthly averages ranging between -12°C and 35°C and a mean annual temperature of 9°C (Begdullayeva et al., 2007). In the winter air from Siberia in the north flows south, which has low humidity. In spring warm air from the Atlantic collides with the air from the north, causing wind storms and cyclones due to temperature differences (Figure 10). The average number of storm days per year varies between 15 and 45 days. The air from the west is humid, allowing for precipitation in these months. The summer is characterised by dry and warm air (Breckle et al., 2012).

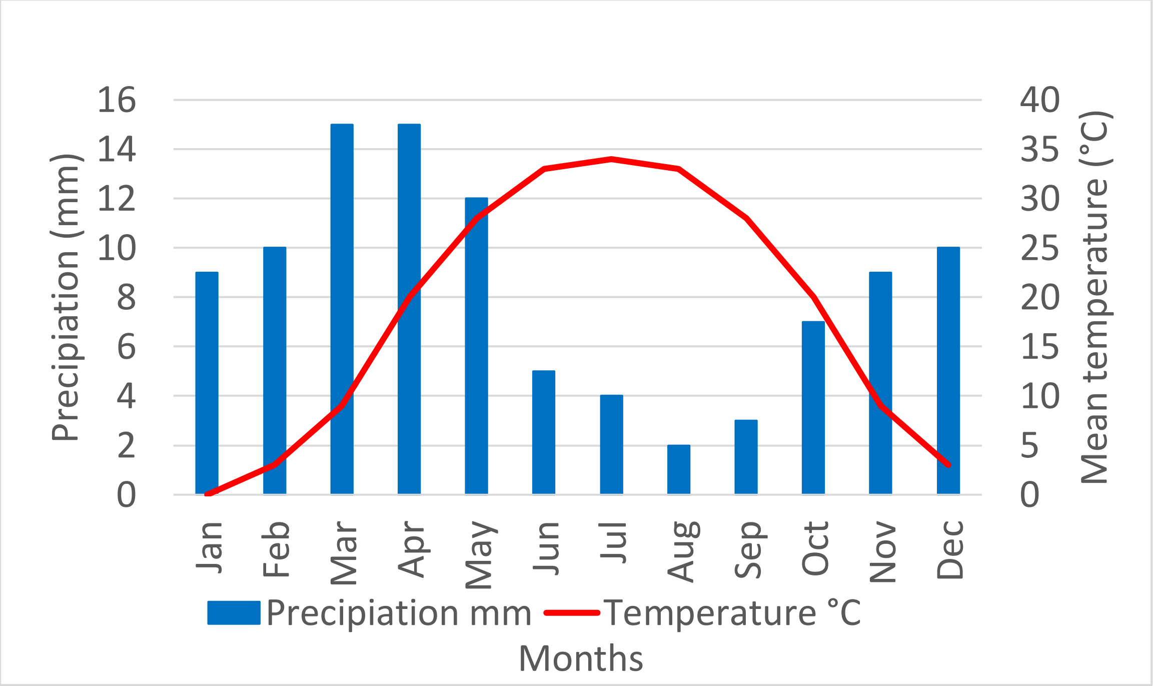


Figure 9 Climate Graph of the region south of the Aral Sea

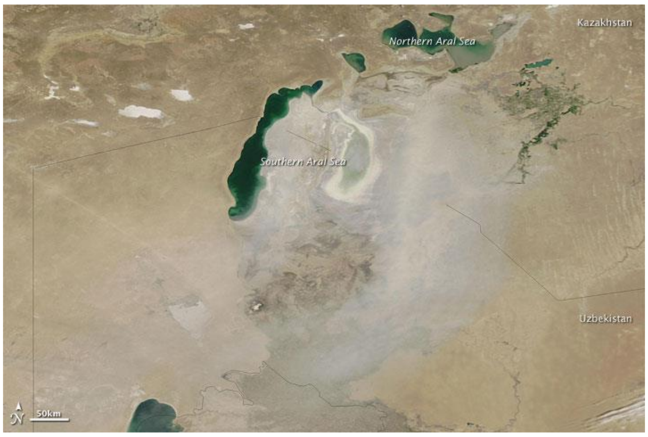


Figure 10 satellite image of a dust storm on 28 June 2009.

2.3 Soils

The lower reaches of the Amu-Darya river are fertile lands containing riparian woodlands and croplands. The region however, is experiencing salinisation, decreasing the soil productivity (Djumaeva et al., 2013). Salinisation of the soil has always been a problem in Uzbekistan as agriculture accounts for 25% of the GDP. The groundwater table is shallow, with a depth ranging from one to two meter in irrigated areas (Ibrakhimov et al., 2007). In 75.3% of Uzbekistan, the salinity of the groundwater ranges between 1.1-5.0 g/L, in 13.1% it is below 1.1 g/L, and in 10.5% the salinity is extremely high (between 5.0-10.0 g/L). Salinisation levels have slowly increased during the period 1996-2015. High salinisation of the groundwater and soil leads to a reduction in crop productivity, posing a threat to the economy (Kulmatov et al., 2020). When the Aral Sea desiccated and receded, it left a salt crust behind. Between 2000 and 2008, the total area of salt crusts has more than tripled (1,266.6 km^2 vs 4,561.7 km^2) (Figure 11). This dried up seabed has groundwater levels ranging between 0.5-10 m below the surface. Due to the groundwater's salinity and capillary rise soils with shallow groundwater stay saline. Soil with deep groundwater levels experiences downward seepage of salt due to precipitation. The soils in the region are therefore related to groundwater levels, as the water level drops the soil changes. When the groundwater level drops, the soil desalinises (Water > Salt crust > Saline Soil > Bare soil / sparsely vegetated soil) (Figure 12). The desalinisation slowly transforms the salt crust into Takyr soils (clay soils with a low concentration of salts) or sandy soils. These soils can accommodate vegetation. Through the changes in these soils, vegetation is being brought to the desiccated Aral Sea. In regions with deeper groundwater levels, the saline soil is still present, but a majority of the soil has changed into degraded sand and Takyr soil, which can accommodate vegetation (Stullina & Sektimenko, 2004).

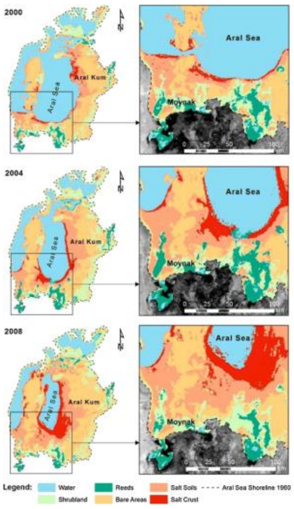


Figure 11 Increase in the area of saline soils near the Aral Sea between 2000 and 2008 (Low et al., 2013)

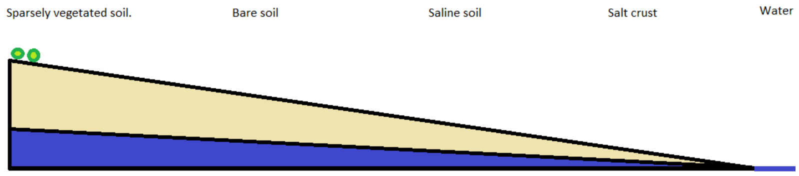


Figure 12 Change in soil with groundwater level

3 Methods

This research was conducted using a combination of remote sensing image analysis and wind sediment transport modelling. To assess previous droughts and changes in vegetation, data was gathered using Google Earth Engine. Scripts were written in R to analyse changes in vegetation and explore the link between these changes and occurrence of drought events. A long-term analysis of strong wind-events was made and compared with recorded dust events. The magnitude of sediment transport in relation to different vegetation densities and patterns was modelled. Multiple vegetated scenarios were created and tested against different storm classes.

3.1 Land-degradation assessment

3.1.1 Impacts of meteorological drought

A long-term trend analysis in temperature and precipitation was carried out. The meteorological changes play an important role in the shrinking of the Aral Sea. This trend analysis was carried out to understand the relationship between drought and meteorological changes in the region. The meteorological shift can drive desertification and changes in vegetation. A statistical analysis was made to accurately determine meteorological changes.

The Standardised Precipitation Index (SPI) (Mckee et al., 1993) and the Palmer Drought Index (PDSI) (Palmer, 1965) are the most widely used indices to assess meteorological change (Wang et al., 2015). The PDSI is prone to seasonality issues and requires more data than the SPI. Therefore, in this study, the SPI and the related SPEI indices were used.

The SPI is a meteorological drought index which is used to assess drought over multiple timescales (Mckee et al., 1993). Dry periods (1-24 months) in a timeframe of 30+ years can be identified with the SPI. Precipitation data is quantified as the standard deviation from the mean. The precipitation data is compared to the same month(s) in previous years. This allows it to compare SPI between different regions and times within the year. The prerequisite is however that a dataset of at least 30 years of monthly data is needed. The problem with the SPI is that it only measures the influx of water and not the outflux. It therefore does not take into account the temperature differences. The Standardised Precipitation Evaporation Index (SPEI) (Vincente-Serrano et al., 2010) was therefore developed, which was also used in this study. This index works similar to the SPI. However, instead of using precipitation, it uses precipitation minus potential evapotranspiration. This drought index was used to analyse the dry and wet periods within a time period with an influence of temperature.

The Terraclimate water balance dataset (Abatzoglou et al., 2015) was used to calculate the SPI and SPEI. The Terraclimate dataset has a 4 km spatial resolution and a monthly temporal resolution. The data range is 1958-2020. The dataset was built using a combination of satellite data and meteorological stations and uses the Penman-Monteith equation to estimate potential evapotranspiration. The SPI and SPEI were calculated for a period of 60 years; 1960-2020. The SPI and SPEI were calculated with precipitation and evapotranspiration aggregated for 1, 3 and 6 months.

Trends in precipitation were studied for different regions in Uzbekistan. The regions are: 1) Aral Sea bed desiccated between 1960-1990 and between 1990-2020 (Figure 13), 2) Irrigated lands within 0-300 and 300-500 km from the centre (Figure 14), 3) Rangelands within a buffer of 0-100, 100-300 and 300-500 km from the centre of the Aral Sea (Figure 15). The Aral sea regions were chosen to determine if there was a difference within the desiccated Aral Sea. The other regions were chosen to test the different changes in the rangelands and irrigated lands, and the effect of proximity to the desiccation zone. Trends in SPI and SPEI were studied between 1990-2020.

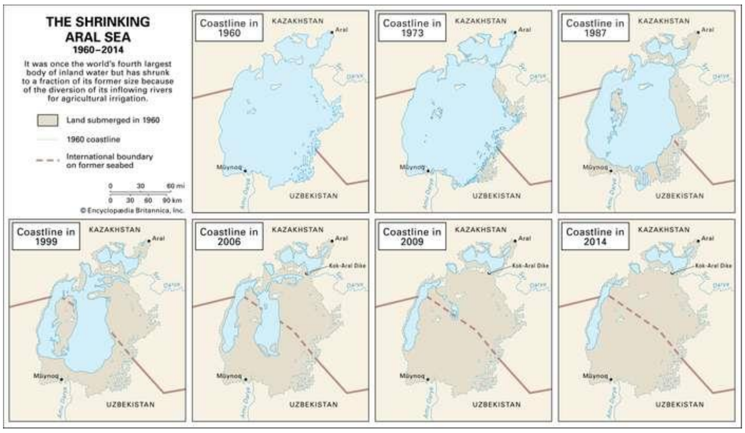

THE SHRINKING ARAL SEA
1960–2014
It was once the world's fourth largest body of inland water but has shrunk to a fraction of its former size because of the diversion of its inflowing rivers for agricultural irrigation.
Land submerged in 1960
1960 coastline
International boundary on former seabed
0 30 60 mi
0 30 60 90 km
© Encyclopædia Britannica, Inc.
Coastline in 1960
Coastline in 1973
Coastline in 1987
Coastline in 1999
Coastline in 2006
Coastline in 2009
Coastline in 2014
KAZAKHSTAN
UZBEKISTAN
Aral
Mŭynoq
Kok-Aral Dike

Figure 13 Change in surface area of the Aral Sea (Micklin., 2017).

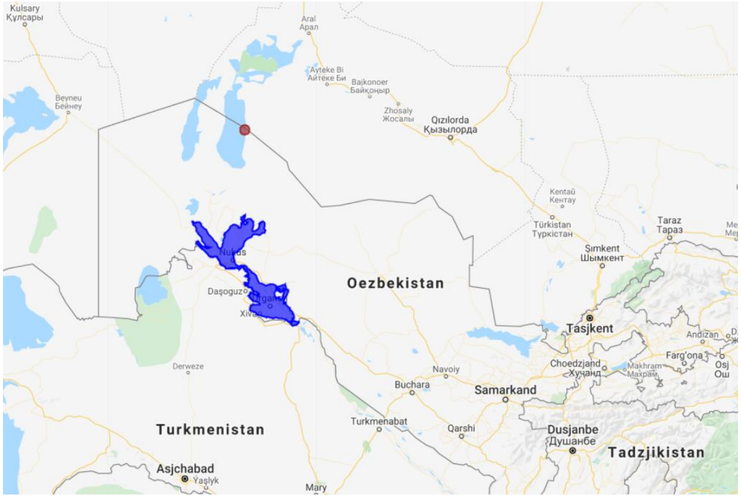


Figure 14 Irrigated lands in Uzbekistan within a 0-500 km buffer zone (black) from the centre of the Aral sea (brown point).

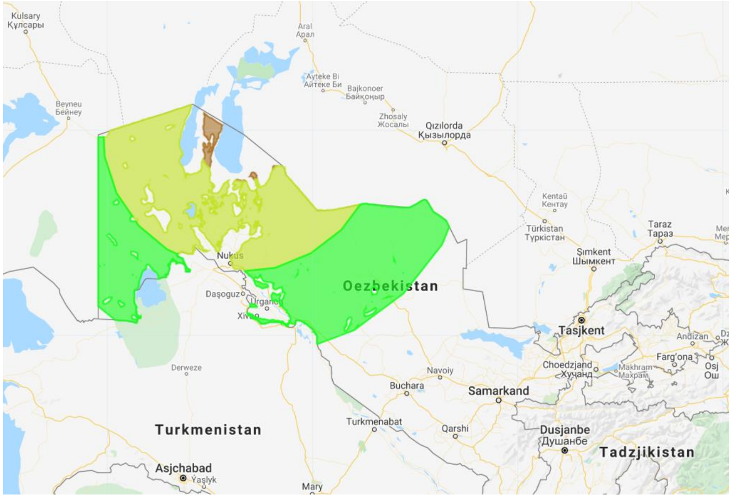


Figure 15 Rangelands in Uzbekistan within a 0-100 km buffer zone (brown), a 100-300 km buffer zone (yellow) and a 300-500 km buffer zone (green) from the centre of the Aral sea.

3.1.2 changes in vegetation

The analysis of vegetation cover changes was done in the same regions as the SPI analysis (Figure 13-15). The normalised vegetation index (NDVI) was used to assess changes in vegetation cover (Elmore et al., 2000). The NDVI is a vegetation index that is based on the difference between red and near infra-red light. Vegetation absorbs red light for photosynthesis and reflects near infra-red light as not to overheat. This gives a sharp change in reflectance value between red and infra-red at 700 nm. This sharp difference is a spectral signature for healthy and or dense vegetation (Tucker, 1979). Certain NDVI values correlate with certain surface covers. If a combination of different soils is present in a grid cell, the resulting NDVI will be in between the NDVI of these surface covers. In the Aral Sea region, these surface covers are present with the following NDVI (Shibuo et al., 2007):

- Water: -0.15
- Salt crust: -0.02
- Salt soil: 0.00
- Bare soil: 0.02
- Shrub land: 0.25
- Grass land: 0.30

The NDVI data were gathered in Google Earth Engine (GEE) using the NOAA CDR AVHRR NDVI product (Eric et al., 2014). This dataset has a 0.05 arc degree resolution and was used as it has continuous data from 1982 till present, with a daily temporal resolution. Other datasets have better spatial resolutions but are either only available since 2000 or contain large gaps. This would lead to using multiple datasets, creating large discrepancies within the final aggregated dataset.

The statistical significance of the trends was tested. The seasonal trends were removed from the NDVI dataset using the function decompose from the R package imputeTS. The decompose function uses moving averages to remove seasonality from the dataset. Linear regression was then used on the time-series without seasonality. The linear regression was then tested for significance using a *t*-test. The trend was deemed significant when the certainty exceeded 97.5%.

3.1.3 vegetation response

To assess the impact meteorological drought has on vegetation, the NDVI was converted into the Standardised Vegetation Index (SVI) (Peters et al., 2002). The SVI is calculated in the same way as the SPI. In contrast to the SPI, the SVI is usually not aggregated for multiple months but is used for a single month only. This is because droughts are a result of multiple relatively dry months and thus, drought can best be identified using longer time series. Vegetative drought or a decline in relative vegetation cover however, is about the current state of the vegetation and therefore best identified for the most recent month (Peters et al., 2002). For the period 1990-2020, the SVI-1 was calculated. The correlations between different SPI and SPEI aggregations and SVI-1 were calculated for different areas using the Pearson correlation coefficient. This indicates which areas are most sensitive to changes in the climate. To test the significance of the correlations, a *t*-test was done. A correlation was deemed significant when the certainty exceeded 97.5%.

3.2 Long term wind & aerosol assessment

To model sediment transport, historical wind erosion events with high wind velocities were evaluated. The occurrence of storms which exceed threshold wind speeds of 10, 15, 20, and 25 m/s (at 2 m height) for a 3-hour average were determined. These values were chosen since wind speeds of 10-15 m/s are needed before wind erosion starts on bare soil, with some variation depending on the soil type (Indoitu et al., 2012). The 20 and 25 m/s thresholds were used to determine the frequency of extreme wind events. To explore the variation in wind speed, 3-hourly data were used from the local meteorological stations in Kungrad and Jaslyk. In addition, the Integrated Surface Hourly Dataset of the National Oceanic and Atmospheric Administration (NOAA) (Lott et al., 2000) was used. This dataset contains average 3-hourly windspeed data, eight 3-hour periods a day, for the years 1980-2020. The data of strong wind velocity events was used for sediment transport modelling.

In this study, aerosols were taken as a proxy for dust in the air. Aerosols are a mixture of dust particles, liquid droplets or smoke and pollution in the air. Their size varies between 0.2-20 µm. Aerosol Optical Depth (AOD) is a measure of the extinction of solar radiation by aerosols. The aerosols can block sunlight by absorbing or by scattering light. AOD is a measure for the absorption or reflection of direct sunlight by these aerosol particles. It is a dimensionless number that is related to the amount of aerosol in the vertical column of the atmosphere over the observation location. A very clear sky would give an AOD of 0.01 while very hazy conditions would correspond to 0.4 AOD. The measured AOD values for the Aral Sea region were taken from the NASA MODIS/Terra + AQUA MAIAC dataset (Lyapustin et al., 2018). It has 1 km spatial resolution, daily temporal resolution and covers the period 2000-2020. A threshold analysis of the AOD was made to determine the months with high AOD. The selected thresholds were 0.2, 0.3 and 0.4 mean daily AOD. A correlation was calculated between AOD and the windspeed threshold exceedance to determine in which months high windspeeds will lead to high dust concentrations.

3.3 Aeolian transport modelling

3.3.1 The model

The aeolian transport model used in this study is based on a previous model (Kersten, 2015). This model was used to quantify the effects of scattered vegetation on sediment transport in the Sahel. The vegetation present in the Sahel consists of 0.5-2 meter high shrubs and trees of 3-25 meter height with a crown diameter of 2-20 meter breadth. The model was changed in this study into a cellular automaton model. Compared to the original model, a few equations were changed to better match the conditions in the Aral Sea region, and grass was added as a vegetation cover to the model of Kersten (2015).

The model calculates the effects of vegetation on wind velocity and sediment transport. Vegetation elements are modelled as obstacles with height, frontal area and drag. To quantify the vegetation, the model calculates the vegetation density. This is calculated as the ratio of the total vegetational frontal area divided by surface area (Figure 16). This dimensionless parameter is an indicator of the relative number of obstacles encountered by the wind.

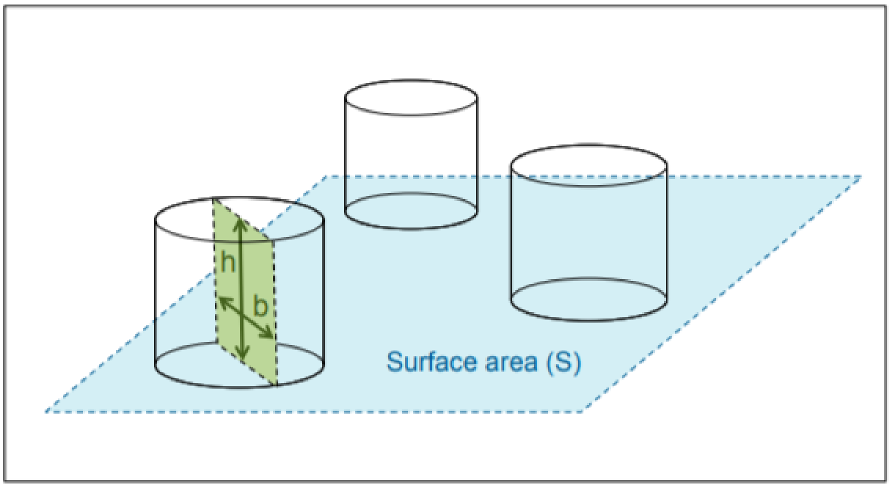


Figure 16 Frontal area and surface area (Mayaud., 2017)

The frontal area of vegetation (A; m^2) is calculated using the Breadth (b; m), porosity (n; fraction between 0 and 1) and height (h; m) of the vegetation. For trees the following equation is used:

$$A_t = \mathrm{b}_\mathrm{t}^2 * \mathrm{n}_\mathrm{t} + B * (h_t - b_t) \qquad 3.1$$

Where B (m) is the breadth of the trunk. For shrubs the frontal area is calculated as:

$$A_s = h_s^2 * \pi * \frac{n_s}{2} \qquad 3.2$$

And finally, for grasses the following equation is used:

$$A_g = h_g * b_g * n_g \qquad 3.3$$

Using the frontal area per vegetation type, the total surface area of the study area (S; m^2) and the quantity of vegetation (Q), the vegetation density (λ) is calculated per vegetation type Lettau (1969). For trees the following equation is used:

$$\lambda_t = \frac{A_t * Q_t}{S} \tag{3.4}$$

For Shrubs the vegetation density is calculated as:

$$\lambda_s = \frac{A_s * Q_s}{S} \tag{3.5}$$

And finally, for grasses the following equation is used:

$$\lambda_t = \frac{A_s * Q_s}{S} \tag{3.6}$$

To calculate the total vegetation density, the different vegetation densities are added.

$$\lambda_v = \lambda_t + \lambda_s + \lambda_g \tag{3.7}$$

Average vegetation height (h; m) is calculated:

$$h_v = \frac{h_t * \lambda_t + h_s * \lambda_s + h_g * \lambda_g}{\lambda_v} \tag{3.8}$$

Average vegetation drag (C) is then calculated:

$$C_v = \frac{C_t * \lambda_t + C_s * \lambda_s + C_g * \lambda_g}{\lambda_v} \tag{3.9}$$

To model wind sediment transport, the friction velocity is required. Friction velocity represents the shear stress at the surface, including bare soil and vegetation elements. Friction velocity in this model is dependent on average wind velocity(u; m/s), aerodynamic roughness length(z_0; m), the height of airflow (z; m) and von Karmann constant (k; 0.4) (Lettau et al., 1969). Aerodynamic roughness length is defined as “The height above the displacement plane at which the mean wind velocity becomes zero when extrapolating the logarithmic wind speed profile downward through the surface layer” (Figure 19). The roughness length represents the roughness of the surface. The roughness of the surface is created by roughness elements such as vegetation. In this model, the only roughness elements are vegetation elements. The roughness length is calculated using the average vegetation height and vegetation density (Wolfe & Nickling, 1993):

$$z_0 = 0.5 * h_v * \lambda_v \qquad 3.10$$

Using the roughness length, the friction velocity is calculated:

$$u_* = u * \frac{k}{\ln\left(\frac{z}{z_0}\right)} \qquad 3.11$$

The friction velocity increases with windspeed and roughness length.

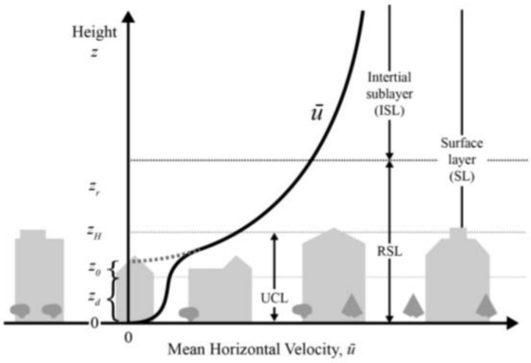


Figure 17 Illustration of roughness length (Bradford 2015)

The total shear stress (t_0; $g*m^{-1}*s^{-2}$) is dependent on the air density (ρ; g/m^3) and friction velocity:

$$t_o = {u_*}^2 * \rho \qquad 3.12$$

The vegetation absorbs a large part of the momentum of the wind. This is how the vegetation protects a sizable area of the ground. This is modelled by partitioning the shear stress. The shear stress is partitioned in two types of stress: Stress on soil and stress on vegetation (Raupach., 1992). The drag is partitioned using the drag ratio (r). The drag ratio is the ratio between vegetation drag and the soil drag coefficient:

$$r = \frac{C_v}{C_{soil}} \qquad 3.13$$

High vegetation densities will result in high stress on vegetation, reducing the stress on the soil. The drag ratio is then used to partition into the shear stress on the soil (t_s; $g*m^{-1}*s^{-2}$)) and the shear stress on vegetation. The drag on the soil can then be calculated:

$$t_s = \frac{t_o}{1 + r * \lambda_v} \qquad 3.14$$

Using the drag on the soil, the effective friction velocity on the soil (u_{*S}; m/s) is calculated.

$$u_{*S} = \sqrt{\frac{t_s}{\rho}} \qquad 3.15$$

The stress or drag on the soil is then used to model the sediment flux. The number of particles brought in saltation is dependent on the threshold friction velocity. Threshold friction velocity is the minimum velocity required to bring soil particles into motion. The particle comes into motion when the drag and lift forces exceed the gravitational and inter-particle cohesive forces. The threshold friction velocity is dependent on, gravity (g; 9.81 m/s^2) the mean grain size (d; m) and a few constants: A_n (0.0123), σ (2163) and f ($10^{-1.8}$).

$$u_{*t} = A_N \sqrt{\left(\sigma * g * d * f + \frac{\Upsilon}{\rho * f * d}\right)} \qquad 3.16$$

When the friction velocity on the soil exceeds the threshold friction velocity, the sediment flux is calculated. Calculating the sediment flux can be done with different equations to calculate the sediment transport. The model was tested with six different transport equations: Kawamura (1964), Kind (1976), Lettau and Lettau (1978), Maegley (1976) and Zingg (1953). For this study, the Kawamura (1964) equation was chosen, since it provided results closest to the expected values. The sediment flux (Q_t; g/m/s) is calculated using the drag on the soil, the threshold friction velocity and the Kawamura constant (C_{ka}; 0.14) (Kawamura., 1964).

$$Q_t = C_{ka} * \left(1 - \frac{u_t}{u_{*S}}\right) * \left(1 + \frac{u_t}{u_{*S}}\right)^2 * \frac{\rho * u^3{}_{*S}}{g} \qquad 3.17$$

3.3.2 CA modelling

The model made by Kersten (2015) is a lumped, field-scale model that represents the area as a single cell. This cell is homogenous, and there is no option to create spatially heterogeneous patterns of vegetation. To make this possible and more accurately model the vegetation, a cellular automata model was created. Cellular Automata (CA) modelling is a form of modelling which uses uniform discrete grid cells which interact and influence each other using predefined rules. Using these simple cells, more complex large-scale dynamics can better be modelled using only simple fundamental principles. The CA approach has shown its potential for analysing eco-geomorphic systems (Trunbull et al., 2014). Using CA requires more computational power than running a model for a single cell. To reduce runtimes the model has therefore been changed into a 1-d model instead of a 2-d model.

The model consists of multiple cells on a 1-dimensional strip. The wind direction of the cells is right to left. The model calculates the sediment mass flux per cell. The outflux of the cell is then calculated by combining the sediment mass flux of the current cell, and the influx of the upwind cell in the previous time-step. The influx in a cell (outflux of the previous cell) is reduced by two factors:

1) The vegetation density in the cell. This is to represent interception of the sand grains by vegetation (Wolfe & Nickling, 1993).

2) A constant to represent the deposition rate in bare soil (Mayaud et al., 2017).

These reductions are used to simulate the particles going out of the cell by suspension. The mass of particles going out by suspension in a 10-meter strip ranges from 20-100% depending on the vegetation density (Mayaud et al., 2017). This is calculated using the following formula:

$$Outflux_{x,t} = Erosionflux_{x,t} + Outflux_{x+1,t} * \left(0.8 - \lambda_{v,x,t}\right) \tag{3.19}$$

Where sediment flux is the sediment generated by the current cell and outflux$_{x+1}$ is the outflux from the previous cell

In CA modelling, each cell has an increased sediment flux compared to the previous model. This is because the outflux of the cell is dependent not only on the current cell but also on the influx of the previous cell. In bare soil this would mean that the sediment flux of each cell would increase to a maximum. This maximum would be five times the sediment output of a bare soil cell, since in bare soil sediment flux between cells is reduced by 20%. To compensate for this, the output of the model is calibrated using a calibration constant (C$_c$; 0.25).

$$Model\ output = Outflux_{1,t} * C_c \tag{3.20}$$

This CA model is able to model a heterogeneous field of patches of grass alternating with rows of shrubs/trees or bare soil.

3.3.3 Model input

To model sediment transport a number of input parameters is required. The single value parameters with base constants are listed in table 3.

Table 3 Model constants

Atmospheric conditions	Soil properties	Physical Constant
Air Temperature T (15 °C)	Surface area S (10 m^2)	Von Karmann constant k (0.4)
	Mean grain size d (65 µm)	Gravity g (9.81 m/s^2)
	Grain density ρ (2650)	Grain size constant A$_k$ (0.1)
	Soil drag coefficient C$_{soil}$ (0.0030)	Reference grain size D (0.25)

The vegetation parameters are listed in Table 4. The vegetation parameters are defined as shown in Figure 18. Porosity is here defined as the fraction of the frontal area which contains leaves.

Table 4 Vegetation parameters

Tree	Shrub	Grass
Height h_t (2m)	Height h_s (1m)	Height h_g (0.15m)
Breadth B_t (1.5 m)		
Breadth Bark B_t (0.2 m)		
Quantity Q (0)	Quantity Q (0)	
Porosity n_t (0.5)	Porosity n_s (0.5)	Porosity n_g (0.5)
Drag coefficient C_t (0.3)	Drag coefficient C_s (0.2)	Drag coefficient C_g (0.1)

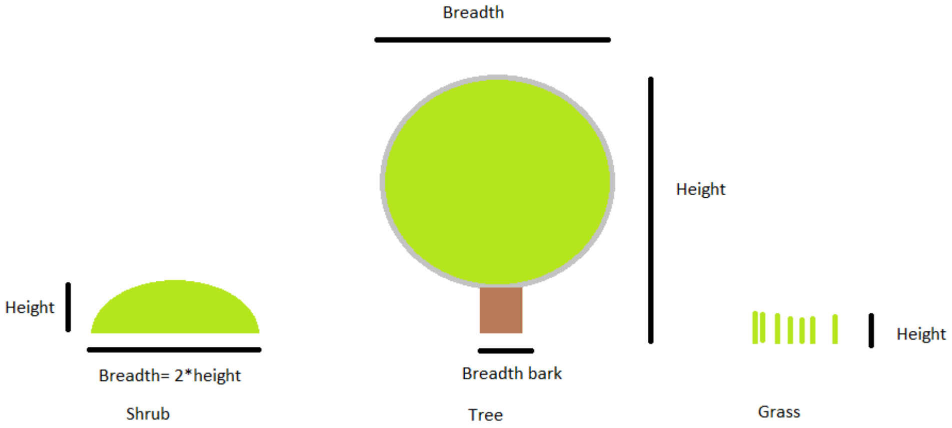


Figure 18 Vegetation parameters

For the Cellular automata model, multiple cells are created per hectare. For each cell the model is run. The cells are 10 m long in a one-dimensional plane with a width of 10 m. The wind direction in this plain is right to left. Each cell has its own vegetation properties (Figure 19). To investigate the effect of different vegetation patterns, six scenarios were chosen. The six scenarios ranged from bare soil to densely vegetated, and have a repeating vegetation pattern of 10 cells.

	1	2	3	4	5	6	7	8	9	10
Bare										
	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
Grass										
	⬜	🟧	🟧	🟧	⬜	🟧	🟧	🟧	⬜	🟧
Shrubs-										
	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨
Shrubs+ & grass										
	🟫	⬜	🟧	🟧	🟫	⬜	🟧	🟧	🟫	⬜
Small trees										
	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Large trees, large shrubs & grass										
	⬛	🟧	⬜	🟧	🟧	🟫	🟧	⬜	🟧	🟧

Bare soil	🟧
Grass	⬜
Small shrub	🟨
Large shrub	🟫
Small tree	🟩
Large tree	⬛

Figure 19 Representation of the different vegetation scenarios

Bare soil: This scenario is fairly simple since it contains no vegetation

Grass (Table 5): Grass patches covering ¼ of the area with a porosity of 0.2 and a height of 0.15 m

Table 5 Vegetation parameters for the grass scenario

Grass	Tree	Shrub	Grass
Height			0.15
Breadth			
Breadth Bark			
Quantity			
Porosity			0.2
Drag coefficient			0.1

Shrubs- (Table 6),: Small shrubs spread out equally with 160 shrubs per hectare

Table 6 Vegetation parameters for the small shrub scenario

Shrubs-	Tree	Shrub	Grass
Height		0.8	
Breadth			
Breadth Bark			
Quantity		160	
Porosity		0.7	
Drag coefficient		0.2	

Shrubs+ & grass (Table 7): Large shrubs in rows with 400 shrubs per hectare. Grass covers ¼ of the area with small patches

Table 7 Vegetation parameters for the large shrub scenario

Shrubs+	Tree	Shrub	Grass
Height		1	0.15
Breadth			
Breadth Bark			
Quantity		400	
Porosity		0.8	0.2
Drag coefficient		0.2	0.1

Small trees (Table 8): 300 trees per hectare equally spread out.

Table 8 Vegetation parameters for the small tree scenario

Tree-	Tree	Shrub	Grass
Height	0.8		
Breadth	0.7		
Breadth Bark	0.05		
Quantity	300		
Porosity	0.6		
Drag coefficient	0.3		

Large trees, large shrubs & grass (Table 9): Large trees in rows. Large shrubs in rows with 400 shrubs per hectare. Grass covers 1/5 of the area with small patches

Table 9 Vegetation parameters for the densely vegetated scenario

Tree+	Tree	Shrub	Grass
Height	3.5	1	0.15
Breadth	3		
Breadth Bark	0.2		
Quantity	365	400	
Porosity	0.6	0.8	0.2
Drag coefficient	0.3	0.2	0.1

3.3.4 Model testing and application

The original plan was to measure model parameters in the field. Due to Covid-19 fieldwork was cancelled. This resulted in the inability to measure different model parameters and the effect of changes in these parameters. This caused the modelling to be based on previous studies. The outcomes of the model were compared to the literature to test whether the model provided realistic values. Due to limited studies done in the region, the values of the parameters are not very precise. There is however a range of confidence for each of these parameters. To review the impact of different parameters, the sensitivity of each of these parameters was tested. The sensitivity analyses provide a better understanding of which parameters have the most impact on the model and provide a margin of error in the vegetation scenarios. The parameters that were tested include Mean grain size, Soil drag coefficient, air density and vegetative parameters.

To determine the best vegetation scenario for vegetating the desiccated Aral Seabed the model was tested for six different scenarios (Bare soil, small shrubs, grass, large shrubs & grass, small trees, large trees, large shrubs & grass). The wind velocity input for the model was taken from storms with different temporal recurrences. Storms were divided into three categories according to temporal recurrence: Once every five years, Once every year, multiple times a year. The different scenarios were tested for each of the storms. Sediment mass fluxes were modelled for each category during the storm. The total eroded quantity per storm and per scenario was calculated, including error margins. The error margins were calculated by changing each of the uncertain vegetation, soil and climatic variables to obtain the minimum and maximum possible sediment flux per scenario per storm. This gave a minimum and maximum possible sediment flux per scenario per storm. The long-term effect of the patchy wood planting scenario is to create vast areas with sparsely vegetated areas. The effect of these sparsely vegetated areas was separately assessed. This was tested by modelling the effects of very sparse vegetation and differences in uniform and striped vegetation patterns.

3.3.5 Dust modelling

The main focus of the modelling was to quantify the wind sediment transport in the desiccated areas. When erosion occurs, sand is transported in the form of creep, saltation and suspension. The majority of the mass is transported in saltation. The type of transport is dependent on the size of the particles: Creep > 500 µm, Saltation 70-500 µm, short term Suspension 20-70 µm, Long term suspension <20 µm (Kok et al., 2012). Since dust storms are a major hazard, it is important to know the relative quantity of dust generated per storm and the impact of the vegetation on the dust generation. Modelling the generation of dust is more complicated and has a larger margin of error than modelling the horizontal sediment flux. A simple estimation of the dust transport was made using the total horizontal sediment transport, wind velocity and the grain size (Alfaro et al., 2014). This model uses a simple approach, and vegetation parameters were therefore calculated using a single cell. The approach assumes that the initial rate of particles being brought into suspension is correlated to the particles brought into saltation. The particles brought into suspension increases with the amount of turbulence and only particles with a certain size can be brought into suspension. The model is dependent on the total sediment flux output of the saltation model, the friction velocity and the mean grain size. Most of the particles are not able to go into long-term suspension and thus a constant (α; 10) is introduced as a measure of the percentage of particles that are able to go into suspension. For calculations the output of the CA model was used, and an average cell was made with averaged vegetation parameters. The dust output can then be calculated for different velocities:

$$Dust\ flux = Sediment\ flux * \frac{(u_{*s} - u_t)^2}{g} * \alpha \qquad 3.21$$

The dust estimation using Alfaro (2014), gives only a rough estimation but is useful to further see the impact of vegetation patterns on dust storms.

4 Results

4.1 Land degradation assessment

The desiccation of the Aral sea has had an effect on climate and in turn on the land and vegetation cover in the region. First, the change in climate (Temperature, Precipitation, SPI and SPEI) will be shown then the change in vegetation and the correlation between the two.

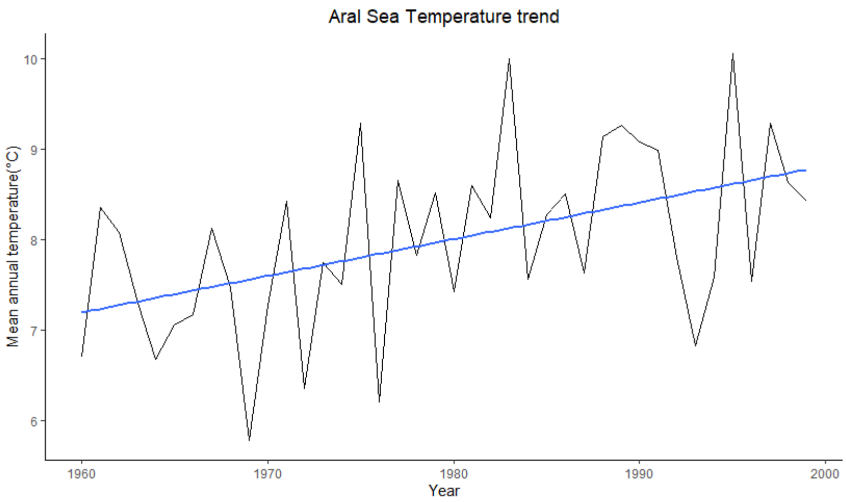


Figure 20 Mean_annual **temperatures in the Aral Sea and desiccated part of the Aral Sea. Data is from the MODIS Aqua Land Surface Temperature and emissivity 8-Day Global 1km Dataset (Kalnay et al., 1996). The mean annual temperature is calculated by taking the mean of all the grid cells covering the Aral Sea and the desiccated Aral Sea.**

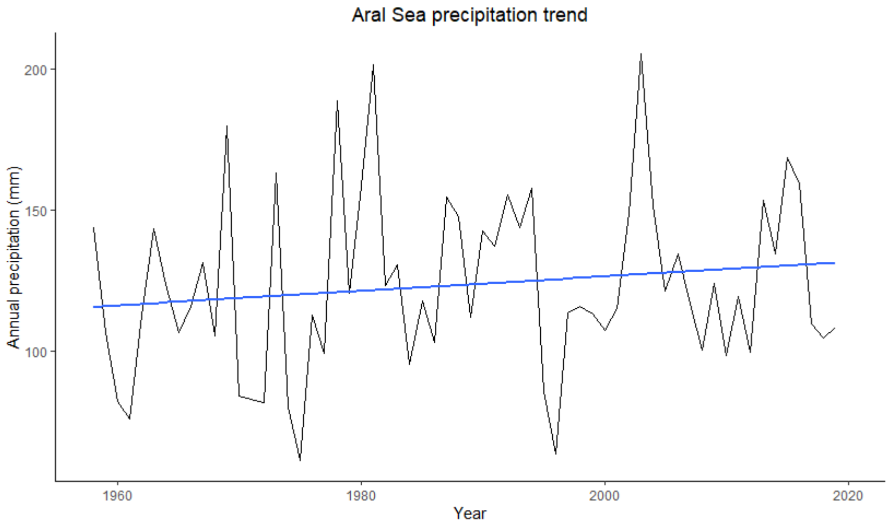


Figure 21 Mean annual precipitation in the Aral Sea region showing an increasing trend. The mean annual precipitation is calculated by taking the mean of all the grid cells covering the Aral Sea and the desiccated Aral Sea.

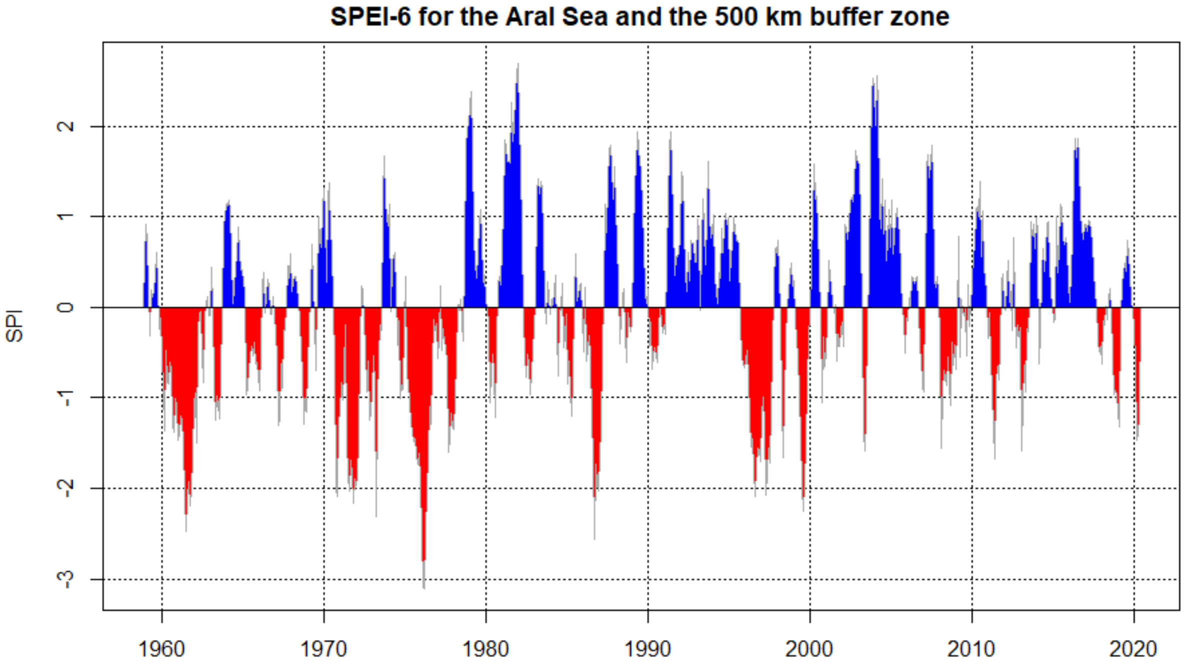


Figure 22 1958-2020 SPI for the area around the Aral Sea. The SPI-6 is calculated by taking the mean precipitation for the entire region as input for the SPI calculations.

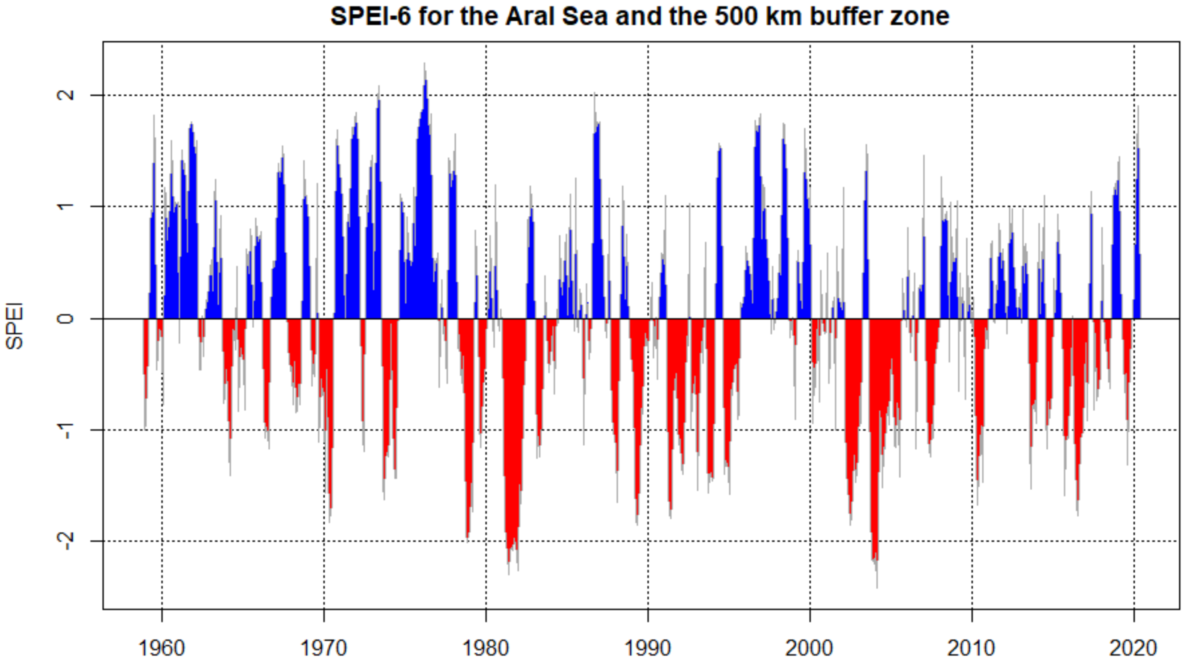


Figure 23 1958-2020 SPEI for the area around the Aral Sea. The SPEI-6 is calculated by taking the mean precipitation for the entire region as input for the SPEI calculations.

The mean annual temperature has risen significantly between 1960 and 2000 with 2 degrees Celsius (Figure 20), which is likely caused by the decrease in actual evapotranspiration due to the reduction in surface water in the region. The SPI shows dry years in the period 1958-1980 and wetter years after this period for areas within 300 km buffer of the Aral Sea, revealing a rising trend in precipitation (Figure 21 &22). The SPEI shows the opposite pattern in the areas in close proximity to the Aral Sea (Figure 23). The SPEI shows more positive(wet) values in the first 20 years and more negative(dry) values in the later years. This means that the potential evapotranspiration has increased in the region to the point that it offsets the increase in precipitation. The increase in temperature due to climate change is the main driver for the increase of potential evapotranspiration in the region. Higher temperatures are likely to also be a cause for the intensification of the hydrological cycle. The Higher temperatures allow for more evapotranspiration and thus more precipitation. Around 1980 however, the increase in temperature does not deliver a corresponding amount of precipitation, causing the SPEI to drop significantly, signalling that the region has become even more water-limited.

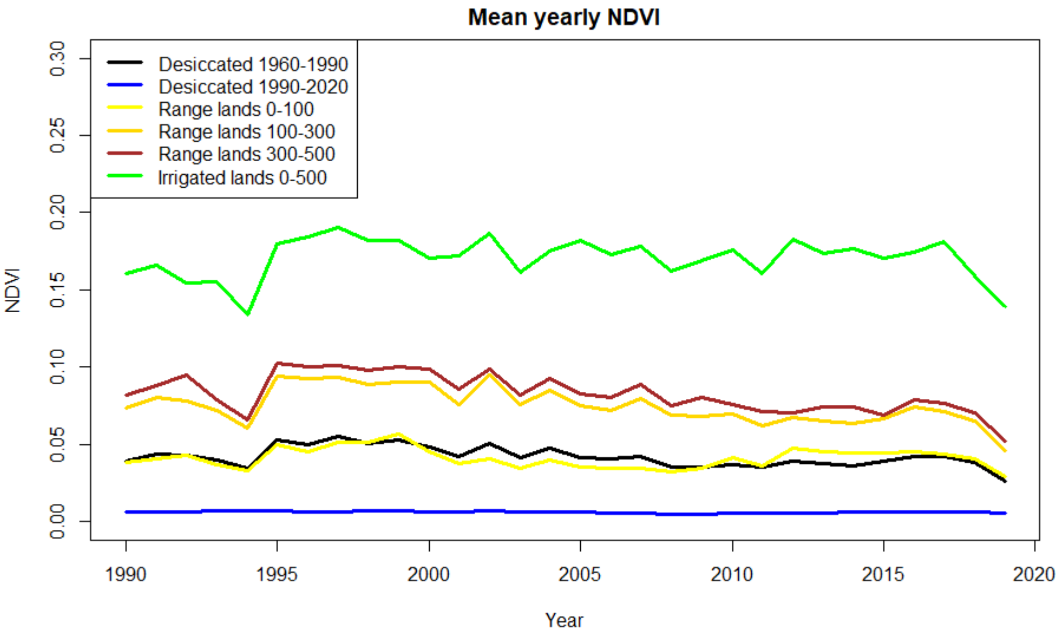


Figure 24 Mean yearly NDVI for different areas around the Aral Sea region

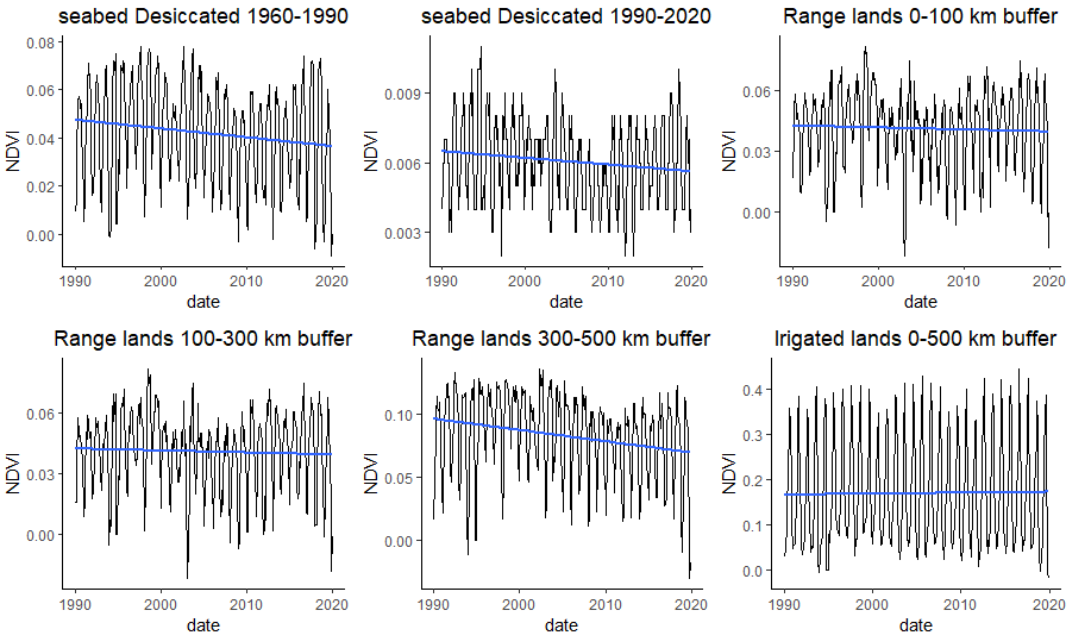


Figure 25 Monthly NDVI for different areas around the Aral Sea region

The entire desiccated seabed has a low vegetation cover (Figure 24), with some seasonal influence (Figure 25). The older part of the desiccated seabed has a higher NDVI and a larger seasonality than the younger desiccated seabed. The older part of the desiccated seabed also showed a significant (p= 0.0083) decline in NDVI. The NDVI in the younger desiccated seabed is quite low (-0.1 – 0.0) which indicates a barren landscape with saline soils. The seasonality and higher NDVI values of the old part of the desiccated seabed show that some, but very little vegetation is present (Figures 24 & 25).

The seasonality and the NDVI of the irrigated lands are large and have no significant (p= 0.7712) change over time. The irrigated lands in Uzbekistan have a slightly higher vegetation cover further away from the Aral Sea (Figures 24 & 25). Irrigated lands have an NDVI of 0.1-0.4 and high seasonality suggesting a normal vegetation cover. There is no significant change in the NDVI or seasonality, meaning that the area does not seem to be impacted by the desiccating of the Aral Sea in this time frame.

The rangelands have higher vegetation cover the further they are away from the Aral Sea. The NDVI is ~0.1 in the 100-500 km buffer zone and ~0.05 in the 0-100 km buffer zone. A significant (p= $9.6*10^{-4}$ & $1.2*10^{-5}$) decrease in NDVI is present in the 100-300 and 300-500 km buffer zones. The low NDVI values in the rangelands and seasonality show a sparse vegetation cover. A NDVI of 0.02 correlates with bare soil, while an NDVI of 0.25-0.3 correlate with grass and shrub cover. The NDVI of 0.1 suggests that these rangelands consist of patches with these soils, with only 20% of the patches covered with vegetation. The decrease in vegetation cover with distance indicates an influence of the desiccated Aral Sea (Figures 24 & 25).

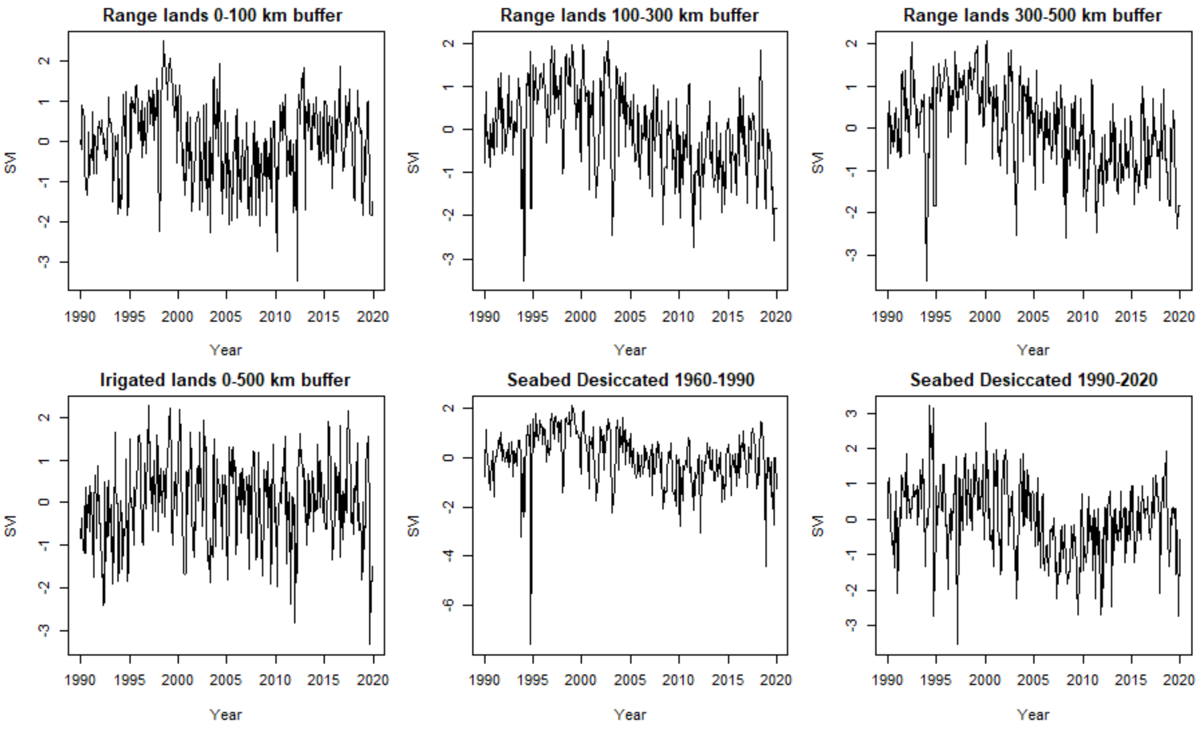


Figure 26 SVI for different areas around the Aral Sea region

Table 10 Correlation SVI-1 and SPI values

Land-class	SPI-1	SPI-3	SPI-6
Rangeland 0-100 km buffer	-0.23	-0.17	-0.09
Rangeland 100 -300km buffer	-0.30	-0.17	-0.03
Rangeland 300-500 km buffer	-0.26	-0.11	-0.01
Irrigated land 0-500 km buffer	-0.42	-0.29	-0.18
Seabed desiccated 1960-1990	-0.25	-0.17	-0.09
Seabed desiccated 1990-2020	0.01	0.02	0.13

Table 11 Correlation SVI-1 and SPEI values

Land-class	SPEI-1	SPEI-3	SPEI-6
Rangeland 0-100 km buffer	0.02	-0.01	-0.03
Rangeland 100 -300km buffer	0.00	-0.12	-0.17
Rangeland 300-500 km buffer	0.06	-0.10	-0.19
Irrigated land 0-500 km buffer	0.26	0.15	-0.02
Seabed desiccated 1960-1990	0.01	-0.06	-0.08
Seabed desiccated 1990-2020	-0.10	-0.16	-0.20

The correlations between SVI-1 (Figure 26) and SPI-1,3 are significant ($p = 1.501 * 10^{-15}$ & $2.097 * 10^{-9}$) for the irrigated lands. They are negative in the irrigated and range zones (Table 10). The negative relation can be explained by increase in groundwater table during extra wet growing seasons. The higher groundwater table would increase the risk of the capillary rise of saltwater which is would occur by heavy precipitation. The precipitation will bring saltwater closer to the surface, then evaporate leaving the salt behind. This process, also known as capillary fringe salinises the soil and could be the reason for the lower NDVI in the irrigated regions. For the rangelands the Correlation is mostly insignificant. This would indicate that the vegetation in the rangelands are unable to use the moisture provided by precipitation.

The SVI-1 - SPEI correlations are much weaker and more positive than the SVI-1 - SPI correlations (Table 11). The correlation with the SPEI-1 and SVI is positive and significant ($p = 7.162 * 10^{-7}$).) for the irrigated lands. The irrigated lands are the only region to have a significant negative correlation with potential evapotranspiration. Since the SPEI is precipitation minus potential evapotranspiration, the correlation with SPEI is positive. The negative correlation with Potential evapotranspiration is because if water evaporates, it leaves salt behind and salinising the soil. In arid regions, this is the main cause of soil salination. A possible reason why the irrigated lands are the only lands to have a significant negative correlation with potential evapotranspiration is that those lands have a high groundwater level due to irrigation. The high groundwater level causes the topsoil to contain water and be susceptible to salinisation through evapotranspiration.

4.2 Long-term wind occurrence and threshold analysis

The windspeed analysis was based on the wind speed data from Kungrad, Uzbekistan. The maximum 3-hourly average windspeed per day was determined for the period 2000-2020. The average times the windspeed thresholds for sediment transport (10, 15, 20 and 25 m/s) were exceeded in the past 20 years were then recorded (Table 12).

Table 12 Days the 3-hour average windspeed exceeded each threshold.

Windspeed(m/s)	Average yearly occurrences	Total 20-year occurrences
>=10	41.6	832
>=15	14.2	284
>=20	1.6	32
>=25	0.25	6

The occurrences of 3-hourly wind events with average speeds over 10, 15, 20 and 25 m/s were plotted monthly (Figure 26) and yearly (Figure 27) to assess the spatial distribution within years and throughout the 20 year time period. Most high wind events occur in spring: March-June, this coincides with most dust storms. The strong wind events fluctuate throughout the years with no clear pattern (Figure 27).

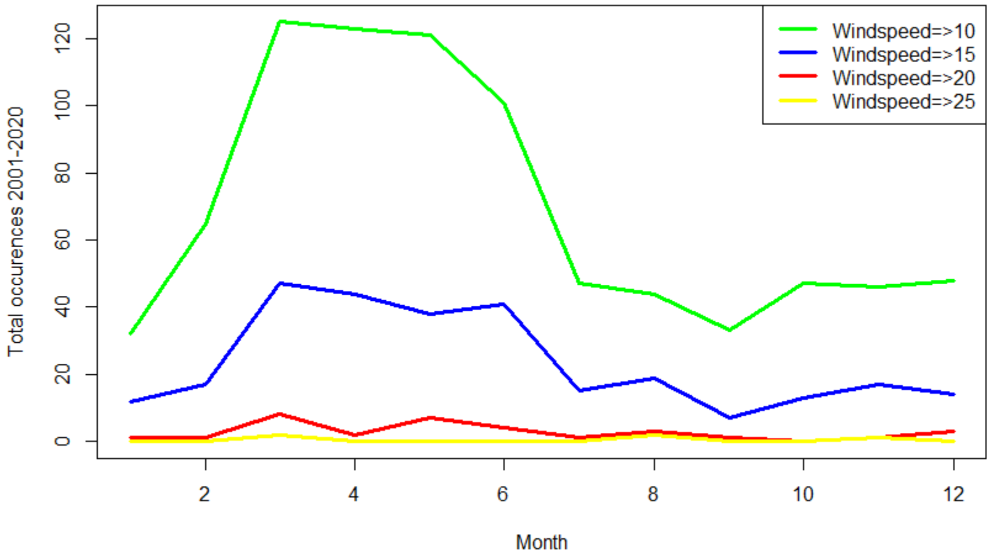


Figure 27 Windspeed threshold exceedance per month for 2001-2020.

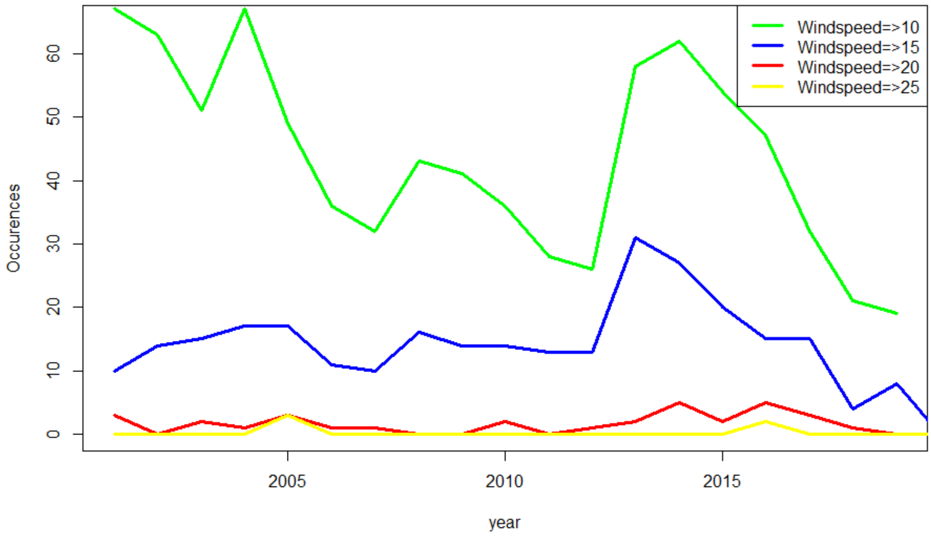


Figure 28 Windspeed threshold exceedance per year for 2001-2020.

4.3 Long-term linkage of wind occurrence with aerosols/dust

Aerosol optical depth (AOD) was used as an index for dust storms. The AOD is an adequate indicator for dust storm events as can be seen in Figures 28 and 29, where the satellite AOD measurements on days with a large dust storm are shown. The images of the dust storms are from a particular day, while the AOD shows the mean values of a four days period in which the dust storm occurred.

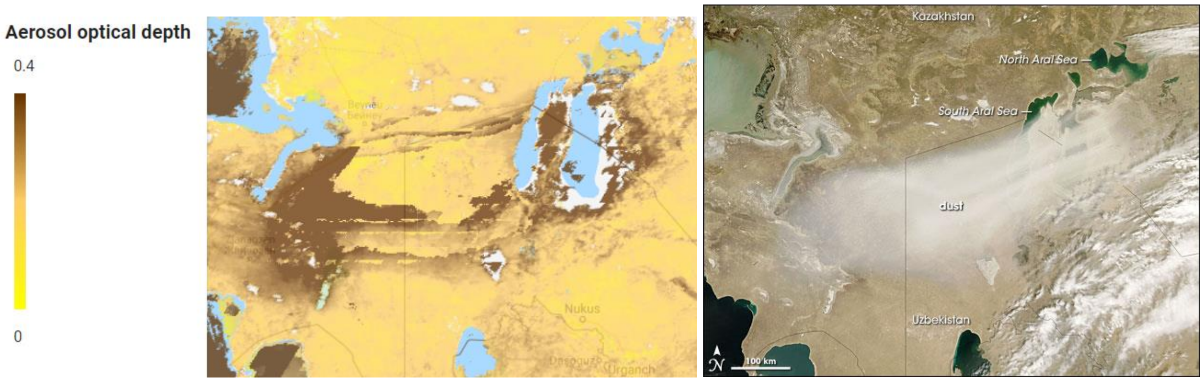


Figure 29 Mean AOD April 27-30 2008 (left), a dust storm on 29 April 2008 (right) (NASA Earth observatory).

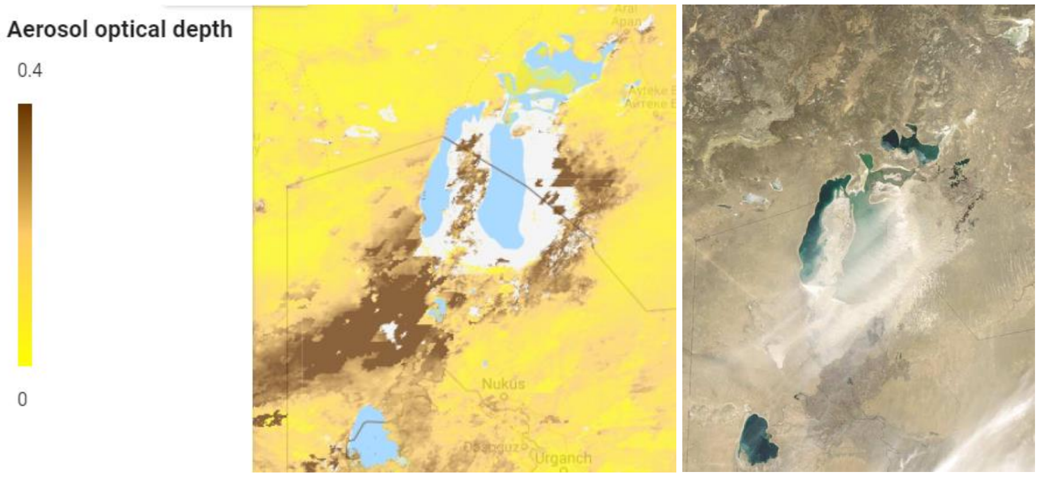


Figure 30   Mean AOD April 17-20 2003 (left), a dust storm on 18 April 2003 (right) (NASA Earth observatory).

The average days per month in which the AOD exceeded 0.2, 0.3 and 0.4 were plotted per year and per month for 2001-2020. The patterns showed higher AOD values during spring and summer, with a small decrease in June (Figure 30). The dusty months coincide with the months which have higher wind speeds.

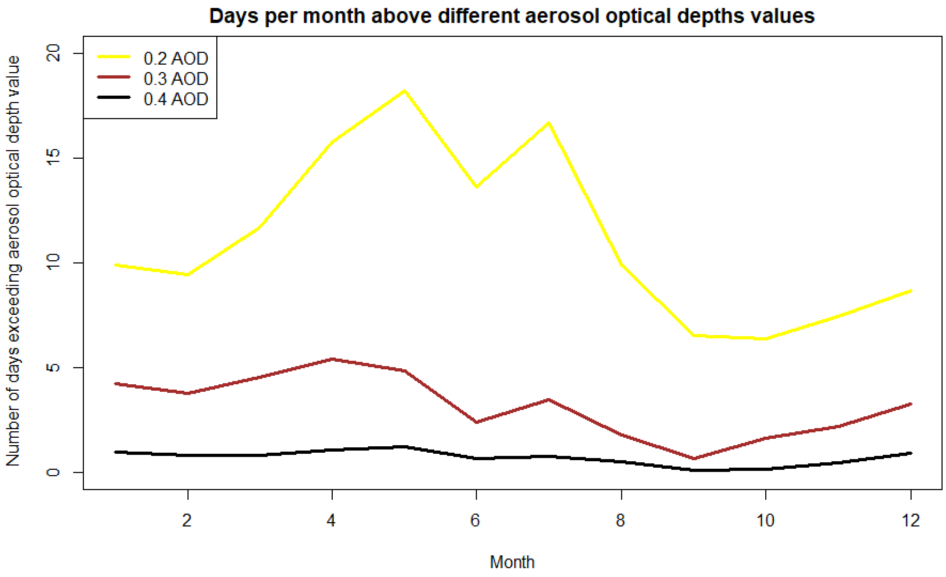


Figure 31 Average number of days per month above a certain AOD threshold calculated for 2001-2020

To assess the influence of the wind speed on the higher AOD, the correlation between maximum 3-hourly windspeed per day and AOD was calculated. The correlation was calculated for days with wind speeds above 8 m/s for different months (Figure 31). This Indicator for dust is not the ideal indicator as it takes all aerosols present in the air, hence the relatively low correlations of 0.0-0.4. A clear correlation is present in Autumn and in spring, while in the winter, the wind speed does not influence the aerosols at all. The region experiences temperatures below zero during winter, during this time, part of the land will be covered in snow, thus reducing wind-erosion and bringing the correlation down -0.02. Likely is that other factors apart from wind speed also influence the formation of dust storms. Other factors could include; days without rainfall, wind direction or location in the Aral Sea region.

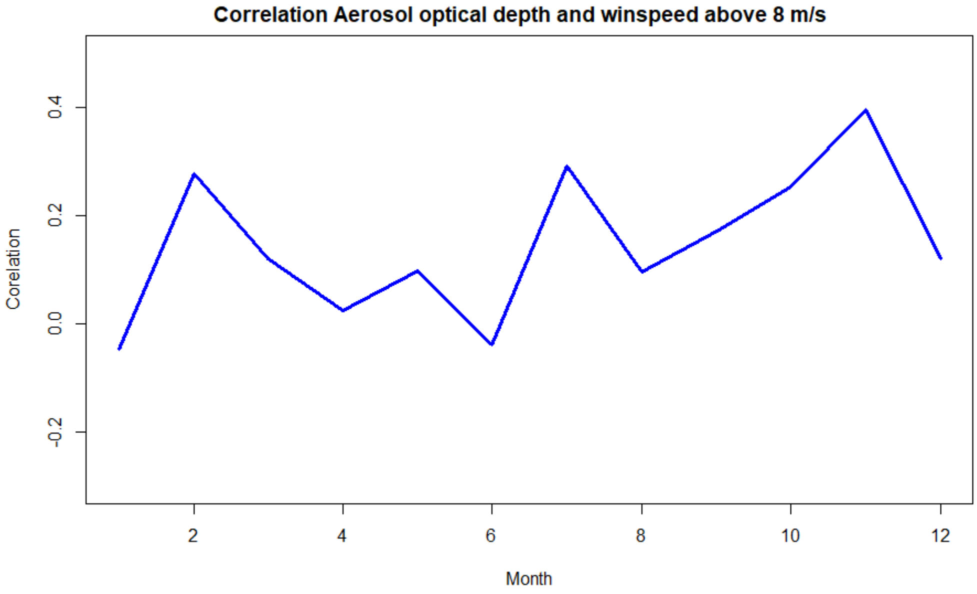


Figure 32 Relation AOD and Wind speed

4.4 Sediment transport model evaluation

To assess whether the values of the CA model are reasonable, different scenarios were modelled with the new CA model and also with the existing field-scale model of Kersten (2015). The outputs were compared to see if the sediment flux values are in reasonable agreement when using the two models. The models were compared with different vegetation covers, namely bare soil, low vegetation cover (shrubs) and medium vegetation cover (trees) for wind speeds of 10, 15 and 20 m/s. The results (Figures 32, 33 & 33) of the two models are quite similar, with slightly lower values for the CA model (0 – 16%). The CA model predicted a lower sediment flux than the Kersten (2015) for higher vegetation covers.

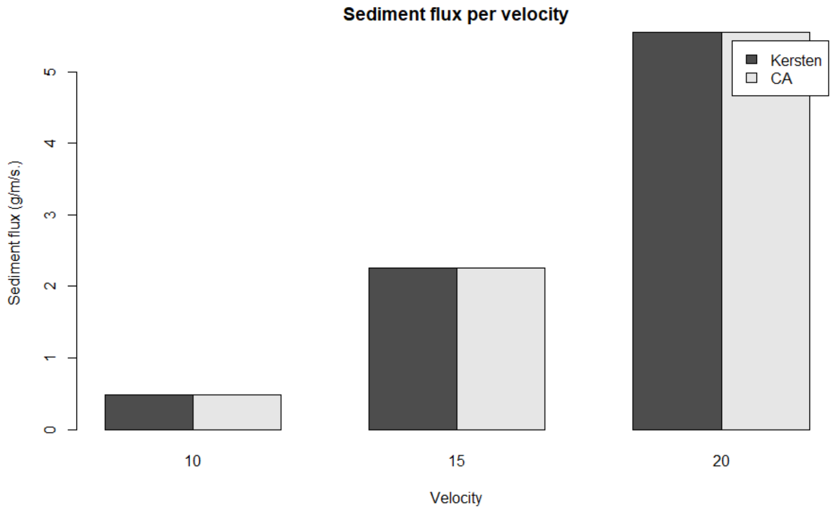


Figure 33 Differences in sediment flux prediction using two models for bare soil conditions

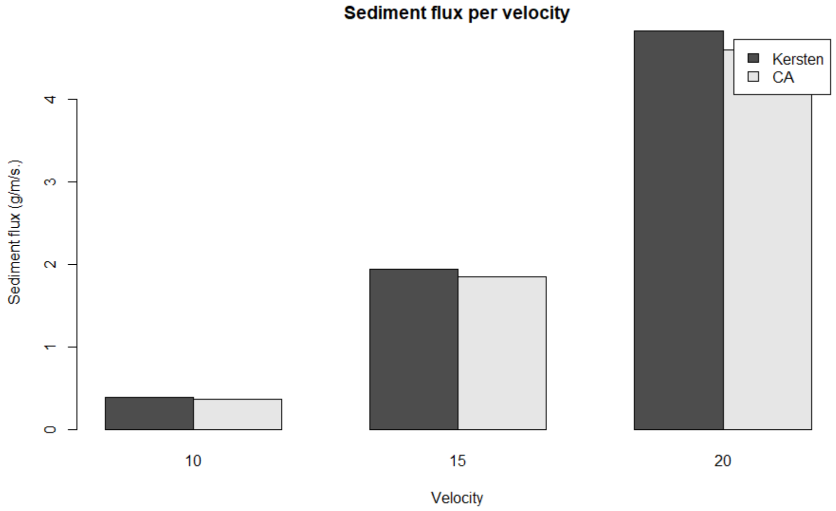


Figure 34 Differences in sediment flux prediction using two models for low vegetation cover

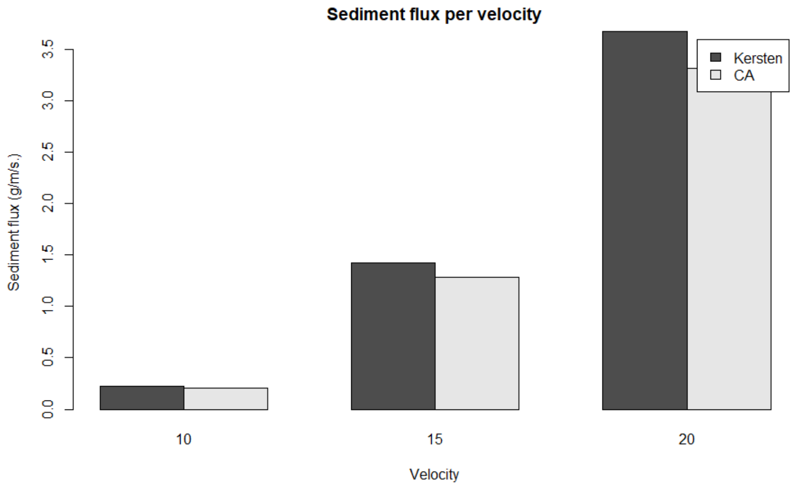


Figure 35 Differences in sediment flux prediction using two models for medium vegetation cover

To determine whether the values are realistic, the modelling results were also compared to experimental results from other countries. The horizontal sediment flux on the bare soil was compared with an experiment on bare soil in Patagonia (Sterk et al., 2012). There the total mass transport during nine different storms was measured. The wind speeds during storms ranged between 7.88 and 13.23 m/s with 10.71 m/s as the median. The sediment flux of these storms ranged between 0.2 and 20 g/m/s with the median sediment flux being ~1.6 g/m/s. If we compare this to our modelling results of 0.5 g/m/s for 10 m/s, we see that the modelling is factor 3 lower than the median. This difference can be explained by either the lower than the median windspeed in the model or the difference in texture. The Patagonian soil consists of mostly (~93%) sand, while a large factor of the soil in the Aral region is clay (~10%) and silt (~30%). Silt and clay have a higher threshold friction velocity which would explain the lower sediment flux. Since the spread in experimental sediment fluxes is large and the differences can be explained, we can confidently claim that the modelling results are realistic.

Compared to the Kersten (2015) model, the modelling of grass was added. To determine whether the CA model correctly simulates the effect of grasses, the results were compared with an experiment done in the desert grasslands of New Mexico (Li et al., 2007). This experiment tested the effect of removing different percentages of grass covers (0, 25, 50, 75 & 100%). The experiment found that removing the last 25% of grass cover has the most effect on the horizontal sediment flux and can increase the erosion up to 200%. In the experiment, the sediment flux did approach zero but never was zero. Using the CA model, scenarios with 0-100% grass cover removal were modelled for wind velocities between 0 and 25 m/s and the total sediment flux summed. For most vegetation covers, the results of the model were similar to the results of the experiment (Figure 35). For 0% vegetation cover removed (Control) the model seems to underestimate the sediment flux.

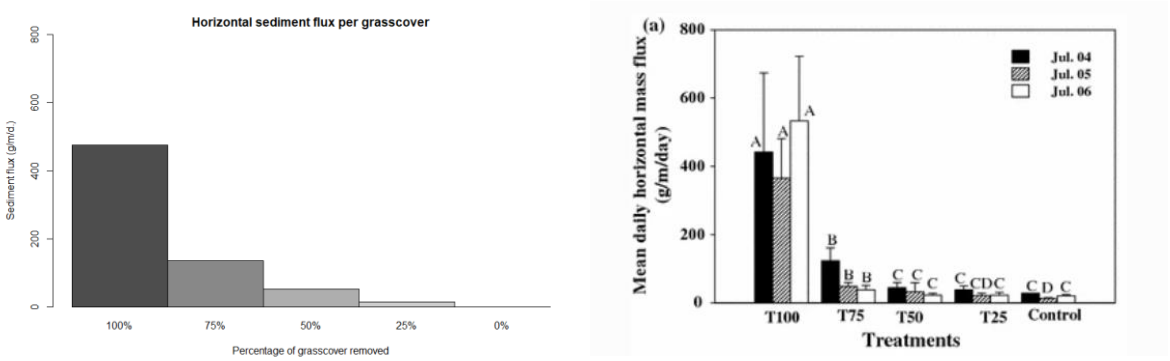


Figure 36 Comparison in horizontal sediment flux between model and experiment (Model findings left, experimental findings Right).

The effect of small amounts of vegetation was tested for 0-50 shrubs with wind speeds of 10, 15 and 20 m/s (Figures 39, 40 & 41). The shrubs modelled were 1 m high, 2 m wide shrubs with a porosity of 0.5. The graphs show the same pattern at different wind velocities. A small number of shrubs (0-28) actually increase the sediment flux. This is due to increased roughness of the surface the vegetation creates. The relation between vegetative elements and surface roughness is logarithmic, meaning that the effect of roughness elements on surface roughness decreases with increasing roughness elements. The roughness creates turbulence and increases shear stress, thus increasing erosion. The vegetative elements also absorb momentum and prevent fully developed wakes this reduces surface drag. When the vegetation density is high enough, the absorption of the shear stress by vegetation will offset the shear stress created by the roughness, and there will be a reduction in Sediment flux.

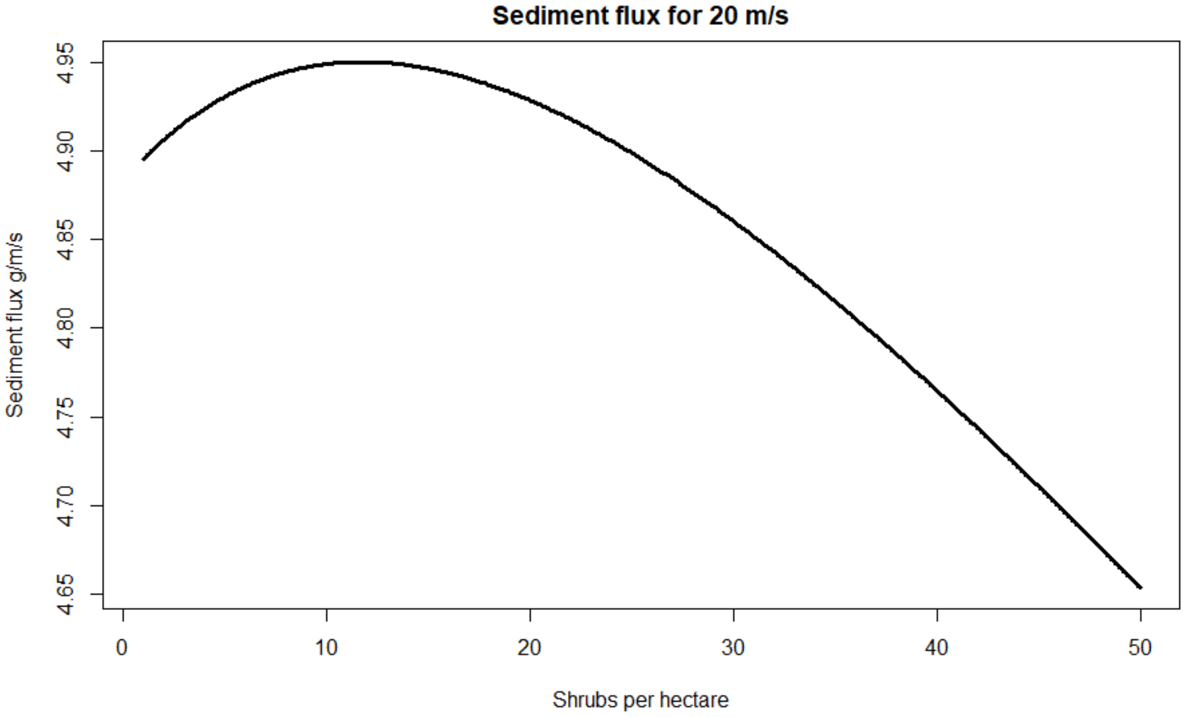


Figure 37 Sediment flux at 20 m/s for low vegetation

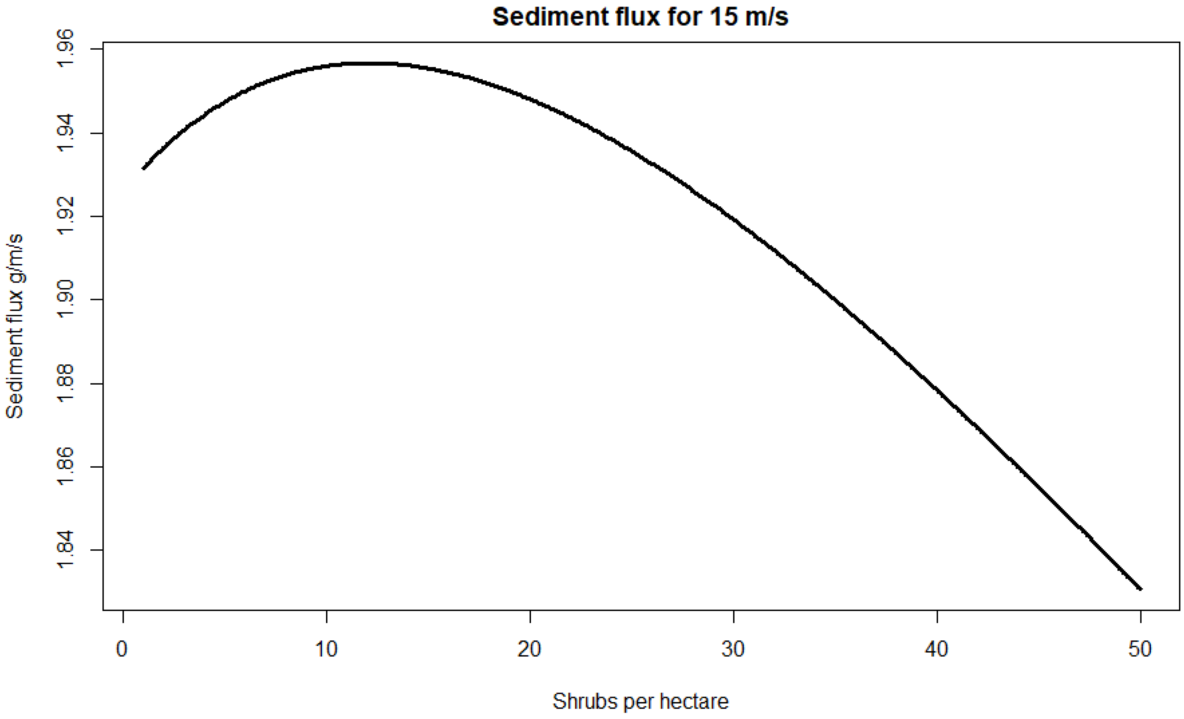


Figure 38 Sediment flux at 15 m/s for low vegetation

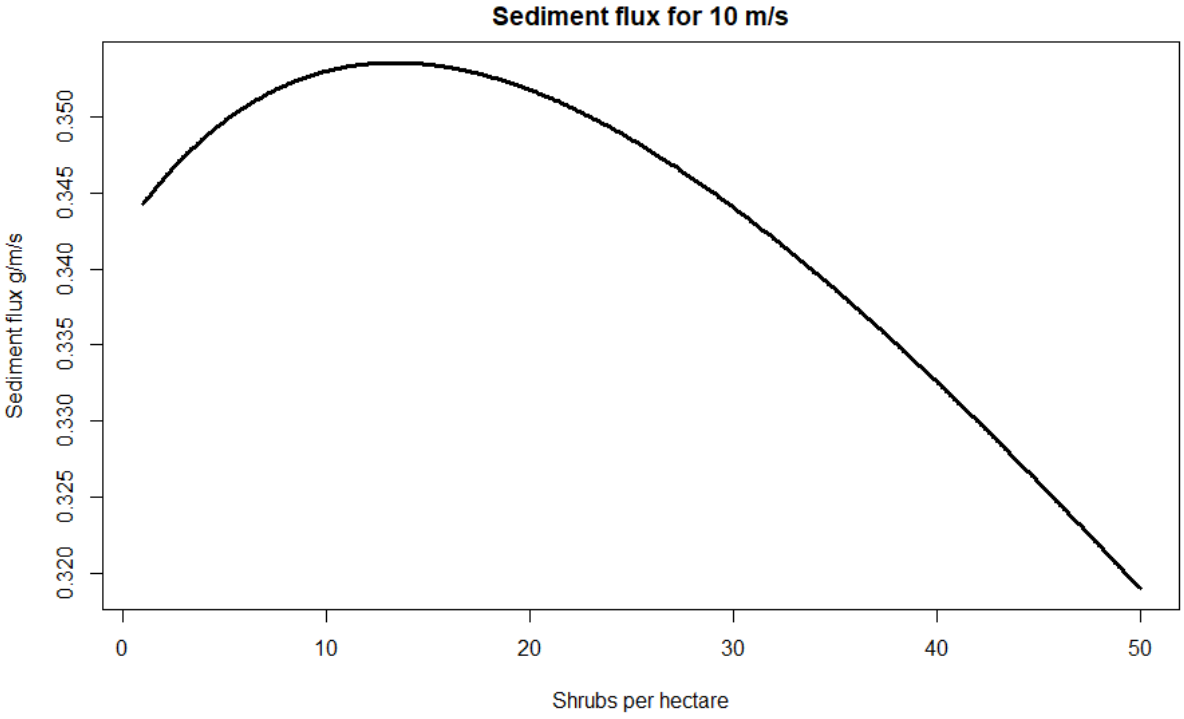


Figure 39 Sediment flux at 10 m/s for low vegetation

The CA model is able to alter spatial patterns with different vegetation densities. The effect of different vegetation patterns on sediment flux for 50 and 100 trees per hectare of 3 m height and 2.5 m breadth was modelled (Figures 42, 43 & 44). The results show that a uniformly spread vegetation pattern is best at stopping erosion at low vegetation density. At lower wind speeds the uniformly spread vegetation is best since it prevents wakes from fully developing and thus lowers the drag on the soil, reducing saltation. At medium vegetation densities, the difference in vegetation pattern is negligible. The relative increase in effectiveness of striped vegetation is likely due to the fact that at medium vegetation the density of the vegetation is high enough to act as a windbreak.

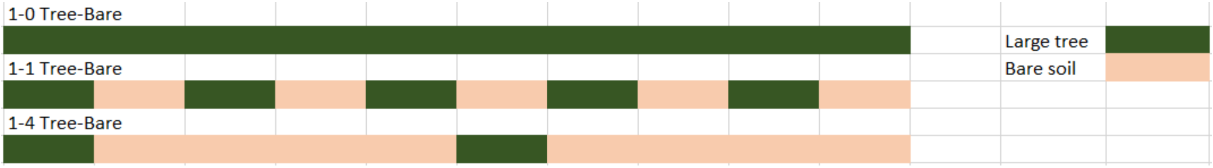


Figure 40 Vegetation patterns

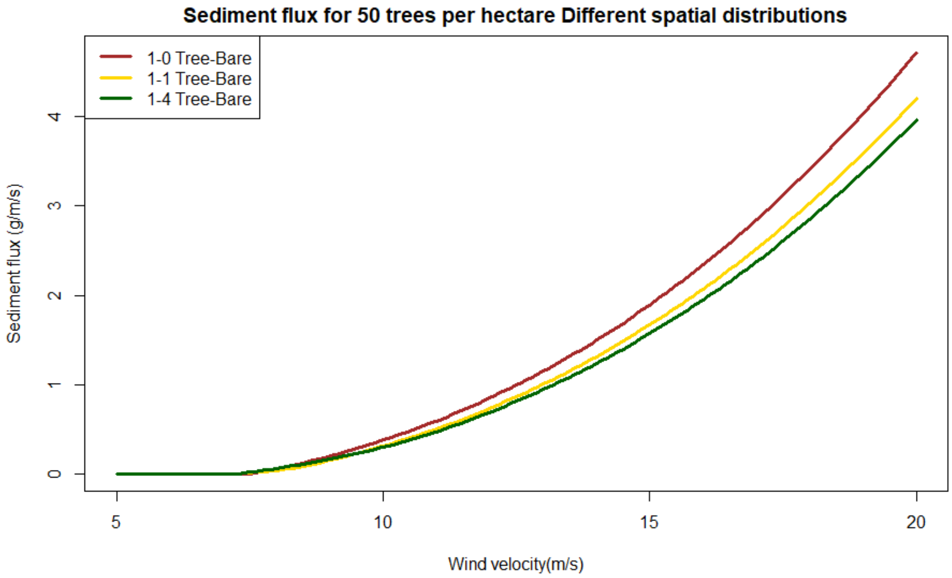


Figure 41 Spatial Effect of striped vegetation on sediment flux.

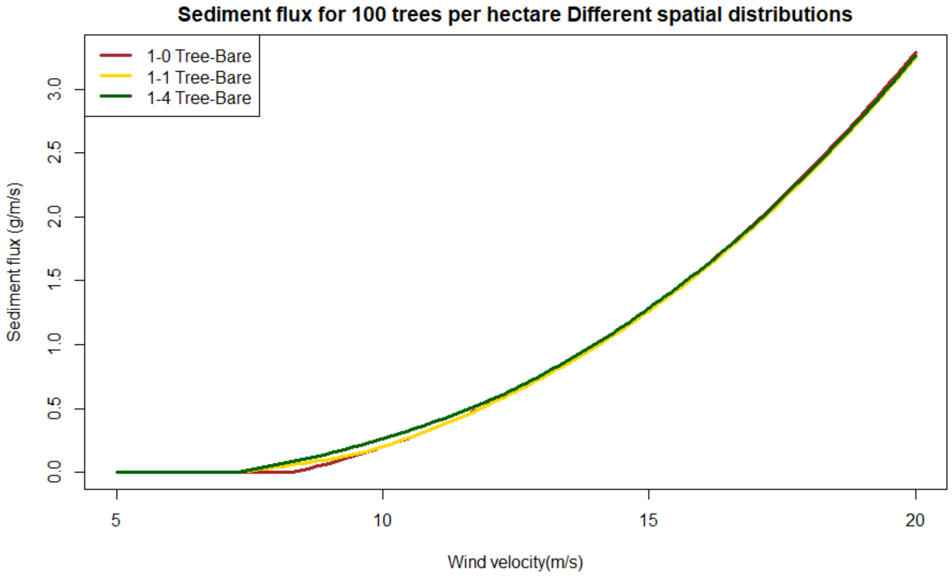


Figure 42 Spatial Effect of striped vegetation on sediment flux.

4.5 Scenario modelling: degraded vs rehabilitated

The effect of the different vegetation cover scenario's (Figure 19) was modelled for three different classes of storms. The three classes were derived from the wind speed analysis in 4.2. Class 1 storm: a storm with a recurrence rate of 14 times per year. Class 2 storm: a storm with a recurrence rate of once 1-2 time per year. Class 3 storm: a storm with a recurrence rate of once every five years. For each class, an average storm was picked from local wind speed measurements. For the storms, the horizontal and vertical fluxes over time were modelled. The total horizontal and vertical sediment flux during the storm were then summed for different parameters, showing the expected values and uncertainties.

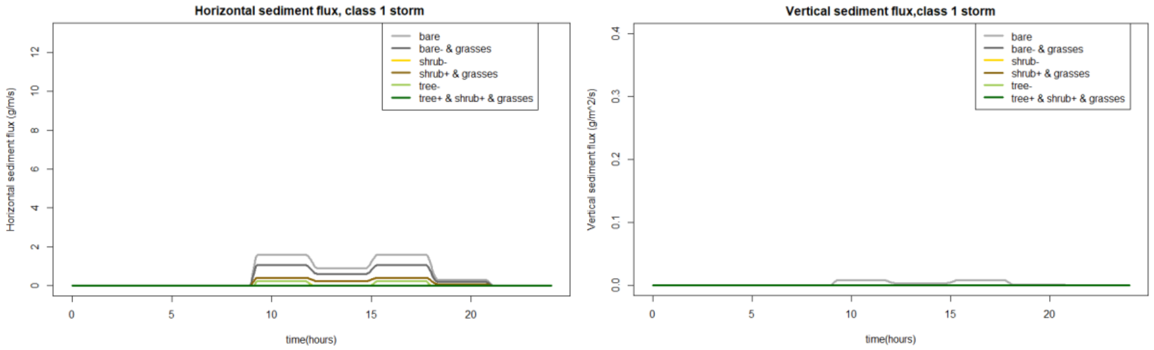


Figure 47 Sediment fluxes during a class 1 storm

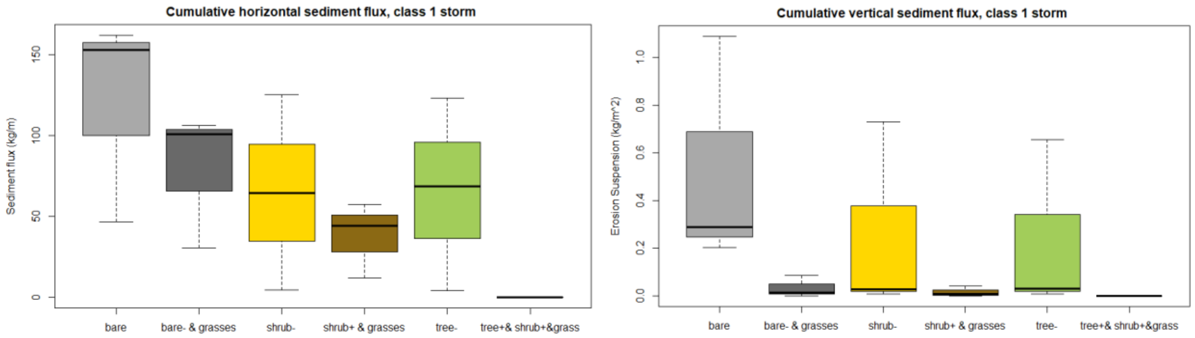


Figure 48 Cumulative Sediment fluxes with confidence intervals for a type 1 storm

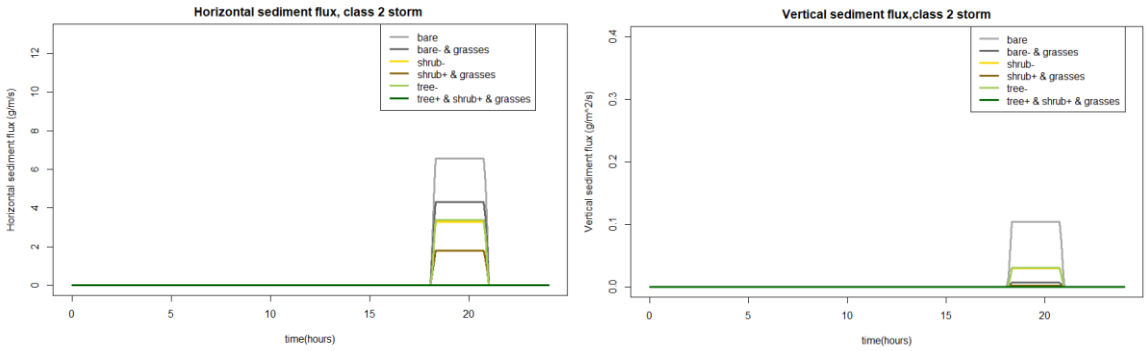


Figure 43 Sediment fluxes during a class 2 storm

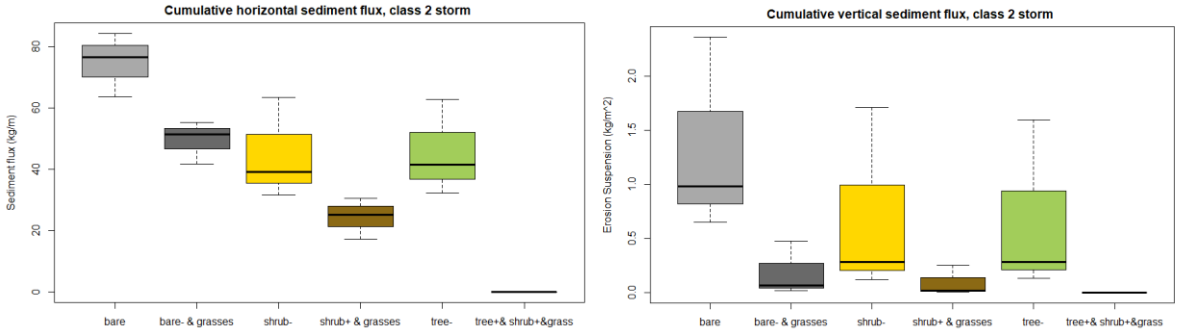


Figure 44 Cumulative sediment flux with confidence intervals for a type 2 storm

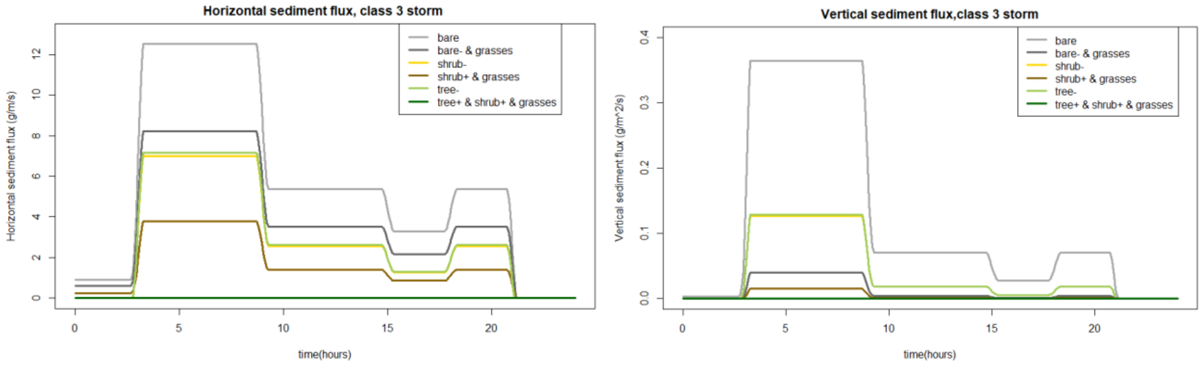


Figure 51 Sediment fluxes during a class 3 storm

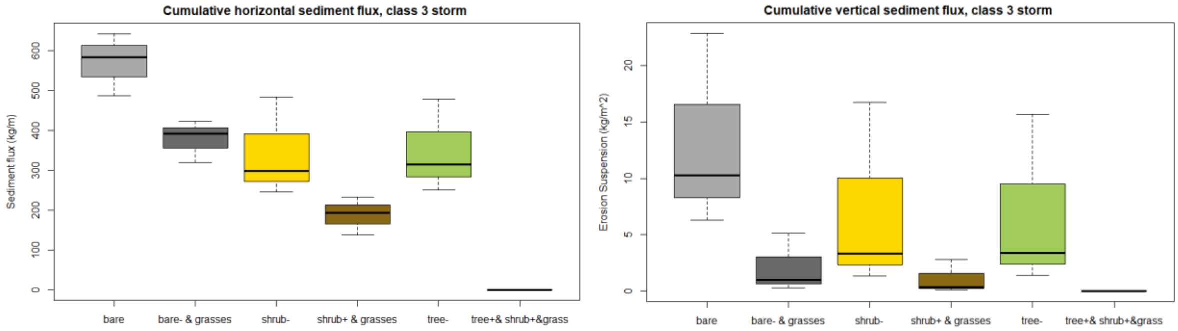


Figure 52 Cumulative sediment fluxes with confidence intervals for a type 3 storm

Table 5 Cumulative sediment flux per scenario and per storm

Cumulative sediment flux (kg/m)	Class 3 storm	Class 2 storm	Class 1 storm
Bare	326.0	66.8	41.0
Grasses	106.9	21.9	13.5
Shrub-	164.9	35.9	10.4
Shrub+ & grasses	66.1	14.0	7.3
Tree-	168.8	36.8	10.4
Tree+	0.0	0.0	0.0

Table 6 Cumulative vertical sediment flux per scenario and per storm

Cumulative vertical sediment flux (kg/m^2)	Class 3 storm	Class 2 storm	Class 1 storm
Bare	6.73	1.27	0.24
Grasses	0.32	0.06	0.00
Shrub-	2.17	0.41	0.03
Shrub+ & grasses	0.17	0.03	0.00
Tree-	2.20	0.42	0.03
Tree+	0.00	0.00	0.00

The difference in sediment flux between the storms is very large (Figures 47-52 & tables 16 & 17). The difference between classes being a factor 8 while the maximum wind speed only differs 5 m/s between classes. This is because the transport of sand has a cubic correlation with wind speed, causing the large contrasts in wind-erosion.

Horizontal sediment flux is a lot higher than in the vertical flux, while the uncertainty is higher for suspension. The mass of particles which go into suspension is based on the horizontal sediment flux, turbulence and particle size. There are more parameters and calculations required for the vertical sediment flux than the horizontal sediment flux. Due to this the uncertainty of suspension is a lot higher. The uncertainty accounts for total sediment flux calculations and suspension calculations.

The patterns of saltation and suspension fluxes are quite similar. Small trees and small shrubs have relatively more transport suspension than saltation. Small sparsely distributed trees cause turbulence, while their reduction in drag on the soil is relatively small. Since the percentage of particles going from saltation to suspension is largely determined by the turbulence, these sparsely vegetated scenarios bring a relatively high amount of particles into suspension.

Different vegetation scenarios show different sediment fluxes. The scenario with trees, grasses and shrubs eliminated all sediment transport. Grasses showed to be very effective at reducing wind sediment transport at all wind speeds, but not able to reduce it to 0%. Small trees and shrubs have the largest uncertainties in the model; they showed a reduction in sediment fluxes for all storms. Compared to grasses, small trees and shrubs create more particles in suspension. This is due to the fact that small shrubs and trees create turbulence which is a driving factor for putting particles in suspension.

All the vegetation scenarios showed a reduction in sediment flux compared to the bare scenario. The combination of different vegetation with high vegetation density, reduced the sediment flux to 0 in even the heaviest of wind events. Sparsely vegetated areas will reduce the total wind erosion but only slightly reduce particles in brought into suspension.

5 Discussion

The analysis of the meteorological data shows an increase in precipitation close to the Aral Sea. This shift towards a wetter climate occurs circa 1980. After this shift towards a more wet climate, there are only annual fluctuations and no clear trend. A study modelling the return flow of the Aral Sea basin estimated a return flow of 96%, which means that 96% of evapotranspiration finds its way back into the Aral Sea basin (Shibuo et al., 2005). This would explain the increase in precipitation in the basin. The large scale irrigation projects in the basin, increased the surface area susceptible to evapotranspiration. The change of 36 million ha into cultivated land has likely offset the loss of 4 million ha of Aral Sea. This increase in evapotranspiration caused an intensification of the hydrological cycle, which led to more precipitation. The SPEI shows a declining trend which means an increase in potential evapotranspiration. The shift to a warmer climate causes this increase in evapotranspiration. This is confirmed by the literature that claims a 2-4 °C increase in average surface temperature around 1980 (Guo., 2018). The effect was most pronounced during the growing season, during this time most of the water was used for irrigation, increasing the surface temperature by 5-6 °C in the Aral Sea (Small et al., 2001). During the same period, however further south of the Aral Sea, a temperature decline was found, this due to the increased evapotranspiration in the irrigated land (Shibuo et al., 2005).

The vegetation of the irrigated areas in Uzbekistan has not seen any significant change in the last 20 years. The rangelands in Uzbekistan and the desiccated seabed however, have seen a considerable decline. The likely cause for this is the increase in temperature causing water stress in the vegetation. Other studies covering the entire Central Asia region have also concluded that increases in water stress are to blame (Xu, 2012). Another reason might be the Dust storms that transport dust and salt up to 500 km from the Aral Sea.

Correlations between the SVI and SPI are low for the rangelands. A larger scale study for Central Asia also concluded this (Nezlin et al., 2005). A likely reason is that rain does not accumulate in the ground and thus does not sustain the development of vegetations. The correlations between SPI and SVI for the irrigated lands in Uzbekistan are significant and negative. During the period 1990-2000, the average depth of the groundwater table was 1.47 m, ranging from 0.04 m to 5.55 m (Awan et al., 2014). During the growing season, the average groundwater level rises with 0.3 m to 1.2 m (Ibrakhimov et al., 2004). In the irrigated lands, the “critical depth”, which results in the accumulation of salts due to capillary rise, is in the range of 1.5-2.0 m below the surface. Taking into account that the “critical depth” varies depending on soil texture and salinity level of soil and groundwater, most of the groundwater in Khorezm is above the critical level throughout the growing season (Ibrakhimov et al., 2004). The water for irrigation is derived from the Amu-Darya river, and the groundwater level of the irrigated lands is thus strongly correlated to the discharge of the Amu-Darya (Ibrakhimov et al., 2007). High precipitation means more river discharge resulting in higher groundwater levels. This means that precipitation has an amplified effect in Khorezm. The precipitation increases the groundwater table locally by infiltration, but also on a larger scale; precipitation occurring upstream will be transported by the river and used as irrigation water in Khorezm. Three case studies in Khorezm where groundwater table, soil and groundwater salinity were closely monitored at the field scale, suggested that the elevated groundwater levels forced soil salinisation by annually adding 3.5–14 t ha^{-1} of salts depending on the position and salinity of the GW table (Ibrakhimov et al., 2007).

During months with high precipitation and high evapotranspiration, the water rises due to capillary rise and evaporates, leaving the salt behind and increasing the salinity, while decreasing the fertility of the soil. A study in Khorezm which experimented with deficit irrigation application showed that with a 20 % of deficit irrigation application, throughout the growth period cotton yield would increase by 2 % (Akhtar et al., 2013). Farmers have a motive to keep groundwater levels high as an emergency water supply since the irrigation system is unreliable (Forkutsa et al., 2009). To reduce the salination of the soil, there are multiple options. Creating a better drainage system in combination with either stronger Water Consumers Associations and water management organisations for more reliable water distribution systems, or small decentralised water storage facilities. This will allow farmers to lower the groundwater and thus reduce salinisation of the soil. Another option is to reduce soil evaporation by adding a surface residue layer to the soil. At this moment soil salinisation is a major problem in the irrigated lands in Uzbekistan.

The wind and dust analysis using satellite and local weather station data gave insight into the temporal variability of high dust and wind events. The interannual variation in strong wind events is large, while the interannual aerosol variation is quite small. No clear trend in aerosol or large wind events is present. The correlation between AOD and high wind events is quite low. This is likely because of the dependence on land use and days without rain (Xu, 2006). The effect of the changes in vegetation and precipitation on aerosol quantities in the atmosphere is unable to be accurately tested. A big problem is the lack of a reliable data set for dust. AOD is one of the better indicators but not perfect. Correlation between wind and aerosols is present in most of the year. In the winter, however not. This is likely because of snow in this part of the year. The snow reduces the surface area available for wind erosion since the soil is covered in snow.

The model evaluation proved that the model accurately able to model the effect of vegetation on sediment fluxes. The modelling results were in a reasonable margin of error with experimental results and calculations. The effect of low-density vegetation is an increase in sediment flux. Only when the number of shrubs per hectare surpasses 28, there is a reduction in sediment flux. This effect was also noticed by Wolfe & Nickling (1993). The shrubs cause wakes (turbulence) when there are few vegetative elements present. The wakes will fully develop and increase erosion. When more vegetative elements are present, the wakes will be broken into smaller turbulent motions unable to erode the sand. This is concerning because of the negative trend in NDVI in the rangelands. It indicates a decrease in vegetation density. Lower vegetation densities have the potential to aggravate the erosion.

The modelling of the different vegetation scenarios showed that the bulk of the sediment transport occurs during extremely high wind events. Every vegetation scenario showed a decrease in sediment flux, with the dense shrubs and trees eliminating sediment flux. The small trees and shrubs showed a small decrease because of the wakes they create. In comparison, the grasses showed a large decrease in sediment flux, due to their ability to reduce wind velocity without causing much turbulence. Previous research also found that grasses provide better protection than shrubs (Engelstaedter et al., 2003). When comparing the plans for planting vegetation, the best plan will be the massive wood planting. This plan will eliminate most of the erosion. The plan to plant patchy wood plantings which will then create low-density vegetation in large swaths of lands will have a lesser effect. The vegetation will reduce erosion in the area where the trees and shrubs are planted. The sparsely vegetated areas which they create will, however, increase the erosion in these regions. The net effect of this approach might thus be negative. The region around the Aral Sea has become increasingly water-limited. Heavily striped and uniform spread is the best planting strategy, as uniform spread prevents wakes from fully developing and breaks them up into smaller motions. In contrast, heavily striped can absorb the momentum of the wind and thus bring sand particles out of saltation. The constraint of heavily striped is that it is only effective at high vegetation densities.

Another important component of the modelling is the particles brought in suspension. The dust storms are responsible for damage to the agricultural sector, multiple airway diseases and other health problems in the general population of Uzbekistan. Dust modelling was based on a fraction of the particles brought in saltation rising in the vertical column, the fraction being depended on the grain size and shear stress (Alfaro et al., 2014). The dust modelling showed patterns similar to saltation modelling. The main differences are that the effects of grass patches on the soil are much more pronounced reducing dust than in reducing saltation. The small trees and small shrubs, however, showed similar results in both aspects. The main reason for this is that small trees and shrubs increase the roughness of the surface, creating wakes. The wakes create a lot of shear stress which is required to bring particles in suspension. Due to the limited increase in surface roughness, grass patches are ideal and will be vital in reducing dust storms.

In 2004 30% of the soils in the Aralkum were available for vegetation due to the lowering of the groundwater table (Stullina & Sektimenko., 2004). As the groundwater table continues to go down, the area suitable for vegetation has likely increased. The lowering will also have the effect of increasing saline surface area susceptible to erosion because lowering of the groundwater table coincides with further desiccating of the Aral Sea. This brings the danger of increased salt and dust storms but also opportunity to revegetate the desiccated seabed. Ironically it could be a great strategy to reduce the groundwater level, letting the Aral desiccate further. This would prevent salinisation due to capillary rise and change the saline soils to bare soils available for revegetation. This would only be a good strategy if there would be a committed effort to planting new plantations as this strategy would also increase the total area of soil susceptible to erosion in the short term.

The model still has limitations. Future research should focus on improving the model and creating a 2D or 3D model instead of a 1D CA model. Using a 2D model would be benefitcial for modelling wind erosion and sediment transport when the wind is parallel to the vegetation lines. Creating such a model would however require an extensive runtime. This model only focusses on small scale sediment transport. A further study could focus on the spread of the sand and dust.

6 conclusion

The temperature has increased in close proximity to the Aral Sea while it has decreased in the irrigated regions further south, far away from the sea. This change in temperature was caused by a shift in land use. The Aral Sea desiccated while lands further south became irrigated. The land susceptible to evapotranspiration shifted south creating this shift in temperature. Meanwhile the increased evapotranspiration has caused an increased in precipitation for the entire region. This change in climate has harmed the vegetation in the rangelands in proximity (100-500 km) to the Aral Sea. The region has become increasingly water-limited, reducing the vegetation density and health. The increased temperatures combined with increased precipitation are the cause of salinisation of the soil, reducing soil fertility even more. Dust storms and wind erosion are substantial problems for the public health and agriculture of Uzbekistan. The dust storms are mainly in the spring when high wind events are present, and there is no snow to protect the soil against erosion. The World Bank and Uzbek government have proposed multiple revegetation scenarios to reduce this problem. The results of the modelling show that dense vegetation covers are able to reduce 100% of the sediment flux. Scattered low-density vegetation covers will be able to decrease the sediment flux by 40% while a combination of shrubs and grasses would be able to decrease the sediment flux by 70%. Vegetation scenarios in which the plan is to let the seeds spread out and create super low-density vegetation covers will result in the creation of fully developed wakes while only slightly protecting the soil against the momentum of the wind. This would lead to an in an increase of the soil erosion and would be a precarious strategy. A more patchy strategy with multiple high density vegetated plantations spread throughout the Aralkum would be best. Heavily striped vegetation is only as good as uniform spread vegetation in the case of high vegetation densities. To prevent dust storms, grass patches will be incredibly beneficial as they reduce shear stress on the soil while not increasing the roughness of the surface. Grass patches with or without other vegetation are able to stop 90% of the generation of dust particles in suspension. To vegetate these regions is essential for multiple reasons. Salt and dust storms salinize the nearby soils. Dust storms endanger local Uzbek population, causing cancer and raspatory problems leading to premature deaths. Action is required to mitigate the impact of this humanitarian and ecological disaster.

References

Abatzoglou, J.T., S.Z. Dobrowski, S.A. Parks, K.C. Hegewisch., 2018. Terraclimate, a high-resolution global dataset of monthly climate and climatic water balance from 1958-2015, Scientific Data 5:170191, doi: 10.1038/sdata.2017.191

Akhtar, F., Tischbein, B., & Awan, U. K., 2013. Optimizing deficit irrigation scheduling under shallow groundwater conditions in lower reaches of Amu Darya River Basin. Water resources management, 27(8), 3165-3178.

Alfaro, S. C., Flores-Aqueveque, V., Foret, G., Caquineau, S., Vargas, G., & Rutllant, J. A., 2011. A simple model accounting for the uptake, transport, and deposition of wind-eroded mineral particles in the hyperarid coastal Atacama Desert of northern Chile. Earth Surface Processes and Landforms, 36(7), 923-932.

Awan, U. K., Tischbein, B., & Martius, C., 2014. A GIS-based approach for up-scaling capillary rise from field to system level under soil–crop–groundwater mix. Irrigation science, 32(6), 449-458.

Begdullayeva, T., Kienzler, K. M., Kan, E., Ibragimov, N., & Lamers, J. P., 2007. Response of Sorghum bicolor varieties to soil salinity for feed and food production in Karakalpakstan, Uzbekistan. Irrigation and Drainage Systems, 21(3-4), 237-250.

Botman, E., 2009. Forest rehabilitation in the Republic of Uzbekistan. Keep Asia green, 253.

Bradford, G. R. (2015). Investigations of surface roughness length modification in Black Rock City, NV (Doctoral dissertation, San Francisco State University).

Breckle, Siegmar-Walter., & Walter Wucherer., 2012. "Climatic conditions in the Aralkum." Aralkum-a Man-Made Desert. Springer, Berlin, Heidelberg,. 49-72.

Breckle, S., 2013. "From Aral Sea to Aralkum: An Ecological Disaster or Halophytes' Paradise." Progress in Botany. Springer, Berlin, Heidelberg. 351-398.

Chander, G., Markham, B. L., & Helder, D. L., 2009. Summary of current radiometric calibration coefficients for Landsat MSS, TM, ETM+, and EO-1 ALI sensors. Remote sensing of environment, 113(5), 893-903.

Cretaux, J., Letolle, R., & Bergé-Nguyen, M., 2013. "History of Aral Sea level variability and current scientific debates." Global and Planetary Change 110 : 99-113.

Dimeyeva, L, A., Breckle, S., & Wucherer, W., 2012. "Flora of the Aralkum." Aralkum-a Man-Made Desert. Springer, Berlin, Heidelberg. 109-126.

Djumaeva, D., Lamers, J. P. A., Khamzina, A., & Vlek, P. L. G., 2013. The benefits of phosphorus fertilization of trees grown on salinized croplands in the lower reaches of Amu Darya, Uzbekistan. Agroforestry systems, 87(3), 555-569.

Elmore, A. J., Mustard, J. F., Manning, S. J., & Lobell, D. B., 2000. Quantifying vegetation change in semiarid environments: precision and accuracy of spectral mixture analysis and the normalized difference vegetation index. Remote sensing of environment, 73(1), 87-102.

Engelstaedter, S., Kohfeld, K ,E., Tegen, I., & Harrison, S. P., 2003. Controls of dust emissions by vegetation and topographic depressions: An evaluation using dust storm frequency data. Geophysical Research Letters, 30(6).

Vermote, E., Justice, C., Csiszar, I., Eidenshink, J., Myneni, R., Baret, F., Masuoka, E., Wolfe, R., Claverie, M., and NOAA CDR Program 2014: NOAA Climate Data Record (CDR) of Normalized Difference Vegetation Index (NDVI), Version 4. [indicate subset used]. NOAA National Climatic Data Center. doi:10.7289/V5PZ56R6

Forkutsa, I., Sommer, R., Shirokova, Y. I., Lamers, J. P. A., Kienzler, K., Tischbein, B., ... & Vlek, P. L. G. 2009. Modeling irrigated cotton with shallow groundwater in the Aral Sea Basin of Uzbekistan: I. Water dynamics. Irrigation Science, 27(4), 331-346

Glendening, J. W. 1977. Aeolian transport and vegetative capture of particulates. Atmospheric science paper; no. 310.

Greeley, R., White, B., Leach, R., Iversen, J., & Pollack, J. 1976. Mars: Wind friction speeds for particle movement. Geophysical Research Letters, 3(8), 417-420.

Guido Kersten., 2015. The effect of vegetation on wind erosion in the African Sahel dessert.

Guo, H., Bao, A., Liu, T., Ndayisaba, F., Jiang, L., Kurban, A., & De Maeyer, P., 2018. Spatial and temporal characteristics of droughts in Central Asia during 1966–2015. Science of the total environment, 624, 1523-1538.

Hua, T., Wang, X., Zhang, C., Lang, L., & Li, H., 2017. Responses of vegetation activity to drought in northern China. Land Degradation & Development, 28(7), 1913-1921..

Ibrakhimov, M., Khamzina, A., Forkutsa, I., Paluasheva, G., Lamers, J. P. A., Tischbein, B., ... & Martius, C. 2007. Groundwater table and salinity: Spatial and temporal distribution and influence on soil salinization in Khorezm region (Uzbekistan, Aral Sea Basin). Irrigation and Drainage systems, 21(3-4), 219-236

Ibrakhimov, M., Park, S. J., & Vlek, P. L. (2004). Development of groundwater salinity in a region of the lower Amu-Darya River, Khorezm, Uzbekistan. In Agriculture in Central Asia: Research for Development. Proceedings of a Symposium held at the American Society of Agronomy Annual Meetings at Indianapolis, Indiana, USA, ICARDA, Aleppo, Syria (pp. 56-75).

Indoitu, R., Orlovsky, L., & Orlovsky, N., 2012. Dust storms in Central Asia: spatial and temporal variations. Journal of Arid Environments, 85, 62-70.

Issanova, G., Abuduwaili, J., Galayeva, O., Semenov, O., & Bazarbayeva, T., 2015. Aeolian transportation of sand and dust in the Aral Sea region. International journal of environmental science and technology, 12(10), 3213-3224.

Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., ... & Zhu, Y. 1996. The NCEP/NCAR 40-year reanalysis project. Bulletin of the American meteorological Society, 77(3), 437-472.

Kawamura, R., 1964. Study of sand movement by wind. Hydraulic Eng. Lab. Rep. Number HEL-2-8, Berkeley, University of California, pp. 99–108.

Kind, R.J., 1976. A critical examination of the requirements for model simulation of wind-induced erosion-deposition phenomena such as snow drifting. Atmospheric Environment 10, 219–227.

Khamraev, Aloviddin Sh., 2003. "Soil organisms and entomocomplexes in Khorezm and Karakalpakstan (Uzbekistan)." ZEF Work Papers for Sustainable Development in Central Asia 6 : 71.

Kok, J. F., Parteli, E. J., Michaels, T. I., & Karam, D. B., 2012. The physics of wind-blown sand and dust. Reports on progress in Physics, 75(10), 106901.

Kulmatov, R., Mirzaev, J., Abuduwaili, J., & Karimov, B. 2020. Challenges for the sustainable use of water and land resources under a changing climate and increasing salinization in the Jizzakh irrigation zone of Uzbekistan. Journal of Arid Land, 12, 90-103.

Létolle, R., Aladin, N., Filipov, I., Boroffka NGO., 2005. The future chemical evolution of the Aral Sea from 2000 to the years 2050. Mitig Adapt Strateg Glob Change 10:51–70

Lettau, H., 1969. Note on aerodynamic roughness-parameter estimation on the basis of roughness-element description. Journal of applied meteorology, 8(5), 828-832.

Lettau, K., Lettau, H.H., 1978. Experimental and micro-meteorological field studies of dune migration, exploring the world's driest climate. IES Report 101, University of Winsconsin-Madison, Institute for Environmental Studies, Madison, USA, pp. 110–147.

Li, J., Okin, G. S., Alvarez, L., & Epstein, H. (2007). Quantitative effects of vegetation cover on wind erosion and soil nutrient loss in a desert grassland of southern New Mexico, USA. Biogeochemistry, 85(3), 317-332.

Lott, Neal., Data Documentation for Federal Climate Complex Integrated Surface Hourly Data. [Asheville, N.C.]: National Climatic Data Center, 2000.

Lyapustin, A., Wang, Y., 2018. MCD19A2 MODIS/Terra+Aqua Land Aerosol Optical Depth Daily L2G Global 1km SIN Grid V006 [Data set]. NASA EOSDIS Land Processes DAAC. Accessed 2020-07-04 from https://doi.org/10.5067/MODIS/MCD19A2.006

MacKay, Joseph., 2009 "Running dry: international law and the management of Aral Sea depletion." Central Asian Survey 28.1 : 17-27.

Max spoor., 1998. The Aral Sea Basin Crisis: Transition and Environment in Former Soviet Central Asia Development and Change [0012-155X] yr: vol:29 iss:3 pg:409 -435

Mayaud, J. R., & Webb, N. P. 2017., Vegetation in drylands: Effects on wind flow and aeolian sediment transport. Land, 6(3), 64.

Mayaud, J. R., Bailey, R. M., & Wiggs, G. F., 2017. A coupled vegetation/sediment transport model for dryland environments. Journal of Geophysical Research: Earth Surface, 122(4), 875-900.

Maegley, W.J., 1976. Saltation and Martian sandstorms. Reviews of Geophysics 14, 135–142.

McKee, T. B., Doesken, N, J., & Kleist, J., 1993. The relationship of drought frequency and duration to time scales. Preprints, Eighth Conf. on Applied Climatology. Anaheim, CA, Amer. Meteor. Soc., 179–184.

Micklin, Philip P., 1988. "Desiccation of the Aral Sea: a water management disaster in the Soviet Union." Science 241.4870 : 1170-1176.

Micklin, Philip., 2007. "The Aral sea disaster." Annu. Rev. Earth Planet. Sci. 35 : 47-72.

Micklin, Philip., 2016. "The future Aral Sea: hope and despair." Environmental Earth Sciences 75.9 : 844.

Micklin, P., Aladin, N. V., & Plotnikov, I., 2016. Aral Sea. Springer-Verlag Berlin An.

Novitskiy, Z. B., 2012. Phytomelioration in the southern Aralkum. In Aralkum-a Man-Made Desert (pp. 387-406). Springer, Berlin, Heidelberg.

Nezlin, N. P., Kostianoy, A. G., & Li, B. L. (2005). Inter-annual variability and interaction of remote-sensed vegetation index and atmospheric precipitation in the Aral Sea region. Journal of Arid Environments, 62(4), 677-700.

Palmer, W. C., 1965: Meteorological droughts. U.S. Department of Commerce, Weather Bureau Research

Parolari, A. J., Li, D., Bou-Zeid, E., Katul, G. G., & Assouline, S., 2016. Climate, not conflict, explains extreme Middle East dust storm. Environmental Research Letters, 11(11), 114013.

Peters, A. J., Walter-Shea, E. A., Ji, L., Vina, A., Hayes, M., & Svoboda, M. D., 2002. Drought monitoring with NDVI-based standardized vegetation index. Photogrammetric engineering and remote sensing, 68(1), 71-75.

Qi, J., Kulmatov, R., 2008. An overview of environmental issues in Central Asia

Environmental Problems of Central Asia and Their Economic, Social and Security Impacts, Springer, pp. 3-14

Raupach, M. 1992. Drag and drag partition on rough surfaces. Boundary-Layer Meteorology, 60(4), 375-395.

Suvdantsetseg, B., H. Fukui, and R. Tsolmon., 2008. "Saxaul forest area determination by remote sensing in Mongolia’s Gobi region." Proceedings of “The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences”, Beijing .

Somayeh, H., 2014. "Impact of aeolian dust accumulation on some biochemical parameters in black saxaul (Haloxylon aphyllum Bunge) leaves: a case study for the Aran-Bidgol region, Iran." International Journal of Forest, Soil and Erosion (IJFSE) 4.1 : 11-15.

Sutton, W. R., Whitford, P., Montanari Stephens, E., Pedroso Galinato, S., Nevel, B., Plonka, B., & Karamete, E., 2008. Integrating environment into agriculture and forestry: progress and prospects in Eastern Europe and Central Asia. The World Bank.

Saiko, T. A., & Zonn, I. S., 2000. Irrigation expansion and dynamics of desertification in the Circum-Aral region of Central Asia. Applied Geography, 20(4), 349-367.https://doi.org/10.1016/S0143-6228(00)00014-X

Shao, Y., 2008. Physics and modelling of wind erosion (Vol. 37). Springer Science & Business Media.

Shibuo, Y., Jarsjö, J., & Destouni, G. (2007). Hydrological responses to climate change and irrigation in the Aral Sea drainage basin. Geophysical Research Letters, 34(21).

Shomurodov, H. F., Saribayeva, S. U., & Akhmedov, A., 2015. Distribution pattern and modern status of rare plant species on the Ustyurt Plateau in Uzbekistan. Arid Ecosystems, 5(4), 261-267.

Singer, A., Zobeck, T., Poberezsky, L., & Argaman, E., 2003. The PM10and PM2· 5 dust generation potential of soils/sediments in the Southern Aral Sea Basin, Uzbekistan. Journal of Arid Environments, 54(4), 705-728.

Small, E. E., Giorgi, F., Sloan, L. C., & Hostetler, S. (2001). The effects of desiccation and climatic change on the hydrology of the Aral Sea. Journal of Climate, 14(3), 300-322.

Sterk, G., Parigiani, J., Cittadini, E., Peters, P., Scholberg, J., & Peri, P. (2012). Aeolian sediment mass fluxes on a sandy soil in Central Patagonia. Catena, 95, 112-123.

Stulina, G., & Sektimenko, V., 2004. The change in soil cover on the exposed bed of the Aral Sea. Journal of marine systems, 47(1-4), 121-125.

Tucker, C. J. (1979). Red and photographic infrared linear combinations for monitoring vegetation. Remote sensing of Environment, 8(2), 127-150.

Turnbull, L., Hochstrasser, T., Wieczorek, M., Baas, A., Wainwright, J., Scarsoglio, S., ... & Mueller, E. N. (2014). Approaches to modelling ecogeomorphic systems. In Patterns of Land Degradation in Drylands (pp. 171-209). Springer, Dordrecht.

Tischbein, B., Awan, U.K., Abdullaev, I., Bobojonov, I., Conrad, C., Jabborov, H., Forkutsa, I., Ibrakhimov, M., Poluasheva, G., Martius, C., 2012. Water management in Khorezm: Current situation and options for improvement (hydrological perspective). In Cotton, Water, Salts and Soums; Martius, C., Rudenko, I., Lamers, J., Vlek, P., Eds.; Springer: Dordrecht, The Netherlands, pp. 69–92

Vicente-Serrano, S. M., Beguería, S., & López-Moreno, J. I., 2010. A multiscalar drought index sensitive to global warming: the standardized precipitation evapotranspiration index. Journal of climate, 23(7), 1696-1718.

Vinogradov, Sergei., & Vance PE Langford., 2001. "Managing transboundary water resources in the Aral Sea Basin: in search of a solution." International Journal of Global Environmental Issues 1.3-4 : 345-362.

Wang, Hongshuo., Jeffrey C. Rogers., and Darla K. Munroe., 2015. "Commonly used drought indices as indicators of soil moisture in China." Journal of Hydrometeorology 16.3 : 1397-1408.

Wolfe, S. A., & Nickling, W. G., 1993. The protective role of sparse vegetation in wind erosion. Progress in physical geography, 17(1), 50-68.

Xu, J. 2006. Sand-dust storms in and around the Ordos Plateau of China as influenced by land use change and desertification. Catena, 65(3), 279-284.

Xu, Hao-jie., Xin-ping Wang., & Xiao-xiao Zhang., 2016. "Decreased vegetation growth in response to summer drought in Central Asia from 2000 to 2012." International journal of applied earth observation and geoinformation 52 : 390-402.

Xi, X., & Sokolik, I. N., 2016. Quantifying the anthropogenic dust emission from agricultural land use and desiccation of the Aral Sea in Central Asia. Journal of Geophysical Research: Atmospheres, 121(20), 12-270. https://doi.org/10.1002/2016JD025556

Youlin Y., Squired V., Qi L., 2002. Global alarm: dust and sand storms from the World's Drylands, UNCCD report, Bangkok, pp 169–201

Zingg, A.W., 1953. Wind tunnel studies of the movement of sedimentary material. Proceedings of the 5th Hydraulic Conference, University of Iowa Studies in Engineering Bull., 34, pp. 111–135.

Dust storm over the Aral Sea. (2008, April 19). Retrieved September, 2020, https://earthobservatory.nasa.gov/images/19853/dust-storm-over-the-aral-sea

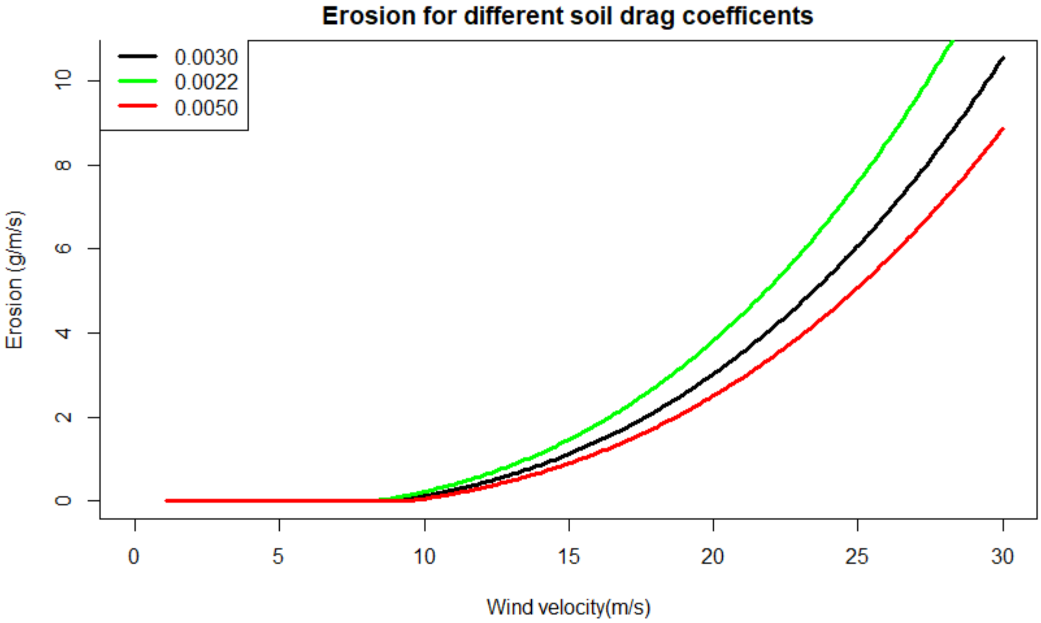
About the SPEI, Drought monitoring and analysis, viewed 2020, https://spei.csic.es/home.html

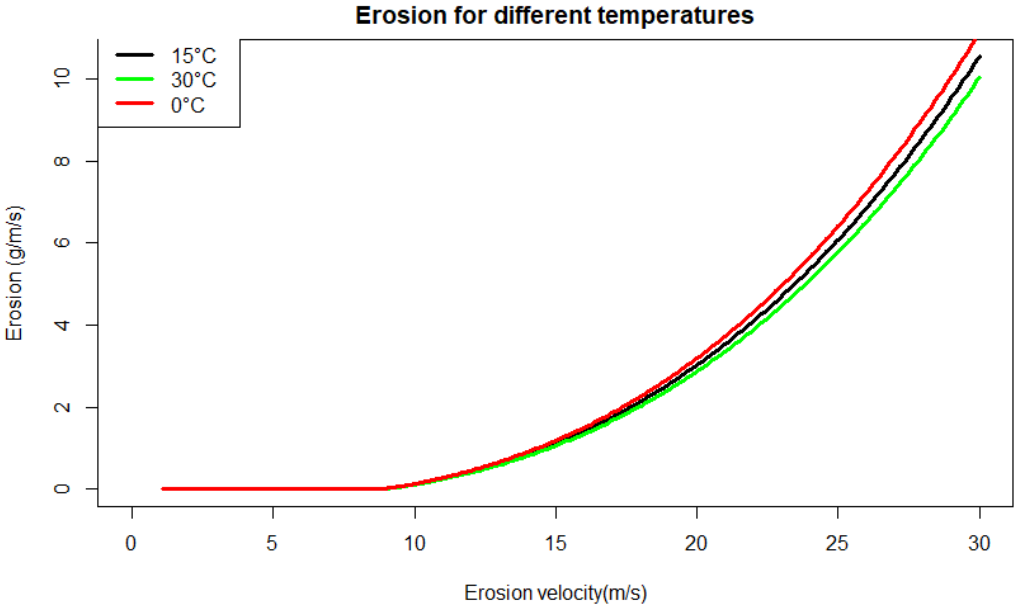
AOD dataset, viewed 2020 , https://disc.gsfc.nasa.gov/datasets/OMAERUV_V003/summary

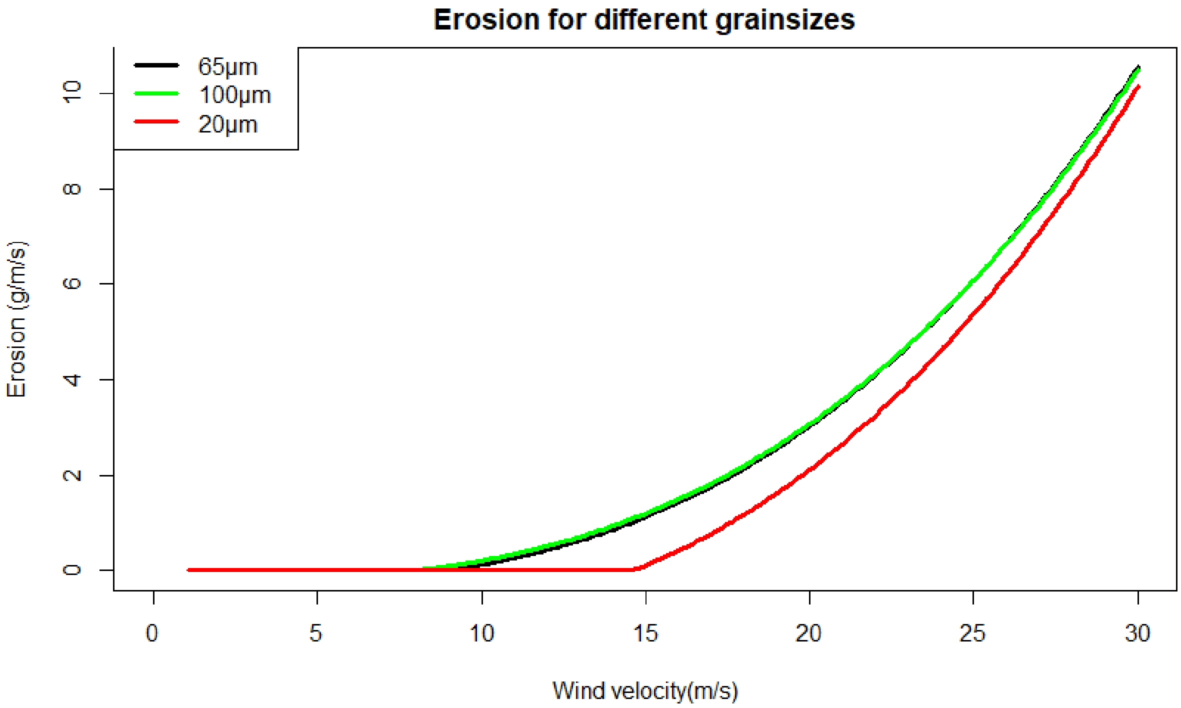
Appendix

Sensitivity analysis

To assess the sensitivity of the model to certain parameters, a sensitivity analysis was done for environmental parameters. The effect of grainsize, air temperature(it controls air density) and surface drag. The model was run for different speeds for the standard value, a high and a low estimate.


Erosion for different soil drag coefficents


Erosion for different temperatures



The sensitivity analysis showed the grainsize has a large effect on the velocity threshold in which grains come into saltation, the wind velocity required to bring grains in saltation has a minimum velocity around 90µm resulting in 65 and 100µm grains having the same velocity-Erosion curve. The curves of 20µm 65/100µm converge at higher windspeeds since smaller grainsizes however harder to get in saltation are much easier to keep in saltation. Temperature has no effect on the threshold of wind-erosion, and has a minor effect on the quantity of wind-erosion at high velocities. Surface drag has a minor effect on the threshold and a relatively large effect on wind erosion at high speeds.

Code

Earth engine

Temperature Uzbekistan

```
// selecting months and time period



var startdate= ee.Date('2000-01-01');

var time_period= 18;//years

var time_step = 3; //months

var enddate= ee.Date(startdate.advance( time_period, 'year').advance(-1, 'second'));
```

```
print(startdate);
print(enddate);


//selecting imagecollection
var CollectionTemp= ee.ImageCollection('NCEP_RE/surface_temp').select('air');
// IDAHO_EPSCOR/TERRACLIMATE
//NASA/GLDAS/V021/NOAH/G025/T3H


//creating imagecollection to later be filled
var collect =ee.ImageCollection([]);

var i;
for (i=0 ; i < time_period ; i++){
 var  begin_interval= startdate.advance(i, 'year');
 var  end_interval= begin_interval.advance(time_step, 'month').advance( -1, 'second');

 var gemidelde= ee.Image(CollectionTemp.filterDate(begin_interval, end_interval).mean());
 var gemidelde= gemidelde.clip(geometry);
 var gemidelde= gemidelde.add(-273);
 gemidelde=gemidelde.set('system:time_start',(begin_interval));
var add_image=ee.ImageCollection([gemidelde]);
collect=collect.merge(add_image);
 }

print(collect.size())
```

```
var www=(ui.Chart.image.series(collect,geometry,ee.Reducer.mean(),5000 ));

print(www);
```

Dust

```
//var begin=ee.Date('2008-4-27')

//var end=ee.Date('2008-4-30')

var begin=ee.Date('2003-04-17')

var end=ee.Date('2003-04-20')

var test=ee.ImageCollection("MODIS/006/MCD19A2_GRANULES").select('Optical_Depth_055').filterDate(begin,end).select('Optical_Depth_055')

var image= test.mean()

image=image.clip(geometry)

var viz = {

  min: 0,

  max: 300,

  palette: [' #ffff00','#ffcc66','#663300'],

  opacity: 0.75

};

Map.addLayer(image,viz)

var viz = {
  min: 0,

  max: 300,

  palette: [' #ffff00','#ffcc66','#663300'],

};



// set position of panel

var legend = ui.Panel({

  style: {

    position: 'bottom-left',

    padding: '8px 15px'

  }

});



// Create legend title

var legendTitle = ui.Label({

  value: 'Aerosol optical depth',

  style: {

    fontWeight: 'bold',

    fontSize: '18px',

    margin: '0 0 4px 0',

    padding: '0'

    }

});


 // Add the title to the panel
```

```
legend.add(legendTitle);


// create the legend image

var lon = ee.Image.pixelLonLat().select('latitude');

var gradient = lon.multiply((viz.max-viz.min)/100.0).add(viz.min);

var legendImage = gradient.visualize(viz);


// create text on top of legend

var panel = ui.Panel({

    widgets: [

      ui.Label(viz['max'])

    ],

  });


legend.add(panel);


// create thumbnail from the image

var thumbnail = ui.Thumbnail({

  image: legendImage,

  params: {bbox:'0,0,10,100', dimensions:'10x200'},

  style: {padding: '1px', position: 'bottom-center'}

});


// add the thumbnail to the legend

legend.add(thumbnail);


// create text on top of legend
```

```
var panel = ui.Panel({
   widgets: [
     ui.Label(viz['min'])
   ],
  });

legend.add(panel);

Map.add(legend);

var begin=ee.Date('2014-01-01')
var end=ee.Date('2015-01-01')

var
test=ee.ImageCollection("MODIS/006/MCD19A2_GRANULES").select('Optical_Depth_055').filterDate
(begin,end).filterBounds(geometry4)
var reduced_dataset=ee.ImageCollection([])
var dailyimage;
var dailyset;
var Days_it_will_run=360;
var j;
for (j = 0; j < Days_it_will_run ; j++) {
  dailyset=test.filterDate(begin.advance(j,'days'),begin.advance(j+1,'days'))
  var time=dailyset.first().get('system:time_start');
  dailyimage=dailyset.mean().set('system:time_start',time)
  dailyimage=dailyimage.clip(geometry4)

  reduced_dataset=reduced_dataset.merge(dailyimage)
```

```
}


print(reduced_dataset.size())

print(test.size())

//reduced_dataset=test


var river_area = reduced_dataset.map(function(image) {
  var anti_crash= image.reduce(ee.Reducer.mean()).rename('river_area');
  anti_crash=anti_crash.clip(geometry1)
  return ee.Image(anti_crash).set('system:time_start', image.get('system:time_start'));
});



var old_aral_sea = reduced_dataset.map(function(image) {
  var anti_crash= image.reduce(ee.Reducer.mean()).rename('old_aral_sea');
  anti_crash=anti_crash.clip(geometry2)
  return ee.Image(anti_crash).set('system:time_start', image.get('system:time_start'));
});


var dessert = reduced_dataset.map(function(image) {
  var anti_crash= image.reduce(ee.Reducer.mean()).rename('dessert');
  anti_crash=anti_crash.clip(geometry3)

  return ee.Image(anti_crash).set('system:time_start', image.get('system:time_start'));

});



var Dust_data=dessert.merge(old_aral_sea)

Dust_data=Dust_data.merge(river_area)


print(ui.Chart.image.series(Dust_data,geometry4,ee.Reducer.mean(),500,'system:time_start' )

.setOptions({

  title: 'Dust_data',

  vAxis: {title: 'optical depth'}}));
```

NDVI

```
var aralcov_1960_polygon = ee.Geometry.Polygon([[58.53100164420937,45.59099079345098]

, [58.36620672233437,45.305815207015286]

, [58.26732976920937,45.06577420855824]

, [58.16845281608437,44.90259069323471]

, [58.19042547233437,44.77014953835887]

, [58.15746648795937,44.4102532181692]

, [58.10253484733437,44.11915438521379]

, [58.03661687858437,44.05602457830358]

, [58.00365789420937,43.921650040063184]

, [58.16845281608437,43.81076024920512]

, [58.27831609733437,43.71554768256729]

, [58.53100164420937,43.755238000846816]
```

,[58.79467351920937,43.739365030366166]

, [59.04735906608437,43.707606462132006]

, [59.66259344108437,43.71554768256729]

, [59.99218328483437,43.755238000846816]

, [61.16571844108437,43.78697131402503]

, [61.14262273795937,43.93747459859828]

, [60.96459539420937,44.2137227578639]

, [61.25136297233437,44.66866562077574]

,[61.48319890983437,44.94148604315535]

, [61.61054265983437,45.15106590751395]

, [61.90380437858437,45.19753493469595]

, [61.77196844108437,45.39846256326186]

, [61.74237859733437,45.49095825896429]

, [61.47646062858437,45.92881174266003]

, [61.45448797233437,46.11133808448483]

, [61.54237859733437,46.41515113500167]

, [61.74999578483437,46.89663096616474]

, [61.34350164420937,46.839872884414595]

, [60.80517156608437,46.777505039913336]

, [60.43163640983437,46.8000717177866]

, [60.25585515983437,46.73234329930523]

, [59.94823797233437,46.59663096616474]

, [59.72851140983437,46.505966783433685]

, [59.48681219108437,46.400000466022156]

, [59.34398992545937,46.29382794831035]

, [59.20116765983437,46.17223539190432]

, [58.98144109733437,46.05799768400151]

```
, [58.83861883170937,45.966437001129194]
,[58.58593328483437,45.77519868628525],[58.53100164420937,45.59099079345098]]);
var aralcov_1960=ee.Image(42).clip(aralcov_1960_polygon);
var test=ee.Image(42)


var middle=ee.Geometry.Point(59.94, 44.99)
var buffer_100 = middle.buffer(100000)
var buffer_300 = middle.buffer(300000)
var buffer_500 = middle.buffer(500000)

var uzbek = ee.FeatureCollection(table4)
var range = ee.FeatureCollection(table3)
var irri = ee.FeatureCollection(table5)

var buffer_100 = buffer_100.intersection(uzbek, ee.ErrorMargin(1));
var buffer_300 = buffer_300.intersection(uzbek, ee.ErrorMargin(1));
var buffer_500 = buffer_500.intersection(uzbek, ee.ErrorMargin(1));
buffer_500=buffer_300.symmetricDifference(buffer_500, ee.ErrorMargin(1))
buffer_300=buffer_100.symmetricDifference(buffer_300, ee.ErrorMargin(1))
//Map.addLayer(buffer_500)

var buffer_100_range = buffer_100.intersection(range, ee.ErrorMargin(1));
var buffer_300_iri = buffer_300.intersection(irri, ee.ErrorMargin(1));
var buffer_300_range = buffer_300.intersection(range, ee.ErrorMargin(1));
var buffer_500_iri = buffer_500.intersection(irri, ee.ErrorMargin(1));
var buffer_500_range = buffer_500.intersection(range, ee.ErrorMargin(1));
var pointy = middle.buffer(10000)

//Map.addLayer(buffer_300_iri, {color: '22EC30'})
//Map.addLayer(buffer_500_iri, {color: '1807FF'})
//Map.addLayer(pointy, {color: '947E0F'})

Map.addLayer(buffer_100_range, {color: 'AC762F'})
Map.addLayer(buffer_300_range, {color: 'C4E22B'})
Map.addLayer(buffer_500_range, {color: '1BFC22'})

var Water_cover_1990=(ee.ImageCollection('JRC/GSW1_1/YearlyHistory').filterDate('1990-05-05','1991-05-05'))
var Water_cover_1990=Water_cover_1990.first()
Water_cover_1990=Water_cover_1990.clip(aralcov_1960_polygon)
// checking if correct date
var date = ee.Date(Water_cover_1990.get('system:time_start'));
//print('Timestamp: ', date); // ee.Date
//boolean map with where water is
var aralcov_1990=Water_cover_1990.eq(3)
var aralcov_1990 = test.updateMask(aralcov_1990);
//Map.addLayer(aralcov_1987);

// creating map of dried up between 1960 and 1987
var aralcov_1960_1990=aralcov_1960.where(aralcov_1990.eq(42) ,420 )
aralcov_1960_1990=aralcov_1960_1990.lt(100)
```

```
aralcov_1960_1990 = test.updateMask(aralcov_1960_1990);


var Water_cover_2017=(ee.ImageCollection('JRC/GSW1_1/YearlyHistory').filterDate('2017-05-05','2020-05-05'))

var Water_cover_2017=Water_cover_2017.first()

Water_cover_2017=Water_cover_2017.clip(aralcov_1960_polygon)

// checking if correct date

var date = ee.Date(Water_cover_2017.get('system:time_start'));

print('Timestamp: ', date); // ee.Date

//boolean map with where water is

var aralcov_2017=Water_cover_2017.eq(3)

var aralcov_2017 = test.updateMask(aralcov_2017);



// creating map of dried up between 1987 and 2009

var aralcov_1990_2017=aralcov_1990.where(aralcov_2017.eq(42) ,420 )

aralcov_1990_2017=aralcov_1990_2017.lt(100)

aralcov_1990_2017 = test.updateMask(aralcov_1990_2017)



//Map.addLayer(aralcov_1960_1990,{color: '02100D'});

//Map.addLayer(aralcov_1990_2017,{color: 'AAB5B3'});


var collection=ee.ImageCollection("MODIS/006/MOD13Q1").select('NDVI')

var collection2=ee.ImageCollection("NOAA/CDR/AVHRR/NDVI/V5").select('NDVI')


var begin_date_1= ee.Date('1990-01-01')

var ndvi_collect=ee.ImageCollection([])
var i;

for (i = 0; i < 1 ; i++) {

  var date_1=begin_date_1.advance(i,'month')

  var date_2=begin_date_1.advance(1+i,'month')

  var collection_image=collection2.filterDate(date_1,date_2)

  var time_image=collection_image.first()

  collection_image=collection_image.mean().set('system:time_start',
time_image.get('system:time_start'))


  //collection_image=collection_image.clip(buffer_300_iri)

  ndvi_collect=ndvi_collect.merge(collection_image)

}




print(ndvi_collect.size())

print(collection.size())


var a1960_1990 = ndvi_collect.map(function(image) {

  var time=image

  image=aralcov_1960_1990.where(aralcov_1960_1990.eq(42),image)

  var anti_crash= image.reduce(ee.Reducer.mean()).divide(10000).rename('1960_1990');

  return ee.Image(anti_crash).set('system:time_start', time.get('system:time_start'));

});


var a1990_2020 = ndvi_collect.map(function(image) {

  var time=image
  image=aralcov_1990_2017.where(aralcov_1990_2017.eq(42),image)
  var anti_crash= image.reduce(ee.Reducer.mean()).divide(10000).rename('1990_2020');
  return ee.Image(anti_crash).set('system:time_start', time.get('system:time_start'));
});


var buffer_300_iri = ndvi_collect.map(function(image) {
  image=image.clip(buffer_300_iri)
  var anti_crash= image.reduce(ee.Reducer.mean()).divide(10000).rename('buffer_300_iri');
  return ee.Image(anti_crash).set('system:time_start', image.get('system:time_start'));
});
var buffer_500_iri = ndvi_collect.map(function(image) {
  image=image.clip(buffer_500_iri)
  var anti_crash= image.reduce(ee.Reducer.mean()).divide(10000).rename('buffer_500_iri');
  return ee.Image(anti_crash).set('system:time_start', image.get('system:time_start'));
});
var buffer_100_range = ndvi_collect.map(function(image) {
  image=image.clip(buffer_100_range)
  var anti_crash= image.reduce(ee.Reducer.mean()).divide(10000).rename('buffer_100_range');
  return ee.Image(anti_crash).set('system:time_start', image.get('system:time_start'));
});
var buffer_300_range = ndvi_collect.map(function(image) {
  image=image.clip(buffer_300_range)
  var anti_crash= image.reduce(ee.Reducer.mean()).divide(10000).rename('buffer_300_range');
  return ee.Image(anti_crash).set('system:time_start', image.get('system:time_start'));
});
var buffer_500_range = ndvi_collect.map(function(image) {
  image=image.clip(buffer_500_range)
  var anti_crash= image.reduce(ee.Reducer.mean()).divide(10000).rename('buffer_500_range');

  return ee.Image(anti_crash).set('system:time_start', image.get('system:time_start'));

});


var merged_data=buffer_300_iri.merge(buffer_500_iri)

 merged_data=merged_data.merge(buffer_100_range)

 merged_data=merged_data.merge(buffer_300_range)

 merged_data=merged_data.merge(buffer_500_range)

 merged_data=merged_data.merge(a1960_1990)

 merged_data=merged_data.merge(a1990_2020)


 a1990_2020


print(ui.Chart.image.series(merged_data,uzbek,ee.Reducer.mean(),5000,'system:time_start' )

.setOptions({

  title: 'test',

  vAxis: {title: 'NDVI'}}));
```

R

5.1

```
setwd("C:/Users/timon/Documents/scriptie")

SPI_scale <- 6


# opening packages if not already openned

bool <- True

if (bool){
```

```
library(SPEI)
library(csvread)
library(imputeTS)
library(forecast)
library(tseries)
library(ggplot2)
library(ggplot)
library(data.table)
library(dplyr)
library(lubridate)
library(basicTrendline)
library(ISwR)
library(dplyr)
library(ggplot2)}
###reading the data


#reading datasets
range_100 = read.csv('100_range.csv', header=TRUE, sep=',')
iri_300 = read.csv('300_iri.csv', header=TRUE, sep=',')
range_300 = read.csv('300_range.csv', header=TRUE, sep=',')
iri_500 = read.csv('500_iri.csv', header=TRUE, sep=',')
range_500 = read.csv('500_range.csv', header=TRUE, sep=',')
sea_precip = read.csv('sea.CSV', header=TRUE, sep=',')



 SPEI_total_pricip <- range_100
 SPEI_total_pricip$aet <- range_100$pr-range_100$aet+1000
 SPEI_total_pricip$pdsi <- range_300$pr-range_300$aet+1000

 SPEI_total_pricip$pr <- range_500$pr-range_500$aet+1000

 SPEI_total_pricip$pds2i<- iri_300$pr-iri_300$aet+1000

 SPEI_total_pricip$pds3i<- iri_500$pr-iri_500$aet+1000

 SPEI_total_pricip$Desiccated  <- sea_precip$pr-sea_precip$aet+1000

 total_pricip <- range_100

 total_pricip$aet <- range_100$pr

 total_pricip$pdsi <- range_300$pr

 total_pricip$pr <- range_500$pr

 total_pricip$pds2i<- iri_300$pr

 total_pricip$pds3i<- iri_500$pr

 total_pricip$Desiccated  <- sea_precip$pr



#giving Data colums names

colnames(total_pricip) <- c( 'dates','range 0-100','range 100-300','range 300-500','iri 100-300','iri 300-500','Desiccated aral seabed')

colnames(SPEI_total_pricip) <- c( 'dates','range 0-100','range 100-300','range 300-500','iri 100-300','iri 300-500','Desiccated aral seabed')



write.csv(total_pricip,"C:/Users/timon/Documents/scriptie/precip_for_regions.csv", row.names = FALSE,append =TRUE)


#removing NA values

total_pricip <- na_interpolation(total_pricip, option = "linear")

SPEI_total_pricip <- na_interpolation(SPEI_total_pricip, option = "linear")

#cutting Precip data for smaller time frame

total_pricip_cut <- total_pricip[385:744,]
```

```
SPEI_total_pricip_cut <- SPEI_total_pricip[385:744,]


# Calculating SPI

iri_spi_300 <- spi(data=total_pricip$'iri 100-300', scale=SPI_scale, kernel = list(type = 'rectangular',
shift = 0),

              distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

              ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

iri_spi_500 <- spi(data=total_pricip$'iri 300-500', scale=SPI_scale, kernel = list(type = 'rectangular',
shift = 0),

           distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

           ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

range_spi_100 <- spi(data=total_pricip$'range 0-100', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

           distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

           ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

range_spi_300 <- spi(data=total_pricip$'range 100-300', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

           distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

           ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

range_spi_500 <- spi(data=total_pricip$'range 300-500', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

           distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

           ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

desiccated <- spi(data=total_pricip$'Desiccated aral seabed', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

# Calculating SPI

SPEI_iri_spi_300 <- spei(data=SPEI_total_pricip$'iri 100-300', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),
                 distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                 ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
SPEI_iri_spi_500 <- spei(data=SPEI_total_pricip$'iri 300-500', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                 distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                 ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
SPEI_range_spi_100 <- spei(data=SPEI_total_pricip$'range 0-100', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
SPEI_range_spi_300 <- spei(data=SPEI_total_pricip$'range 100-300', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
SPEI_range_spi_500 <- spei(data=SPEI_total_pricip$'range 300-500', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
SPEI_desiccated <- spei(data=SPEI_total_pricip$'Desiccated aral seabed', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
           distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
           ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)



par(mfcol = c(2, 3))
# plotting SPI
plot(iri_spi_300,main='Irrigated lands in the 0-300 km buffer')
plot(iri_spi_500,main='Irrigated lands in the 300-500 km buffer')
plot(range_spi_100,main='Range lands in the 0-100 km buffer')
plot(range_spi_300,main='Range lands in the 100-300 km buffer')

plot(range_spi_500,main='Range lands in the 300-500 km buffer')

plot(desiccated,main='Desiccated aral seabed')


#calculating SPI for smaller timeframe

iri_spi_300 <- spi(data=total_pricip_cut$'iri 100-300', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

iri_spi_500 <- spi(data=total_pricip_cut$'iri 300-500', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

range_spi_100 <- spi(data=total_pricip_cut$'range 0-100', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

                    distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                    ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

range_spi_300 <- spi(data=total_pricip_cut$'range 100-300', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

                    distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                    ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

range_spi_500 <- spi(data=total_pricip_cut$'range 300-500', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

                    distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                    ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

desiccated <- spi(data=total_pricip_cut$'Desiccated aral seabed', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

                 distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                 ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
```

```
SPEI_iri_spi_300 <- spei(data=SPEI_total_pricip_cut$'iri 100-300', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),

                   distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                   ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

SPEI_iri_spi_500 <- spei(data=SPEI_total_pricip_cut$'iri 300-500', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),

                   distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                   ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

SPEI_range_spi_100 <- spei(data=SPEI_total_pricip_cut$'range 0-100', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),

                     distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                     ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

SPEI_range_spi_300 <- spei(data=SPEI_total_pricip_cut$'range 100-300', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),

                     distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                     ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

SPEI_range_spi_500 <- spei(data=SPEI_total_pricip_cut$'range 300-500', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),

                     distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                     ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)

SPEI_desiccated <- spei(data=SPEI_total_pricip_cut$'Desiccated aral seabed', scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),

                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)




# plotting SPI for smaller timeframe

plot(iri_spi_300,main='Irrigated lands in the 0-300 km buffer')
plot(iri_spi_500,main='Irrigated lands in the 300-500 km buffer')

plot(range_spi_100,main='Range lands in the 0-100 km buffer')

plot(range_spi_300,main='Range lands in the 100-300 km buffer')

plot(range_spi_500,main='Range lands in the 300-500 km buffer')

plot(desiccated,main='Desiccated aral seabed')


####################

#reading the Vegation NDVI data and dates

date_1 = read.csv('C:/Users/timon/Documents/scriptie/1990_2005.csv', header=TRUE)

date_2 = read.csv('C:/Users/timon/Documents/scriptie/2005_2020.csv', header=TRUE)

date <- rbind(date_1,date_2)

data_veg1 <- csvread('C:/Users/timon/Documents/scriptie/1990_2005.csv', header=TRUE,
coltypes=c("string","double","double","double","double","double","double","double"))

data_veg2 <- csvread('C:/Users/timon/Documents/scriptie/2005_2020.csv', header=TRUE,
coltypes=c("string","double","double","double","double","double","double","double"))

data_veg <- rbind(data_veg1,data_veg2)

data_veg$system.time_start  <- date$system.time_start



data_veg$bad_Date <- format(data_veg$system.time_start, format="%b %d, %Y")

head(data_veg$bad_Date)

class(data_veg$bad_Date)

### making the date better

data_veg$Date <- as.Date(data_veg$bad_Date,format="%b %d, %Y")

head(data_veg$Date)

class(data_veg$Date)

data_veg$system.time_start <- NULL

data_veg$bad_Date <- NULL
```

```
grouped <- group_by(data_veg,Date)


data_veg1$buffer_500_iri[0:50]


#grouping the data


grouped_veg <-  summarize(grouped,X1960_1990=mean(X1960_1990,
na.rm=TRUE),X1990_2020=mean(X1990_2020, na.rm=TRUE),

                    buffer_100_range=mean(buffer_100_range,
na.rm=TRUE),buffer_300_range=mean(buffer_300_range, na.rm=TRUE),

                    buffer_500_range=mean(buffer_500_range,
na.rm=TRUE),buffer_300_iri=mean(buffer_300_iri, na.rm=TRUE),

                    buffer_500_iri=mean(buffer_500_iri, na.rm=TRUE))

# removing NAN

grouped_veg$X1960_1990[is.nan(grouped_veg$X1960_1990)] <- 0

grouped_veg$X1990_2020[is.nan(grouped_veg$X1990_2020)] <- 0

grouped_veg$buffer_100_range[is.nan(grouped_veg$buffer_100_range)] <- 0

grouped_veg$buffer_300_range[is.nan(grouped_veg$buffer_300_range)] <- 0

grouped_veg$buffer_500_range[is.nan(grouped_veg$buffer_500_range)] <- 0

grouped_veg$buffer_300_iri[is.nan(grouped_veg$buffer_300_iri)] <- 0

grouped_veg$buffer_500_iri[is.nan(grouped_veg$buffer_500_iri)] <- 0


# calculating SVI

SPI_scale <- 1


iri_svi_300 <- spi(data=grouped_veg$buffer_300_iri, scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),

               distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,

               ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
```

```
iri_svi_500 <- spi(data=grouped_veg$buffer_500_iri, scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                   distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                   ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
range_svi_100 <- spi(data=grouped_veg$buffer_100_range, scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                     distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                     ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
range_svi_300 <- spi(data=grouped_veg$buffer_300_range, scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                     distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                     ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
range_svi_500 <- spi(data=grouped_veg$buffer_500_range, scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                     distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                     ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
desiccated_1960 <- spi(data=grouped_veg$X1960_1990, scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
desiccated_1990 <- spi(data=grouped_veg$X1990_2020, scale=SPI_scale, kernel = list(type = 'rectangular', shift = 0),
                  distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
                  ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
plot(iri_svi_300)
#plotting SVI
#plot(iri_svi_300)
#plot(iri_svi_500)
#plot(range_svi_100)
#plot(range_svi_300)
```

```
#plot(range_svi_500)

#plot(desiccated_1960)

#plot(desiccated_1990)


# removing INF

iri_svi_300 <- replace(iri_svi_300$fitted, iri_svi_300$fitted=='-Inf', 0)

iri_svi_500 <- replace(iri_svi_500$fitted, iri_svi_500$fitted=='-Inf', 0)

range_svi_100 <- replace(range_svi_100$fitted, range_svi_100$fitted=='-Inf', 0)

range_svi_300 <- replace(range_svi_300$fitted, range_svi_300$fitted=='-Inf', 0)

range_svi_500 <- replace(range_svi_500$fitted, range_svi_500$fitted=='-Inf', 0)

desiccated_1960 <- replace(desiccated_1960$fitted, desiccated_1960$fitted=='-Inf', 0)

desiccated_1990 <- replace(desiccated_1990$fitted, desiccated_1990$fitted=='-Inf', 0)




(X1960_1990 <- ggplot(grouped_veg, aes(x = Date, y = X1960_1990)) +

   geom_line() +

   geom_smooth(method = "lm", se = FALSE, span = 0.6) +

   theme_classic() +

   ggtitle('Seabed Desiccated 1960-1990 ')+

   xlab('date') +

   ylab('NDVI')+

   theme(plot.title = element_text(hjust = 0.5)))
```

```
((a2$coefficients[2]*365*30)+a2$coefficients[1])/a2$coefficients[1]

# plotting



X1960_1990 <- ggplot(grouped_veg, aes(x = Date, y = X1960_1990)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE, span = 0.6) +
  theme_classic() +
  ggtitle('seabed Desiccated 1960-1990')+
  xlab('date') +
  ylab('NDVI')+
  theme(plot.title = element_text(hjust = 0.5))
X1990_2020 <- ggplot(grouped_veg, aes(x = Date, y = X1990_2020)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE, span = 0.6) +
  theme_classic() +
  ggtitle('seabed Desiccated 1990-2020')+
  xlab('date') +
  ylab('NDVI')+
  theme(plot.title = element_text(hjust = 0.5))

buffer_500_iri <- ggplot(grouped_veg, aes(x = Date, y = buffer_500_iri)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE, span = 0.6) +
  theme_classic() +
  ggtitle('Irigated lands 0-500 km buffer')+
  xlab('date') +
  ylab('NDVI')+
  theme(plot.title = element_text(hjust = 0.5))
buffer_100_range <- ggplot(grouped_veg, aes(x = Date, y = buffer_100_range)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE, span = 0.6) +
  theme_classic() +
  ggtitle('Range lands 0-100 km buffer')+
  xlab('date') +
  ylab('NDVI')+
  theme(plot.title = element_text(hjust = 0.5))
buffer_300_range <- ggplot(grouped_veg, aes(x = Date, y = buffer_100_range)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE, span = 0.6) +
  theme_classic() +
  ggtitle('Range lands 100-300 km buffer')+
  xlab('date') +
  ylab('NDVI')+
  theme(plot.title = element_text(hjust = 0.5))

buffer_500_range <- ggplot(grouped_veg, aes(x = Date, y = buffer_500_range)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE, span = 0.6) +
  theme_classic() +
  ggtitle('Range lands 300-500 km buffer')+
  xlab('date') +
  ylab('NDVI')+
  theme(plot.title = element_text(hjust = 0.5))
```

```
a2 <- lm(X1960_1990~Date,data=grouped_veg)

summary(a2)


a2 <- lm(X1990_2020~Date,data=grouped_veg)

summary(a2)


a2 <- lm(buffer_500_iri~Date,data=grouped_veg)

summary(a2)


a2 <- lm(buffer_100_range~Date,data=grouped_veg)

summary(a2)


a2 <- lm(buffer_300_range~Date,data=grouped_veg)

summary(a2)


a2 <- lm(buffer_500_range~Date,data=grouped_veg)

summary(a2)



library("ggpubr")


ggarrange(X1960_1990,X1990_2020, buffer_100_range,
buffer_300_range,buffer_500_range,buffer_500_iri,

          ncol = 3, nrow = 2)
```

```
plot(grouped_veg)



year_veg <- grouped_veg


year_veg$Date<- format(as.Date(year_veg$Date, "%Y-%m-%d"), "%Y")

year_veg <- group_by(year_veg,Date)


year_veg <-  summarize(year_veg,X1960_1990=mean(X1960_1990,
na.rm=TRUE),X1990_2020=mean(X1990_2020, na.rm=TRUE),

                    buffer_100_range=mean(buffer_100_range,
na.rm=TRUE),buffer_300_range=mean(buffer_300_range, na.rm=TRUE),

                    buffer_500_range=mean(buffer_500_range,
na.rm=TRUE),buffer_300_iri=mean(buffer_300_iri, na.rm=TRUE),

                    buffer_500_iri=mean(buffer_500_iri, na.rm=TRUE))

par(mfcol = c(1, 1))

plot(year_veg$Date,year_veg$X1960_1990,type='l',ylim =
c(0,0.3),lwd='3',col='black',xlab='Year',ylab='NDVI',main='Mean yearly NDVI')

lines(year_veg$Date,year_veg$X1990_2020,lwd='3',col='blue')

lines(year_veg$Date,year_veg$buffer_100_range,lwd='3',col='yellow')

lines(year_veg$Date,year_veg$buffer_300_range,lwd='3',col='gold')

lines(year_veg$Date,year_veg$buffer_500_range,lwd='3',col='Brown')

lines(year_veg$Date,(year_veg$buffer_500_iri),lwd='3',col='green')

legend(1988.84,0.312,legend=c('Desiccated 1960-1990','Desiccated 1990-2020','Range lands 0-
100','Range lands 100-300','Range lands 300-500','Irrigated lands 0-
500'),col=c('black','blue','yellow','gold','brown','green')
```

```
    , lty=1, cex=1,,lwd =3)



Year_vector <- seq(1990,(2020-1/12),(1/12))

plot(Year_vector,range_svi_100,xlab = 'Year',ylab='SVI',type='l',main='Range lands 0-100 km buffer ')

plot(Year_vector,range_svi_300,xlab = 'Year',ylab='SVI',type='l',main='Range lands 100-300 km buffer')

plot(Year_vector,range_svi_500,xlab = 'Year',ylab='SVI',type='l',main='Range lands 300-500 km buffer')

plot(Year_vector,iri_svi_500,xlab = 'Year',ylab='SVI',type='l',main='Irigated lands 0-500 km buffer')

plot(Year_vector,desiccated_1960,xlab = 'Year',ylab='SVI',type='l',main='Seabed Desiccated 1960-1990')

plot(Year_vector,desiccated_1990,xlab = 'Year',ylab='SVI',type='l',main='Seabed Desiccated 1990-2020')
```

```
test <- ts(grouped_veg$buffer_500_iri,start = 1990, end = 2020, freq = 12)  #  measurement frequency (monthly = 12)

# Decompose using `stl()`

test1 <- stl(test, s.window = "period")

# Generate plots

plot(test1)
```

```
# removing NA data


desiccated$fitted[is.na(desiccated$fitted)] <- 0

iri_spi_500$fitted[is.na(iri_spi_500$fitted)] <- 0

iri_spi_300$fitted[is.na(iri_spi_300$fitted)] <- 0

range_spi_500$fitted[is.na(range_spi_500$fitted)] <- 0

range_spi_300$fitted[is.na(range_spi_300$fitted)] <- 0

range_spi_100$fitted[is.na(range_spi_100$fitted)] <- 0


#desiccated_1960$fitted[is.na(desiccated_1960$fitted)] <- 0

##desiccated_1990$fitted[is.na(desiccated_1990$fitted)] <- 0
```

```
#iri_svi_500$fitted[is.na(iri_svi_500$fitted)] <- 0
#iri_svi_300$fitted[is.na(iri_svi_300$fitted)] <- 0
#range_svi_500$fitted[is.na(range_svi_500$fitted)] <- 0
#range_svi_300$fitted[is.na(range_svi_300$fitted)] <- 0
#range_svi_100$fitted[is.na(range_svi_100$fitted)] <- 0

SPI_scale <- 1

iri_svi_300 <- spi(data=grouped_veg$buffer_300_iri, scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),
         distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
         ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
iri_svi_300 <- replace(iri_svi_300$fitted, iri_svi_300$fitted=='-Inf', 0)

iri_spi_300 <- spi(data=total_pricip_cut$'iri 100-300', scale=SPI_scale, kernel = list(type =
'rectangular', shift = 0),
         distribution = 'Gamma', fit = 'ub-pwm', na.rm = FALSE,
         ref.start=NULL, ref.end=NULL, x=FALSE, params=NULL)
iri_spi_300$fitted[is.na(iri_spi_300$fitted)] <- 0


testspi <- iri_spi_300$fitted
testsvi <- iri_svi_300

testyy <- c()
bebe <- seq(1,12,1)
for (abab in bebe){
grow <- c(abab)
totgrow <- c()
vectorgrowing <- seq(0,29,1)

for (numbeer in vectorgrowing) {
  vary <- grow+numbeer*12
  totgrow <- c(totgrow,vary)
}

corri <- cor(testsvi[totgrow],testspi[totgrow])
testyy <- c(testyy,corri)
}
plot(testyy,type='l',xlab='months',ylab='Cor')
# calculating correlations

print(cor(range_spi_100$fitted,range_svi_100))
print(cor(range_spi_300$fitted,range_svi_300))
print(cor(range_spi_500$fitted,range_svi_500))
print(cor(iri_spi_300$fitted,iri_svi_300))
print(cor(iri_spi_500$fitted,iri_svi_500))
print(cor(desiccated$fitted,desiccated_1960))
print(cor(desiccated$fitted,desiccated_1990))


cor.test(range_spi_100$fitted,range_svi_100)
cor.test(range_spi_300$fitted,range_svi_300)
cor.test(range_spi_500$fitted,range_svi_500)
cor.test(iri_spi_300$fitted,iri_svi_300)
cor.test(iri_spi_500$fitted,iri_svi_500)

cor.test(desiccated$fitted,desiccated_1960)

cor.test(desiccated$fitted,desiccated_1990)


preci <- total_pricip

preci_month <- total_pricip

preci_year <- total_pricip


preci_month$dates<- format(as.Date(preci$dates, "%b %d, %Y"), "%m")


grouped_month <- group_by(preci_month,dates)

grouped_month <-  summarize(grouped_month,`range 0-100`=mean(`range 0-100`,
na.rm=TRUE),`range 100-300`=mean(`range 100-300`, na.rm=TRUE),

                        `range 300-500`=mean(`range 300-500`, na.rm=TRUE),`iri 100-300`=mean(`iri 100-
300`, na.rm=TRUE),

                        `iri 300-500`=mean(`iri 300-500`, na.rm=TRUE),`Desiccated aral
seabed`=mean(`Desiccated aral seabed`, na.rm=TRUE),)

grouped_month$mean <- (grouped_month$`range 0-100`+grouped_month$`range 100-
300`+grouped_month$`range 300-500`+grouped_month$`iri 100-300`+grouped_month$`iri 300-
500`+grouped_month$`Desiccated aral seabed`)

grouped_month$mean <- grouped_month$mean/6


plot(grouped_month$dates,grouped_month$mean,type='l',col='dark
blue',lwd=3,xlab='Month',ylab='Mean monhtly precipitation (mm)',main='Aral Sea seasonal
precipitation trend')



preci_year$dates<- format(as.Date(preci$dates, "%b %d, %Y"), "%Y")


grouped_year <- group_by(preci_year,dates)
grouped_year <-  summarize(grouped_year,`range 0-100`=sum(`range 0-100`, na.rm=TRUE),`range
100-300`=sum(`range 100-300`, na.rm=TRUE),

                    `range 300-500`=sum(`range 300-500`, na.rm=TRUE),`iri 100-300`=sum(`iri 100-300`,
na.rm=TRUE),

                    `iri 300-500`=sum(`iri 300-500`, na.rm=TRUE),`Desiccated aral
seabed`=sum(`Desiccated aral seabed`, na.rm=TRUE),)


grouped_year$mean <- (grouped_year$`range 0-100`+grouped_year$`range 100-
300`+grouped_year$`range 300-500`+grouped_year$`iri 100-300`+grouped_year$`iri 300-
500`+grouped_year$`Desiccated aral seabed`)

grouped_year$mean <- grouped_year$mean/6



fit <- lm(dates~mean, data=grouped_year)

plot(grouped_year$dates,grouped_year$mean,type='l',col='dark
blue',lwd=3,xlab='Year',ylab='Annual precipitation (mm)',main='Aral Sea precipitation trend')

#lines(grouped_year$dates, grouped_year$dates*fit$coefficients[2]+fit$coefficients[2])


2020-1958

grouped_year_cut <- grouped_year[1:63,]

grouped_year_cut$dates <- seq(1958,2020,1)

(trendy <- ggplot(grouped_year_cut, aes(x = dates, y = mean,group = 1)) +

   geom_line() +

   geom_smooth(method = "lm", se = FALSE, span = 0.6) +

   theme_classic() +

   ggtitle('Aral Sea precipitation trend')+

   xlab('Year') +

   ylab('Annual precipitation (mm)')+

   theme(plot.title = element_text(hjust = 0.5)))
```

```
plot(grouped_year$dates,grouped_year$`range 0-100`,type='l',ylim=c(50,250),col='dark red',lwd=3)
lines(grouped_year$dates,grouped_year$`range 100-300`,col='gold',lwd=3)
lines(grouped_year$dates,grouped_year$`range 300-500`,col='black',lwd=3)
lines(grouped_year$dates,grouped_year$`iri 100-300`,col='dark green',lwd=3)
lines(grouped_year$dates,grouped_year$`iri 300-500`,col='black',lwd=3)
lines(grouped_year$dates,grouped_year$`Desiccated aral seabed`,lwd=3)

SPEI_total_pricip$`range 100-300`
total_pricip
plot(spei(ts(SPEI_total_pricip[,'range 100-300'], freq=12, start=c(1958,6)), 6),main='SPEI-6 for the Aral Sea and the 500 km buffer zone')
plot(spi(ts(total_pricip[,'range 100-300'], freq=12, start=c(1958,6)), 6),main='SPEI-6 for the Aral Sea and the 500 km buffer zone')


plot(range_spi_300,main='SPI-6 for the Aral Sea and the 500 km buffer zone',start=c(1900,1))
plot(SPEI_range_spi_300,main='SPEI-6 for the Aral Sea and the 500 km buffer zone')
```

5.2+5.3

```
getwd()
setwd('C:/Users/timon/Documents/scriptie/Data')


# activating packages if not already enabled
bool <- FALSE
if (bool){
  library(SPEI)
  library(csvread)
  library(imputeTS)
  library(forecast)
  library(tseries)
  library(ggplot2)
  library(dplyr)
  library(lubridate)
  library(basicTrendline)
  library(ISwR)
  library(ggplot2)
}


# combined wind data is read
all_wind_data <- read.csv("C:/Users/timon/Documents/scriptie/Data/Windspeeds.csv")
hourly_wind_data <- all_wind_data
hourly_wind_data$wdsp.chimbaj <- NULL
hourly_wind_data$mxwspd.kungrad <- NULL
hourly_wind_data$wdsp.kungrad <- NULL
hourly_wind_data$mxwdsp.chimbaj <- NULL

#8*30*4+18*8*365

# wind data is cleaned from no data
hourly_wind_clean <- hourly_wind_data
```

```
hourly_wind_clean$kungrad[hourly_wind_clean$kungrad >  999] <- 0

hourly_wind_clean <- hourly_wind_clean[1:53900,]

group_clean <- group_by(hourly_wind_clean,Date)

group_clean <- summarize(group_clean,kungrad=max(kungrad, na.rm=TRUE))



# probability of windspeed exceeding a treshold is examined

windspeed_vector <- seq(0,30,1)

prob_vector <- c()

tresholds_vector <- c()

for (speed in windspeed_vector){

Kungrad_treshold <- group_clean[ which(group_clean$kungrad>=speed), ]

Nomral_length <- length(group_clean$kungrad)

treshold_length <- length(Kungrad_treshold$kungrad)


propability_exceedence <- treshold_length/Nomral_length*100

prob_vector <- c(prob_vector,propability_exceedence)

tresholds_vector <- c(tresholds_vector,treshold_length)

}

table1 <- cbind(windspeed_vector,tresholds_vector)



# exceedence is poltted

plot(windspeed_vector,prob_vector,type = 'l',lwd =3,xlab= 'windspeed(m/s)',ylab='Exceedance probability(%)',main='Maximum 3-hour average windspeed')


#   threshold exceedence for different treshold is extracted

over_10 <-  group_clean[ which(group_clean$kungrad>=10), ]
```

```
over_15 <-  group_clean[ which(group_clean$kungrad>=15), ]
over_20 <-  group_clean[ which(group_clean$kungrad>=20), ]
over_25 <-  group_clean[ which(group_clean$kungrad>=25), ]


# exceedence of the treshold per year is measuered
year_seq <- seq(2001,2020,1)
number_storms <- rep(c(0),20)
table2 <- cbind(year_seq,number_storms)
table3 <- cbind(year_seq,number_storms)
table22<- cbind(year_seq,number_storms)
table33<- cbind(year_seq,number_storms)
for(year in year_seq){
  year_str <- toString(year)
  for(datum in over_10$Date){
    if(substr(datum, start = 1, stop = 4)==year_str){
      table2[(year-2000),2] <- table2[(year-2000),2]+1
    }
    }
}
for(year in year_seq){
  year_str <- toString(year)
  for(datum in over_15$Date){
    if(substr(datum, start = 1, stop = 4)==year_str){
      table3[(year-2000),2] <- table3[(year-2000),2]+1
    }
  }
}
for(year in year_seq){
  year_str <- toString(year)
  for(datum in over_20$Date){
    if(substr(datum, start = 1, stop = 4)==year_str){
      table22[(year-2000),2] <- table22[(year-2000),2]+1
    }
  }
}

for(year in year_seq){
  year_str <- toString(year)
  for(datum in over_25$Date){
    if(substr(datum, start = 1, stop = 4)==year_str){
      table33[(year-2000),2] <- table33[(year-2000),2]+1
    }
  }
}



plot(table2,type='l',xlab='year',ylab='Occurences',col='green',lwd =3)
lines(table3,col='blue',lwd =3)
lines(table22,col='red',lwd =3)
lines(table33,col='yellow',lwd =3)
legend(2015.60,69.6,legend=c('Windspeed=>10','Windspeed=>15','Windspeed=>20','Windspeed=>25'),col=c('green','blue','red','yellow')
       , lty=1, cex=1,lwd =3)
```

```
#######################################



# treshold exceedence per month is extracted
month_seq <- seq(1,12,1)
number_storms <- rep(c(0),12)
table4 <- cbind(month_seq,number_storms)
table5 <- cbind(month_seq,number_storms)
table44 <- cbind(month_seq,number_storms)
table55 <- cbind(month_seq,number_storms)

for(month in month_seq){
  if(month<10){
    month_str <- toString(month)
    month_str <- paste('0',month_str,sep='')

  }else{
    month_str <- toString(month)
  }
  for(datum in over_10$Date){
    if(substr(datum, start = 6, stop = 7)==month_str){
      table4[(month),2] <- table4[(month),2]+1
    }
  }
}
for(month in month_seq){
  if(month<10){
    month_str <- toString(month)
    month_str <- paste('0',month_str,sep='')

  }else{
    month_str <- toString(month)
  }
  for(datum in over_15$Date){
    if(substr(datum, start = 6, stop = 7)==month_str){
      table5[(month),2] <- table5[(month),2]+1
    }
  }
}

for(month in month_seq){
  if(month<10){
    month_str <- toString(month)
    month_str <- paste('0',month_str,sep='')

  }else{
    month_str <- toString(month)
  }
  for(datum in over_20$Date){
    if(substr(datum, start = 6, stop = 7)==month_str){
      table44[(month),2] <- table44[(month),2]+1
    }
  }
}
```

```
for(month in month_seq){
  if(month<10){
    month_str <- toString(month)
    month_str <- paste('0',month_str,sep='')

  }else{
    month_str <- toString(month)
  }
  for(datum in over_25$Date){
    if(substr(datum, start = 6, stop = 7)==month_str){
      table55[(month),2] <- table55[(month),2]+1
    }
  }
}

plot(table4,type='l',xlab='Month',lwd =3,ylab='Total occurences 2001-
2020',col='green',ylim=c(0,125))
lines(table5,col='blue',lwd =3)
lines(table44,col='red',lwd =3)
lines(table55,col='yellow',lwd =3)
legend(9.45,130,legend=c('Windspeed=>10','Windspeed=>15','Windspeed=>20','Windspeed=>25'),c
ol=c('green','blue','red','yellow')
       , lty=1, cex=1,lwd =3)

###########################################################
#0.55 ??m aerosol optical depth
total_for_dates <- read.csv('total_data.CSV')
setwd("C:/Users/timon/Documents/scriptie/0_dustdata")
```

```
#reading total dust data


total <-
csvread("total_data.CSV",header=TRUE,na.strings=".",coltypes=c("string","double","double","double
"))




# Setting dates in R


total$system.time_start <- total_for_dates$system.time_start


total$bad_Date <- format(total$system.time_start, format="%b %d, %Y")

head(total$bad_Date)

class(total$bad_Date)

### making the date better

total$Date <- as.Date(total$bad_Date,format="%b %d, %Y")

head(total$Date)

class(total$Date)


dust_group <- group_by(total,Date)

dust_dated <-  summarize(dust_group,dessert=max(as.numeric(dessert, na.rm=TRUE)),
old_aral_sea=max(as.numeric(old_aral_sea, na.rm=TRUE)),

                        river_area=max(as.numeric(river_area, na.rm=TRUE)))


dust_dated$river_area <- dust_dated$river_area^1


#creaing empty data sets to be filled with Aerosol exceedence
```

```
month_seq <- seq(1,12,1)
number_high_aerosols <- rep(c(0),12)
Mean_aerosols <-  rep(c(0),12)
Aero_11 <- cbind(month_seq,number_high_aerosols)
Aero_12 <- cbind(month_seq,number_high_aerosols)
Aero_13 <- cbind(month_seq,number_high_aerosols)
Aero_2 <- cbind(month_seq,Mean_aerosols)

#setting Aerosol treshold
dust_dated_100 <-  dust_dated[ which(dust_dated$old_aral_sea>=100), ]
dust_dated_200 <-  dust_dated[ which(dust_dated$old_aral_sea>=200), ]
dust_dated_300 <-  dust_dated[ which(dust_dated$old_aral_sea>=300), ]

# calculating exceedences

Date_vector <- seq(1,length(dust_dated_100$Date),1)
for(month in month_seq){
  if(month<10){
    month_str <- toString(month)
    month_str <- paste('0',month_str,sep='')

  }else{
    month_str <- toString(month)
  }
  for(datum in Date_vector){
    if(substr(dust_dated_100$Date[datum], start = 6, stop = 7)==month_str){
      Aero_11[(month),2] <- Aero_11[(month),2]+1
    }
  }
}
Date_vector <- seq(1,length(dust_dated_200$Date),1)
for(month in month_seq){
  if(month<10){
    month_str <- toString(month)
    month_str <- paste('0',month_str,sep='')


  }else{
    month_str <- toString(month)
  }
  for(datum in Date_vector){
    if(substr(dust_dated_200$Date[datum], start = 6, stop = 7)==month_str){
      Aero_12[(month),2] <- Aero_12[(month),2]+1
    }
  }
}
Date_vector <- seq(1,length(dust_dated_300$Date),1)
for(month in month_seq){
  if(month<10){
    month_str <- toString(month)
    month_str <- paste('0',month_str,sep='')


  }else{
    month_str <- toString(month)
  }
  for(datum in Date_vector){
    if(substr(dust_dated_300$Date[datum], start = 6, stop = 7)==month_str){
      Aero_13[(month),2] <- Aero_13[(month),2]+1
    }
  }
}
#########################
Date_vector <- seq(1,length(dust_dated$Date),1)
for(month in month_seq){
  if(month<10){
    month_str <- toString(month)
    month_str <- paste('0',month_str,sep='')

  }else{
    month_str <- toString(month)
  }
  for(datum in Date_vector){
    if(substr(dust_dated$Date[datum], start = 6, stop = 7)==month_str){
      Aero_2[(month),2] <- Aero_2[(month),2]+dust_dated$old_aral_sea[datum]
    }
  }
}

Aero_2 <- Aero_2/(30*20)
Aero_11 <- Aero_11/20
Aero_12 <- Aero_12/20
Aero_13 <- Aero_13/20
```

```
plot(month_seq,Aero_11[,2], col='yellow',lwd =3,main='Nr of days Per month above  0.55 ??m Aerosol optical depth',xlab= 'Month',ylab='Nr of days per month above aerosol optical depth',ylim=c(0,30),type='l')

lines(month_seq,Aero_12[,2],col='Brown',lwd =3)

lines(month_seq,Aero_13[,2], col='Black',lwd =3)

legend(0.55,31.2,legend=c('100 ','200 ','300 '),col=c('yellow','brown','black'),lty=1, cex=1,lwd =3)



plot(month_seq,Aero_2[,2], xlab= 'Month',ylab='Aerosol optical depth',main='Average daily aerosols per month')


##########################################################


#creating empty datasets

year_seq <- seq(2001,2020,1)

number_high_aerosols <- rep(c(0),20)

Aero_21 <- cbind(year_seq,number_high_aerosols)

Aero_22 <- cbind(year_seq,number_high_aerosols)

Aero_23 <- cbind(year_seq,number_high_aerosols)


# calculating exceedences

Date_vector <- seq(1,length(dust_dated_100$Date),1)

for(year in year_seq){

  year_str <- toString(year)


  for(datum in Date_vector){

    if(substr(dust_dated_100$Date[datum], start = 1, stop = 4)==year_str){

      Aero_21[(year-2000),2] <- Aero_21[(year-2000),2]+1
    }
  }
}
Date_vector <- seq(1,length(dust_dated_200$Date),1)
for(year in year_seq){
  year_str <- toString(year)


  for(datum in Date_vector){
    if(substr(dust_dated_200$Date[datum], start = 1, stop = 4)==year_str){
      Aero_22[(year-2000),2] <- Aero_22[(year-2000),2]+1
    }
  }
}
Date_vector <- seq(1,length(dust_dated_300$Date),1)
for(year in year_seq){
  year_str <- toString(year)


  for(datum in Date_vector){
    if(substr(dust_dated_300$Date[datum], start = 1, stop = 4)==year_str){
      Aero_23[(year-2000),2] <- Aero_23[(year-2000),2]+1
    }
  }
}



plot(year_seq,Aero_21[,2], col='yellow',lwd =3,xlab= 'Year',main='Nr of days Per year above  0.55
??m Aerosol optical depth',ylab='Nr of days per year above Aerosol optical
depth',ylim=c(0,365),type='l')
```

```
lines(year_seq,Aero_22[,2],col='Brown',lwd =3)

lines(year_seq,Aero_23[,2], col='Black',lwd =3)

legend(2017.99,380,legend=c('100 ','200 ','300 '),col=c('yellow','brown','black'),lty=1, cex=1,lwd =3)



typeof(dust_dated$Date[1])



group_clean$bad_Date <- format(group_clean$Date, format="%Y-%m-%d")

head(group_clean$bad_Date)

class(group_clean$bad_Date)

### making the date better

group_clean$Date <- as.Date(group_clean$bad_Date,format="%Y-%m-%d")

head(group_clean$Date)

class(group_clean$Date)

group_clean$bad_Date <- NULL

typeof(group_clean$Date[1])



#removing NA data


wind_dust <- merge(x= group_clean, y= dust_dated, by= 'Date', all.x= T)


wind_dust$old_aral_sea <- na_interpolation(wind_dust$old_aral_sea, option = "linear")
```

```
wind_dust$dessert <- na_interpolation(wind_dust$dessert, option = "linear")

wind_dust$river_area <- na_interpolation(wind_dust$river_area, option = "linear")

wind_dust$kungrad <- na_interpolation(wind_dust$kungrad, option = "linear")



wind_dust_month <- wind_dust

wind_dust_month$Date <- format(as.Date(wind_dust_month$Date,"%Y-%m-%d"), "%m")



#calculating Correlations


first_wind_dust1 <-
wind_dust_month[which(wind_dust_month$kungrad>0&wind_dust_month$old_aral_sea>50),]

first_wind_dust2 <-
wind_dust_month[which(wind_dust_month$kungrad>0&wind_dust_month$old_aral_sea>100),]

first_wind_dust3 <-
wind_dust_month[which(wind_dust_month$kungrad>0&wind_dust_month$old_aral_sea>150),]

first_wind_dust4 <-
wind_dust_month[which(wind_dust_month$kungrad>0&wind_dust_month$old_aral_sea>200),]

cor_vector1 <- c()

cor_vector2 <- c()

cor_vector3 <- c()

cor_vector4 <- c()

for(month in month_seq){

 if(month<10){

   month_str <- toString(month)

   month_str <- paste('0',month_str,sep='')


 }else{
   month_str <- toString(month)
 }
 filterr1 <- first_wind_dust1[which(first_wind_dust1$Date==month_str),]
 filterr2 <- first_wind_dust2[which(first_wind_dust2$Date==month_str),]
 filterr3 <- first_wind_dust3[which(first_wind_dust3$Date==month_str),]
 filterr4 <- first_wind_dust4[which(first_wind_dust4$Date==month_str),]
 corre1 <- cor(filterr1$kungrad,filterr1$old_aral_sea)
 corre2 <- cor(filterr2$kungrad,filterr2$old_aral_sea)
 corre3 <- cor(filterr3$kungrad,filterr3$old_aral_sea)
 corre4 <- cor(filterr4$kungrad,filterr4$old_aral_sea)
 cor_vector1 <- c(cor_vector1,corre1)
 cor_vector2 <- c(cor_vector2,corre2)
 cor_vector3 <- c(cor_vector3,corre3)
 cor_vector4 <- c(cor_vector4,corre4)
}
plot(month_seq,cor_vector1,col='blue',lwd =3,ylab='Corelation',xlab='Month',type='l',ylim=c(-0.1,0.4),main='Correlation 0.55 ??m Aerosol optical depth and winspeed   ')
lines(month_seq,cor_vector2,col='green',lwd =3)
lines(month_seq,cor_vector3,col='yellow',lwd =3)
lines(month_seq,cor_vector4,col='red',lwd =3)
legend(0.55,0.42,legend=c('50 ','100 ','150 ','200'),col=c('blue','green','yellow','red'),lty=1,lwd =3, cex=1)
```

5.4

```
Grassy <- 0.5

#The_model is the main funciton which calculates total erosion in g/m/s
```

```
The_model <-
function(mean_grainsize=0.1,tree_quantity=0,tree_drag=0.3,tree_breidth=2.5,tree_porosity=0.7,tree_height=3,


wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=15,

                    drag_coeff = 0.0033,type_of_model='kawamura',  grass_drag=0.1,
grass_porosity=0,grass_height=0.15,

                    Test_area=10^2){

  #Empty strings are created for each type of sediment flux calculation. They will be filled with data

  erosion <- c()

  erosion_1 <- c()

  erosion_2 <- c()

  erosion_3 <- c()

  erosion_4 <- c()

  erosion_5 <- c()


  # If the model is ran for a string of velocities the model will be run multiple times in this loop

  # for CA modelling done with the scenarios Wind_vel_vector is a vector of size 1

  for( wind_vel in wind_vel_vector){


    if (wind_vel<1){

      wind_vel <- 1}



    ##Atmospheric conditions are set

    R <- 287.058/1000

    air_dens <- 101.325 /(R*(273.15+Temp))
```

```
# Height of wind velocity is set
Height_z <- 100
## Soil specifications are set
Area <- Test_area
grain_dens <- 2650

##vegetation Densities and frontal area are calculated

#trees
tree_frontalarea <- (tree_breidth/2)^2*pi
tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area

#shrubs
shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity
shrub_density <- Shrub_quantity*shrub_frontalarea/Area

#grasses

# grasses are added when there is a grass porosity higher than 0
if(grass_porosity>0){
  grass_quantity <- 100*1000 # means 1000 per square meter
}else{
  grass_quantity <- 0

}
grass_width <- 0.01
grass_frontalarea <- grass_height*grass_width*grass_porosity
```

```
  grass_density <- grass_quantity*grass_frontalarea/Area


  # average vegetation specifics
  vegetation_density <- grass_density+shrub_density+tree_density
  if((grass_density+shrub_density+tree_density)==0){
    average_veg_height <- 0
  }else{
    average_veg_height <-
(grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density
+shrub_density+tree_density)
  }


  #treshold velocity is calculated
  mean_grainsize_adapt <- mean_grainsize*10^-4.8
  treshhold_vel <- 7*sqrt(0.0123*(grain_dens/air_dens*9.81*mean_grainsize_adapt+(3*(10^-
4)/(mean_grainsize_adapt*grain_dens))))



  #constants
  Karman <- 0.4
  grav <- 9.81
  Ak <- 0.1
  ref_grainsize <- 0.25


  #wind profile and friction velocity are calculated
  z0 <- 0.001+(0.5*vegetation_density*average_veg_height)
  friction_vel <- (wind_vel*Karman)/(log(Height_z/(z0)))
  Heights <- seq(1,100,1)
  vel_per_height <- (friction_vel/Karman*(log(Heights)/z0))
  ##Drag is partitioned

  tot_drag <- (friction_vel^2)*air_dens

  if((tree_quantity+grass_quantity+Shrub_quantity)==0){

    veg_drag <- drag_coeff^2

  }else{veg_drag <- (grass_drag*grass_density+
shrub_drag*shrub_density+tree_drag*tree_density)/(vegetation_density)}


  drag_ratio <- veg_drag/drag_coeff

  drag_on_veg <- (drag_ratio*vegetation_density*tot_drag)/(1+drag_ratio*vegetation_density)

  drag_on_soil <- tot_drag/(1+drag_ratio*vegetation_density)


  ## Friction velcoity on surface is calculated


  grainsize_dependent_Fd <- 1.41+4.98*exp((-0.5)*(log((mean_grainsize/(1.55*ref_grainsize))^2)))

  treshhold_vel <- treshhold_vel

  friction_vel_on_surface <- sqrt(drag_on_soil/air_dens)

  ## flux in g/m/s


  if((1-treshhold_vel/friction_vel_on_surface)<0){

    surplus_erosie <- 0

  }else{

    surplus_erosie <- 1-(treshhold_vel/friction_vel_on_surface)

  }


  # constants for different sediment flux formulas are calculated

  constant_Z <- 0.5

  constant_Ka <- 0.14
```

```
   constant_M <- 0.5

   constant_Kl <-0.5

   Constant_L <- 0.9


   #sediment fluxes are calculated using different formulas

   sediment_flux <-
10*grainsize_dependent_Fd*(surplus_erosie^1)*(air_dens/grav)*(friction_vel_on_surface^3)

   sediment_flux_1 <-
1000*constant_Z*((mean_grainsize/ref_grainsize)^0.75)*air_dens*(friction_vel_on_surface^3)/9.81

   sediment_flux_2 <- 1000*constant_Ka*(1-
treshhold_vel/friction_vel_on_surface)*((1+treshhold_vel/friction_vel_on_surface)^2)*air_dens*(fri
ction_vel_on_surface^3)/9.81

   sediment_flux_3 <-
1000*(air_dens*(friction_vel_on_surface^3)/9.81)*constant_M*((mean_grainsize/ref_grainsize)^0.7
5)*(1-(treshhold_vel/friction_vel_on_surface)^13.72)

   sediment_flux_4 <- 1000*constant_Kl*(1-
(treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81

   sediment_flux_5 <- 1000*Constant_L
*((mean_grainsize/ref_grainsize)^0.5)*(air_dens*(friction_vel_on_surface^3)/9.81)*(friction_vel_on
_surface-treshhold_vel)




   erosion <- c(erosion,sediment_flux)

   erosion_1 <- c(erosion_1,sediment_flux_1)

   erosion_2 <- c(erosion_2,sediment_flux_2)

   erosion_3 <- c(erosion_3,sediment_flux_3)

   erosion_4 <- c(erosion_4,sediment_flux_4)

   erosion_5 <- c(erosion_5,sediment_flux_5)
   #some formulas give negative values when wind velocity is low. If this is the case the erosion is set
to 0

   erosion[erosion<0] <- 0

   erosion_1[erosion_1<0] <- 0

   erosion_2[erosion_2<0] <- 0

   erosion_3[erosion_3<0] <- 0

   erosion_4[erosion_4<0] <- 0

   erosion_5[erosion_5<0] <- 0


   # factor for CA_modelling is added

   erosion <- erosion/5

   erosion_1 <- erosion_1/5

   erosion_2 <- erosion_2/5

   erosion_3 <- erosion_3/5

   erosion_4 <- erosion_4/5

   erosion_5 <- erosion_5/5





   erosions <- cbind(erosion,erosion_1,erosion_2,erosion_3,erosion_4,erosion_5)

  }


  # The type of erosion used in the model is returned

  if(type_of_model=='old'){

   return(erosion)}

  else if(type_of_model=='zingg'){

   return(erosion_1)}
  else if(type_of_model=='kawamura'){
    return(erosion_2)}
  else if(type_of_model=='maegley'){
    return(erosion_3)}
  else if(type_of_model=='kind'){
    return(erosion_4)}
  else if(type_of_model=='lettau'){
    return(erosion_5)}
  else if(type_of_model=='all'){
    return(erosions)}


}



dustfunc <-
function(production=0,alpha=1,mean_grainsize=0.02,tree_quantity=0,tree_drag=0.3,tree_breidth=2.
5,tree_porosity=0.7,tree_height=3,


wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=1
5,
                  drag_coeff = 0.0033,type_of_model='kawamura',  grass_drag=0.1,
grass_porosity=0,grass_height=0.15,
                  Test_area=10^2){


  #empty datasets are created
  erosion <- c()
  erosion_1 <- c()
  erosion_2 <- c()
  erosion_3 <- c()
```

```
erosion_4 <- c()

erosion_5 <- c()

# If there is a string of windvelocity it is looped here

for( wind_vel in wind_vel_vector){

 if (wind_vel<1){

   wind_vel <- 1

 }


 ##Atmospheric conditions

 R <- 287.058/1000

 air_dens <- 101.325 /(R*(273.15+Temp))

 Height_z <- 100

 Area <- Test_area

 grain_dens <- 2650


 ##vegetation densities and frontal areas are calculated


 #trees

 tree_frontalarea <- (tree_breidth/2)^2*pi

 tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area


 #shrubs

 shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity

 shrub_density <- Shrub_quantity*shrub_frontalarea/Area


 #grasses

 # if grass porosity is higher than 0 grass is added
```

```
  if(grass_porosity>0){
    grass_quantity <- 100*1000 # means 1000 per square meter
  }else{
    grass_quantity <- 0
  }

  grass_width <- 0.01
  grass_frontalarea <- grass_height*grass_width*grass_porosity
  grass_density <- grass_quantity*grass_frontalarea/Area


  # vegetation averaged density and height is calculated
  vegetation_density <- grass_density+shrub_density+tree_density
  if((grass_density+shrub_density+tree_density)==0){
    average_veg_height <- 0
  }else{
    average_veg_height <-
(grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density
+shrub_density+tree_density)
  }

  # treshold velocirty is calculated
  mean_grainsize_adapt <- mean_grainsize*8*10^-5.5
  treshhold_vel <- 7*sqrt(0.0123*(grain_dens/air_dens*9.81*mean_grainsize_adapt+(3*(10^-
4)/(mean_grainsize_adapt*grain_dens))))

  #constants
  Karman <- 0.4
  grav <- 9.81
  Ak <- 0.1

  ref_grainsize <- 0.25



  #wind profile and friction velcoty are calulated

  z0 <- 0.001+(0.5*vegetation_density*average_veg_height)

  friction_vel <- (wind_vel*Karman)/(log(Height_z/(z0)))



  Heights <- seq(1,100,1)

  vel_per_height <- (friction_vel/Karman*(log(Heights)/z0))



  ##Drag is partitioned

  tot_drag <- (friction_vel^2)*air_dens

  if((tree_quantity+grass_quantity+Shrub_quantity)==0){

    veg_drag <- drag_coeff^2

  }else{veg_drag <- (grass_drag*grass_density+
shrub_drag*shrub_density+tree_drag*tree_density)/(vegetation_density)}


  drag_ratio <- veg_drag/drag_coeff

  drag_on_veg <- (drag_ratio*vegetation_density*tot_drag)/(1+drag_ratio*vegetation_density)

  drag_on_soil <- tot_drag/(1+drag_ratio*vegetation_density)


  #friction on surface is calculated


  grainsize_dependent_Fd <- 1.41+4.98*exp((-0.5)*(log((mean_grainsize/(1.55*ref_grainsize))^2)))

  treshhold_vel <- treshhold_vel
  friction_vel_on_surface <- sqrt(drag_on_soil/air_dens)

  ## flux in g/m/s


  if((1-treshhold_vel/friction_vel_on_surface)<0){

    surplus_erosie <- 0

  }else{

    surplus_erosie <- 1-(treshhold_vel/friction_vel_on_surface)

  }


  #constants for different formulas are given


  constant_Z <- 0.5

  constant_Ka <- 0.14

  constant_M <- 0.5

  constant_Kl <-0.5

  Constant_L <- 0.9


  # Sediment fluxes are calculated for different forumlas



  sediment_flux <- 10*grainsize_dependent_Fd*(surplus_erosie^1)*(air_dens/grav)*(friction_vel_on_surface^3)

  sediment_flux_1 <- 1000*constant_Z*((mean_grainsize/ref_grainsize)^0.75)*air_dens*(friction_vel_on_surface^3)/9.81

  sediment_flux_2 <- 1000*constant_Ka*(1-treshhold_vel/friction_vel_on_surface)*((1+treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81

  sediment_flux_3 <- 1000*(air_dens*(friction_vel_on_surface^3)/9.81)*constant_M*((mean_grainsize/ref_grainsize)^0.75)*(1-(treshhold_vel/friction_vel_on_surface)^13.72)
  sediment_flux_4 <- 1000*constant_KI*(1-(treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81
  sediment_flux_5 <- 1000*Constant_L *((mean_grainsize/ref_grainsize)^0.5)*(air_dens*(friction_vel_on_surface^3)/9.81)*(friction_vel_on_surface-treshhold_vel)



  if((friction_vel-treshhold_vel)<0){
    sediment_flux <- 0
    sediment_flux_1 <- 0
    sediment_flux_2 <- 0
    sediment_flux_3 <- 0
    sediment_flux_4 <- 0
    sediment_flux_5 <- 0
  }else{


    #sediment brought into suspension is calculated

    alpha <- alpha*10
    sediment_flux <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_1 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_2 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_3 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_4 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_5 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
  }

  erosion <- c(erosion,sediment_flux)
  erosion_1 <- c(erosion_1,sediment_flux_1)

  erosion_2 <- c(erosion_2,sediment_flux_2)

  erosion_3 <- c(erosion_3,sediment_flux_3)

  erosion_4 <- c(erosion_4,sediment_flux_4)

  erosion_5 <- c(erosion_5,sediment_flux_5)


  # If a sediment flux is lower than 0 it is set to 0


  erosion[erosion<0] <- 0

  erosion_1[erosion_1<0] <- 0

  erosion_2[erosion_2<0] <- 0

  erosion_3[erosion_3<0] <- 0

  erosion_4[erosion_4<0] <- 0

  erosion_5[erosion_5<0] <- 0



  erosions <- cbind(erosion,erosion_1,erosion_2,erosion_3,erosion_4,erosion_5)

}


#erosion is returned based on type of formula used


if(type_of_model=='old'){

  return(erosion)}

else if(type_of_model=='zingg'){

  return(erosion_1)}

else if(type_of_model=='kawamura'){

  return(erosion_2)}
  else if(type_of_model=='maegley'){
    return(erosion_3)}
  else if(type_of_model=='kind'){
    return(erosion_4)}
  else if(type_of_model=='lettau'){
    return(erosion_5)}
  else if(type_of_model=='all'){
    return(erosions)}


}




veg_density_func <- function(mean_grainsize=0.02,tree_quantity=0,tree_drag=0.3,tree_breidth=2.5,tree_porosity=0.7,tree_height=3,

wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=15,
                             drag_coeff = 0.0033,type_of_model='kawamura',  grass_drag=0.1, grass_porosity=0,grass_height=0.15,
                             Test_area=10^2){



  ##Atmospheric conditions
  R <- 287.058/1000
  air_dens <- 101.325 /(R*(273.15+Temp))
  Height_z <- 100
  ## Soil specification are given
```

```
Area <- Test_area

bulk_density <- 1.5#g/cm3

grain_dens <- 2650


##vegetation specifications are calculated

#trees

tree_frontalarea <- (tree_breidth/2)^2*pi

tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area




#shrubs

shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity

shrub_density <- Shrub_quantity*shrub_frontalarea/Area


#grasses

# if grass porosity is higher than 0 grass is added

if(grass_porosity>0){

  grass_quantity <- 100*1000 # means 1000 per square meter

}else{

  grass_quantity <- 0


}

grass_width <- 0.01

grass_frontalarea <- grass_height*grass_width*grass_porosity

grass_density <- grass_quantity*grass_frontalarea/Area


# A verage vegetation density and heigth is calculated
```

```
  vegetation_density <- grass_density+shrub_density+tree_density
  if((grass_density+shrub_density+tree_density)==0){
    average_veg_height <- 0
  }else{
    average_veg_height <-
(grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density
+shrub_density+tree_density)
  }
  # vegtation density is set to a max of 0.8, due to model workings this is necessary.
  vegetation_density <- vegetation_density
  if(vegetation_density>0.8){
    vegetation_density <- 0.8
  }

  return(vegetation_density)
}



# vector of windspeed is made
scaling_vector <- seq(0,25,0.1)
scaling_vector <- rep(10,1)
#setting the wind vector used
wind_vel_vector <- scaling_vector



####################################


# Ca model is run for the different scenarios and a single high wind event.
```

```
#The Dust/Suspension model is then run for the 6 scenarios and a single high wind event.

#results are saved in a variable and later plotted


##################################

# vegetation grids are set for different scenarios.

# set vegetation grids how you wouuld like

# 100 cells per vector are required

# using the rep function these lengths are created

outflux_vector <- c()

wind_vel_vector <- 10

grass_porosity_grid <- rep(c(0,0,0.0,0,0),20)

shrub_quantitygrid <- rep(c(0),100)

shrub_porositygrid <- rep(c(0.5),100)

shrub_heightgrid <- rep(c(1),100)

tree_quantitygrid <- rep(c(0),100)

tree_breidthgrid <- rep(c(2.5),100)

tree_porositygrid <- rep(c(0.7),100)

tree_heightgrid<- rep(c(3),100)


# timestep is set to 1

Time_step <- 1

# an empty grid for the model(erosiongird) is made, (this grid is one larger than it needs to be becouse it needs to add from a previous time step which must exist)

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)

# model is run for wind velocity vector

for (wind_vel in wind_vel_vector) {

  # a time-step in the grid is taken
```

```
 Time_step <- Time_step+1
 # The lateral placement is set to 1
 count <- length(grass_porosity_grid)


 # loop to calculate erosion for different cells in the model
 for (grass_porosity in grass_porosity_grid){
  #generation of erosion is calculated for this cell
  production <-
The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygr
id[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
  # Vegetation density for intertception of erosion is calculated
  vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
  #transport is calculated
  transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
  #erosion grid is updated
  erosiongrid[Time_step,count] <- production+transport
  #Lateral step in the gird is taken
  count <- count-1
 }
 #Outflux out of the model is calculated
 outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
```

```
  #Outfluxvector is updated
  outflux_vector <- c(outflux_vector,outflux)
}
#Outcome for scenario 1 is set
Scenario_test_1 <- outflux_vector



windvel=wind_vel_vector
count <- 1
kersten1 <-
The_model(grass_porosity=grass_porosity_grid[1],wind_vel_vector=wind_vel,tree_quantity=tree_qu
antitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
# vegetation grids are set for different scenarios.
# set vegetation grids how you wouuld like
# 100 cells per vector are required
# using the rep function these lengths are created
outflux_vector <- c()
wind_vel_vector <- 15



# timestep is set to 1
Time_step <- 1
# an empty grid for the model(erosiongird) is made, (this grid is one larger than it needs to be
becouse it needs to add from a previous time step which must exist)
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)
```

```
# model is run for wind velocity vector
for (wind_vel in wind_vel_vector) {
  # a time-step in the grid is taken
  Time_step <- Time_step+1
  # The lateral placement is set to 1
  count <- length(grass_porosity_grid)


  # loop to calculate erosion for different cells in the model
  for (grass_porosity in grass_porosity_grid){
    #generation of erosion is calculated for this cell
    production <- The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    # Vegetation density for intertception of erosion is calculated
    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    #transport is calculated
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    #erosion grid is updated
    erosiongrid[Time_step,count] <- production+transport
    #Lateral step in the gird is taken
    count <- count-1
```

```
 }

 #Outflux out of the model is calculated

 outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

 #Outfluxvector is updated

 outflux_vector <- c(outflux_vector,outflux)

}

#Outcome for scenario 1 is set

Scenario_test_2 <- outflux_vector




windvel=wind_vel_vector

count <- 1

kersten2 <- The_model(grass_porosity=grass_porosity_grid[1],wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

# vegetation grids are set for different scenarios.

# set vegetation grids how you wouuld like

# 100 cells per vector are required

# using the rep function these lengths are created

outflux_vector <- c()

wind_vel_vector <- 20




# timestep is set to 1

Time_step <- 1
```

```
# an empty grid for the model(erosiongird) is made, (this grid is one larger than it needs to be
becouse it needs to add from a previous time step which must exist)

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

# model is run for wind velocity vector

for (wind_vel in wind_vel_vector) {

 # a time-step in the grid is taken

 Time_step <- Time_step+1

 # The lateral placement is set to 1

 count <- length(grass_porosity_grid)


 # loop to calculate erosion for different cells in the model

 for (grass_porosity in grass_porosity_grid){

  #generation of erosion is calculated for this cell

  production <-
The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygr
id[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

  # Vegetation density for intertception of erosion is calculated

  vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

  #transport is calculated

  transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

  #erosion grid is updated
```

```
    erosiongrid[Time_step,count] <- production+transport

    #Lateral step in the gird is taken

    count <- count-1

   }

  #Outflux out of the model is calculated

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  #Outfluxvector is updated

  outflux_vector <- c(outflux_vector,outflux)

}

#Outcome for scenario 1 is set

Scenario_test_3 <- outflux_vector




windvel=wind_vel_vector

count <- 1

kersten3 <- The_model(grass_porosity=grass_porosity_grid[1],wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])




kersten <- cbind(kersten1,kersten2,kersten3)*4

Model <- cbind(Scenario_test_1,Scenario_test_2,Scenario_test_3)

table1 <- rbind(kersten,Model)


barplot(height=table1,width=1,main="Wind erosion per velocity",
    xlab="Velocity",beside=TRUE,ylab='Erosion (g/m/s.)',
    names.arg=c('10','15','20'),legend = c('Kersten','CA'))
```

kersten/Model

```
Grassy <- 0.15

#The_model is the main funciton which calculates total erosion in g/m/s
The_model <- function(mean_grainsize=0.02,tree_quantity=0,tree_drag=0.3,tree_breidth=2.5,tree_porosity=0.7,tree_height=3,

wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=15,
            drag_coeff = 0.0033,type_of_model='kawamura',  grass_drag=0.1, grass_porosity=0,grass_height=0.15,
            Test_area=10^2){
  #Empty strings are created for each type of sediment flux calculation. They will be filled with data
  erosion <- c()
  erosion_1 <- c()
  erosion_2 <- c()
  erosion_3 <- c()
  erosion_4 <- c()
  erosion_5 <- c()

  # If the model is ran for a string of velocities the model will be run multiple times in this loop
# for CA modelling done with the scenarios Wind_vel_vector is a vector of size 1
for( wind_vel in wind_vel_vector){


 if (wind_vel<1){
   wind_vel <- 1}




 ##Atmospheric conditions are set
 R <- 287.058/1000
 air_dens <- 101.325 /(R*(273.15+Temp))


 # Height of wind velocity is set
 Height_z <- 100
 ## Soil specifications are set
 Area <- Test_area
 grain_dens <- 2650


 ##vegetation Densities and frontal area are calculated


 #trees
 tree_frontalarea <- (tree_breidth/2)^2*pi
 tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area


 #shrubs
 shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity
 shrub_density <- Shrub_quantity*shrub_frontalarea/Area
```

```
  #grasses


  # grasses are added when there is a grass porosity higher than 0
  if(grass_porosity>0){
    grass_quantity <- 100*1000 # means 1000 per square meter
  }else{
    grass_quantity <- 0


  }
  grass_width <- 0.01
  grass_frontalarea <- grass_height*grass_width*grass_porosity
  grass_density <- grass_quantity*grass_frontalarea/Area


  # average vegetation specifics
  vegetation_density <- grass_density+shrub_density+tree_density
  if((grass_density+shrub_density+tree_density)==0){
    average_veg_height <- 0
  }else{
    average_veg_height <-
(grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density
+shrub_density+tree_density)
  }


  #treshold velocity is calculated
  mean_grainsize_adapt <- mean_grainsize*10^-4.8
  treshhold_vel <- 7*sqrt(0.0123*(grain_dens/air_dens*9.81*mean_grainsize_adapt+(3*(10^-
4)/(mean_grainsize_adapt*grain_dens))))
```

```
#constants
Karman <- 0.4
grav <- 9.81
Ak <- 0.1
ref_grainsize <- 0.25

#wind profile and friction velocity are calculated
z0 <- 0.001+(0.5*vegetation_density*average_veg_height)
friction_vel <- (wind_vel*Karman)/(log(Height_z/(z0)))
Heights <- seq(1,100,1)
vel_per_height <- (friction_vel/Karman*(log(Heights)/z0))

##Drag is partitioned
tot_drag <- (friction_vel^2)*air_dens
if((tree_quantity+grass_quantity+Shrub_quantity)==0){
  veg_drag <- drag_coeff^2
}else{veg_drag <- (grass_drag*grass_density+
shrub_drag*shrub_density+tree_drag*tree_density)/(vegetation_density)}

drag_ratio <- veg_drag/drag_coeff
drag_on_veg <- (drag_ratio*vegetation_density*tot_drag)/(1+drag_ratio*vegetation_density)
drag_on_soil <- tot_drag/(1+drag_ratio*vegetation_density)

## Friction velcoity on surface is calculated

grainsize_dependent_Fd <- 1.41+4.98*exp((-0.5)*(log((mean_grainsize/(1.55*ref_grainsize))^2)))
treshhold_vel <- treshhold_vel
friction_vel_on_surface <- sqrt(drag_on_soil/air_dens)
```

```
  ## flux in g/m/s


  if((1-treshhold_vel/friction_vel_on_surface)<0){
    surplus_erosie <- 0
  }else{
    surplus_erosie <- 1-(treshhold_vel/friction_vel_on_surface)
  }


  # constants for different sediment flux formulas are calculated
  constant_Z <- 0.5
  constant_Ka <- 0.14
  constant_M <- 0.5
  constant_KI <-0.5
  Constant_L <- 0.9


  #sediment fluxes are calculated using different formulas
  sediment_flux <-
10*grainsize_dependent_Fd*(surplus_erosie^1)*(air_dens/grav)*(friction_vel_on_surface^3)
  sediment_flux_1 <-
1000*constant_Z*((mean_grainsize/ref_grainsize)^0.75)*air_dens*(friction_vel_on_surface^3)/9.81
  sediment_flux_2 <- 1000*constant_Ka*(1-
treshhold_vel/friction_vel_on_surface)*((1+treshhold_vel/friction_vel_on_surface)^2)*air_dens*(fri
ction_vel_on_surface^3)/9.81
  sediment_flux_3 <-
1000*(air_dens*(friction_vel_on_surface^3)/9.81)*constant_M*((mean_grainsize/ref_grainsize)^0.7
5)*(1-(treshhold_vel/friction_vel_on_surface)^13.72)
  sediment_flux_4 <- 1000*constant_KI*(1-
(treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81
  sediment_flux_5 <- 1000*Constant_L
*((mean_grainsize/ref_grainsize)^0.5)*(air_dens*(friction_vel_on_surface^3)/9.81)*(friction_vel_on
_surface-treshhold_vel)
```

```
   erosion <- c(erosion,sediment_flux)
   erosion_1 <- c(erosion_1,sediment_flux_1)
   erosion_2 <- c(erosion_2,sediment_flux_2)
   erosion_3 <- c(erosion_3,sediment_flux_3)
   erosion_4 <- c(erosion_4,sediment_flux_4)
   erosion_5 <- c(erosion_5,sediment_flux_5)

   #some formulas give negative values when wind velocity is low. If this is the case the erosion is set to 0
   erosion[erosion<0] <- 0
   erosion_1[erosion_1<0] <- 0
   erosion_2[erosion_2<0] <- 0
   erosion_3[erosion_3<0] <- 0
   erosion_4[erosion_4<0] <- 0
   erosion_5[erosion_5<0] <- 0

   # factor for CA_modelling is added
   erosion <- erosion/5
   erosion_1 <- erosion_1/5
   erosion_2 <- erosion_2/5
   erosion_3 <- erosion_3/5
   erosion_4 <- erosion_4/5
   erosion_5 <- erosion_5/5
```

```
  erosions <- cbind(erosion,erosion_1,erosion_2,erosion_3,erosion_4,erosion_5)
 }

 # The type of erosion used in the model is returned
 if(type_of_model=='old'){
  return(erosion)}
 else if(type_of_model=='zingg'){
  return(erosion_1)}
 else if(type_of_model=='kawamura'){
  return(erosion_2)}
 else if(type_of_model=='maegley'){
  return(erosion_3)}
 else if(type_of_model=='kind'){
  return(erosion_4)}
 else if(type_of_model=='lettau'){
  return(erosion_5)}
 else if(type_of_model=='all'){
  return(erosions)}

}



dustfunc <-
function(production=0,alpha=1,mean_grainsize=0.02,tree_quantity=0,tree_drag=0.3,tree_breidth=2.
5,tree_porosity=0.7,tree_height=3,
wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=15,
          drag_coeff = 0.0033,type_of_model='kawamura',  grass_drag=0.1, grass_porosity=0,grass_height=0.15,
          Test_area=10^2){


 #empty datasets are created
 erosion <- c()
 erosion_1 <- c()
 erosion_2 <- c()
 erosion_3 <- c()
 erosion_4 <- c()
 erosion_5 <- c()
 # If there is a string of windvelocity it is looped here
 for( wind_vel in wind_vel_vector){
  if (wind_vel<1){
    wind_vel <- 1
  }

  ##Atmospheric conditions
  R <- 287.058/1000
  air_dens <- 101.325 /(R*(273.15+Temp))
  Height_z <- 100
  Area <- Test_area
  grain_dens <- 2650

  ##vegetation densities and frontal areas are calculated
```

```
#trees
tree_frontalarea <- (tree_breidth/2)^2*pi
tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area

#shrubs
shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity
shrub_density <- Shrub_quantity*shrub_frontalarea/Area

#grasses
# if grass porosity is higher than 0 grass is added
if(grass_porosity>0){
  grass_quantity <- 100*1000 # means 1000 per square meter
}else{
  grass_quantity <- 0
}

grass_width <- 0.01
grass_frontalarea <- grass_height*grass_width*grass_porosity
grass_density <- grass_quantity*grass_frontalarea/Area


# vegetation averaged density and height is calculated
vegetation_density <- grass_density+shrub_density+tree_density
if((grass_density+shrub_density+tree_density)==0){
  average_veg_height <- 0
}else{
      average_veg_height <-
(grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density
+shrub_density+tree_density)
    }


    # treshold velocirty is calculated
    mean_grainsize_adapt <- mean_grainsize*8*10^-5.5
    treshhold_vel <- 7*sqrt(0.0123*(grain_dens/air_dens*9.81*mean_grainsize_adapt+(3*(10^-
4)/(mean_grainsize_adapt*grain_dens))))


    #constants
    Karman <- 0.4
    grav <- 9.81
    Ak <- 0.1
    ref_grainsize <- 0.25



    #wind profile and friction velcoty are calulated
    z0 <- 0.001+(0.5*vegetation_density*average_veg_height)
    friction_vel <- (wind_vel*Karman)/(log(Height_z/(z0)))



    Heights <- seq(1,100,1)
    vel_per_height <- (friction_vel/Karman*(log(Heights)/z0))



    ##Drag is partitioned
    tot_drag <- (friction_vel^2)*air_dens
    if((tree_quantity+grass_quantity+Shrub_quantity)==0){
```

```
    veg_drag <- drag_coeff^2

  }else{veg_drag <- (grass_drag*grass_density+
shrub_drag*shrub_density+tree_drag*tree_density)/(vegetation_density)}


  drag_ratio <- veg_drag/drag_coeff

  drag_on_veg <- (drag_ratio*vegetation_density*tot_drag)/(1+drag_ratio*vegetation_density)

  drag_on_soil <- tot_drag/(1+drag_ratio*vegetation_density)


  #friction on surface is calculated


  grainsize_dependent_Fd <- 1.41+4.98*exp((-0.5)*(log((mean_grainsize/(1.55*ref_grainsize))^2)))

  treshhold_vel <- treshhold_vel

  friction_vel_on_surface <- sqrt(drag_on_soil/air_dens)

  ## flux in g/m/s


  if((1-treshhold_vel/friction_vel_on_surface)<0){

    surplus_erosie <- 0

  }else{

    surplus_erosie <- 1-(treshhold_vel/friction_vel_on_surface)

  }


  #constants for different formulas are given


  constant_Z <- 0.5

  constant_Ka <- 0.14

  constant_M <- 0.5

  constant_Kl <-0.5

  Constant_L <- 0.9
  # Sediment fluxes are calculated for different forumlas

  sediment_flux <- 10*grainsize_dependent_Fd*(surplus_erosie^1)*(air_dens/grav)*(friction_vel_on_surface^3)

  sediment_flux_1 <- 1000*constant_Z*((mean_grainsize/ref_grainsize)^0.75)*air_dens*(friction_vel_on_surface^3)/9.81

  sediment_flux_2 <- 1000*constant_Ka*(1-treshhold_vel/friction_vel_on_surface)*((1+treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81

  sediment_flux_3 <- 1000*(air_dens*(friction_vel_on_surface^3)/9.81)*constant_M*((mean_grainsize/ref_grainsize)^0.75)*(1-(treshhold_vel/friction_vel_on_surface)^13.72)

  sediment_flux_4 <- 1000*constant_KI*(1-(treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81

  sediment_flux_5 <- 1000*Constant_L *((mean_grainsize/ref_grainsize)^0.5)*(air_dens*(friction_vel_on_surface^3)/9.81)*(friction_vel_on_surface-treshhold_vel)



  if((friction_vel-treshhold_vel)<0){

    sediment_flux <- 0

    sediment_flux_1 <- 0

    sediment_flux_2 <- 0

    sediment_flux_3 <- 0

    sediment_flux_4 <- 0

    sediment_flux_5 <- 0

  }else{



    #sediment brought into suspension is calculated
    alpha <- alpha*10
    sediment_flux <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_1 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_2 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_3 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_4 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
    sediment_flux_5 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
  }

  erosion <- c(erosion,sediment_flux)
  erosion_1 <- c(erosion_1,sediment_flux_1)
  erosion_2 <- c(erosion_2,sediment_flux_2)
  erosion_3 <- c(erosion_3,sediment_flux_3)
  erosion_4 <- c(erosion_4,sediment_flux_4)
  erosion_5 <- c(erosion_5,sediment_flux_5)

  # If a sediment flux is lower than 0 it is set to 0

  erosion[erosion<0] <- 0
  erosion_1[erosion_1<0] <- 0
  erosion_2[erosion_2<0] <- 0
  erosion_3[erosion_3<0] <- 0
  erosion_4[erosion_4<0] <- 0
  erosion_5[erosion_5<0] <- 0
```

```
   erosions <- cbind(erosion,erosion_1,erosion_2,erosion_3,erosion_4,erosion_5)
  }

  #erosion is returned based on type of formula used

  if(type_of_model=='old'){
    return(erosion)}
  else if(type_of_model=='zingg'){
    return(erosion_1)}
  else if(type_of_model=='kawamura'){
    return(erosion_2)}
  else if(type_of_model=='maegley'){
    return(erosion_3)}
  else if(type_of_model=='kind'){
    return(erosion_4)}
  else if(type_of_model=='lettau'){
    return(erosion_5)}
  else if(type_of_model=='all'){
    return(erosions)}

}




veg_density_func <-
function(mean_grainsize=0.02,tree_quantity=0,tree_drag=0.3,tree_breidth=2.5,tree_porosity=0.7,tr
ee_height=3,
wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=15,
                         drag_coeff = 0.0033,type_of_model='kawamura',  grass_drag=0.1, grass_porosity=0,grass_height=0.15,
                         Test_area=10^2){



  ##Atmospheric conditions
  R <- 287.058/1000
  air_dens <- 101.325 /(R*(273.15+Temp))
  Height_z <- 100
  ## Soil specification are given
  Area <- Test_area
  bulk_density <- 1.5#g/cm3
  grain_dens <- 2650

  ##vegetation specifications are calculated
  #trees
  tree_frontalarea <- (tree_breidth/2)^2*pi
  tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area



  #shrubs
  shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity
  shrub_density <- Shrub_quantity*shrub_frontalarea/Area

  #grasses
  # if grass porosity is higher than 0 grass is added
  if(grass_porosity>0){
    grass_quantity <- 100*1000 # means 1000 per square meter
  }else{
    grass_quantity <- 0


  }
  grass_width <- 0.01
  grass_frontalarea <- grass_height*grass_width*grass_porosity
  grass_density <- grass_quantity*grass_frontalarea/Area

  # A verage vegetation density and heigth is calculated
  vegetation_density <- grass_density+shrub_density+tree_density
  if((grass_density+shrub_density+tree_density)==0){
    average_veg_height <- 0
  }else{
    average_veg_height <-
(grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density
+shrub_density+tree_density)
  }
  # vegtation density is set to a max of 0.8, due to model workings this is necessary.
  vegetation_density <- vegetation_density
  if(vegetation_density>0.8){
    vegetation_density <- 0.8
  }

  return(vegetation_density)
}
```

```
# vector of windspeed is made

scaling_vector <- seq(0,25,0.1)


#setting the wind vector used

wind_vel_vector <- scaling_vector



##################################


# Ca model is run for the different scenarios and a single high wind event.

#The Dust/Suspension model is then run for the 6 scenarios and a single high wind event.

#results are saved in a variable and later plotted


##################################

# vegetation grids are set for different scenarios.

# set vegetation grids how you wouuld like

# 100 cells per vector are required

# using the rep function these lengths are created

outflux_vector <- c()

grass_porosity_grid <- rep(c(0,0,0.0,0,0),20)

shrub_quantitygrid <- rep(c(0,0,0,0,0,0,0,0,0,0,0),10)

shrub_porositygrid <- rep(c(0),100)

shrub_heightgrid <- rep(c(0),100)

tree_quantitygrid <- rep(c(0,0,0,0,0,0,0,0,0,0),10)

tree_breidthgrid <- rep(c(0),100)

tree_porositygrid <- rep(c(0),100)
```

```
tree_heightgrid<- rep(c(0),100)


# timestep is set to 1

Time_step <- 1

# an empty grid for the model(erosiongird) is made, (this grid is one larger than it needs to be becouse it needs to add from a previous time step which must exist)

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)

# model is run for wind velocity vector

for (wind_vel in wind_vel_vector) {

  # a time-step in the grid is taken

  Time_step <- Time_step+1

  # The lateral placement is set to 1

  count <- length(grass_porosity_grid)


  # loop to calculate erosion for different cells in the model

  for (grass_porosity in grass_porosity_grid){

    #generation of erosion is calculated for this cell

    production <- The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    # Vegetation density for intertception of erosion is calculated

    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    #transport is calculated
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    #erosion grid is updated
    erosiongrid[Time_step,count] <- production+transport
    #Lateral step in the gird is taken
    count <- count-1
  }
  #Outflux out of the model is calculated
  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
  #Outfluxvector is updated
  outflux_vector <- c(outflux_vector,outflux)
}
#Outcome for scenario 1 is set
Scenario_test_1 <- outflux_vector
#plot(wind_vel_vector,Scenario_6,type='l')
#plot(erosiongrid[160,],type='l',main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')



#empty vector is created
dust_vector <- c()
count <- 0
#loop for suspension
for (wind_vel in wind_vel_vector) {
  #time step is taken
  count <- count+1
  #model is run

  Dust <-
dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass
_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  #output is added to the vector

  dust_vector <- c(dust_vector,Dust)

}

#Outcome of scenario 1 is set

scenario_test_dust_1 <- dust_vector




################################################################

outflux_vector <- c()


grass_porosity_grid <- rep(c(0,0,0,Grassy),25)


# timestep is set to 1

Time_step <- 1

# an empty grid for the model(erosiongird) is made, (this grid is one larger than it needs to be
becouse it needs to add from a previous time step which must exist)

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

# model is run for wind velocity vector

for (wind_vel in wind_vel_vector) {

  # a time-step in the grid is taken

  Time_step <- Time_step+1
```

```
 # The lateral placement is set to 1
 count <- length(grass_porosity_grid)


 # loop to calculate erosion for different cells in the model
 for (grass_porosity in grass_porosity_grid){
   #generation of erosion is calculated for this cell
   production <-
The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygr
id[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
   # Vegetation density for intertception of erosion is calculated
   vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
   #transport is calculated
   transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
   #erosion grid is updated
   erosiongrid[Time_step,count] <- production+transport
   #Lateral step in the gird is taken
   count <- count-1
 }
 #Outflux out of the model is calculated
 outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
 #Outfluxvector is updated
```

```
  outflux_vector <- c(outflux_vector,outflux)
}
#Outcome for scenario 1 is set
Scenario_test_2 <- outflux_vector
#plot(wind_vel_vector,Scenario_6,type='l')
#plot(erosiongrid[160,],type='l',main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')


#empty vector is created
dust_vector <- c()
count <- 0
#loop for suspension
for (wind_vel in wind_vel_vector) {
  #time step is taken
  count <- count+1
  #model is run
  Dust <- dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),

Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))
  #output is added to the vector
  dust_vector <- c(dust_vector,Dust)
}
#Outcome of scenario 1 is set
scenario_test_dust_2 <- dust_vector
```

```
####################################################
grass_porosity_grid <- rep(c(0,0,Grassy,Grassy),25)
outflux_vector <- c()


# timestep is set to 1
Time_step <- 1
# an empty grid for the model(erosiongird) is made, (this grid is one larger than it needs to be becouse it needs to add from a previous time step which must exist)
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)
# model is run for wind velocity vector
for (wind_vel in wind_vel_vector) {
 # a time-step in the grid is taken
 Time_step <- Time_step+1
 # The lateral placement is set to 1
 count <- length(grass_porosity_grid)


 # loop to calculate erosion for different cells in the model
 for (grass_porosity in grass_porosity_grid){
  #generation of erosion is calculated for this cell
  production <- The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
  # Vegetation density for intertception of erosion is calculated
  vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],
Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    #transport is calculated
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    #erosion grid is updated
    erosiongrid[Time_step,count] <- production+transport
    #Lateral step in the gird is taken
    count <- count-1
  }
  #Outflux out of the model is calculated
  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
  #Outfluxvector is updated
  outflux_vector <- c(outflux_vector,outflux)
}
#Outcome for scenario 1 is set
Scenario_test_3 <- outflux_vector
#plot(wind_vel_vector,Scenario_6,type='l')
#plot(erosiongrid[160,],type='l',main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')



#empty vector is created
dust_vector <- c()
count <- 0
#loop for suspension
for (wind_vel in wind_vel_vector) {
  #time step is taken
  count <- count+1
  #model is run
  Dust <-
dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass
_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),

Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))
  #output is added to the vector
  dust_vector <- c(dust_vector,Dust)
}
#Outcome of scenario 1 is set
scenario_test_dust_3 <- dust_vector




##################################################
outflux_vector <- c()

grass_porosity_grid <- rep(c(0,Grassy,Grassy,Grassy),25)
# timestep is set to 1
Time_step <- 1
# an empty grid for the model(erosiongird) is made, (this grid is one larger than it needs to be
becouse it needs to add from a previous time step which must exist)
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)
# model is run for wind velocity vector
for (wind_vel in wind_vel_vector) {
  # a time-step in the grid is taken
  Time_step <- Time_step+1
  # The lateral placement is set to 1
  count <- length(grass_porosity_grid)

  # loop to calculate erosion for different cells in the model
  for (grass_porosity in grass_porosity_grid){
    #generation of erosion is calculated for this cell
    production <- The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

 Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    # Vegetation density for intertception of erosion is calculated
    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

 Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    #transport is calculated
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    #erosion grid is updated
    erosiongrid[Time_step,count] <- production+transport
    #Lateral step in the gird is taken
    count <- count-1
  }
  #Outflux out of the model is calculated
```

```
  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
  #Outfluxvector is updated
  outflux_vector <- c(outflux_vector,outflux)
}
#Outcome for scenario 1 is set
Scenario_test_4 <- outflux_vector
#plot(wind_vel_vector,Scenario_6,type='l')
#plot(erosiongrid[160,],type='l',main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')



#empty vector is created
dust_vector <- c()
count <- 0
#loop for suspension
for (wind_vel in wind_vel_vector) {
  #time step is taken
  count <- count+1
  #model is run
  Dust <- dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),

Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))
  #output is added to the vector
  dust_vector <- c(dust_vector,Dust)
}
#Outcome of scenario 1 is set
```

```
scenario_test_dust_4 <- dust_vector



##################################################
outflux_vector <- c()


grass_porosity_grid <- rep(c(Grassy,Grassy,Grassy,Grassy),25)
# timestep is set to 1
Time_step <- 1
# an empty grid for the model(erosiongird) is made, (this grid is one larger than it needs to be
becouse it needs to add from a previous time step which must exist)
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)
# model is run for wind velocity vector
for (wind_vel in wind_vel_vector) {
  # a time-step in the grid is taken
  Time_step <- Time_step+1
  # The lateral placement is set to 1
  count <- length(grass_porosity_grid)


  # loop to calculate erosion for different cells in the model
  for (grass_porosity in grass_porosity_grid){
    #generation of erosion is calculated for this cell
    production <-
The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
  # Vegetation density for intertception of erosion is calculated
  vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
  #transport is calculated
  transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
  #erosion grid is updated
  erosiongrid[Time_step,count] <- production+transport
  #Lateral step in the gird is taken
  count <- count-1
 }
 #Outflux out of the model is calculated
 outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
 #Outfluxvector is updated
 outflux_vector <- c(outflux_vector,outflux)
}
#Outcome for scenario 1 is set
Scenario_test_5 <- outflux_vector
#plot(wind_vel_vector,Scenario_6,type='l')
#plot(erosiongrid[160,],type='l',main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')



#empty vector is created
```

```
dust_vector <- c()

count <- 0

#loop for suspension

for (wind_vel in wind_vel_vector) {

  #time step is taken

  count <- count+1

  #model is run

  Dust <-
dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))

  #output is added to the vector

  dust_vector <- c(dust_vector,Dust)

}

#Outcome of scenario 1 is set

scenario_test_dust_5 <- dust_vector




#################################################################


plot(wind_vel_vector,Scenario_test_1,xlab = 'Wind velocity',ylab='Erosion(saltation)' ,main='Model results',type='l', lwd=3,col='brown')

lines(wind_vel_vector,Scenario_test_2,col='black')

lines(wind_vel_vector,Scenario_test_3, col='dark green')

lines(wind_vel_vector,Scenario_test_4, col = 'gold')


sum0 <- sum(Scenario_test_1)
```

```
sum1 <- sum(Scenario_test_2)
sum2 <- sum(Scenario_test_3)
sum3 <- sum(Scenario_test_4)
sum4 <- sum(Scenario_test_5)

sumbind <- rbind(sum0,sum1,sum2,sum3,sum4)
barplot(height=sumbind,,width=1,main="Wind erosion per grasscover in the horizontal flux",
    xlab="Percentage of grasscover removed",beside=TRUE,ylab='Erosion (g/m/d.)',ylim=c(0,800),
    names.arg=c("100%", "75%", "50%",'25%','0%'))

#plot(wind_vel_vector,scenario_test_dust_1,xlab = 'Wind
velocity',ylab='Erosion(Suspension)',main='Model results',type='l', lwd=3,col='gold')
#lines(wind_vel_vector,scenario_test_dust_2)



dsum0 <- sum(scenario_test_dust_1)
dsum1 <- sum(scenario_test_dust_2)
dsum2 <- sum(scenario_test_dust_3)
dsum3 <- sum(scenario_test_dust_4)
dsum4 <- sum(scenario_test_dust_5)

dsumbind <- rbind(dsum0,dsum1,dsum2,dsum3,dsum4)
#barplot(height=dsumbind,width=1,main="Wind erosion per grasscover in the Vertical flux",
#       xlab="percentage of grass",beside=TRUE,ylab='Erosion (g/m/d.)',
#       names.arg=c("0%", "25%", "50%",'75%','100%'))
```

5.5

```
#The_model is the main funciton which calculates total erosion in g/m/s

The_model <-
function(mean_grainsize=0.02,tree_quantity=0,tree_drag=0.3,tree_breidth=2.5,tree_porosity=0.7,tr
ee_height=3,


wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=1
5,

                  drag_coeff = 0.0030,type_of_model='kawamura',  grass_drag=0.1,
grass_porosity=0,grass_height=0.15,

                  Test_area=10^2){

 #Empty strings are created for each type of sediment flux calculation. They will be filled with data

 erosion <- c()

 erosion_1 <- c()

 erosion_2 <- c()

 erosion_3 <- c()

 erosion_4 <- c()

 erosion_5 <- c()


 # If the model is ran for a string of velocities the model will be run multiple times in this loop

 # for CA modelling done with the scenarios Wind_vel_vector is a vector of size 1

 for( wind_vel in wind_vel_vector){
```

```
if (wind_vel<1){
  wind_vel <- 1}



##Atmospheric conditions are set
R <- 287.058/1000
air_dens <- 101.325 /(R*(273.15+Temp))

# Height of wind velocity is set
Height_z <- 100
## Soil specifications are set
Area <- Test_area
grain_dens <- 2650

##vegetation Densities and frontal area are calculated

#trees
tree_frontalarea <- (tree_breidth/2)^2*pi
tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area

#shrubs
shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity
shrub_density <- Shrub_quantity*shrub_frontalarea/Area

#grasses

# grasses are added when there is a grass porosity higher than 0
```

```
  if(grass_porosity>0){
    grass_quantity <- 100*1000 # means 1000 per square meter
  }else{
    grass_quantity <- 0

  }
  grass_width <- 0.01
  grass_frontalarea <- grass_height*grass_width*grass_porosity
  grass_density <- grass_quantity*grass_frontalarea/Area

  # average vegetation specifics
  vegetation_density <- grass_density+shrub_density+tree_density
  if((grass_density+shrub_density+tree_density)==0){
    average_veg_height <- 0
  }else{
    average_veg_height <-
(grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density
+shrub_density+tree_density)
  }

  #treshold velocity is calculated
  mean_grainsize_adapt <- mean_grainsize*10^-4.8
  treshhold_vel <- 7*sqrt(0.0123*(grain_dens/air_dens*9.81*mean_grainsize_adapt+(3*(10^-
4)/(mean_grainsize_adapt*grain_dens))))

  #constants
  Karman <- 0.4
  grav <- 9.81
  Ak <- 0.1
  ref_grainsize <- 0.25


  #wind profile and friction velocity are calculated

  z0 <- 0.001+(0.5*vegetation_density*average_veg_height)

  friction_vel <- (wind_vel*Karman)/(log(Height_z/(z0)))

  Heights <- seq(1,100,1)

  vel_per_height <- (friction_vel/Karman*(log(Heights)/z0))


  ##Drag is partitioned

  tot_drag <- (friction_vel^2)*air_dens

  if((tree_quantity+grass_quantity+Shrub_quantity)==0){

    veg_drag <- drag_coeff^2

  }else{veg_drag <- (grass_drag*grass_density+
shrub_drag*shrub_density+tree_drag*tree_density)/(vegetation_density)}


  drag_ratio <- veg_drag/drag_coeff

  drag_on_veg <- (drag_ratio*vegetation_density*tot_drag)/(1+drag_ratio*vegetation_density)

  drag_on_soil <- tot_drag/(1+drag_ratio*vegetation_density)


  ## Friction velcoity on surface is calculated


  grainsize_dependent_Fd <- 1.41+4.98*exp((-0.5)*(log((mean_grainsize/(1.55*ref_grainsize))^2)))

  treshhold_vel <- treshhold_vel

  friction_vel_on_surface <- sqrt(drag_on_soil/air_dens)

  ## flux in g/m/s


  if((1-treshhold_vel/friction_vel_on_surface)<0){

    surplus_erosie <- 0
  }else{
    surplus_erosie <- 1-(treshhold_vel/friction_vel_on_surface)
  }


  # constants for different sediment flux formulas are calculated
  constant_Z <- 0.5
  constant_Ka <- 0.14
  constant_M <- 0.5
  constant_KI <-0.5
  Constant_L <- 0.9


  #sediment fluxes are calculated using different formulas
  sediment_flux <- 10*grainsize_dependent_Fd*(surplus_erosie^1)*(air_dens/grav)*(friction_vel_on_surface^3)
  sediment_flux_1 <- 1000*constant_Z*((mean_grainsize/ref_grainsize)^0.75)*air_dens*(friction_vel_on_surface^3)/9.81
  sediment_flux_2 <- 1000*constant_Ka*(1-treshhold_vel/friction_vel_on_surface)*((1+treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81
  sediment_flux_3 <- 1000*(air_dens*(friction_vel_on_surface^3)/9.81)*constant_M*((mean_grainsize/ref_grainsize)^0.75)*(1-(treshhold_vel/friction_vel_on_surface)^13.72)
  sediment_flux_4 <- 1000*constant_KI*(1-(treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81
  sediment_flux_5 <- 1000*Constant_L *((mean_grainsize/ref_grainsize)^0.5)*(air_dens*(friction_vel_on_surface^3)/9.81)*(friction_vel_on_surface-treshhold_vel)




  erosion <- c(erosion,sediment_flux)
  erosion_1 <- c(erosion_1,sediment_flux_1)
  erosion_2 <- c(erosion_2,sediment_flux_2)
  erosion_3 <- c(erosion_3,sediment_flux_3)
  erosion_4 <- c(erosion_4,sediment_flux_4)
  erosion_5 <- c(erosion_5,sediment_flux_5)

  #some formulas give negative values when wind velocity is low. If this is the case the erosion is set to 0
  erosion[erosion<0] <- 0
  erosion_1[erosion_1<0] <- 0
  erosion_2[erosion_2<0] <- 0
  erosion_3[erosion_3<0] <- 0
  erosion_4[erosion_4<0] <- 0
  erosion_5[erosion_5<0] <- 0

  # factor for CA_modelling is added
  erosion <- erosion/4
  erosion_1 <- erosion_1/4
  erosion_2 <- erosion_2/4
  erosion_3 <- erosion_3/4
  erosion_4 <- erosion_4/4
  erosion_5 <- erosion_5/4




  erosions <- cbind(erosion,erosion_1,erosion_2,erosion_3,erosion_4,erosion_5)
 }
```

```
# The type of erosion used in the model is returned
  if(type_of_model=='old'){
    return(erosion)}
  else if(type_of_model=='zingg'){
    return(erosion_1)}
  else if(type_of_model=='kawamura'){
    return(erosion_2)}
  else if(type_of_model=='maegley'){
    return(erosion_3)}
  else if(type_of_model=='kind'){
    return(erosion_4)}
  else if(type_of_model=='lettau'){
    return(erosion_5)}
  else if(type_of_model=='all'){
    return(erosions)}


}



dustfunc <- function(production=0,alpha=10,mean_grainsize=0.065,tree_quantity=0,tree_drag=0.3,tree_breidth=2.5,tree_porosity=0.7,tree_height=3,


wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=15,

                    drag_coeff = 0.0030,type_of_model='kawamura',  grass_drag=0.1, grass_porosity=0,grass_height=0.15,

                    Test_area=10^2){
```

```
#empty datasets are created
erosion <- c()
erosion_1 <- c()
erosion_2 <- c()
erosion_3 <- c()
erosion_4 <- c()
erosion_5 <- c()
# If there is a string of windvelocity it is looped here
for( wind_vel in wind_vel_vector){
  if (wind_vel<1){
    wind_vel <- 1
  }

  ##Atmospheric conditions
  R <- 287.058/1000
  air_dens <- 101.325 /(R*(273.15+Temp))
  Height_z <- 100
  Area <- Test_area
  grain_dens <- 2650

  ##vegetation densities and frontal areas are calculated

  #trees
  tree_frontalarea <- (tree_breidth/2)^2*pi
  tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area
  #shrubs
  shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity
  shrub_density <- Shrub_quantity*shrub_frontalarea/Area


  #grasses
  # if grass porosity is higher than 0 grass is added
  if(grass_porosity>0){
    grass_quantity <- 100*1000 # means 1000 per square meter
  }else{
    grass_quantity <- 0
  }


  grass_width <- 0.01
  grass_frontalarea <- grass_height*grass_width*grass_porosity
  grass_density <- grass_quantity*grass_frontalarea/Area



  # vegetation averaged density and height is calculated
  vegetation_density <- grass_density+shrub_density+tree_density
  if((grass_density+shrub_density+tree_density)==0){
    average_veg_height <- 0
  }else{
    average_veg_height <-
(grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density
+shrub_density+tree_density)
  }


  # treshold velocirty is calculated
```

```
mean_grainsize_adapt <- mean_grainsize*8*10^-5.5
treshhold_vel <- 9.5*sqrt(0.0123*(grain_dens/air_dens*9.81*mean_grainsize_adapt+(3*(10^-4)/(mean_grainsize_adapt*grain_dens))))


#constants
Karman <- 0.4
grav <- 9.81
Ak <- 0.1
ref_grainsize <- 0.25



#wind profile and friction velcoty are calulated
z0 <- 0.001+(0.5*vegetation_density*average_veg_height)
friction_vel <- (wind_vel*Karman)/(log(Height_z/(z0)))



Heights <- seq(1,100,1)
vel_per_height <- (friction_vel/Karman*(log(Heights)/z0))



##Drag is partitioned
tot_drag <- (friction_vel^2)*air_dens
if((tree_quantity+grass_quantity+Shrub_quantity)==0){
  veg_drag <- drag_coeff^2
}else{veg_drag <- (grass_drag*grass_density+ shrub_drag*shrub_density+tree_drag*tree_density)/(vegetation_density)}


drag_ratio <- veg_drag/drag_coeff
```

```
drag_on_veg <- (drag_ratio*vegetation_density*tot_drag)/(1+drag_ratio*vegetation_density)

drag_on_soil <- tot_drag/(1+drag_ratio*vegetation_density)


#friction on surface is calculated


grainsize_dependent_Fd <- 1.41+4.98*exp((-0.5)*(log((mean_grainsize/(1.55*ref_grainsize))^2)))

treshhold_vel <- treshhold_vel

friction_vel_on_surface <- sqrt(drag_on_soil/air_dens)

## flux in g/m/s


if((1-treshhold_vel/friction_vel_on_surface)<0){

  surplus_erosie <- 0

}else{

  surplus_erosie <- 1-(treshhold_vel/friction_vel_on_surface)

}


#constants for different formulas are given


constant_Z <- 0.5

constant_Ka <- 0.14

constant_M <- 0.5

constant_Kl <-0.5

Constant_L <- 0.9


# Sediment fluxes are calculated for different forumlas


sediment_flux <-
10*grainsize_dependent_Fd*(surplus_erosie^1)*(air_dens/grav)*(friction_vel_on_surface^3)
```

```
  sediment_flux_1 <- 1000*constant_Z*((mean_grainsize/ref_grainsize)^0.75)*air_dens*(friction_vel_on_surface^3)/9.81

  sediment_flux_2 <- 1000*constant_Ka*(1-treshhold_vel/friction_vel_on_surface)*((1+treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81

  sediment_flux_3 <- 1000*(air_dens*(friction_vel_on_surface^3)/9.81)*constant_M*((mean_grainsize/ref_grainsize)^0.75)*(1-(treshhold_vel/friction_vel_on_surface)^13.72)

  sediment_flux_4 <- 1000*constant_KI*(1-(treshhold_vel/friction_vel_on_surface)^2)*air_dens*(friction_vel_on_surface^3)/9.81

  sediment_flux_5 <- 1000*Constant_L *((mean_grainsize/ref_grainsize)^0.5)*(air_dens*(friction_vel_on_surface^3)/9.81)*(friction_vel_on_surface-treshhold_vel)



  if((friction_vel-treshhold_vel)<0){

   sediment_flux <- 0

   sediment_flux_1 <- 0

   sediment_flux_2 <- 0

   sediment_flux_3 <- 0

   sediment_flux_4 <- 0

   sediment_flux_5 <- 0

     }else{



     #sediment brought into suspension is calculated


  sediment_flux <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha

  sediment_flux_1 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha

  sediment_flux_2 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha

  sediment_flux_3 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha
  sediment_flux_4 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha

  sediment_flux_5 <- production*((friction_vel_on_surface-treshhold_vel)/grav)^2*alpha

      }


  erosion <- c(erosion,sediment_flux)

  erosion_1 <- c(erosion_1,sediment_flux_1)

  erosion_2 <- c(erosion_2,sediment_flux_2)

  erosion_3 <- c(erosion_3,sediment_flux_3)

  erosion_4 <- c(erosion_4,sediment_flux_4)

  erosion_5 <- c(erosion_5,sediment_flux_5)


  # If a sediment flux is lower than 0 it is set to 0


  erosion[erosion<0] <- 0

  erosion_1[erosion_1<0] <- 0

  erosion_2[erosion_2<0] <- 0

  erosion_3[erosion_3<0] <- 0

  erosion_4[erosion_4<0] <- 0

  erosion_5[erosion_5<0] <- 0



  erosions <- cbind(erosion,erosion_1,erosion_2,erosion_3,erosion_4,erosion_5)

}


#erosion is returned based on type of formula used


if(type_of_model=='old'){
    return(erosion)}
  else if(type_of_model=='zingg'){
    return(erosion_1)}
  else if(type_of_model=='kawamura'){
    return(erosion_2)}
  else if(type_of_model=='maegley'){
    return(erosion_3)}
  else if(type_of_model=='kind'){
    return(erosion_4)}
  else if(type_of_model=='lettau'){
    return(erosion_5)}
  else if(type_of_model=='all'){
    return(erosions)}


}





veg_density_func <-
function(mean_grainsize=0.065,tree_quantity=0,tree_drag=0.3,tree_breidth=2.5,tree_porosity=0.7,tree_height=3,

wind_vel_vector=15,Shrub_quantity=0,shrub_drag=0.2,shrub_porosity=0.5,shrub_height=1,Temp=15,
                         drag_coeff = 0.0033,type_of_model='old',  grass_drag=0.1,
grass_porosity=0,grass_height=0.15,
                         Test_area=10^2){
```

```
##Atmospheric conditions
R <- 287.058/1000
air_dens <- 101.325 /(R*(273.15+Temp))
Height_z <- 100
## Soil specification are given
Area <- Test_area
bulk_density <- 1.5#g/cm3
grain_dens <- 2650

##vegetation specifications are calculated
#trees
tree_frontalarea <- (tree_breidth/2)^2*pi
tree_density <- tree_quantity*tree_frontalarea*tree_porosity/Area



#shrubs
shrub_frontalarea <- 3.14159265359*(shrub_height^2)/2*shrub_porosity
shrub_density <- Shrub_quantity*shrub_frontalarea/Area

#grasses
# if grass porosity is higher than 0 grass is added
if(grass_porosity>0){
  grass_quantity <- 100*1000 # means 1000 per square meter
}else{
  grass_quantity <- 0

}
```

```
  grass_width <- 0.01

  grass_frontalarea <- grass_height*grass_width*grass_porosity

  grass_density <- grass_quantity*grass_frontalarea/Area


  # A verage vegetation density and heigth is calculated

  vegetation_density <- grass_density+shrub_density+tree_density

  if((grass_density+shrub_density+tree_density)==0){

    average_veg_height <- 0

  }else{

    average_veg_height <- (grass_height*grass_density+shrub_density*shrub_height+tree_height*tree_density)/(grass_density+shrub_density+tree_density)

  }

  # vegtation density is set to a max of 0.8, due to model workings this is necessary.

  vegetation_density <- vegetation_density

  if(vegetation_density>0.8){

    vegetation_density <- 0.8

  }


  return(vegetation_density)

}





#plot(wind_vel_vector=seq(5,30,0.1),The_model(wind_vel_vector=seq(5,30,0.1)))

#################################################


#wind data is read
```

```
all_wind_data <- read.csv("C:/Users/timon/Documents/scriptie/Data/Windspeeds.csv")


#unused coloms are set ot 0

hourly_wind_data <- all_wind_data

hourly_wind_data$wdsp.chimbaj <- NULL

hourly_wind_data$mxwspd.kungrad <- NULL

hourly_wind_data$wdsp.kungrad <- NULL

hourly_wind_data$mxwdsp.chimbaj <- NULL


#8*30+8*15*365

#hourly_wind_data[51668:51674,]


#April 29, 2008.

#28 March 2010

pick_year <- '2004'

pick_month <- '04'

pick_day <- '24'


windhours <-  hourly_wind_data[ which(substr(hourly_wind_data$Date, start = 1, stop =
4)==pick_year&substr(hourly_wind_data$Date, start = 6, stop = 7)

==pick_month & substr(hourly_wind_data$Date, start = 9, stop = 10)==pick_day), ]

windhours




test_3<- windhours


hourly_wind_data[ which(hourly_wind_data$kungrad >15),]
```

```
# Wind data for different classes wind events are exracted from the dataset

################## class. 1
Date_25_feb_2007 <- hourly_wind_data$kungrad[17627:17634]
Date_12_jan_13 <- hourly_wind_data$kungrad[34478:34485]

################## class 2
Date_15_april_01 <- hourly_wind_data$kungrad[814:821]
Date_30_may_01 <- hourly_wind_data$kungrad[1163:1170]

################## class 3

Date_17_march_16 <- hourly_wind_data$kungrad[43321:43328]



# function to change 3-hourly vector in to a 10-min vector

vectormaker <- function(windvector){
  speedvector <- c()
  for (speed in windvector){
    speeds <- c(NA,NA,NA,rep(speed,30),NA,NA,NA)
    speedvector <- c(speedvector,speeds)
  }

  speedvector <- na_interpolation(speedvector, option = "stine" )
  return(speedvector)

}




#################### classes are set to the new standart

#Class 1

Date_25_feb_2007 <- vectormaker(Date_25_feb_2007)

Date_12_jan_13 <- vectormaker(Date_12_jan_13)

#Class 2

Date_15_april_01 <- vectormaker(Date_15_april_01)

Date_30_may_01 <- vectormaker(Date_30_may_01)

#Class3

Date_17_march_16 <- vectormaker(Date_17_march_16)


#1 2005-08-02      25

#2 2005-08-21      25

#3 2005-11-28      26

#4 2016-03-16      25

#5 2016-03-17      25


scaling_vector <- seq(0,25,0.1)


#setting the wind vector used

wind_vel_vector <- Date_25_feb_2007
```

```
##################################

#grids are set for different scenarios.
# an empty grid for the model(erosiongird) is made
# Ca model is run for the different scenarios and a single high wind event.
#The Dust/Suspension model is then run for the 6 scenarios and a single high wind event.
#results are saved in a variable and later plotted

##################################
outflux_vector <- c()
grass_porosity_grid <- rep(c(0,0,0.2,0,0),20)
shrub_quantitygrid <- rep(c(0,0,0,0,0,0,15.6,0,0,0,0),10)
shrub_porositygrid <- rep(c(0.7),100)
shrub_heightgrid <- rep(c(0.5),100)
tree_quantitygrid <- rep(c(36.5,0,0,0,0,0,0,0,0,0,0),10)
tree_breidthgrid <- rep(c(3),100)
tree_porositygrid <- rep(c(0.6),100)
tree_heightgrid<- rep(c(3.5),100)


Time_step <- 1
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)
for (wind_vel in wind_vel_vector) {
 Time_step <- Time_step+1
 count <- 1
 for (grass_porosity in grass_porosity_grid){
    production <-
The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygr
id[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step-1,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count+1

  }

  outflux <- erosiongrid[Time_step-1,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_6 <- outflux_vector

#plot(wind_vel_vector,Scenario_6,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <-
dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass
_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),
Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_6_dust <- dust_vector

##########################

outflux_vector <- c()

grass_porosity_grid <- rep(c(0.0,0.0),50)

shrub_quantitygrid <- rep(c(0),100)

shrub_porositygrid <- rep(c(0.0),100)

shrub_heightgrid <- rep(c(0),100)

tree_quantitygrid <- c(rep(c(3),100))

tree_breidthgrid <- rep(c(0.7),100)

tree_porositygrid <- rep(c(0.6),100)

tree_heightgrid<- rep(c(0.8),100)


Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <-
The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_5 <- outflux_vector



dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <-
dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),

Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}
```

```
scenario_5_dust <- dust_vector

#plot(wind_vel_vector,Scenario_5,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')

#####################################

outflux_vector <- c()

grass_porosity_grid <- rep(c(0,0.2,0,0),25)

shrub_quantitygrid <- rep(c(4,0,0,0),25)

shrub_porositygrid <- rep(c(0.8),100)

shrub_heightgrid <- rep(c(1),100)

tree_quantitygrid <- rep(c(0),100)

tree_breidthgrid <- rep(c(0.0),100)

tree_porositygrid <- rep(c(0.0),100)

tree_heightgrid<- rep(c(0.0),100)

Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <- The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],
Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_4 <- outflux_vector



dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <-
dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass
_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_4_dust <- dust_vector

#plot(wind_vel_vector,Scenario_4,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')
```

```
################################


outflux_vector <- c()

grass_porosity_grid <- rep(c(0.0,0.0),50)

shrub_quantitygrid <- rep(c(1.6),100)

shrub_porositygrid <- rep(c(0.8),100)

shrub_heightgrid <- rep(c(0.8),100)

tree_quantitygrid <- rep(c(0),100)

tree_breidthgrid <- rep(c(0.0),100)

tree_porositygrid <- rep(c(0.0),100)

tree_heightgrid<- rep(c(0.0),100)


Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <-
The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygr
id[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],
Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_3 <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_3_dust <- dust_vector

#plot(wind_vel_vector,Scenario_3,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')
```

```
#################################

outflux_vector <- c()
grass_porosity_grid <- rep(c(0.2,0.0,0.0,0.0),25)
shrub_quantitygrid <- rep(c(0.0),100)
shrub_porositygrid <- rep(c(0.0),100)
shrub_heightgrid <- rep(c(0.0),100)
tree_quantitygrid <- rep(c(0),100)
tree_breidthgrid <- rep(c(0.0),100)
tree_porositygrid <- rep(c(0.0),100)
tree_heightgrid<- rep(c(0.0),100)

Time_step <- 1
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)
for (wind_vel in wind_vel_vector) {
  Time_step <- Time_step+1
  count <- length(grass_porosity_grid)
  for (grass_porosity in grass_porosity_grid){
    production <-
The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_2 <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- 
dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_2_dust <- dust_vector

#plot(wind_vel_vector,Scenario_2,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')


################################
```

```
outflux_vector <- c()
grass_porosity_grid <- rep(c(0.0,0.0),50)
shrub_quantitygrid <- rep(c(0.0),100)
shrub_porositygrid <- rep(c(0.0),100)
shrub_heightgrid <- rep(c(0.0),100)
tree_quantitygrid <- rep(c(0),100)
tree_breidthgrid <- rep(c(0.0),100)
tree_porositygrid <- rep(c(0.0),100)
tree_heightgrid<- rep(c(0.0),100)

Time_step <- 1
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)
for (wind_vel in wind_vel_vector) {
  Time_step <- Time_step+1
  count <- length(grass_porosity_grid)
  for (grass_porosity in grass_porosity_grid){
    production <- The_model(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
```

```
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_1 <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <-
dustfunc(type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass
_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_1_dust <- dust_vector

#plot(wind_vel_vector,Scenario_1,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')



################################
```

```
#plot(wind_vel_vector,Scenario_1,ylim=c(0,20),main='scenarios', type='l', col='black',ylab = 'Sediment transport (g/m/s)',xlab='Wind velocity' )
#lines(wind_vel_vector,Scenario_2, col='violet')
#lines(wind_vel_vector,Scenario_3, col='orange')
#lines(wind_vel_vector,Scenario_4, col='yellow')
#lines(wind_vel_vector,Scenario_5, col='green')
#lines(wind_vel_vector,Scenario_6, col='blue')
#legend(6,15,legend=c('bare','bare+grasses','shrub-','shrub+grasses','tree-','tree+'),col=c('black','violet','orange','yellow','green','blue')
#      , lty=1, cex=1)


#A time vector is created
Time <- seq(0,24,(24/(length(wind_vel_vector)-1)))


#The results are plotted
plot(Time,Scenario_1,lwd =3,ylim=c(0,13),main='Horizontal sediment flux, class 1 storm ', type='l', col='darkgray',ylab = 'Horizontal sediment flux (g/m/s)',xlab='time(hours)' )
lines(Time,Scenario_2,lwd =3, col='dimgray')
lines(Time,Scenario_3,lwd =3, col='gold')
lines(Time,Scenario_4,lwd =3, col='darkgoldenrod4')
lines(Time,Scenario_5,lwd =3, col='darkolivegreen3')
lines(Time,Scenario_6,lwd =3, col='darkgreen')
legend(15.54,13.5,legend=c('bare','bare- & grasses','shrub-','shrub+ & grasses','tree-','tree+ & shrub+ & grasses'),col=c('darkgray','dimgray','gold','goldenrod4','darkolivegreen3','darkgreen')
       , lty=1, cex=1,lwd =3)
#15.54
```

```
plot(Time,scenario_1_dust ,lwd =3, ylim=c(0,0.4),main='Vertical sediment flux,class 1 storm', type='l',
col='darkgray',ylab = 'Vertical sediment flux (g/m^2/s)',xlab='time(hours)' )

lines(Time,scenario_2_dust,lwd =3, col='dimgray')

lines(Time,scenario_3_dust,lwd =3, col='gold')

lines(Time,scenario_4_dust,lwd =3, col='darkgoldenrod4')

lines(Time,scenario_5_dust,lwd =3, col='darkolivegreen3')

lines(Time,scenario_6_dust,lwd =3, col='darkgreen')

legend(16.24,0.416,legend=c('bare','bare- & grasses','shrub-','shrub+ & grasses','tree-','tree+ &
shrub+ & grasses'),col=c('darkgray','dimgray','gold','goldenrod4','darkolivegreen3','darkgreen')

       , lty=1, cex=1,,lwd =3)



#plot(wind_vel_vector,Scenario_1,type='l')


# cumulative erosion and suspended particle generation are calculated

Scenario_1_sum <- sum(Scenario_1)*5*60/1000

Scenario_2_sum <- sum(Scenario_2)*5*60/1000

Scenario_3_sum <- sum(Scenario_3)*5*60/1000

Scenario_4_sum <- sum(Scenario_4)*5*60/1000

Scenario_5_sum <- sum(Scenario_5)*5*60/1000

Scenario_6_sum <- sum(Scenario_6)*5*60/1000


print(Scenario_1_sum)

print(Scenario_2_sum)

print(Scenario_3_sum)

print(Scenario_4_sum)

print(Scenario_5_sum)
print(Scenario_6_sum)


print('dust')

Scenario_1_sum_dust <- sum(scenario_1_dust)*5*60/1000

Scenario_2_sum_dust <- sum(scenario_2_dust)*5*60/1000

Scenario_3_sum_dust <- sum(scenario_3_dust)*5*60/1000

Scenario_4_sum_dust <- sum(scenario_4_dust)*5*60/1000

Scenario_5_sum_dust <- sum(scenario_5_dust)*5*60/1000

Scenario_6_sum_dust <- sum(scenario_6_dust)*5*60/1000


print(Scenario_1_sum_dust)

print(Scenario_2_sum_dust)

print(Scenario_3_sum_dust)

print(Scenario_4_sum_dust)

print(Scenario_5_sum_dust)

print(Scenario_6_sum_dust)



shrub_vec <- seq(0,10,0.1)

dust_vector <- c()

for (shrub in shrub_vec) {

  Dust <- dustfunc(type_of_model='kawamura',Shrub_quantity=shrub, wind_vel_vector = 20)

  dust_vector <- c(dust_vector,Dust)

}


# The same process is done for maximum and minimum expected values to give an indication of the uncertainty of the results.
```

```
############################################################################boxplot stuff

outflux_vector <- c()

grass_porosity_grid <- rep(c(0,0,0.2,0,0),20)

shrub_quantitygrid <- rep(c(0,0,0,0,0,0,15.6,0,0,0,0),10)

shrub_porositygrid <- rep(c(0.7),100)

shrub_heightgrid <- rep(c(0.5),100)

tree_quantitygrid <- rep(c(36.5,0,0,0,0,0,0,0,0,0),10)

tree_breidthgrid <- rep(c(3),100)

tree_porositygrid <- rep(c(0.6),100)

tree_heightgrid<- rep(c(3.5),100)




Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <- The_model(drag_coeff = 0.0022,Temp=0,mean_grainsize =
0.065,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[co
unt],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],
Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_6_low <- outflux_vector

#plot(wind_vel_vector,Scenario_6,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(alpha=5,drag_coeff = 0.0022,Temp=0,mean_grainsize = 0.065,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_6_dust_low <- dust_vector

########################
```

```
outflux_vector <- c()
grass_porosity_grid <- rep(c(0.0,0.0),50)
shrub_quantitygrid <- rep(c(0),100)
shrub_porositygrid <- rep(c(0.0),100)
shrub_heightgrid <- rep(c(0),100)
tree_quantitygrid <- c(rep(c(3),100))
tree_breidthgrid <- rep(c(0.7),100)
tree_porositygrid <- rep(c(0.6),100)
tree_heightgrid<- rep(c(0.8),100)


Time_step <- 1
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)
for (wind_vel in wind_vel_vector) {
  Time_step <- Time_step+1
  count <- length(grass_porosity_grid)
  for (grass_porosity in grass_porosity_grid){
    production <- The_model(drag_coeff = 0.0022,Temp=0,mean_grainsize =
0.065,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[co
unt],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    erosiongrid[Time_step,count] <- production+transport


    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_5_low <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(alpha=5,drag_coeff = 0.0022,Temp=0,mean_grainsize =
0.065,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_po
rosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_5_dust_low <- dust_vector

#plot(wind_vel_vector,Scenario_5,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')

####################################

outflux_vector <- c()

grass_porosity_grid <- rep(c(0,0.2,0,0),25)

shrub_quantitygrid <- rep(c(4,0,0,0),25)

shrub_porositygrid <- rep(c(0.8),100)
```

```
shrub_heightgrid <- rep(c(1),100)
tree_quantitygrid <- rep(c(0),100)
tree_breidthgrid <- rep(c(0.0),100)
tree_porositygrid <- rep(c(0.0),100)
tree_heightgrid<- rep(c(0.0),100)
Time_step <- 1
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)
for (wind_vel in wind_vel_vector) {
  Time_step <- Time_step+1
  count <- length(grass_porosity_grid)
  for (grass_porosity in grass_porosity_grid){
    production <- The_model(drag_coeff = 0.0022,Temp=0,mean_grainsize = 0.065,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    erosiongrid[Time_step,count] <- production+transport


    count <- count-1
  }
  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_4_low <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(alpha=5,drag_coeff = 0.0022,Temp=0,mean_grainsize =
0.065,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_po
rosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_4_dust_low <- dust_vector

#plot(wind_vel_vector,Scenario_4,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')



################################



outflux_vector <- c()

grass_porosity_grid <- rep(c(0.0,0.0),50)

shrub_quantitygrid <- rep(c(1.6),100)

shrub_porositygrid <- rep(c(0.8),100)

shrub_heightgrid <- rep(c(0.8),100)

tree_quantitygrid <- rep(c(0),100)

tree_breidthgrid <- rep(c(0.0),100)
```

```
tree_porositygrid <- rep(c(0.0),100)
tree_heightgrid<- rep(c(0.0),100)


Time_step <- 1
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)
for (wind_vel in wind_vel_vector) {
  Time_step <- Time_step+1
  count <- length(grass_porosity_grid)
  for (grass_porosity in grass_porosity_grid){
    production <- The_model(drag_coeff = 0.0022,Temp=0,mean_grainsize = 0.065,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height =shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height =shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    erosiongrid[Time_step,count] <- production+transport


    count <- count-1
  }
  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
  outflux_vector <- c(outflux_vector,outflux)
}
```

```
Scenario_3_low <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(alpha=5,drag_coeff = 0.0022,Temp=0,mean_grainsize = 0.065,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_3_dust_low <- dust_vector

#plot(wind_vel_vector,Scenario_3,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')



################################



outflux_vector <- c()

grass_porosity_grid <- rep(c(0.2,0.0,0.0,0.0),25)

shrub_quantitygrid <- rep(c(0.0),100)

shrub_porositygrid <- rep(c(0.0),100)

shrub_heightgrid <- rep(c(0.0),100)

tree_quantitygrid <- rep(c(0),100)

tree_breidthgrid <- rep(c(0.0),100)

tree_porositygrid <- rep(c(0.0),100)

tree_heightgrid<- rep(c(0.0),100)
```

```
Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <- The_model(drag_coeff = 0.0022,Temp=0,mean_grainsize =
0.065,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[co
unt],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_2_low <- outflux_vector

dust_vector <- c()
```

```
count <- 0
for (wind_vel in wind_vel_vector) {
  count <- count+1
  Dust <- dustfunc(alpha=5,drag_coeff = 0.0022,Temp=0,mean_grainsize = 0.065,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),

Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))
  dust_vector <- c(dust_vector,Dust)
}
scenario_2_dust_low <- dust_vector
#plot(wind_vel_vector,Scenario_2,type='l')
#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')

################################

outflux_vector <- c()
grass_porosity_grid <- rep(c(0.0,0.0),50)
shrub_quantitygrid <- rep(c(0.0),100)
shrub_porositygrid <- rep(c(0.0),100)
shrub_heightgrid <- rep(c(0.0),100)
tree_quantitygrid <- rep(c(0),100)
tree_breidthgrid <- rep(c(0.0),100)
tree_porositygrid <- rep(c(0.0),100)
tree_heightgrid<- rep(c(0.0),100)

Time_step <- 1
```

```
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <- The_model(drag_coeff = 0.0022,Temp=0,mean_grainsize =
0.065,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[co
unt],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_1_low <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {
  count <- count+1

  Dust <- dustfunc(alpha=5,drag_coeff = 0.0022,Temp=0,mean_grainsize =
0.065,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_po
rosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_1_dust_low <- dust_vector

#plot(wind_vel_vector,Scenario_1,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')


################################


#plot(wind_vel_vector,Scenario_1,type='l')


Scenario_1_sum_low <- sum(Scenario_1_low)*5*60/1000

Scenario_2_sum_low <- sum(Scenario_2_low)*5*60/1000

Scenario_3_sum_low <- sum(Scenario_3_low)*5*60/1000

Scenario_4_sum_low <- sum(Scenario_4_low)*5*60/1000

Scenario_5_sum_low <- sum(Scenario_5_low)*5*60/1000

Scenario_6_sum_low <- sum(Scenario_6_low)*5*60/1000


print('dust')

Scenario_1_sum_dust_low <- sum(scenario_1_dust_low)*5*60/1000

Scenario_2_sum_dust_low <- sum(scenario_2_dust_low)*5*60/1000

Scenario_3_sum_dust_low <- sum(scenario_3_dust_low)*5*60/1000
```

```
Scenario_4_sum_dust_low <- sum(scenario_4_dust_low)*5*60/1000

Scenario_5_sum_dust_low <- sum(scenario_5_dust_low)*5*60/1000

Scenario_6_sum_dust_low <- sum(scenario_6_dust_low)*5*60/1000

#######################################################boxplot


##################################




outflux_vector <- c()

grass_porosity_grid <- rep(c(0,0,0.2,0,0),20)

shrub_quantitygrid <- rep(c(0,0,0,0,0,0,15.6,0,0,0,0),10)

shrub_porositygrid <- rep(c(0.7),100)

shrub_heightgrid <- rep(c(0.5),100)

tree_quantitygrid <- rep(c(36.5,0,0,0,0,0,0,0,0,0),10)

tree_breidthgrid <- rep(c(3),100)

tree_porositygrid <- rep(c(0.6),100)

tree_heightgrid<- rep(c(3.5),100)




Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <- The_model(drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count
],
Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_6_high <- outflux_vector

#plot(wind_vel_vector,Scenario_6,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(alpha=20,drag_coeff = 0.0050,Temp=30,mean_grainsize = 0.1,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_6_dust_high <- dust_vector

#########################

outflux_vector <- c()

grass_porosity_grid <- rep(c(0.0,0.0),50)

shrub_quantitygrid <- rep(c(0),100)

shrub_porositygrid <- rep(c(0.0),100)

shrub_heightgrid <- rep(c(0),100)

tree_quantitygrid <- c(rep(c(3),100))

tree_breidthgrid <- rep(c(0.7),100)

tree_porositygrid <- rep(c(0.6),100)

tree_heightgrid<- rep(c(0.8),100)


Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <- The_model(drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count
],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
```

```
    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    erosiongrid[Time_step,count] <- production+transport


    count <- count-1
  }
  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
  outflux_vector <- c(outflux_vector,outflux)
}
Scenario_5_high <- outflux_vector
dust_vector <- c()
count <- 0
for (wind_vel in wind_vel_vector) {
  count <- count+1
  Dust <- dustfunc(alpha=20,drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),

Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))
  dust_vector <- c(dust_vector,Dust)
}
scenario_5_dust_high <- dust_vector
#plot(wind_vel_vector,Scenario_5,type='l')
```

```
#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')

#####################################

outflux_vector <- c()

grass_porosity_grid <- rep(c(0,0.2,0,0),25)

shrub_quantitygrid <- rep(c(4,0,0,0),25)

shrub_porositygrid <- rep(c(0.8),100)

shrub_heightgrid <- rep(c(1),100)

tree_quantitygrid <- rep(c(0),100)

tree_breidthgrid <- rep(c(0.0),100)

tree_porositygrid <- rep(c(0.0),100)

tree_heightgrid<- rep(c(0.0),100)

Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <- The_model(drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count
],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])

    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_4_high <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(alpha=20,drag_coeff = 0.0050,Temp=30,mean_grainsize = 0.1,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_porosity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygrid),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_4_dust_high <- dust_vector

#plot(wind_vel_vector,Scenario_4,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab = 'distance from mode output (10^1 m)')


################################
```

```
outflux_vector <- c()

grass_porosity_grid <- rep(c(0.0,0.0),50)

shrub_quantitygrid <- rep(c(1.6),100)

shrub_porositygrid <- rep(c(0.8),100)

shrub_heightgrid <- rep(c(0.8),100)

tree_quantitygrid <- rep(c(0),100)

tree_breidthgrid <- rep(c(0.0),100)

tree_porositygrid <- rep(c(0.0),100)

tree_heightgrid<- rep(c(0.0),100)


Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

  Time_step <- Time_step+1

  count <- length(grass_porosity_grid)

  for (grass_porosity in grass_porosity_grid){

    production <- The_model(drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count
],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

    vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])
```

```
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

    erosiongrid[Time_step,count] <- production+transport



    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_3_high <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(alpha=20,drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_poro
sity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_3_dust_high <- dust_vector

#plot(wind_vel_vector,Scenario_3,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')



################################



outflux_vector <- c()
```

```
grass_porosity_grid <- rep(c(0.2,0.0,0.0,0.0),25)
shrub_quantitygrid <- rep(c(0.0),100)
shrub_porositygrid <- rep(c(0.0),100)
shrub_heightgrid <- rep(c(0.0),100)
tree_quantitygrid <- rep(c(0),100)
tree_breidthgrid <- rep(c(0.0),100)
tree_porositygrid <- rep(c(0.0),100)
tree_heightgrid<- rep(c(0.0),100)


Time_step <- 1
erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data =0)
for (wind_vel in wind_vel_vector) {
  Time_step <- Time_step+1
  count <- length(grass_porosity_grid)
  for (grass_porosity in grass_porosity_grid){
    production <- The_model(drag_coeff = 0.0050,Temp=30,mean_grainsize = 0.1,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    vegdens <- veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count],

Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[count],tree_height=tree_heightgrid[count])
    transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))
    erosiongrid[Time_step,count] <- production+transport
    count <- count-1

  }

  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))

  outflux_vector <- c(outflux_vector,outflux)

}

Scenario_2_high <- outflux_vector

dust_vector <- c()

count <- 0

for (wind_vel in wind_vel_vector) {

  count <- count+1

  Dust <- dustfunc(alpha=20,drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_poro
sity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),


Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))

  dust_vector <- c(dust_vector,Dust)

}

scenario_2_dust_high <- dust_vector

#plot(wind_vel_vector,Scenario_2,type='l')

#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')


################################


outflux_vector <- c()

grass_porosity_grid <- rep(c(0.0,0.0),50)

shrub_quantitygrid <- rep(c(0.0),100)
```

```
shrub_porositygrid <- rep(c(0.0),100)

shrub_heightgrid <- rep(c(0.0),100)

tree_quantitygrid <- rep(c(0),100)

tree_breidthgrid <- rep(c(0.0),100)

tree_porositygrid <- rep(c(0.0),100)

tree_heightgrid<- rep(c(0.0),100)


Time_step <- 1

erosiongrid = matrix(, nrow = length(wind_vel_vector)+1, ncol = length(grass_porosity_grid)+1,data
=0)

for (wind_vel in wind_vel_vector) {

 Time_step <- Time_step+1

 count <- length(grass_porosity_grid)

 for (grass_porosity in grass_porosity_grid){

   production <- The_model(drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_quantitygrid[count
],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

   vegdens <-
veg_density_func(grass_porosity=grass_porosity,wind_vel_vector=wind_vel,tree_quantity=tree_qua
ntitygrid[count],


Shrub_quantity=shrub_quantitygrid[count],shrub_porosity=shrub_porositygrid[count],shrub_height
=shrub_heightgrid[count],tree_breidth=tree_breidthgrid[count],tree_porosity=tree_porositygrid[cou
nt],tree_height=tree_heightgrid[count])

   transport <- erosiongrid[Time_step,count+1]*(1-(0.2+vegdens))

   erosiongrid[Time_step,count] <- production+transport



   count <- count-1
```

```
  }
  outflux <- erosiongrid[Time_step,1]*(1-(0.2+vegdens))
  outflux_vector <- c(outflux_vector,outflux)
}
Scenario_1_high <- outflux_vector
dust_vector <- c()
count <- 0
for (wind_vel in wind_vel_vector) {
  count <- count+1
  Dust <- dustfunc(alpha=20,drag_coeff = 0.0050,Temp=30,mean_grainsize =
0.1,type_of_model='kawamura',production=outflux_vector[count],grass_porosity=mean(grass_poro
sity_grid),wind_vel_vector=wind_vel,tree_quantity=mean(tree_quantitygrid),

Shrub_quantity=mean(shrub_quantitygrid),shrub_porosity=mean(shrub_porositygrid),shrub_height=
mean(shrub_heightgrid),tree_breidth=mean(tree_breidthgrid),tree_porosity=mean(tree_porositygri
d),tree_height=mean(tree_heightgrid))
  dust_vector <- c(dust_vector,Dust)
}
scenario_1_dust_high <- dust_vector
#plot(wind_vel_vector,Scenario_1,type='l')
#plot(erosiongrid[100,],main='transport for rows of vegetation ',ylab = 'transport  g/m/s',xlab =
'distance from mode output (10^1 m)')


#################################




#plot(wind_vel_vector,Scenario_1,type='l')


Scenario_1_sum_high <- sum(Scenario_1_high)*5*60/1000
```

```
Scenario_2_sum_high <- sum(Scenario_2_high)*5*60/1000

Scenario_3_sum_high <- sum(Scenario_3_high)*5*60/1000

Scenario_4_sum_high <- sum(Scenario_4_high)*5*60/1000

Scenario_5_sum_high <- sum(Scenario_5_high)*5*60/1000

Scenario_6_sum_high <- sum(Scenario_6_high)*5*60/1000




print('dust')

Scenario_1_sum_dust_high <- sum(scenario_1_dust_high)*5*60/1000

Scenario_2_sum_dust_high <- sum(scenario_2_dust_high)*5*60/1000

Scenario_3_sum_dust_high <- sum(scenario_3_dust_high)*5*60/1000

Scenario_4_sum_dust_high <- sum(scenario_4_dust_high)*5*60/1000

Scenario_5_sum_dust_high <- sum(scenario_5_dust_high)*5*60/1000

Scenario_6_sum_dust_high <- sum(scenario_6_dust_high)*5*60/1000



Scenario_1box <- c(Scenario_1_sum_high,Scenario_1_sum,Scenario_1_sum_low)

Scenario_2box <- c(Scenario_2_sum_high,Scenario_2_sum,Scenario_2_sum_low)

Scenario_3box <- c(Scenario_3_sum_high,Scenario_3_sum,Scenario_3_sum_low)

Scenario_4box <- c(Scenario_4_sum_high,Scenario_4_sum,Scenario_4_sum_low)

Scenario_5box <- c(Scenario_5_sum_high,Scenario_5_sum,Scenario_5_sum_low)

Scenario_6box <- c(Scenario_6_sum_high,Scenario_6_sum,Scenario_6_sum_low)


boxy <-
cbind(Scenario_1box,Scenario_2box,Scenario_3box,Scenario_4box,Scenario_5box,Scenario_6box)

boxplot(boxy,,main='Cumulative horizontal sediment flux, class 1 storm ',ylab='Sediment flux (kg/m)',col=c('darkgray','dimgray','gold','goldenrod4','darkolivegreen3','darkgreen',title='Total'),names=c('bare','bare- & grasses','shrub-','shrub+ & grasses','tree-','tree+& shrub+&grass'))
```

```
Scenario_1box_dust <-
c(Scenario_1_sum_dust_high,Scenario_1_sum_dust,Scenario_1_sum_dust_low)

Scenario_2box_dust <-
c(Scenario_2_sum_dust_high,Scenario_2_sum_dust,Scenario_2_sum_dust_low)

Scenario_3box_dust <-
c(Scenario_3_sum_dust_high,Scenario_3_sum_dust,Scenario_3_sum_dust_low)

Scenario_4box_dust <-
c(Scenario_4_sum_dust_high,Scenario_4_sum_dust,Scenario_4_sum_dust_low)

Scenario_5box_dust <-
c(Scenario_5_sum_dust_high,Scenario_5_sum_dust,Scenario_5_sum_dust_low)

Scenario_6box_dust <-
c(Scenario_6_sum_dust_high,Scenario_6_sum_dust,Scenario_6_sum_dust_low)


boxy_dust <-
cbind(Scenario_1box_dust,Scenario_2box_dust,Scenario_3box_dust,Scenario_4box_dust,Scenario_5
box_dust,Scenario_6box_dust)

boxplot(boxy_dust,main='Cumulative vertical sediment flux, class 1 storm ',ylab='Erosion Suspension
(kg/m^2)',col=c('darkgray','dimgray','gold','goldenrod4','darkolivegreen3','darkgreen',title='Total'),na
mes=c('bare','bare- & grasses','shrub-','shrub+ & grasses','tree-','tree+& shrub+&grass'))


print('Erosion')

print(Scenario_1_sum)

print(Scenario_2_sum)

print(Scenario_3_sum)

print(Scenario_4_sum)

print(Scenario_5_sum)

print(Scenario_6_sum)
print('Suspension')

print(Scenario_1_sum_dust)

print(Scenario_2_sum_dust)

print(Scenario_3_sum_dust)

print(Scenario_4_sum_dust)

print(Scenario_5_sum_dust)

print(Scenario_6_sum_dust)
```