Predicting forced migration by extreme weather patterns in Sudan and South Sudan

Environment and Climate Crisis

Michael Albert

153400150

AS1

23 April 2021

Word count: 2,187

Introduction

The relationship between climate change and migration has been contested among scientists, influencing how policymakers may respond. The predictions by climate migration researchers are distributed on a large spectrum, from forecasting billions of climate refugees[1] to saying there is no causal relationship between climate and large-scale migration.[2] There are several debates about the multi-causal dynamics that drive migration, the accuracy and viability of long-term predictions, as well as the overall concern of whether such predictions do more harm than help solving the issue.

In this paper, I will seek to unify some of the concerns of critics with the chances offered by the ability to predict forced migration, not least in the response to humanitarian crises. Machine learning methods will be used to make inferences about correlations between extreme weather patterns and forced migration, while controlling for other confounders such as human conflict. Building on highly detailed survey data from the *International Organisation of Migration*, I will be able to demonstrate a causal relationship between extreme weather events and migration.

My analysis will avoid controversial assumptions of existing quantitative analysis: the claim of accuracy over long-term predictions, the dependency on correlations, and generalisations over many states and continents. Instead, my model will employ a local case study—the Sudan and South Sudan—and making inferences focusing on the quality of data and robustness of statistical tools used. This localised approach will add a piece to the big puzzle of predicting climate migration which is yet to be solved by climate scientists and statisticians.

Literature Review

Over the past thirty years, reports have suggested a wide range of possible climate migration scenarios. In their influential work *Environmental Exodus*[3], biodiversity specialists Myers and Kent predicted 200 million climate refugees by 2050, while a study

[1] Christian Aid, 'Human Tide: The Real Migration Crisis' (London: Christian Aid, May 2007), https://www.christianaid.org.uk/sites/default/files/2017-08/human-tide-the-real-migration-crisis-may-2007.pdf.
[2] Gunvor Jónsson, 'The Environmental Factor in Migration Dynamics: A Review of African Case Studies', *International Migration Institute*, no. 21 (2010): 1–34.
[3] Norman Myers and Jennifer Kent, *Environmental Exodus: An Emergent Crisis in the Global Arena* (Washington D.C.: Climate Institute, 1995).

from 2007 by a British NGO put the figure as high as 1 billion by the same date.[4] Since then, statistical methods to estimate the link between climate and migration have become more sophisticated and robust. In a recently published paper, Chen and Caldeira presented an attempt to simulate the future incentives for migration. [5] Using weather as well as socio-economic data, they inferred the attractiveness of remaining in one's home country versus migrating. Their results suggest that by 2050, 0.6 to 1.9 billion people may face serious incentives to migrate due to shifting weather patterns.[6]

The methodology and discursive implications of these quantitative analyses to predict future migration has been subject to critique. Methodologically, scholars have questioned whether there was, in fact, a (strong) causal link between climate change and migration. Some argued that the reasons for migrating are more complex than they are assumed to be in the quantitative literature.[7] In other words, the real world contains more independent variables and interaction factors that affect migration.

> "It is instead influenced by a mix of climatic, socio-economic, cultural and political factors. Even when climate change does play a role, it remains difficult to determine the extent of its influence."[8]

Other scholars state that climate change is a "slow-onset phenomenon, which gives people time to adapt to resulting environmental stresses"[9], and in cases of floods or other sudden crises, "the vast majority of people move over short distances, such as to the next neighborhood, village or town"[10]. Furthermore, due to humans' adaptation strategies, e.g., flood defence systems, future migration is more uncertain to predict.

With respect to the discursive effects of predicting large migration patterns, scholars have criticised that it may be misused by political actors to securitise migration. Securitisation refers to the strategy of framing something (e.g., migrants) as a security

[4] Christian Aid, 'Human Tide'.
[5] Min Chen and Ken Caldeira, 'Climate Change as an Incentive for Future Human Migration', *Earth System Dynamics* 11, no. 4 (22 October 2020): 875–83, https://doi.org/10.5194/esd-11-875-2020.
[6] Ibid., 878.
[7] Foresight, 'Migration and Global Environmental Change: Future Challenges and Opportunities' (London: UK Government Office for Science, 2011), https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/287717/11-1116-migration-and-global-environmental-change.pdf.
[8] Ingrid Boas et al., 'Climate Migration Myths', *Nature Climate Change* 9 (December 2019): 902.
[9] Hein de Haas, 'Climate Refugees: The Fabrication of a Migration Threat', *Hein de Haas* (blog), 31 January 2020, https://heindehaas.blogspot.com/2020/01/climate-refugees-fabrication-of.html.
[10] Ibid.

threat (e.g., as a "human tide"[11]) to some object (e.g., domestic security, national identity) in order to justify extreme countermeasures (e.g., militarisation of borders).[12]

Indeed, scientists ought to be careful with their predictions when informing policymakers and the public. However, simply not pursuing quantitative analysis because its result may be appropriated by actors on the other end of the political spectrum would be to bury one's head in the sand. Furthermore, quantitative analysis enables statistical inference, which is needed to address the root causes of forced migration and rising inequality levels and can help to respond quickly to climate disasters. Yet, to do this, we must overcome the conceptual difficulties, particularly related to causality.

Analysis

This paper aims to provide a quantitative framework that avoids the pitfalls described above by being localised, inclusive of cofounders, and grounded in data that enables *causal* inference. It uses high-quality survey data from migrants in the Sudan and South Sudan, collected by the International Organisation for Migration from January 2020 to January 2021.[13] The dataset includes the places and times of departure and destination, mode of transport, reasons for migration and 41 more survey response variables. Its 91,420 observations (i.e., individual surveys) of 550,530 individuals were collected in 50 different places in the region. The data will be aggregated for each of the 268 towns or small regions in Sudan and South Sudan and paired with weather data for each respective town from 2009 to 2021 that is pulled from the API of World Weather Online.[14] After adding demographic[15], socio-economic and conflict data[16] to account for any non-

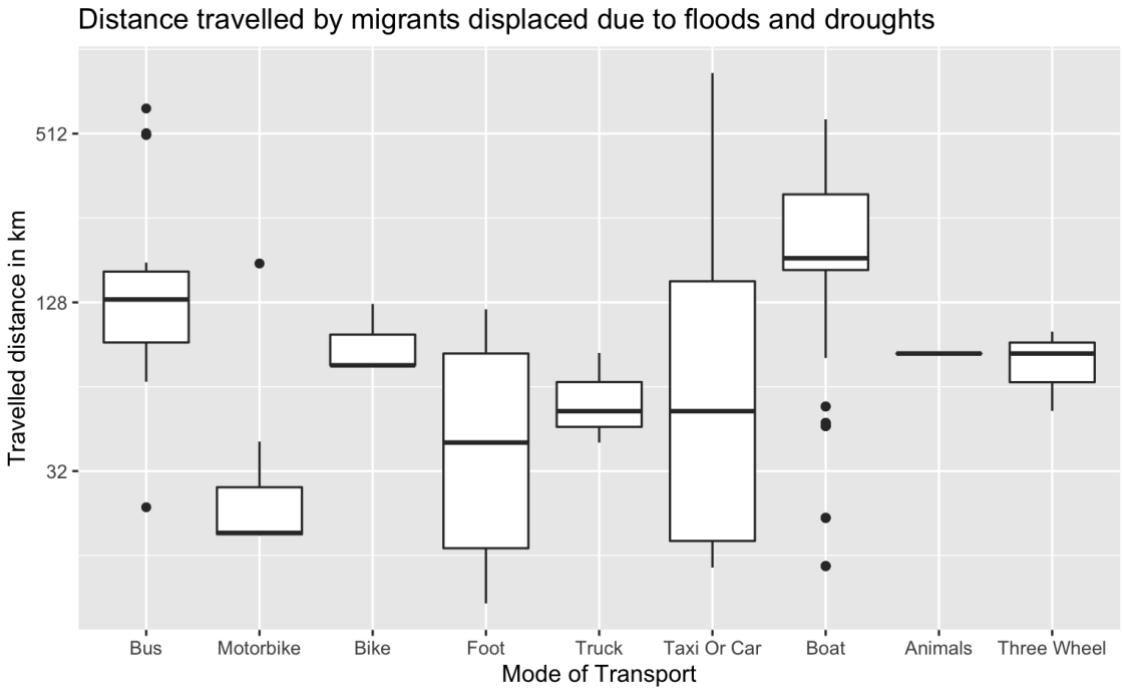
[11] Christian Aid, 'Human Tide'.
[12] Barry Buzan, Ole Waever, and Jaap de Wilde, *Security: A New Framework for Analysis* (Boulder, Colo.: Lynne Rienner, 1998).
[13] IOM (International Organization for Migration), 'South Sudan Population Movement - [Migrants] - [IOM DTM]', *The Humanitarian Data Exchange*, 2021, https://data.humdata.org/dataset/dtm-south-sudan-flow-monitoring-migration-population-movement-iom.
[14] World Weather Online, 'Local City and Town Weather API', *For Developers*, accessed 1 April 2021, https://www.worldweatheronline.com/developer/api/local-city-town-weather-api.aspx#monthly_averages.
[15] IOM (International Organization for Migration), 'South Sudan - Subnational Population Statistics', *The Humanitarian Data Exchange*, 2020, https://data.humdata.org/dataset/dtm-south-sudan-flow-monitoring-migration-population-movement-iom.
[16] Therese Pettersson and Magnus Öberg, 'Organized Violence, 1989-2019', *Journal of Peace Research* 57, no. 4 (2020); Ralph Sundberg and Erik Melander, 'Introducing the UCDP Georeferenced Event Dataset', *Journal of Peace Research* 50, no. 4 (2013).

weather related migration and filtering for individuals who responded that they were "forced migrants", we have one observation for each town and each month from January 2020 to January 2021.

In 2020, Sudan and South Sudan experienced extreme drought and floods. While the beginning of the year had been dry, the last third of the year saw strong precipitation which led to floods and the highest water level of the Nile in nearly a century. Affecting at least 1.6 million people in both states, the flood devastated residential and agricultural lands.[17] In many areas, the water has not yet run off eight months after the rain has started, causing food shortages and hunger.

Using this detailed data, we can assess some of the claims made by scholars. Through a simple data categorisation (**Figure 1**), we observe that displaced individuals travel further than simply the next neighbourhood or village, as argued by de Haas.[18] The average distance travelled is 113 kilometres or 70 miles.

Figure 1:



[17] Susan Martinez, 'Drowned Land: Hunger Stalks South Sudan's Flooded Villages', *The Guardian*, 19 March 2021, https://www.theguardian.com/global-development/2021/mar/19/drowned-land-hunger-stalks-south-sudans-flooded-villages.
[18] de Haas, 'Climate Refugees'.

Examining the migration data for exemplary regions, we can see a clear correlation between the relative precipitation, that is, the daily average of rain in 2020 subtracted by the 10-year monthly average from 2009 to 2019, and the number of forced migrations per million people. **Figures 2a and 2b** show this relationship taking as an example two towns in South Sudan: Bor and Canal/Pigi.

Figure 2a: Precipitation compared to 10-year average and migration in Bor, South Sudan

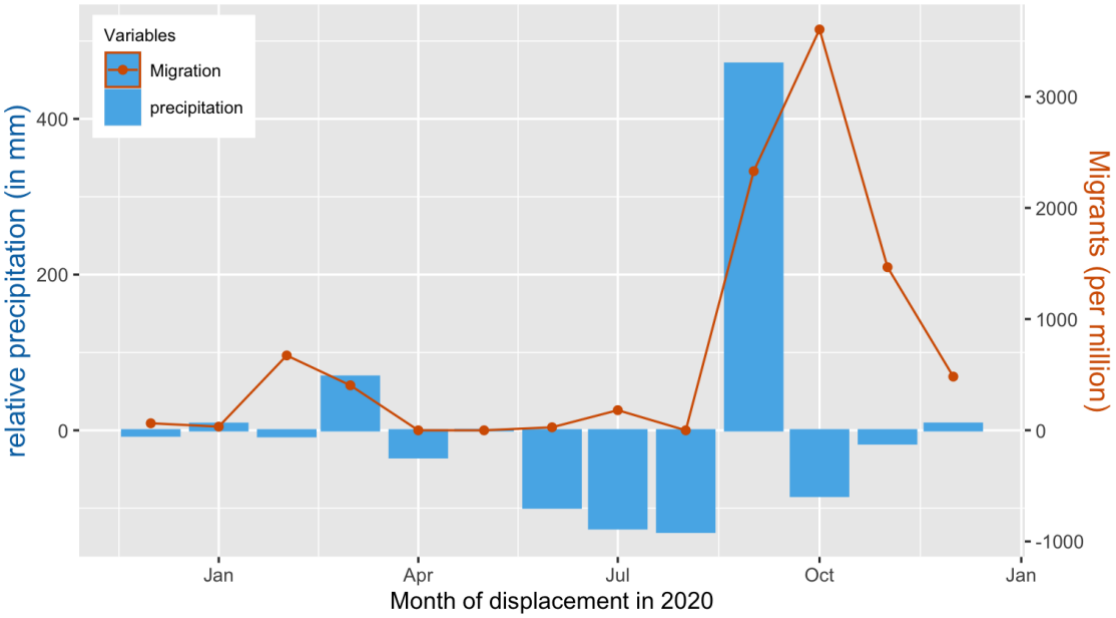
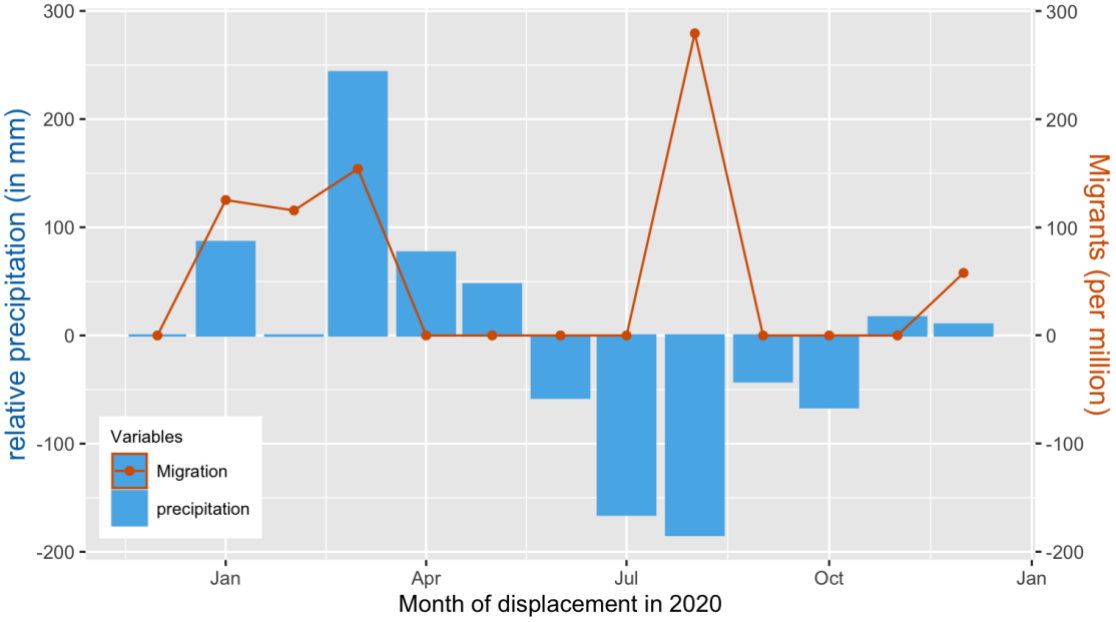


Figure 2b: Precipitation compared to 10-year average and migration in Canal and Pigi, South Sudan



The graphs show a clear correlation between a large diversion from the historical monthly average and an increase in forced migration. Bor, which suffered under an unprecedented flood in September, showed an uptick in displacement in the same and subsequent month. On the other hand, Canal/Pigi had a drought from June to August, which also correlated with an increase in migration. In the survey, the affected individuals said that they had fled because of natural disasters in 82% of cases, food insecurity in 17% of cases—particularly during the drought in Canal/Pigi in August—and conflict violence in 2% (only in January). This proves that there is indeed a causal relationship between the extreme weather patterns and the number of displaced people.

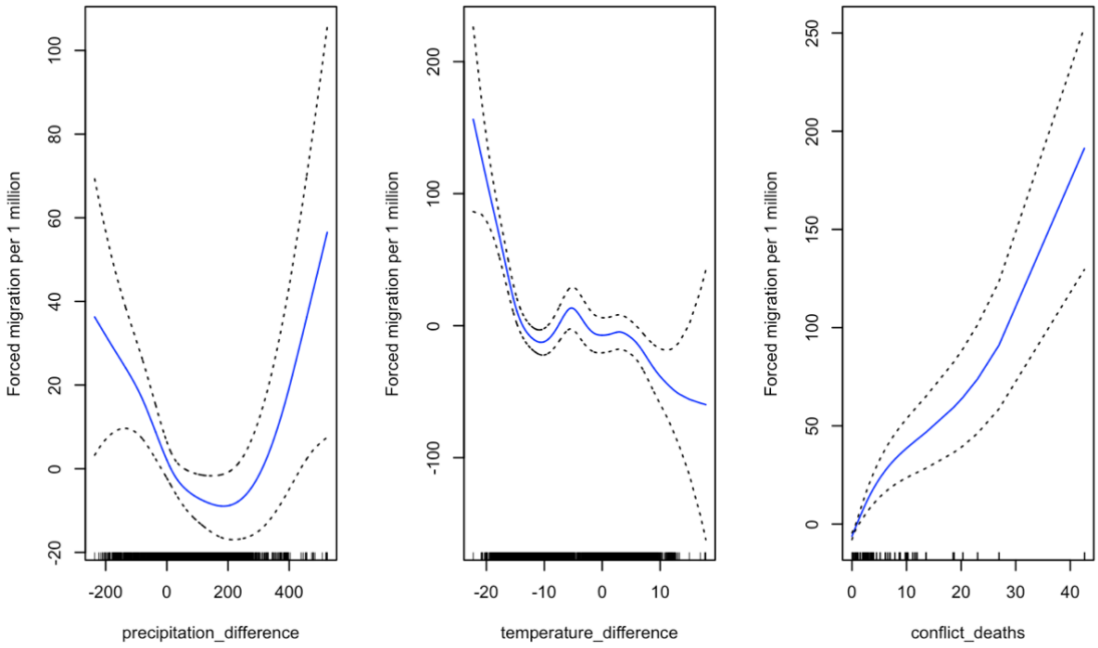
Having established this causality, we can now approximate the causal relationship between weather and displacement by using machine learning. Using a General Additive Model (GAM), we will fit a model onto our original dataset containing migration data from all 268 towns and regions for each month. The algorithm will fit a model of the form

$$forced\ mig._i = \beta_0 + f_1(precipitation_i) + f_2(temperature_i) + f_3(conflict\ deaths) + \epsilon_i$$

where f_1, f_2, and f_3 are non-linear functions of each independent variable and β_0 is an intercept term. A GAM is well-suited for this statistical problem, as it maintains additivity, meaning that predictor values are adding onto each other to produce an estimated result. While being a highly parameterised machine learning model, it is still very interpretable, as we can distinguish the individual effects of each variable.[19]

[19] Gareth James et al., eds., *An Introduction to Statistical Learning: With Applications in R*, Springer Texts in Statistics 103 (New York: Springer, 2013), 282.

Figure 3: General Additive Model

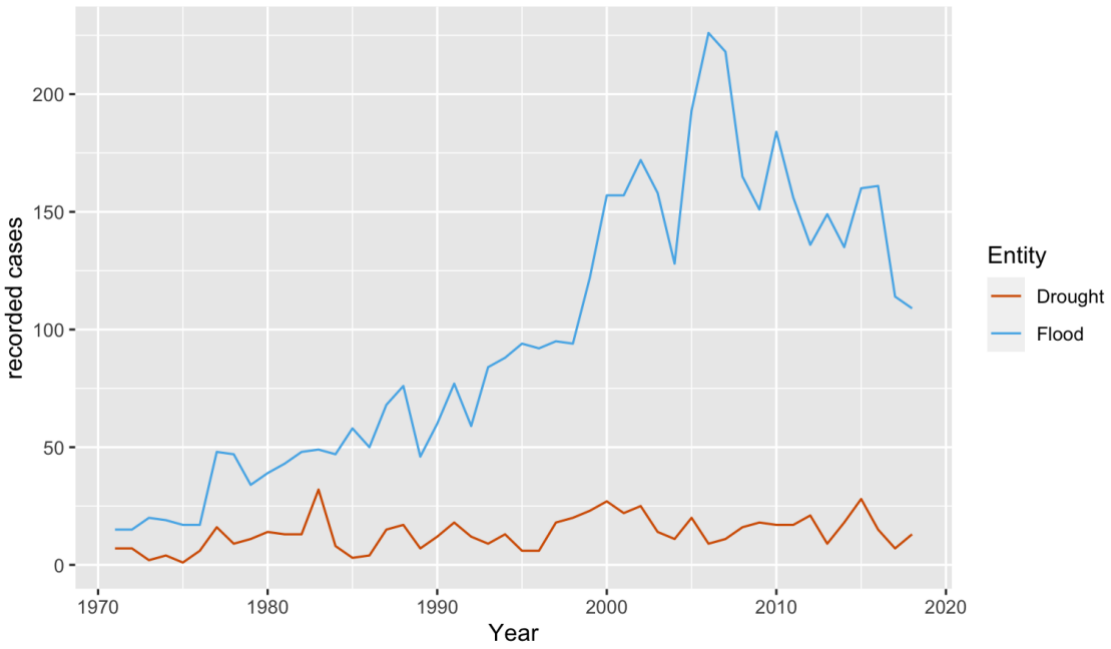


Each plot in **Figure 3** keeps the other variables constant to delineate the effect of the respective variable plotted in blue. The dotted lines are the 95% confidence interval. Relative precipitation per day in mm per square metre (left) shows a strong relationship with forced migration. Particularly when precipitation is lower than in the 10-year average, suggesting a drought and when it is far higher, resulting in floods. The maximum temperature in Celsius compared to the 10-year average (middle) shows that migration is higher when temperatures are more than 15 degrees lower than usual. This is likely due to decreased temperatures during long rainfalls and floods. The per capita number of conflict deaths in the respective areas is unsurprisingly positively related to migration in that same area. The conflict variable is included to balance the model for the severity of conflict in that region. The overall model performs well against benchmark tests and all variables have significant p-values (<0.001).

Purposely, this model does not include a time variable, hence, it is not possible to make predictions about the future of migration from Sudan and South Sudan. Current machine learning algorithms are not able to make reasonable predictions of life

outcomes of individuals in the distant future.[20] Nevertheless, by using time series data (panel data) collectively[21], we were able to observe fixed effects which suggest that there is indeed a strong causal link between extreme weather patterns and migration. Extreme droughts and floods are becoming more frequent worldwide (**Figure 4**).

Figure 4: Number of droughts and floods worldwide



Sudan is particularly vulnerable to climate change as scientists have observed a "trend of decreasing annual rainfall in the last 60 years and increased rainfall variability"[22], causing more droughts and floods at the same time. Connecting this trend with the relationship between extremely low and extremely high precipitation (see left graph on **Figure 3**), we can conclude that Sudan and South Sudan are likely going to see more waves of forced migration in the future.

[20] Matthew J. Salganik et al., 'Measuring the Predictability of Life Outcomes with a Scientific Mass Collaboration', *Proceedings of the National Academy of Sciences* 117, no. 15 (14 April 2020): 8398–8403, https://doi.org/10.1073/pnas.1915006117.
[21] Scott Cunningham, *Causal Inference: The Mixtape* (New Haven: Yale University Press, 2021), 386.
[22] Mutasim Bashir Nimir and Ismail A. Elgizouli, 'Climate Change Adaptation and Decision Making in the Sudan', *World Resources Report*, 2021, https://www.wri.org/our-work/project/world-resources-report/climate-change-adaptation-and-decision-making-sudan.

Conclusion

Statistical inference and prediction require special attention to causality. Correlations must be underpinned by qualitative analysis and rigorous causal inference to qualify as causality. I have sought to provide this evidence, firstly, by using survey data and two model regions to prove that people are indeed fleeing their homes due to natural disasters. Secondly, by using time series panel data that is particularly capable of keeping variables fixed, I show causal relationships between one predictor and migration *ceteris paribus*.

My quantitative analysis concludes that there is indeed a causal relationship between extreme weather patterns and forced migration (over a long distance), in line with the predictions by many climate migration scholars, but contrary to what some critical scholars had suggested. Given the increased variability of precipitation and more extreme weather patterns in Sudan and South Sudan, it becomes clear how—in this particular case study—climate change is (and will be) forcing people to leave their homes.

Bibliography

Boas, Ingrid, Carol Farbotko, Helen Adams, Harald Sterly, Simon Bush, Kees van der Geest, and Hanne Wiegel. 'Climate Migration Myths'. *Nature Climate Change* 9 (December 2019): 898–903.

Buzan, Barry, Ole Waever, and Jaap de Wilde. *Security: A New Framework for Analysis*. Boulder, Colo.: Lynne Rienner, 1998.

Chen, Min, and Ken Caldeira. 'Climate Change as an Incentive for Future Human Migration'. *Earth System Dynamics* 11, no. 4 (22 October 2020): 875–83. https://doi.org/10.5194/esd-11-875-2020.

Christian Aid. 'Human Tide: The Real Migration Crisis'. London: Christian Aid, May 2007. https://www.christianaid.org.uk/sites/default/files/2017-08/human-tide-the-real-migration-crisis-may-2007.pdf.

Cunningham, Scott. *Causal Inference: The Mixtape*. New Haven: Yale University Press, 2021.

Foresight. 'Migration and Global Environmental Change: Future Challenges and Opportunities'. London: UK Government Office for Science, 2011. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/287717/11-1116-migration-and-global-environmental-change.pdf.

Haas, Hein de. 'Climate Refugees: The Fabrication of a Migration Threat'. *Hein de Haas* (blog), 31 January 2020. https://heindehaas.blogspot.com/2020/01/climate-refugees-fabrication-of.html.

IOM (International Organization for Migration). 'South Sudan - Subnational Population Statistics'. *The Humanitarian Data Exchange*, 2020. https://data.humdata.org/dataset/dtm-south-sudan-flow-monitoring-migration-population-movement-iom.

———. 'South Sudan Population Movement - [Migrants] - [IOM DTM]'. *The Humanitarian Data Exchange*, 2021. https://data.humdata.org/dataset/dtm-south-sudan-flow-monitoring-migration-population-movement-iom.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani, eds. *An Introduction to Statistical Learning: With Applications in R*. Springer Texts in Statistics 103. New York: Springer, 2013.

Jónsson, Gunvor. 'The Environmental Factor in Migration Dynamics: A Review of African Case Studies'. *International Migration Institute*, no. 21 (2010): 1–34.

Martinez, Susan. 'Drowned Land: Hunger Stalks South Sudan's Flooded Villages'. *The Guardian*. 19 March 2021. https://www.theguardian.com/global-development/2021/mar/19/drowned-land-hunger-stalks-south-sudans-flooded-villages.

Myers, Norman, and Jennifer Kent. *Environmental Exodus: An Emergent Crisis in the Global Arena*. Washington D.C.: Climate Institute, 1995.

Nimir, Mutasim Bashir, and Ismail A. Elgizouli. 'Climate Change Adaptation and Decision Making in the Sudan'. *World Resources Report*, 2021. https://www.wri.org/our-work/project/world-resources-report/climate-change-adaptation-and-decision-making-sudan.

Pettersson, Therese, and Magnus Öberg. 'Organized Violence, 1989-2019'. *Journal of Peace Research* 57, no. 4 (2020).

Salganik, Matthew J., Ian Lundberg, Alexander T. Kindel, Caitlin E. Ahearn, Khaled Al-Ghoneim, Abdullah Almaatouq, Drew M. Altschul, et al. 'Measuring the Predictability of Life Outcomes with a Scientific Mass Collaboration'. *Proceedings of the National Academy of Sciences* 117, no. 15 (14 April 2020): 8398–8403. https://doi.org/10.1073/pnas.1915006117.

Sundberg, Ralph, and Erik Melander. 'Introducing the UCDP Georeferenced Event Dataset'. *Journal of Peace Research* 50, no. 4 (2013).

World Weather Online. 'Local City and Town Weather API'. *For Developers*. Accessed 1 April 2021. https://www.worldweatheronline.com/developer/api/local-city-town-weather-api.aspx#monthly_averages.

Appendix

In the following, I have provided the code for the statistical programming language **R** which has been used to collect, clean, analyse, and plot data. It is divided into Cleaning code, code used to download historical weather data from APIs, and the data analysis including the machine learning models and graphs. The sources for the compiled datasets are provided in the paper above.

Data Cleaning

Sebastian Dodt

21/03/2021

Data Cleaning

```
library(readxl)
library(zoo)
library(sjlabelled)
library(tidyverse)
require(jsonlite)
require(dplyr)
require(tidyr)
library(zoo)
library(ggplot2)
library(geosphere)
library(lubridate)
```

Importing the datasets from the International Organisation of Migration

```
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Machine Learning/Project/Datasets")
jan20 <- read_xlsx("dtm-south-sudan-flow-monitoring-jan20.xlsx")
feb20 <- read_xlsx("dtm-south-sudan-flow-monitoring-feb20.xlsx")
mar20 <- read_xlsx("dtm-south-sudan-flow-monitoring-mar20.xlsx")
apr20 <- read_xlsx("dtm-south-sudan-flow-monitoring-apr20.xlsx")
may20 <- read_xlsx("dtm-south-sudan-flow-monitoring-may20.xlsx")
jun20 <- read_xlsx("dtm-south-sudan-flow-monitoring-jun20.xlsx")
jul20 <- read_xlsx("dtm-south-sudan-flow-monitoring-jul20.xlsx")
sep20 <- read_xlsx("dtm-south-sudan-flow-monitoring-sep20.xlsx")
oct20 <- read_xlsx("dtm-south-sudan-flow-monitoring-oct20.xlsx")
nov20 <- read_xlsx("dtm-south-sudan-flow-monitoring-nov20.xlsx")
dec20 <- read_xlsx("dtm-south-sudan-flow-monitoring-dec20.xlsx")
jan21 <- read_xlsx("dtm-south-sudan-flow-monitoring-jan21.xlsx")
```

Deleting the first row

```
jan20 <- jan20[-1,]
feb20 <- feb20[-1,]
mar20 <- mar20[-1,]
apr20 <- apr20[-1,]
may20 <- may20[-1,]
jun20 <- jun20[-1,]
jul20 <- jul20[-1,]
```

```
sep20 <- sep20[-1,]
oct20 <- oct20[-1,]
nov20 <- nov20[-1,]
dec20 <- dec20[-1,]
jan21 <- jan21[-1,]
```

changing classes

```
col_with_factors <- c(2,5:29,40:42)
col_with_numeric <- c(1,3,4)
col_with_integer <- c(30:39,43:46)
for (i in col_with_factors) {
  jan20[,i] <- as_factor(unlist(jan20[,i]))
  feb20[,i] <- as_factor(unlist(feb20[,i]))
  mar20[,i] <- as_factor(unlist(mar20[,i]))
  apr20[,i] <- as_factor(unlist(apr20[,i]))
  may20[,i] <- as_factor(unlist(may20[,i]))
  jun20[,i] <- as_factor(unlist(jun20[,i]))
  jul20[,i] <- as_factor(unlist(jul20[,i]))
  sep20[,i] <- as_factor(unlist(sep20[,i]))
  oct20[,i] <- as_factor(unlist(oct20[,i]))
  nov20[,i] <- as_factor(unlist(nov20[,i]))
  dec20[,i] <- as_factor(unlist(dec20[,i]))
  jan21[,i] <- as_factor(unlist(jan21[,i]))
}
for (i in col_with_numeric) {
  jan20[,i] <- as.numeric(unlist(jan20[,i]))
  feb20[,i] <- as.numeric(unlist(feb20[,i]))
  mar20[,i] <- as.numeric(unlist(mar20[,i]))
  apr20[,i] <- as.numeric(unlist(apr20[,i]))
  may20[,i] <- as.numeric(unlist(may20[,i]))
  jun20[,i] <- as.numeric(unlist(jun20[,i]))
  jul20[,i] <- as.numeric(unlist(jul20[,i]))
  sep20[,i] <- as.numeric(unlist(sep20[,i]))
  oct20[,i] <- as.numeric(unlist(oct20[,i]))
  nov20[,i] <- as.numeric(unlist(nov20[,i]))
  dec20[,i] <- as.numeric(unlist(dec20[,i]))
  jan21[,i] <- as.numeric(unlist(jan21[,i]))
}
for (i in col_with_integer) {
  jan20[,i] <- as.integer(unlist(jan20[,i]))
  feb20[,i] <- as.integer(unlist(feb20[,i]))
  mar20[,i] <- as.integer(unlist(mar20[,i]))
  apr20[,i] <- as.integer(unlist(apr20[,i]))
  may20[,i] <- as.integer(unlist(may20[,i]))
  jun20[,i] <- as.integer(unlist(jun20[,i]))
  jul20[,i] <- as.integer(unlist(jul20[,i]))
  sep20[,i] <- as.integer(unlist(sep20[,i]))
  oct20[,i] <- as.integer(unlist(oct20[,i]))
  nov20[,i] <- as.integer(unlist(nov20[,i]))
  dec20[,i] <- as.integer(unlist(dec20[,i]))
  jan21[,i] <- as.integer(unlist(jan21[,i]))
}
```

```
jan20$interview.month <- as.Date(jan20$interview.month) - 25569
jan20$interview.month[c(11527,11526)] <- "2020-01-01"
feb20$interview.month <- as.Date(feb20$interview.month) - 25569
mar20$interview.month <- as.Date(mar20$interview.month) - 25569
apr20$interview.month <- as.Date(apr20$interview.month) - 25569
may20$interview.month <- as.Date(may20$interview.month) - 25569
jun20$interview.month <- as.Date(jun20$interview.month) - 25569
jul20$interview.month <- as.Date(jul20$interview.month) - 25569
sep20$interview.month <- as.Date(sep20$interview.month) - 25569
oct20$interview.month <- as.Date(oct20$interview.month) - 25569
nov20$interview.month <- as.Date(nov20$interview.month) - 25569
dec20$interview.month <- as.Date(dec20$interview.month) - 25569
jan21$interview.month <- as.Date(jan21$interview.month) - 25569
```

combining all data

```
dtm.alldata <- rbind(jan20,feb20,mar20,apr20,may20,jun20,jul20,sep20,oct20,nov20,dec20,jan21)
```

Overview of our data. The dataset contains 1,723 surveys with a total of 16,446 individuals.

```
#creating matrix with only forced migrants who are not coming from a refugee camp
forced.migrants <- dtm.alldata %>%
  filter(forced.displacement == "Yes") %>%
  filter(dep.iscamp=="No")
#number of observations
nrow(forced.migrants)
```

```
## [1] 1723
```

```
#number of individuals
sum(forced.migrants$total.ind)
```

```
## [1] 16446
```

```
#summary of all reasons for travel/displacement
reason_summary <- summary(dtm.alldata$reason)
#summary of reasons for displaced refugees
reason_summary_1 <- dtm.alldata %>%
  filter(forced.displacement == "Yes") %>%
  pull(reason) %>% summary()
#summary of reasons for economic migrants
reason_summary_2 <- dtm.alldata %>%
  filter(reason == "Economic") %>%
  pull(reason.subtype) %>% summary()
#summary of departure countries
dep_country_summary <- forced.migrants %>%
  pull(dep.country) %>%
  summary()
```

adding the geo-location to our dataset `dtm.alldata`.

```
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Machine Learning/Project/Datasets")
geodata_ss <- read_csv("southsudan_centroids_lonlat.csv")
geodata_ss$ADM2_PCODE <- as.factor(geodata_ss$ADM2_PCODE)
geodata_ss$lon <- as.numeric(geodata_ss$lon)
geodata_ss$lat <- as.numeric(geodata_ss$lat)
geodata_ss <- geodata_ss[,c(4,15,16)]
geodata_s <- read_csv("sudan_centroids_lonlat.csv")
geodata_s$lon <- as.numeric(geodata_s$lon)
geodata_s$lat <- as.numeric(geodata_s$lat)
geodata_s <- geodata_s[,c(5,20,21)]
geodata <- rbind(geodata_s,geodata_ss)
all_admin2 <- c(as.character(geodata_ss$ADM2_PCODE),geodata_s$ADM2_PCODE)
```

adding geodata to our dataset and exporting it

```
dep.lon <- vector(length = nrow(dtm.alldata))
dep.lat <- vector(length = nrow(dtm.alldata))
for (i in 1:nrow(dtm.alldata)) {
  row <- which(dtm.alldata$dep.adm2.pcode[i] == geodata_ss$ADM2_PCODE)
  row <- as.integer(row)
  dep.lon[i] <- as.numeric(geodata_ss[row,2])
  dep.lat[i] <- as.numeric(geodata_ss[row,3])
}
dtm.alldata.geo <- cbind(dtm.alldata,dep.lon,dep.lat)
forced.migrants.geo <- dtm.alldata.geo %>%
  filter(forced.displacement == "Yes") %>%
  filter(dep.iscamp=="No")
#export
#write.csv(dtm.alldata.geo,"all_travellers.csv")
#write.csv(forced.migrants.geo,"forced_migrants.csv")
```

Distance between departure location and place of survey.

```
distance_travelled <- vector(length = nrow(dtm.alldata.geo))
for (i in 1:nrow(dtm.alldata.geo)) {
  distance_travelled[i] <-
    distm(c(dtm.alldata.geo$fmp.longitude[i],dtm.alldata.geo$fmp.latitude[i]),
          c(dtm.alldata.geo$dep.lon[i],dtm.alldata.geo$dep.lat[i]),
          fun = distHaversine)
}
summary(distance_travelled)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    9325   25568   73795  113283  143442  911273   39968
```

```
dtm.alldata.geo <- cbind(dtm.alldata.geo, distance_travelled)
```

The average person travelled 113 km.

Mode of transports

```
summary(dtm.alldata.geo$transport)
```

```
##         Air         Bus   Motorbike        Bike        Foot       Truck
##        1137       17292       12032        7634       11354       15027
## Taxi Or Car        Boat     Animals Three Wheel Three-wheel    J. Train
##       23240        2187         341         618         552           4
##        NA's
##           2
```

Unify `Three-wheel` and `Three Wheel`

```
for (i in 1:nrow(dtm.alldata.geo)) {
  if (is.na(dtm.alldata.geo$transport[i]) == TRUE) dtm.alldata.geo$transport[i] <- NA else
    if (dtm.alldata.geo$transport[i] == "Three-wheel") dtm.alldata.geo$transport[i] <- "Three Wheel"
}
```

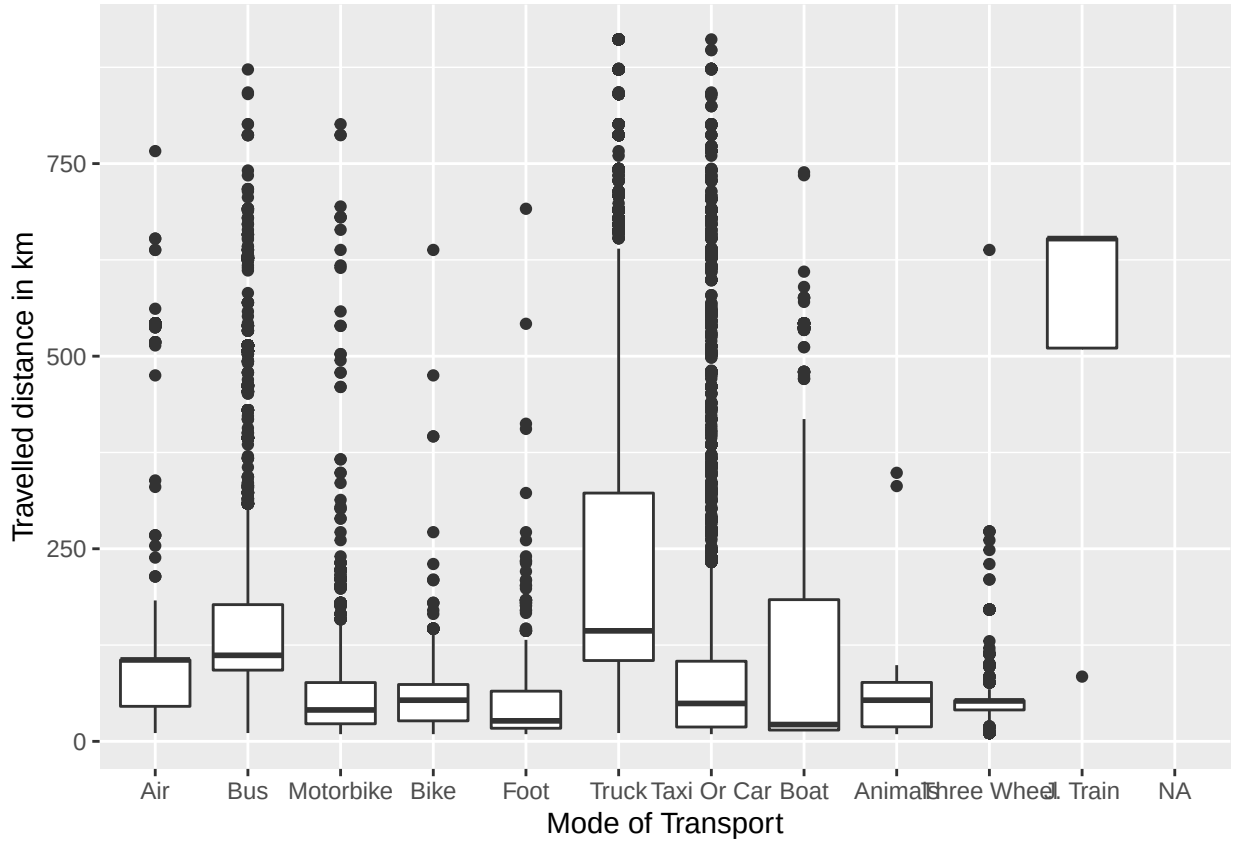
Estimating travelled distance with each mode of transport per day.

```
basic_unit <- 1 # e.g., 1 could stand for 1 km
air <- basic_unit * 300
bus <- basic_unit * 50
motorbike <- basic_unit * 50
bike <- basic_unit * 20
foot <- basic_unit * 5
truck <- basic_unit * 50
taxi.car <- basic_unit * 100
boat <- basic_unit * 25
animals <- basic_unit * 10
three.wheel <- basic_unit * 20
train <- basic_unit * 100
transport.na <- basic_unit * 20
```

Plotting the travelled distance for all forced migrants.

```
ggplot(dtm.alldata.geo, aes(x = transport, y = distance_travelled/1000)) +
  geom_boxplot() +
  ylab("Travelled distance in km") +
  xlab("Mode of Transport")
```

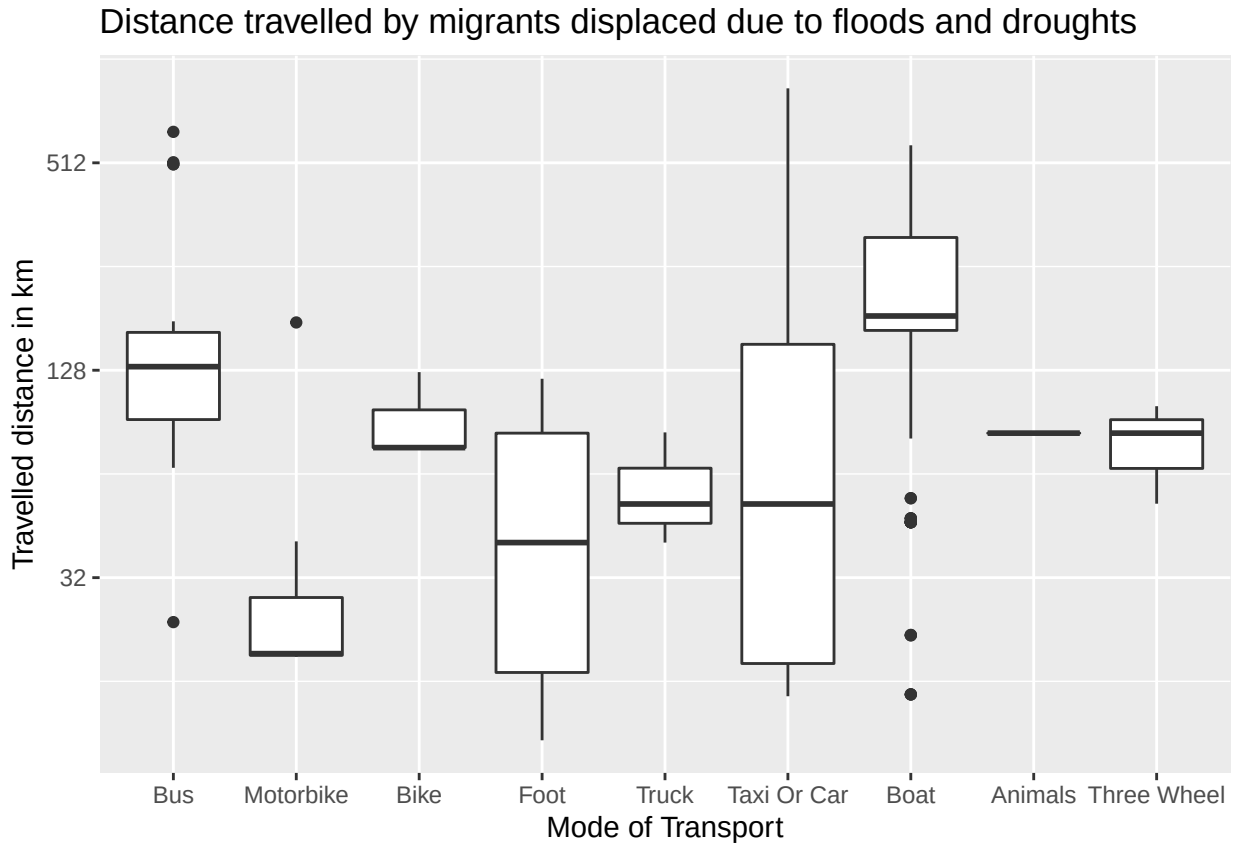
```
## Warning: Removed 39968 rows containing non-finite values (stat_boxplot).
```



Plotting travelled distance for all migrants who have fled from natural disasters, i.e., floods and droughts.

```
dtm.alldata.geo %>% filter(reason == "Disaster Displacement") %>%
  ggplot(aes(x = transport, y = distance_travelled/1000)) +
    geom_boxplot() +
    scale_y_continuous(trans = 'log2') +
    ylab("Travelled distance in km") +
    xlab("Mode of Transport") +
    ggtitle("Distance travelled by migrants displaced due to floods and droughts")
```

```
## Warning: Removed 46 rows containing non-finite values (stat_boxplot).
```



Estimating the day of departure

```
default.dep.date <- 15 #i.e., at the first day of the month during which the interview happened
default.distance <- mean(dtm.alldata.geo$distance_travelled, na.rm=TRUE)
dep.date <- rep(default.dep.date, nrow(dtm.alldata.geo))
for (i in 1:nrow(dtm.alldata.geo)) {
  if(is.na(dtm.alldata.geo$transport[i]) == TRUE) distance_per_day <- transport.na else {
    if(dtm.alldata.geo$transport[i] == "Air") distance_per_day <- air
    if(dtm.alldata.geo$transport[i] == "Bus") distance_per_day <- bus
    if(dtm.alldata.geo$transport[i] == "Motorbike") distance_per_day <- motorbike
    if(dtm.alldata.geo$transport[i] == "Bike") distance_per_day <- bike
    if(dtm.alldata.geo$transport[i] == "Foot") distance_per_day <- foot
    if(dtm.alldata.geo$transport[i] == "Truck") distance_per_day <- truck
    if(dtm.alldata.geo$transport[i] == "Taxi Or Car") distance_per_day <- taxi.car
    if(dtm.alldata.geo$transport[i] == "Boat") distance_per_day <- boat
    if(dtm.alldata.geo$transport[i] == "Animals") distance_per_day <- animals
    if(dtm.alldata.geo$transport[i] == "Three Wheel") distance_per_day <- three.wheel
    if(dtm.alldata.geo$transport[i] == "J. Train") distance_per_day <- train
  }
  if(is.na(dtm.alldata.geo$distance_travelled[i]) == TRUE) {
    dep.date[i] <- dtm.alldata.geo$interview.month[i] + 15 -
      (default.distance/1000) / distance_per_day }
  else {
    dep.date[i] <- dtm.alldata.geo$interview.month[i] + 15 -
      (dtm.alldata.geo$distance_travelled[i]/1000) / distance_per_day
  }
```

```
}
dep.date <- as.Date(dep.date)
```

Preparing a column with the name of the starting month to store values in monthly bins.

```
dep.month <- floor_date(dep.date, unit = "month")
dtm.alldata.geo <- cbind(dtm.alldata.geo, dep.date, dep.month)
```

Filtering for only forced migration.

```
forced.migrants.geo <- dtm.alldata.geo %>%
  filter(forced.displacement == "Yes")
```

Creating an aggregated matrix

```
agg.fm.matrix <- matrix(ncol = ncol(forced.migrants.geo), nrow = length(all_admin2)*14)
rownames(agg.fm.matrix) <- paste0(all_admin2[rep(c(1:length(all_admin2)), each = 14)]," ",
                                  month.abb[c(12,1:12,1)]," ",
                                  c(rep(19,1),rep(20,12),21))
colnames(agg.fm.matrix) <- colnames(forced.migrants.geo)
agg.fm.matrix <- data.frame(agg.fm.matrix)
```

Labelling observations: First row is Admin 2 code and second row is first day of month of departure.

```
admin2codes <- all_admin2[rep(c(1:length(all_admin2)), each = 14)]
month.dep <- seq(as.Date("2019-12-01"), length=14, by="1 month")
agg.fm.matrix <- cbind(admin2codes, month.dep, agg.fm.matrix)
agg.fm.matrix$admin2codes <- as.factor(agg.fm.matrix$admin2codes)
```

Aggregate the remaining values

```
columns_to_take_most_frequent <- c(1,2,3,4,5,6,7,8,9,10,12,13,14,15,16,17,19,22,23,27,28,29,40:42,51)
columns_to_take_percentage_of_factor_YES <- c(11,18,20,21,24,25,26)
columns_to_take_sum <- c(30,31:39,43:46)
columns_to_take_average <- c(49,50)
for(i in 1:nrow(agg.fm.matrix)) {
  pcode <- as.character(agg.fm.matrix[i,1])
  month <- agg.fm.matrix[i,2]
  subset <- forced.migrants.geo %>%
    filter(dep.adm2.pcode == pcode) %>%
    filter(dep.month == month)
  if(nrow(subset) != 0) {
    for(j in columns_to_take_most_frequent) {
      agg.fm.matrix[i,j+2] <- names(sort(table(subset[,j]), decreasing = TRUE)[1])
    }
    for(j in columns_to_take_percentage_of_factor_YES) {
      agg.fm.matrix[i,j+2] <- mean(subset[,j] == "Yes")
    }
    for(j in columns_to_take_sum) {
```

```
      agg.fm.matrix[i,j+2] <- sum(subset[,j])
    }
    for(j in columns_to_take_average) {
      agg.fm.matrix[i,j+2] <- mean(subset[,j])
    }
    #if(i/30 == round(i/30)) print(i/nrow(agg.fm.matrix))
  }
}
```

Import weather data

```
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Machine Learning/Project/displacement_su
precipitation <- read.csv("precipitation_all.csv")
temperature <- read.csv("temperature_all.csv")
heat <- read.csv("heat_all.csv")
windchill <- read.csv("windchill_all.csv")
maxwind <- read.csv("maxwind_all.csv")
meanwind <- read.csv("meanwind_all.csv")
```

Calculate weather variables

```
rain.dep.month <- vector(length = nrow(agg.fm.matrix))
rain.month.before <- vector(length = nrow(agg.fm.matrix))
rain.two.months.before <- vector(length = nrow(agg.fm.matrix))
for(i in 1:nrow(agg.fm.matrix)) {
  pcode <- as.character(agg.fm.matrix[i,1])
  pcode_col <- which(pcode == all_admin2) + 2
  month <- agg.fm.matrix[i,2]
  month_row <- which(month == seq(as.Date("2019-10-01"),
                                  as.Date("2021-01-31"),
                                  1))
  #rain in departure month
  rain.dep.month[i] <- sum(precipitation[c(month_row:(month_row+29)), pcode_col])
  rain.month.before[i] <- sum(precipitation[c((month_row-30):(month_row-1)),
                                                              pcode_col])
  #rain two months before
  if(month_row < 60) rain.two.months.before[i] <- NA
  if(month_row > 59) rain.two.months.before[i] <- sum(precipitation[c((month_row-60):(month_row-31)),
                                                              pcode_col])
}
#rain sum two months
rain.sum.2.months <- rain.dep.month + rain.month.before
#rain sum three months
rain.sum.3.months <- rain.dep.month + rain.month.before + rain.two.months.before
#average temperature
temp.dep.month <- vector(length = nrow(agg.fm.matrix))
temp.month.before <- vector(length = nrow(agg.fm.matrix))
temp.two.months.before <- vector(length = nrow(agg.fm.matrix))
for(i in 1:nrow(agg.fm.matrix)) {
  pcode <- as.character(agg.fm.matrix[i,1])
  pcode_col <- which(pcode == all_admin2) + 2
  month <- agg.fm.matrix[i,2]
  month_row <- which(month == seq(as.Date("2019-10-01"),
```

```
                                  as.Date("2021-01-31"),
                                  1))
  temp.dep.month[i] <- mean(heat[c(month_row:(month_row+29)), pcode_col])
  temp.month.before[i] <- mean(heat[c((month_row-30):(month_row-1)),
                                    pcode_col])
  #rain two months before
  if(month_row < 60) temp.two.months.before[i] <- NA
  if(month_row > 59) temp.two.months.before[i] <- mean(heat[c((month_row-60):(month_row-31)),
                                                            pcode_col])
}
#temp sum two months
temp.sum.2.months <- (temp.dep.month+temp.month.before)/2
#temp sum three months
temp.sum.3.months <- (temp.dep.month+temp.month.before+temp.two.months.before)/3
```

Adding weather variables to the dataset

```
agg.fm.matrix <- cbind(agg.fm.matrix,
                       rain.dep.month,rain.month.before,rain.two.months.before,
                       rain.sum.2.months,rain.sum.3.months,
                       temp.dep.month,temp.month.before,temp.two.months.before,
                       temp.sum.2.months,temp.sum.3.months)
#converting South Sudan data from inches to mm
agg.fm.matrix <- agg.fm.matrix %>%
  mutate(rain.dep.month = ifelse(substr(admin2codes,1,2) == "SS",
                                 rain.dep.month/0.03937007874,
                                 rain.dep.month)) %>%
  mutate(rain.month.before = ifelse(substr(admin2codes,1,2) == "SS",
                                 rain.month.before/0.03937007874,
                                 rain.month.before)) %>%
  mutate(rain.two.months.before = ifelse(substr(admin2codes,1,2) == "SS",
                                 rain.two.months.before/0.03937007874,
                                 rain.two.months.before)) %>%
  mutate(rain.sum.2.months = ifelse(substr(admin2codes,1,2) == "SS",
                                 rain.sum.2.months/0.03937007874,
                                 rain.sum.2.months)) %>%
  mutate(rain.sum.3.months =ifelse(substr(admin2codes,1,2) == "SS",
                                 rain.sum.3.months/0.03937007874,
                                 rain.sum.3.months))
```

Adding `total.ind = 0` if there is no recorded migration.

```
for(i in 1:nrow(agg.fm.matrix)) {
  if(is.na(agg.fm.matrix$total.ind[i]) == TRUE) agg.fm.matrix$total.ind[i] <- 0
}
```

Adding the total population data

```
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Machine Learning/Project/Datasets")
SSpop <- read_xlsx("20201102_2020_2021_south_sudan_cod-population_endorsed_v2.xlsx")
SSpop[,4] <- as.factor(SSpop$admin2Pcod)
SDpop <- read_xlsx("sudan_hno-2021-baseline-data.xlsx")[-1,]
```

```
## New names:
## * `` -> ...37
```

```
SDpop[,3] <- as.factor(SDpop$`LOCLITY PCODE`)
SDpop <- SDpop[-73,]
population2021 <- vector(length = nrow(agg.fm.matrix))
for (i in 1:nrow(agg.fm.matrix)) {
  if (sum(agg.fm.matrix$admin2codes[i] == SSpop$admin2Pcod) == 1) {
    row <- which(agg.fm.matrix$admin2codes[i] == SSpop$admin2Pcod)
    population2021[i] <- as.integer(SSpop[row,5])
  }
  if (sum(agg.fm.matrix$admin2codes[i] == SDpop$`LOCLITY PCODE`) == 1) {
    row <- which(agg.fm.matrix$admin2codes[i] == SDpop$`LOCLITY PCODE`)
    population2021[i] <- as.integer(SDpop[row,5])
  }
}
agg.fm.matrix <- cbind(agg.fm.matrix, population2021)
```

Calculating the share of the population that fled.

```
agg.fm.matrix <- agg.fm.matrix %>% mutate(mig.per.1000000 = total.ind/population2021*1000000)
```

Importing the historical temperature and precipitation data.

```
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Machine Learning/Project/displacement_su
historical.precipitation.sd <- array(dim = c(189,16,11))
for (i in 2009:2019) {
  file_name <- paste0("TerraClimate_ppt_",i,".nc_ppt_SD.csv")
  imported_data <- read.csv(file_name)
  historical.precipitation.sd[,,i-2008]  <- as.matrix(imported_data)
}
historical.precipitation.ss <- array(dim = c(79,16,11))
for (i in 2009:2019) {
  file_name <- paste0("TerraClimate_ppt_",i,".nc_ppt_SSD.csv")
  imported_data <- read.csv(file_name)
  historical.precipitation.ss[,,i-2008]  <- as.matrix(imported_data)
}
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Machine Learning/Project/displacement_su
historical.temp.sd <- array(dim = c(189,16,7))
years_available <- c(2011,2014:2019)
j <- 0
for (i in years_available) {
  j <- j+1
  file_name <- paste0("TerraClimate_tmax_",i,".nc_temp_SD.csv")
  imported_data <- read.csv(file_name)
  historical.temp.sd[,,j]  <- as.matrix(imported_data)
}
historical.temp.ss <- array(dim = c(79,16,7))
j <- 0
for (i in years_available) {
  j <- j+1
  file_name <- paste0("TerraClimate_tmax_",i,".nc_temp_SSD.csv")
  imported_data <- read.csv(file_name)
```

```
  historical.temp.ss[,,j]  <- as.matrix(imported_data)
}
```

Taking mean from 2009 to 2019.

```
mean.historical.precipitation <- matrix(nrow = 189+79, ncol = 15)
mean.historical.precipitation[,1:3] <- rbind(historical.precipitation.sd[,1:3,1],
                                             historical.precipitation.ss[,1:3,1])
mean.historical.precipitation[,4:15] <- as.numeric(mean.historical.precipitation[,4:15])
for (i in 4:15) {
  for (j in 1:189) {
    mean.historical.precipitation[j,i] <- mean(as.numeric(historical.precipitation.sd[j,i,]))
  }
  for (j in 190:268) {
    mean.historical.precipitation[j,i] <- mean(as.numeric(historical.precipitation.ss[j-189,i,]))
  }
}
#mean.historical.precipitation[,4:15] <- as.numeric(mean.historical.precipitation[,4:15])
mean.historical.temp <- matrix(nrow = 189+79, ncol = 15)
mean.historical.temp[,1:3] <- rbind(historical.temp.sd[,1:3,1],
                                    historical.temp.ss[,1:3,1])
for (i in 4:15) {
  for (j in 1:189) {
    mean.historical.temp[j,i] <- mean(as.numeric(historical.temp.sd[j,i,]))
  }
  for (j in 190:268) {
    mean.historical.temp[j,i] <- mean(as.numeric(historical.temp.ss[j-189,i,]))
  }
}
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Machine Learning/Project/displacement_su
size.ss <- read.csv("area_pcode_southsudan.csv")
size.sd <- read.csv("area_pcode_sudan.csv")
size <- rbind(size.sd, size.ss)
mean.historical.precipitation <- cbind(mean.historical.precipitation, size[,4])
mean.historical.precipitation <- data.frame(mean.historical.precipitation)
names(mean.historical.precipitation) <- c("index", "pcode", "country", "jan", "feb",
                                          "mar", "apr", "may", "jun", "jul", "aug",
                                          "sep", "oct", "nov", "dec", "size")
for (i in 4:16) mean.historical.precipitation[,i] <- as.numeric(mean.historical.precipitation[,i])
mean.historical.precipitation <- mean.historical.precipitation %>%
  mutate(jan = jan/size) %>%
  mutate(feb = feb/size) %>%
  mutate(mar = mar/size) %>%
  mutate(apr = apr/size) %>%
  mutate(may = may/size) %>%
  mutate(jun = jun/size) %>%
  mutate(jul = jul/size) %>%
  mutate(aug = aug/size) %>%
  mutate(sep = sep/size) %>%
  mutate(oct = oct/size) %>%
  mutate(nov = nov/size) %>%
  mutate(dec = dec/size)
mean(mean.historical.precipitation)
```

```
## [1] NA
```

Calculating difference to 2020.

```
rel.rain.dep.month <- vector(length = length(agg.fm.matrix))
rel.rain.month.before <- vector(length = length(agg.fm.matrix))
rel.rain.two.months.before <- vector(length = length(agg.fm.matrix))
rel.rain.sum.2.months <- vector(length = length(agg.fm.matrix))
rel.rain.sum.3.months <- vector(length = length(agg.fm.matrix))
rel.temp.dep.month <- vector(length = length(agg.fm.matrix))
rel.temp.month.before <- vector(length = length(agg.fm.matrix))
rel.temp.two.months.before <- vector(length = length(agg.fm.matrix))
rel.temp.sum.2.months <- vector(length = length(agg.fm.matrix))
rel.temp.sum.3.months <- vector(length = length(agg.fm.matrix))
for(i in 1:nrow(agg.fm.matrix)) {
  pcode <- as.character(agg.fm.matrix[i,1])
  pcode_row <- which(pcode == mean.historical.precipitation[,2])
  month_col <- as.integer(substr(agg.fm.matrix[i,2],6,7)) + 3

  #precipitation
  rel.rain.dep.month[i] <- agg.fm.matrix$rain.dep.month[i] -
    as.numeric(mean.historical.precipitation[pcode_row,month_col])
  month.before <- month_col - 1
  if (month_col == 4) month.before <- 15
  rel.rain.month.before[i] <- agg.fm.matrix$rain.month.before[i] -
    as.numeric(mean.historical.precipitation[pcode_row,month.before])
  month.2.before <- month_col - 2
  if (month_col == 4) month.2.before <- 14
  if (month_col == 5) month.2.before <- 15
  rel.rain.two.months.before[i] <- agg.fm.matrix$rain.two.months.before[i] -
    as.numeric(mean.historical.precipitation[pcode_row,month.2.before])
  rel.rain.sum.2.months[i] <- rel.rain.month.before[i] + rel.rain.two.months.before[i]
  rel.rain.sum.3.months[i] <- rel.rain.sum.2.months[i] + rel.rain.two.months.before[i]

  #temperature
  rel.temp.dep.month[i] <- agg.fm.matrix$temp.dep.month[i] -
    as.numeric(mean.historical.temp[pcode_row,month_col])
  month.before <- month_col - 1
  if (month_col == 4) month.before <- 15
  rel.temp.month.before[i] <- agg.fm.matrix$temp.month.before[i] -
    as.numeric(mean.historical.temp[pcode_row,month.before])
  month.2.before <- month_col - 2
  if (month_col == 4) month.2.before <- 14
  if (month_col == 5) month.2.before <- 15
  rel.temp.two.months.before[i] <- agg.fm.matrix$temp.two.months.before[i] -
    as.numeric(mean.historical.temp[pcode_row,month.2.before])
  rel.temp.sum.2.months[i] <- (rel.temp.month.before[i] + rel.temp.dep.month[i])/2
  rel.temp.sum.3.months[i] <- (rel.temp.month.before[i] + rel.temp.dep.month[i] +
                                 rel.temp.two.months.before[i])/3
}
```

Addint historical weather comparison to the main dataset.

```
agg.fm.matrix <- cbind(agg.fm.matrix, rel.rain.dep.month, rel.rain.month.before,
                       rel.rain.two.months.before, rel.rain.sum.2.months, rel.rain.sum.3.months,
                       rel.temp.dep.month, rel.temp.month.before, rel.temp.two.months.before,
                       rel.temp.sum.2.months, rel.temp.sum.3.months)
```

Exporting the aggregated table

```
#write.csv(agg.fm.matrix,"monthly_data_3.csv")
#agg.fm.matrix <- read.csv("monthly_data_4.csv")
```

Adding conflict deaths

```
conflict_deaths_data <- read.csv("monthly_data_deaths.csv")
conflict_deaths <- vector(length = nrow(agg.fm.matrix))
for(i in 1:nrow(agg.fm.matrix)) {
  pcode <- as.character(agg.fm.matrix[i,2])
  pcode_row <- which(pcode == conflict_deaths_data[,2])
  pcode_row <- pcode_row[1]
  if (is.na(pcode_row) == TRUE) conflict_deaths[i] <- 0 else
    if (is.na(conflict_deaths_data[pcode_row,67]) == TRUE) {
      if (is.na(conflict_deaths_data[pcode_row,68]) ==TRUE) conflict_deaths[i] <- 0 else
        conflict_deaths[i] <- conflict_deaths_data[pcode_row,68]
    } else
      conflict_deaths[i] <- conflict_deaths_data[pcode_row,67]
}
agg.fm.matrix <- cbind(agg.fm.matrix, conflict_deaths)
#write.csv(agg.fm.matrix,"monthly_data_4.csv")
```

Weather API second try

Sebastian Dodt

19/03/2021

API precipitation download

Pulling historical weather data from the OpenWeather API

I registered 26 email addresses with the Weather data provider.

```
APIs <- c(rep("ab4267cc8a0441f28b3203229211903",10), "e77f0b2e8976478984422130921 1903",
          "6be53899edcb4093a97202124212603", "186a9a0985bc4998a08200749212603",
          "81e1f1325cbf4463863201019212603", "afdd19d5752f44aa9d4200728212603",
          "51f476ad99e041e0b9a201710212603", "f330eddb08a743a1afa201014212603",
          "0b5116dce7b34ac4923202542212603", "b277e3557ed54b4b982201702212603",
          "e635f71fb57c4560a88201957212603", "26140a2010984694be7202857212603",
          "6e4db4b0b7fb481dbf5202059212603", "e1d05549506f4a5d948202946212603",
          "9cd62d7b89c54883bcd202206212603", "003b4ccb714f48b2809203029212603",
          "d070519d4ab648f8824213054212603", "5e5beb0f7bce4b99bb8213029212603",
          "69ef44d1abcb49d7b9f213121212603", "31de8235592d4bc789f213113212603",
          "b1f32d39949045efaa3213047212603", "5f88e4c90dd14724b62213038212603",
          "ec14565c48364240be9213042212603", "5084b09f1c084b5993e213306212603",
          "a25d32d89f584a98a66213332212603", "c9afac8d57384bc49e6233402212603")
```

Functions to pull precipitation and temperature for a certain day at a certain place defined by longitude and latitude.

```
precip_and_temp <- function(lon, lat, day, API) {
  options(warn=-1)
  url <- paste0("http://api.worldweatheronline.com/premium/v1/past-weather.ashx?key=",
                API,
                "&q=",
                lon,",",
                lat,
                "&format=json&date=",
                day,
                "&includelocation=yes&tp=1") # this is the API url
  response <- fromJSON(url)
  response <- data.frame(response)
  response <- unnest(response)
  max.temperature <- max(as.numeric(response$tempC))
  precipitation.daily <- sum(as.numeric(response$precipMM))
  heat.index <- max(as.numeric(response$HeatIndexC))
  max.feels.like <- max(as.numeric(response$WindChillC))
  max.wind <- max(as.numeric(response$WindGustKmph))
```

```
  mean.wind <- mean(as.numeric(response$WindGustKmph))
  output <- c(precipitation.daily, max.temperature, heat.index, max.feels.like, max.wind, mean.wind)
  output
}
```

Loading location data

Preparing vectors to store downloaded results.

```
bulk.adm2 <- all_admin2
precipitation.matrix <- matrix(nrow = 489, ncol = length(bulk.adm2) + 1)
max.temp.matrix <- matrix(nrow = 489, ncol = length(bulk.adm2) + 1)
heat.index.matrix <- matrix(nrow = 489, ncol = length(bulk.adm2) + 1)
max.windchill.matrix <- matrix(nrow = 489, ncol = length(bulk.adm2) + 1)
max.wind.matrix <- matrix(nrow = 489, ncol = length(bulk.adm2) + 1)
mean.wind.matrix <- matrix(nrow = 489, ncol = length(bulk.adm2) + 1)
colnames(precipitation.matrix) <- c("Date",bulk.adm2)
colnames(max.temp.matrix) <- c("Date",bulk.adm2)
colnames(heat.index.matrix) <- c("Date",bulk.adm2)
colnames(max.windchill.matrix) <- c("Date",bulk.adm2)
colnames(max.wind.matrix) <- c("Date",bulk.adm2)
colnames(mean.wind.matrix) <- c("Date",bulk.adm2)
```

Pulling data from API server for the period Oct 2019 until Jan 2021 for all 268 regions in Sudan and South Sudan.

Changing first column to date class

```
precipitation.matrix <- data.frame(precipitation.matrix)
precipitation.matrix[,1] <- as.Date.numeric(precipitation.matrix[,1])
max.temp.matrix <- data.frame(max.temp.matrix)
max.temp.matrix[,1] <- as.Date.numeric(max.temp.matrix[,1])
heat.index.matrix <- data.frame(heat.index.matrix)
heat.index.matrix[,1] <- as.Date.numeric(heat.index.matrix[,1])
max.windchill.matrix <- data.frame(max.windchill.matrix)
max.windchill.matrix[,1] <- as.Date.numeric(max.windchill.matrix[,1])
max.wind.matrix <- data.frame(max.wind.matrix)
max.wind.matrix[,1] <- as.Date.numeric(max.wind.matrix[,1])
mean.wind.matrix <- data.frame(mean.wind.matrix)
mean.wind.matrix[,1] <- as.Date.numeric(mean.wind.matrix[,1])
```

Export weather data

```
write.csv(precipitation.matrix,"precipitation_knit.csv")
write.csv(max.temp.matrix,"temperature_knit.csv")
write.csv(heat.index.matrix,"heat_knit.csv")
write.csv(max.windchill.matrix,"windchill_knit.csv")
write.csv(max.wind.matrix,"maxwind_knit.csv")
write.csv(mean.wind.matrix,"meanwind_knit.csv")
#precipitation.matrix <- read.csv("precipitation.csv")
```

Data Analysis

659605

10/04/2021

R Markdown

```
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Machine Learning/Project/displacement_su
dataset <- read.csv("monthly_data_4.csv")
```

Selecting variables

```
relevant_data <- dataset %>%
  select(mig.per.1000000, rain.dep.month, rain.month.before, rain.two.months.before,
         rain.sum.2.months, rain.sum.3.months,
         temp.dep.month, temp.month.before, temp.two.months.before,
         temp.sum.2.months, temp.sum.3.months,
         rel.rain.dep.month, rel.rain.month.before, rel.rain.two.months.before,
         rel.rain.sum.2.months, rel.rain.sum.3.months,
         rel.temp.dep.month, rel.temp.month.before, rel.temp.two.months.before,
         rel.temp.sum.2.months, rel.temp.sum.3.months, conflict_deaths,
         reason, dep.country)
```

dealing with missing values

```
for (i in 1:nrow(relevant_data)) {
  if (is.na(relevant_data$mig.per.1000000[i]) == TRUE) relevant_data$mig.per.1000000[i] <- 0
}
```

Splitting data into training and testing data

```
set.seed(1)
splits <- initial_split(relevant_data) #, strata = mig.per.1000000)
dataset_train <- training(splits)
dataset_test <- testing(splits)
```

Baseline model to compare more sophisticated models against.

```
lm_model <- lm(mig.per.1000000 ~. -reason -dep.country, dataset_train)
tidy(lm_model)
```

```
## # A tibble: 22 x 5
##    term                   estimate std.error statistic  p.value
```

```
##    <chr>                    <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)            -1150.      684.      -1.68   0.0937
##  2 rain.dep.month             2.77      1.32     2.10   0.0367
##  3 rain.month.before          1.65      1.63     1.01   0.312
##  4 rain.two.months.before    -3.49      1.31    -2.67   0.00808
##  5 rain.sum.2.months         NA        NA       NA     NA
##  6 rain.sum.3.months         NA        NA       NA     NA
##  7 temp.dep.month            15.8      23.1      0.684  0.495
##  8 temp.month.before         40.3      33.2      1.21   0.227
##  9 temp.two.months.before   -19.6      24.4     -0.802  0.423
## 10 temp.sum.2.months         NA        NA       NA     NA
## # ... with 12 more rows
```

Quadratic model

```
lm_model <- lm(mig.per.1000000 ~ (rel.rain.dep.month)^2 + temp.dep.month, dataset_train)
tidy(lm_model)
```

```
## # A tibble: 3 x 5
##   term                estimate std.error statistic  p.value
##   <chr>                  <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)          56.6       9.21       6.14 9.33e-10
## 2 rel.rain.dep.month  -0.0373    0.0188     -1.98 4.73e- 2
## 3 temp.dep.month      -1.10      0.311      -3.53 4.25e- 4
```

Figure 2a

```
graph_data <- dataset %>%
  filter(admin2codes == "SS0303") %>%
  filter(month.dep != "2021-01-01")%>%
  data.frame() %>%
  mutate(month.dep = as.Date(month.dep))

p <- ggplot(graph_data, aes(x = month.dep)) +
  scale_x_date(date_labels = "%b")
p <- p + geom_bar(aes(y = rel.rain.dep.month,
                      colour = "precipitation"),
                  stat = "identity",
                  fill = "#56B4E9")
p <- p + geom_line(aes(y = mig.per.1000000/7,
                       colour = "Migration")) +
  geom_point(aes(y = mig.per.1000000/7,
                       colour = "Migration"))
p <- p + scale_y_continuous(sec.axis = sec_axis(~.*7, name = "Migrants (per million)"))
p <- p + scale_colour_manual(values = c("#D55E00", "#56B4E9"))
p <- p + labs(y = "relative precipitation (in mm)",
              x = "Month of displacement in 2020",
              colour = "Variables")

p <- p + theme(legend.position = c(0.1, 0.87),
               legend.title = element_text(size = 8),
               legend.text = element_text(size = 8))
p <- p + theme(
```

```
    axis.title.y = element_text(color = "#0072B2", size=13),
    axis.title.y.right = element_text(color = "#D55E00", size=13)
  )
p <- p + ggtitle("Figure 2a: Precipitation compared to 10-year average \n and migration in Bor, South Su
p
```

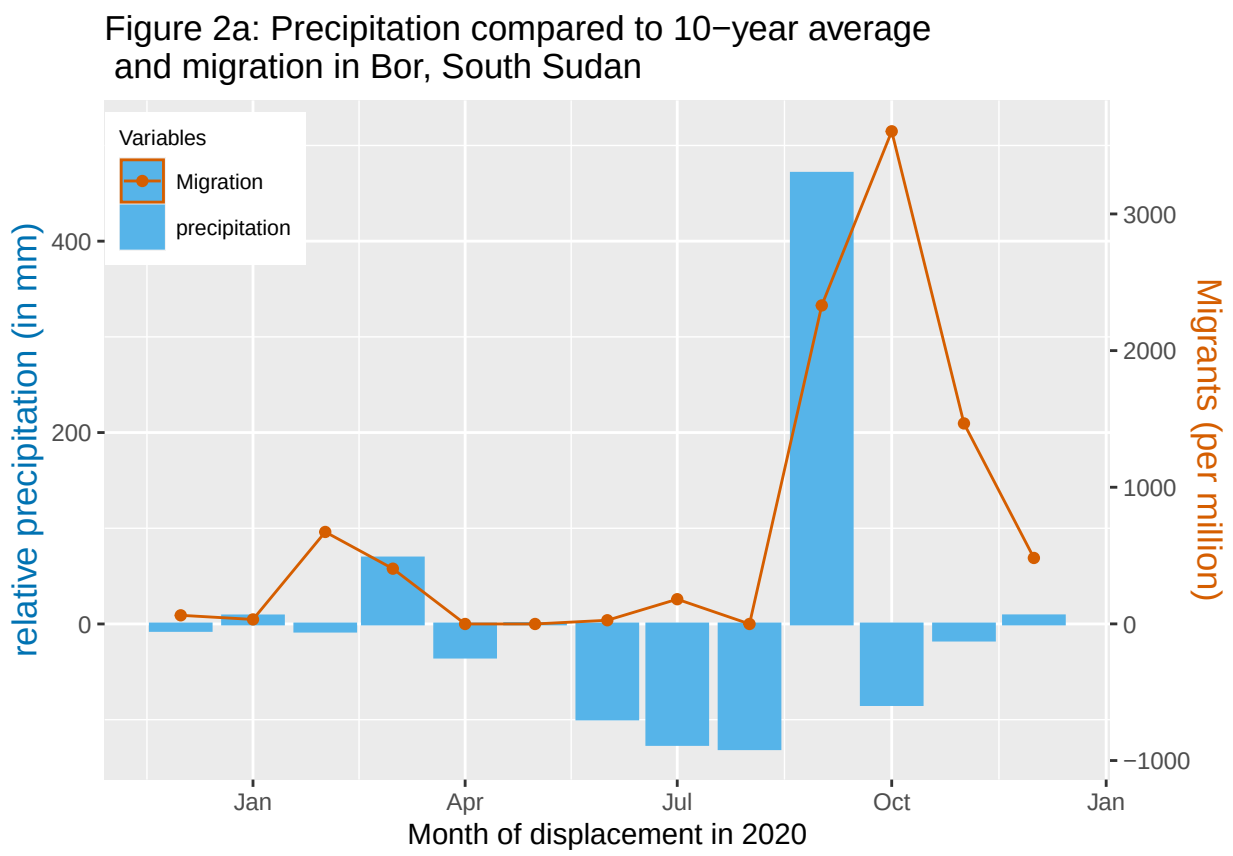
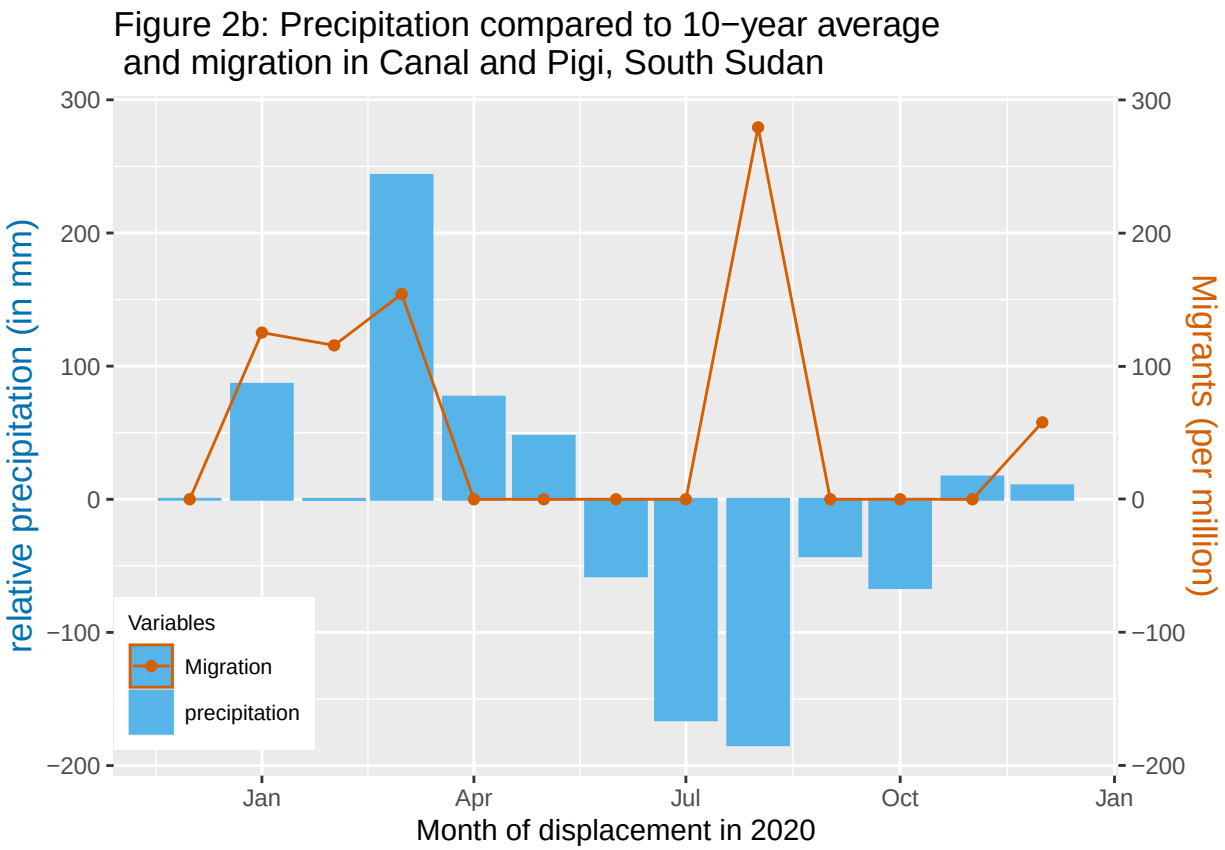


Figure 2b

```
graph_data <- dataset %>%
  filter(admin2codes == "SS0304") %>%
  filter(month.dep != "2021-01-01") %>%
  data.frame() %>%
  mutate(month.dep = as.Date(month.dep))

p <- ggplot(graph_data, aes(x = month.dep)) +
  scale_x_date(date_labels = "%b")
p <- p + geom_bar(aes(y = rel.rain.dep.month,
                      colour = "precipitation"),
                  stat = "identity",
                  fill = "#56B4E9")
p <- p + geom_line(aes(y = mig.per.1000000,
                       colour = "Migration")) +
  geom_point(aes(y = mig.per.1000000,
                       colour = "Migration"))
p <- p + scale_y_continuous(sec.axis = sec_axis(~., name = "Migrants (per million)"))
p <- p + scale_colour_manual(values = c("#D55E00", "#56B4E9"))
```

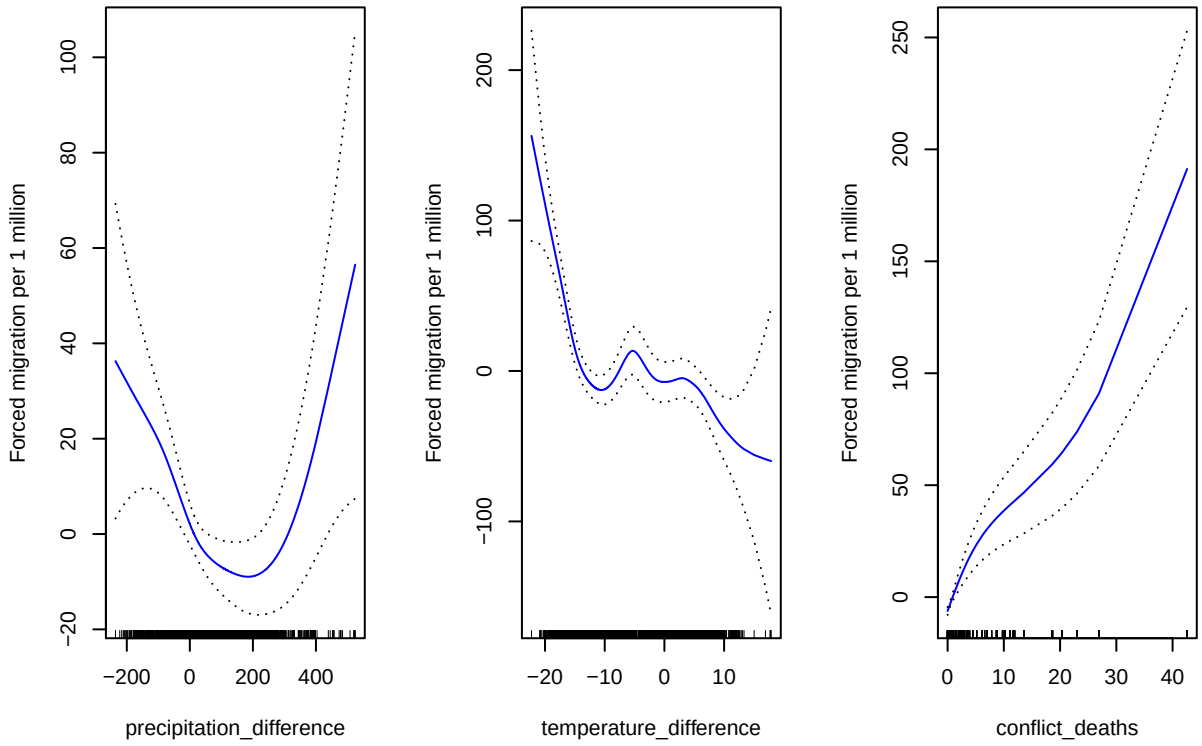
```
p <- p + labs(y = "relative precipitation (in mm)",
              x = "Month of displacement in 2020",
              colour = "Variables")

p <- p + theme(legend.position = c(0.1, 0.15),
               legend.title = element_text(size = 8),
               legend.text = element_text(size = 8))
p <- p + theme(
    axis.title.y = element_text(color = "#0072B2", size=13),
    axis.title.y.right = element_text(color = "#D55E00", size=13)
  )
p <- p + ggtitle("Figure 2b: Precipitation compared to 10-year average \n and migration in Canal and Pi
p
```



Fitting a GAM

```
restricted <- dataset_train %>%
  filter(rel.rain.dep.month < 550) %>%
  rename(precipitation_difference = rel.rain.dep.month) %>%
  rename(temperature_difference = rel.temp.dep.month) %>%
  mutate(conflict_deaths = conflict_deaths/12)
gam1 <- gam(mig.per.1000000 ~ s(precipitation_difference,3) +
              s(temperature_difference,8) +
              s(conflict_deaths, 2),
            data = restricted)
par(mfrow = c(1,3))
```

```
gam.plots <- plot(gam1, se = TRUE, col = "blue",
     ylab = "Forced migration per 1 million")
```



Testing the performance of the GAM

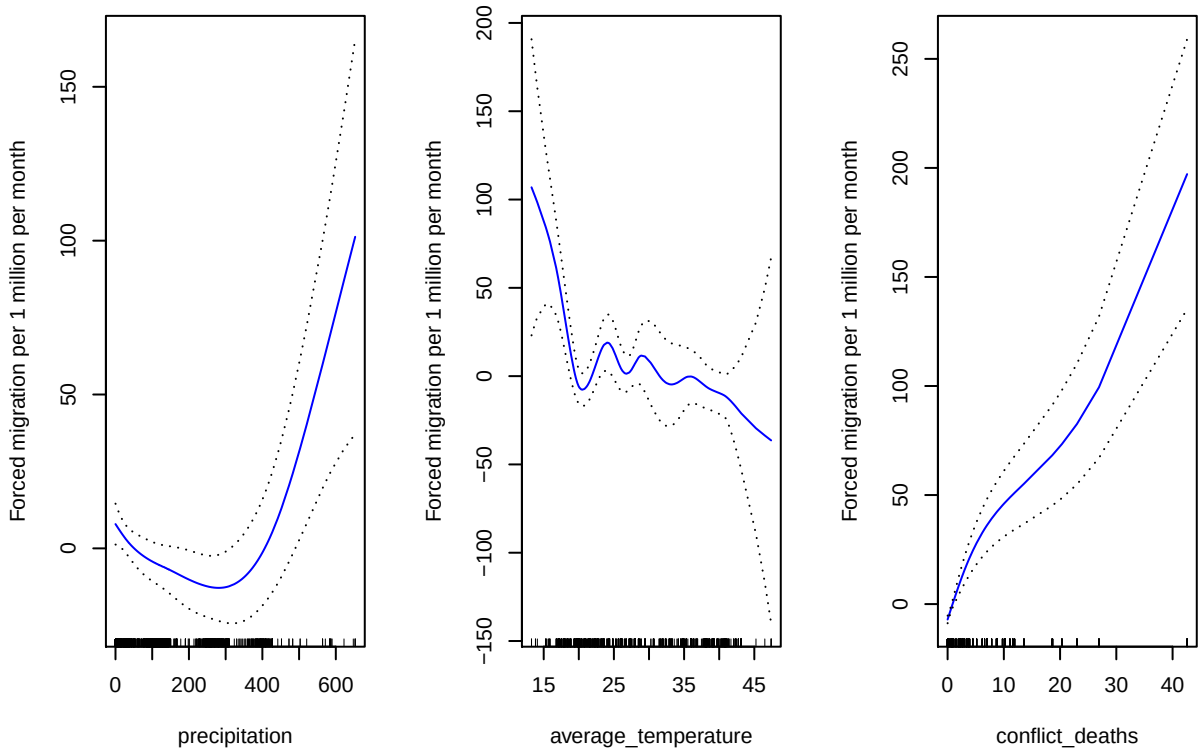
```
test.restricted <- dataset_test %>%
  rename(precipitation_difference = rel.rain.dep.month) %>%
  rename(temperature_difference = rel.temp.dep.month) %>%
  mutate(conflict_deaths = conflict_deaths/12)
gam.fit <- predict(gam1, newdata = test.restricted)
rmse <- mean(sqrt((gam.fit-dataset_test$mig.per.1000000)^2))
rmse
```

```
## [1] 41.1246
```

Checking GAM of absolute data (rather than relative) as above. It does not perform as well.

```
restricted <- dataset_train %>%
  filter(rel.rain.dep.month < 550) %>%
  rename(precipitation = rain.dep.month) %>%
  rename(average_temperature = temp.dep.month) %>%
  mutate(conflict_deaths = conflict_deaths/12)
gam2 <- gam(mig.per.1000000 ~ s(precipitation,3) +
              s(average_temperature,8) +
              s(conflict_deaths, 2),
```

```
           data = restricted)
par(mfrow = c(1,3))
gam.plots <- plot(gam2, se = TRUE, col = "blue",
     ylab = "Forced migration per 1 million per month")
```




```
test.restricted <- dataset_test %>%
  rename(precipitation = rain.dep.month) %>%
  rename(average_temperature = temp.dep.month) %>%
  mutate(conflict_deaths = conflict_deaths/12)
gam.fit <- predict(gam2, newdata = test.restricted)
rmse <- mean(sqrt((gam.fit-dataset_test$mig.per.1000000)^2))
rmse
```

```
## [1] 41.69248
```

Fitting a GAM on the data from the month before the migration occured. This performs better.

```
restricted <- dataset_train %>%
  filter(rel.rain.dep.month < 550) %>%
  rename(precipitation = rel.rain.month.before) %>%
  rename(average_temperature = rel.temp.month.before) %>%
  mutate(conflict_deaths = conflict_deaths/12)
gam3 <- gam(mig.per.1000000 ~ s(precipitation,3) +
              s(average_temperature,8) +
```

```
            s(conflict_deaths, 2),
          data = restricted)
test.restricted <- dataset_test %>%
  rename(precipitation = rel.rain.month.before) %>%
  rename(average_temperature = rel.temp.month.before) %>%
  mutate(conflict_deaths = conflict_deaths/12)
gam.fit <- predict(gam3, newdata = test.restricted)
rmse <- mean(sqrt((gam.fit-dataset_test$mig.per.1000000)^2))
rmse
```

```
## [1] 38.60995
```

Every variable is significant

```
tidy(gam1)
```

```
## # A tibble: 4 x 6
##   term                            df     sumsq   meansq statistic   p.value
##   <chr>                        <dbl>     <dbl>    <dbl>     <dbl>     <dbl>
## 1 s(precipitation_difference, 3)    1    218826.  218826.      13.4  2.56e- 4
## 2 s(temperature_difference, 8)      1    557880.  557880.      34.2  5.63e- 9
## 3 s(conflict_deaths, 2)             1   1010499. 1010499.      61.9  5.13e-15
## 4 Residuals                      2782. 45415588.   16325.      NA   NA
```

The GAM passes the test against a baseline model

```
rmse_baseline <- mean(sqrt((mean(dataset_test$mig.per.1000000)-dataset_test$mig.per.1000000)^2))
rmse_baseline
```

```
## [1] 46.72526
```

Figure 4

```
setwd("~/OneDrive/Uni/SOAS University of London/Modules/year 3/Environment and Climate Crisis/AS1")
natural_disasters <- read.csv("number-of-natural-disaster-events.csv")
natural_disasters %>%
  filter(Year > 1970) %>%
  filter(Entity == "Drought" | Entity == "Flood") %>%
  ggplot(aes(x = Year, y = Number.of.reported.natural.disasters..reported.disasters.)) +
  geom_line(aes(color = Entity)) +
  scale_color_manual(values = c("#D55E00", "#56B4E9")) +
  ggtitle("Figure 4: Number of droughts and floods worldwide") +
  ylab("recorded cases") +
  xlab("Year")
```

Figure 4: Number of droughts and floods worldwide