Impact of socio-economic determinants on the speed of epidemic diseases.

Gilles Dufrénot Ewen Gallic Pierre Michel
Norgile Midopkè Bonou Ségui Gnaba Iness Slaoui

2024-01-09

Table of contents

Introduction

This ebook is the online supplementary materials for the article titled "*Impact of socio-economic determinants on the speed of epidemic diseases. A comparative analysis*".

The document is divided in three parts. The first part provides the codes for downloading the data (Chapter 1). The second part provides the codes for estimating the parameters of a generalized phenomenological Richards model by nonlinear least squares (NLS) method (Chapter 2). Lastly, the third part presents the codes that explain the heterogeneity of the epidemic parameters using socioeconomic variables (Chapters 3, 4, and 5).

Part I

Data

1 Load Data

This chapter provides the codes used to obtain the final dataset used in the article. It contains two sections. The first one concerns the Covid-19 epidemic, the second one is devoted to the 2009-2010 H1N1 epidemic.

The {`tidyverse`} package is needed, as well as {`lubridate`}:

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(lubridate)
```

For graphical output, we can define a theme:

```
#' Theme for ggplot2
#'
#' @param ... arguments passed to the theme function
#' @export
#' @importFrom ggplot2 element_rect element_text element_blank element_line unit
#'   rel
theme_paper <- function (...) {
  theme(
    text = element_text(family = "Times"),
    plot.background = element_rect(fill = "transparent", color = NA),
    panel.background = element_rect(fill = "transparent", color = NA),
```

```
    panel.border = element_rect(fill = NA, colour = "grey50", linewidth = 1),
    axis.text = element_text(),
    legend.text = element_text(size = rel(1.1)),
    legend.title = element_text(size = rel(1.1)),
    legend.background = element_rect(fill = "transparent", color = NULL),
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.box = "vertical",
    legend.key = element_blank(),
    panel.spacing = unit(1, "lines"),
    panel.grid.major = element_line(colour = "grey90"),
    panel.grid.minor = element_blank(),
    plot.title = element_text(hjust = 0, size = rel(1.3), face = "bold"),
    plot.title.position = "plot",
    plot.margin = unit(c(1, 1, 1, 1), "lines"),
    strip.background = element_rect(fill = NA, colour = NA),
    strip.text = element_text(size = rel(1.1))
  )
}
```

1.1 Covid-19

This section is devoted to Covid-19 data. The first part presents the data relative to the number of cases while the second part shows the socio-economic data associated to each country during the period of the epidemic.

1.1.1 Epidemic Data

The epidemic data come from the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (GitHub).

To handle dates properly, some adjustments need to be done:

```
# source("themes.R")
if (.Platform$OS.type == "unix") {
  Sys.setlocale("LC_ALL", "en_US.UTF-8")
} else {
  Sys.setlocale("LC_ALL", "English_United States")
}
```

```
[1] "en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8"
```

Let us define the following function (as suggested by markus on stackoverflow) to remove trailing NAs:

```
rm_NA_headtail <- function(x) {
  cumsum(!is.na(x)) != 0 & rev(cumsum(!is.na(rev(x)))) != 0
}
```

We can create a table, `list_countries`, with two columns: one that contains the name of the countries, and another one that gives the group the corresponding country belongs to.

```
mena <- c(
  "Algeria", "Bahrain", "Egypt", "Iran", "Iraq", "Israel", "Jordan", "Kuwait",
  "Lebanon", "Morocco", "Oman", "Qatar", "Saudi Arabia", "Syria", "Tunisia",
  "Yemen"
)
ssa <- c(
  "Angola", "Benin", "Burkina Faso", "Burundi","Cape Verde",
  "Cameroon", "Central African Republic", "Chad",
  "Democratic Republic of Congo", "Congo", "Cote d'Ivoire", "Djibouti",
  "Ethiopia", "Gabon", "Gambia", "Ghana", "Guinea", "Kenya",
  "Lesotho", "Liberia", "Madagascar", "Malawi", "Mali", "Mauritania",
  "Mauritius", "Mozambique", "Namibia", "Niger", "Nigeria", "Rwanda",
  "Senegal", "Seychelles", "Sierra Leone", "Somalia", "South Africa",
  "South Sudan", "Sudan", "Eswatini", "Tanzania", "Togo", "Uganda",
  "Zimbabwe"
)
latin_america <- c(
  "Argentina", "Bolivia", "Brazil", "Chile", "Colombia", "Ecuador",
  "El Salvador", "Guatemala", "Haiti", "Honduras", "Jamaica", "Mexico",
  "Nicaragua", "Panama", "Paraguay", "Peru", "Uruguay", "Venezuela"
)
asia <- c(
  "Afghanistan", "Bangladesh", "Bhutan", "Brunei", "Cambodia", "India",
  "Indonesia", "Laos", "Malaysia", "Mongolia", "Myanmar", "Nepal", "Pakistan",
  "Philippines", "Russia", "Solomon Islands", "South Korea", "Sri Lanka",
  "Thailand", "Timor-Leste", "Vanuatu", "Vietnam", "China"
)
indus <- c(
  "Australia", "Austria", "Belgium", "Canada", "Denmark", "France", "Germany",
  "Greece", "Ireland", "Italy", "Japan", "Netherlands", "Spain", "Sweden",
```

```
    "United Kingdom", "United States"
  )

  list_countries <-
    tibble(country = mena, group = "MENA") |>
    bind_rows(tibble(country = ssa, group = "Sub-Saharan Africa")) |>
    bind_rows(tibble(country = latin_america, group = "Latin America")) |>
    bind_rows(tibble(country = asia, group = "Asia")) |>
    bind_rows(tibble(country = indus, group = "Industrialized"))

  list_countries
```

```
# A tibble: 115 x 2
   country group
   <chr>   <chr>
 1 Algeria MENA
 2 Bahrain MENA
 3 Egypt   MENA
 4 Iran    MENA
 5 Iraq    MENA
 6 Israel  MENA
 7 Jordan  MENA
 8 Kuwait  MENA
 9 Lebanon MENA
10 Morocco MENA
# i 105 more rows
```

```
  save(list_countries, file = "./data/output/list_countries.rda")
```

We can also create a variable with the name of all countries:

```
  names_countries <- list_countries$country |> unique()
```

The data can be obtained directly from GitHub:

```
  # Run once
  df_oxford <- str_c(
    "https://raw.githubusercontent.com/OxCGRT/covid-policy-tracker/master/data/",
    "OxCGRT_latest.csv"
  ) |>
    read.csv()
```

```
# Save the raw data
save(df_oxford, file = "./data/raw/df_oxford.rda")
```

To avoid downloading the data each time we run the program, we can save them and only load the previously downloaded data as follows:

```
load("./data/raw/df_oxford.rda")
```

We want to focus only on data at the country level: `RegionName` must be equal to the empty string.

```
confirmed_df <-
  df_oxford |>
  filter(RegionName == "") |>
  select(
    country = CountryName, country_code = CountryCode,
    date = Date,
    value = ConfirmedCases, stringency_index = StringencyIndex) |>
  filter(country %in% names_countries) |>
  as_tibble() |>
  mutate(
    date = ymd(date),
    days_since_2020_01_22 = lubridate::interval(
      lubridate::ymd("2020-01-22"), date
    ) / lubridate::ddays(1)
    )
```

The cumulative number of cases is contained in `value`. There are some bizarre values corresponding to statistical adjustments. We can smooth the series of the number of cases using a 7 days rolling windows. To do so, we compute the first difference of the cumulative number of cases to get daily cases, then smooth the data. Lastly, we compute the cumulative number of cases from the smoothed values.

```
confirmed_df <-
  confirmed_df |>
  group_by(country, country_code) |>
  mutate(c = value - lag(value)) |>
  mutate(c_7d = zoo::rollmean(c, k = 7, fill = NA)) |>
  mutate(c_7d = ifelse(c_7d < 0, yes = 0, no = c_7d)) |>
  group_by(country, country_code) |>
  mutate(cumul = ifelse(is.na(c_7d), yes = 0, no = c_7d),
         cumul = cumsum(cumul),
```

```
          cumul = ifelse(is.na(value), yes = NA, no = cumul)) |>
  ungroup()
```

Then, we replace cumulative number of confirmed cases with the value obtained using a moving average:

```
confirmed_df <-
  confirmed_df |>
  mutate(value = cumul)
```

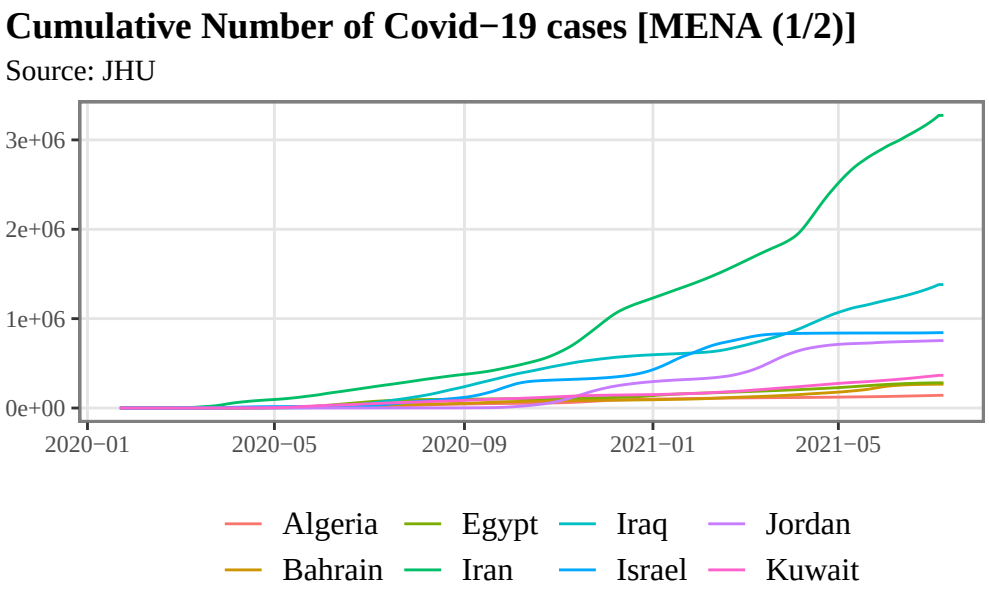
The trailing `NA` values can be removed as follows:

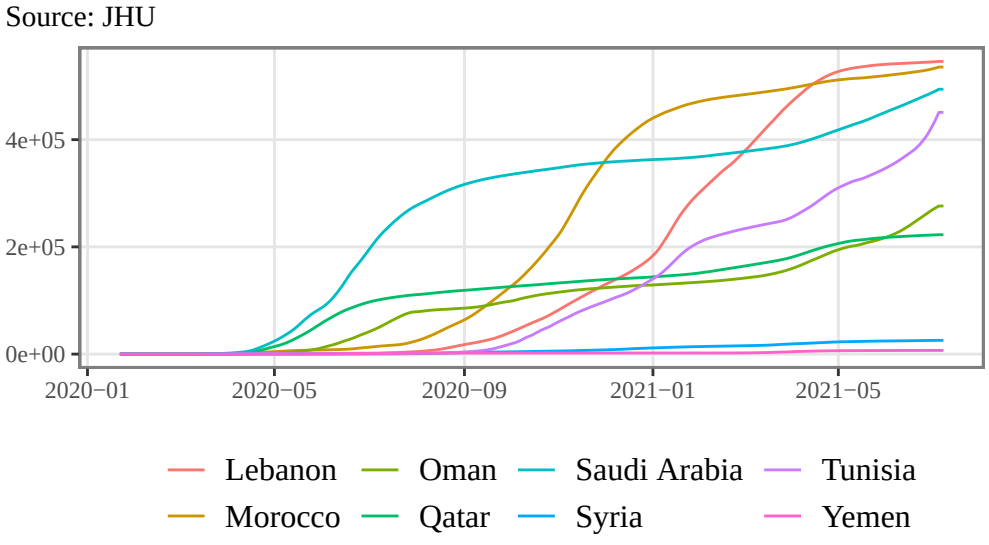
```
confirmed_df <-
  confirmed_df |>
  group_by(country) |>
  filter(rm_NA_headtail(value)) |>
  ungroup()
```

The series can be plotted. For Mena countries:

```
split_chunks <- function(x,n) {
  split(x, cut(seq_along(x), n, labels = FALSE))
}

g <- split_chunks(1:length(mena), n = 2)

for (i in 1:length(g)) {
  p <-
    ggplot(
      data = confirmed_df |>
        filter(country %in% mena[g[[i]]]),
      mapping = aes(x = date, y = value, colour = country)
    ) +
    geom_line() +
    scale_colour_discrete(NULL) +
    scale_x_date(
      breaks = ymd(pretty_dates(confirmed_df$date, n = 5)),
      date_labels = "%Y-%m"
    ) +
    labs(
      x = NULL, y = NULL,
      title = str_c(
```

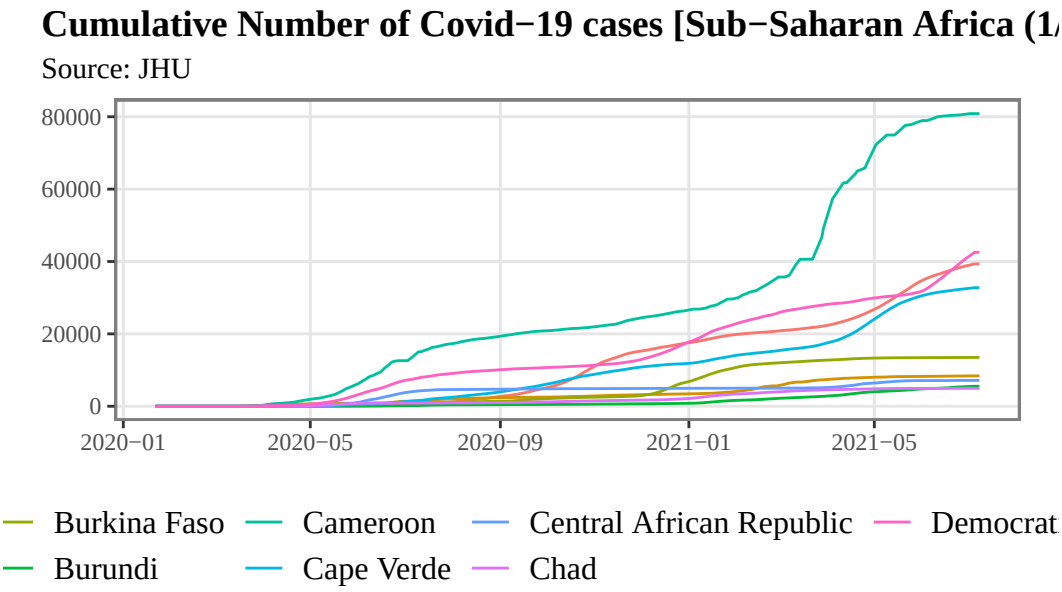
```
        "Cumulative Number of Covid-19 cases [MENA (", i, "/", length(g), ")]"
      ),
      subtitle = "Source: JHU"
    ) +
    theme_paper()
  print(p)
}
```





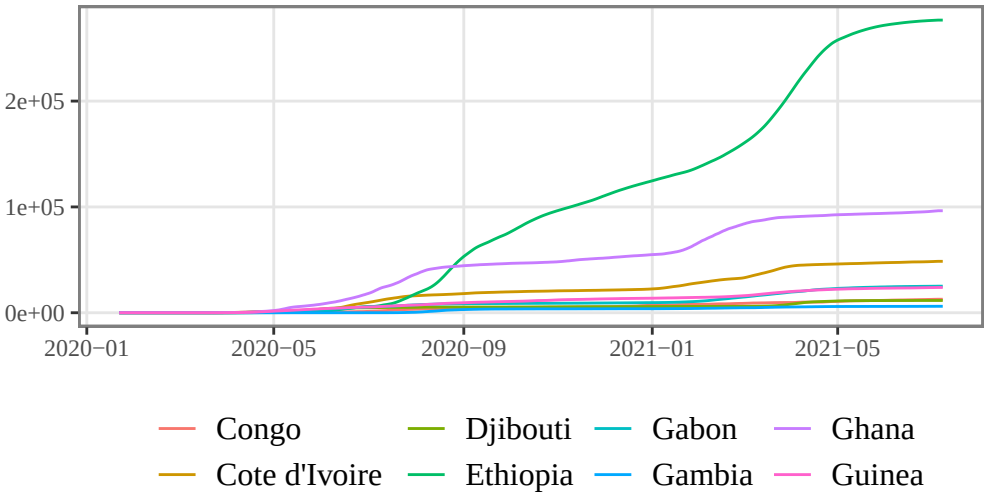
For Sub-Saharan countries:

```
g <- split_chunks(1:length(ssa), n = 5)

for (i in 1:length(g)) {
  p <- ggplot(
    data = confirmed_df |> filter(country %in% ssa[g[[i]]]),
    mapping = aes(x = date, y = value, colour = country)
  ) +
    geom_line() +
    scale_colour_discrete(NULL) +
    scale_x_date(
      breaks = ymd(pretty_dates(confirmed_df$date, n = 5)),
      date_labels = "%Y-%m"
    ) +
    labs(
      x = NULL, y = NULL,
      title = str_c(
        "Cumulative Number of Covid-19 cases [Sub-Saharan Africa (",  i, "/",
        length(g), ")]"
      ),
      subtitle = "Source: JHU") +
    theme_paper()
  print(p)
}
```



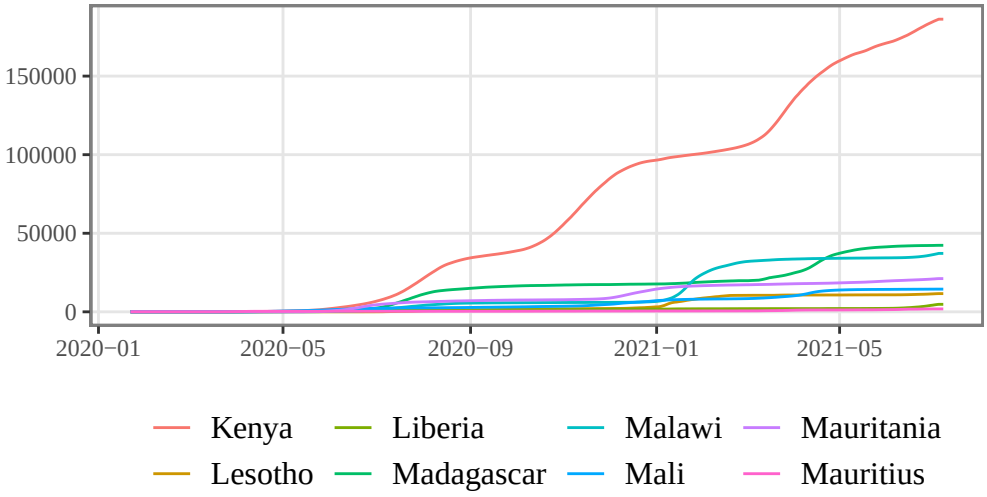
Cumulative Number of Covid−19 cases [Sub−Saharan Africa (2/

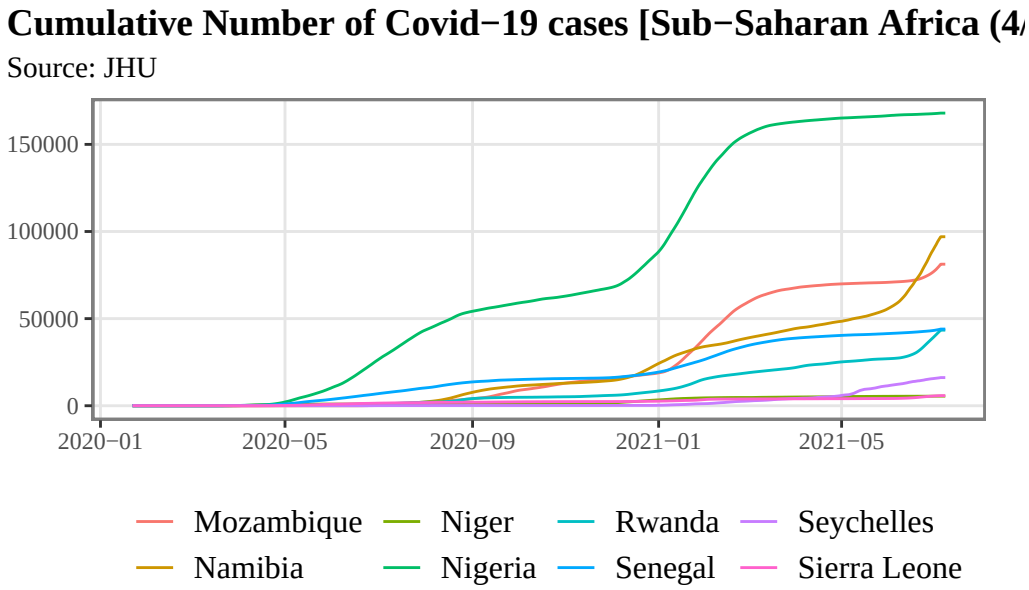
Source: JHU

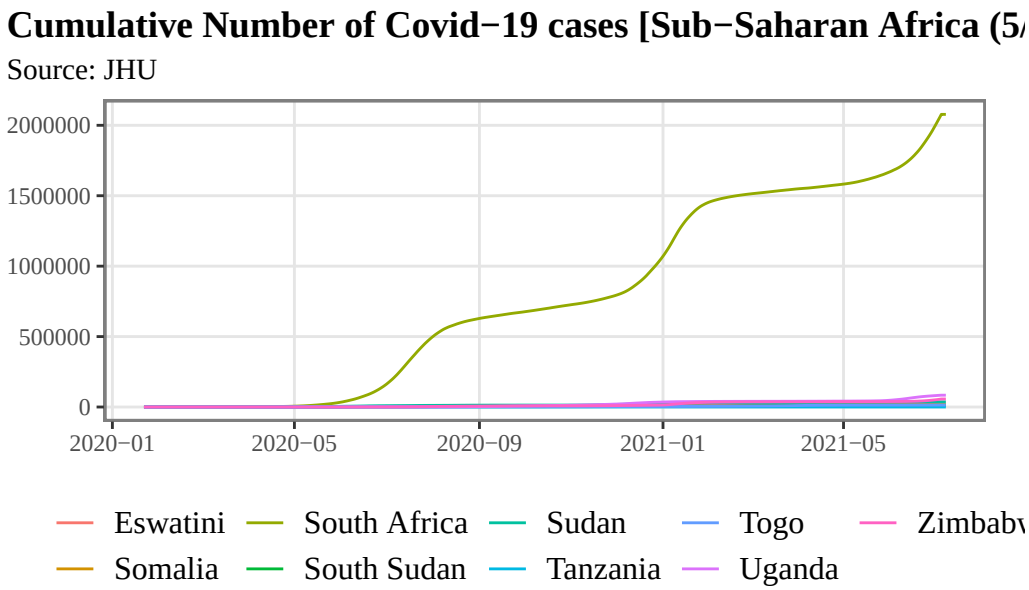


Cumulative Number of Covid−19 cases [Sub−Saharan Africa (3/

Source: JHU



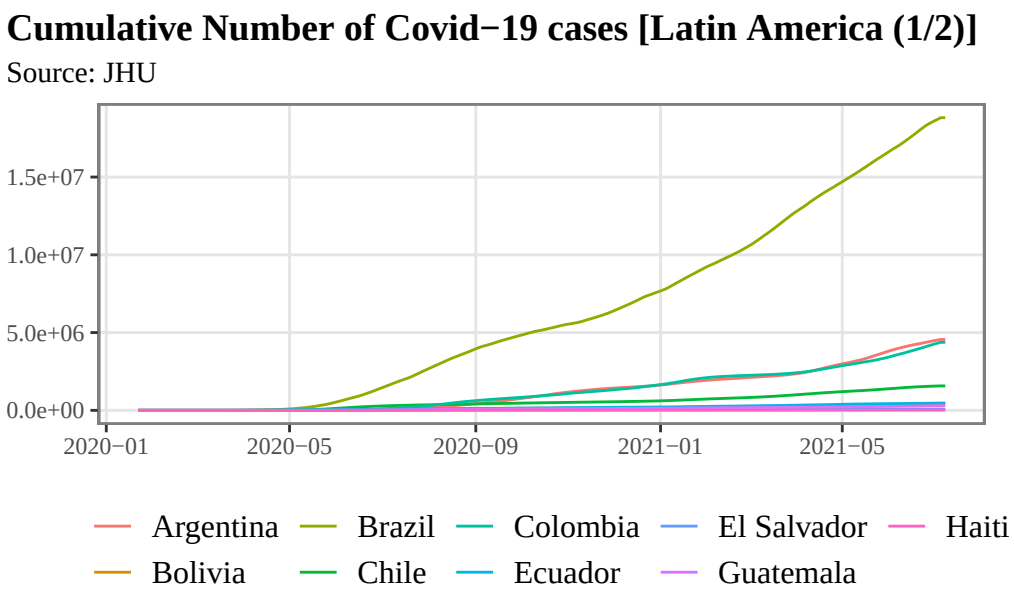


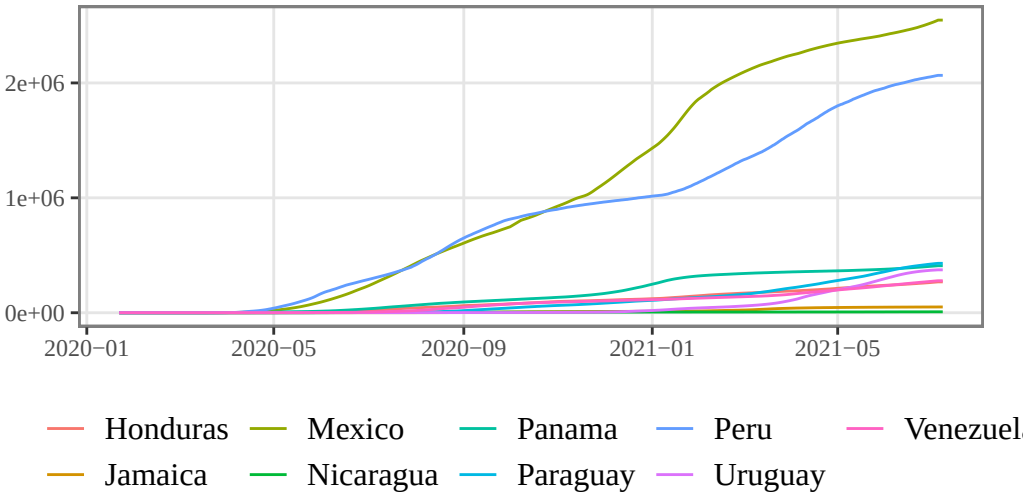


For Latin America:

```
g <- split_chunks(1:length(latin_america), n = 2)

for (i in 1:length(g)) {
  p <- ggplot(
    data = confirmed_df |> filter(country %in% latin_america[g[[i]]]),
```

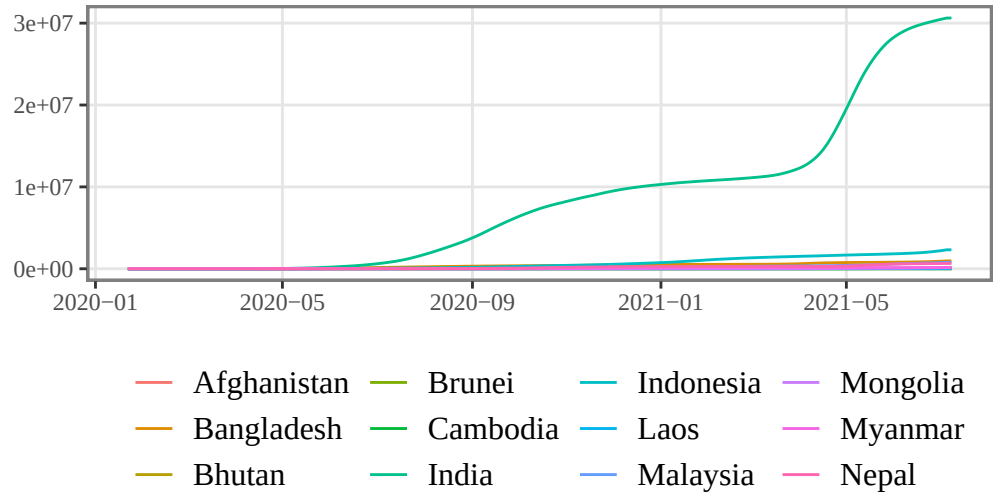
```
    mapping = aes(x = date, y = value, colour = country)
  ) +
    geom_line() +
    scale_colour_discrete(NULL) +
    scale_x_date(breaks = ymd(pretty_dates(confirmed_df$date, n = 5)),
                 date_labels = "%Y-%m") +
    labs(
      x = NULL, y = NULL,
      title = str_c(
        "Cumulative Number of Covid-19 cases [Latin America (",
        i, "/", length(g), ")]"
      ),
      subtitle = "Source: JHU"
    ) +
    theme_paper()
  print(p)
}
```

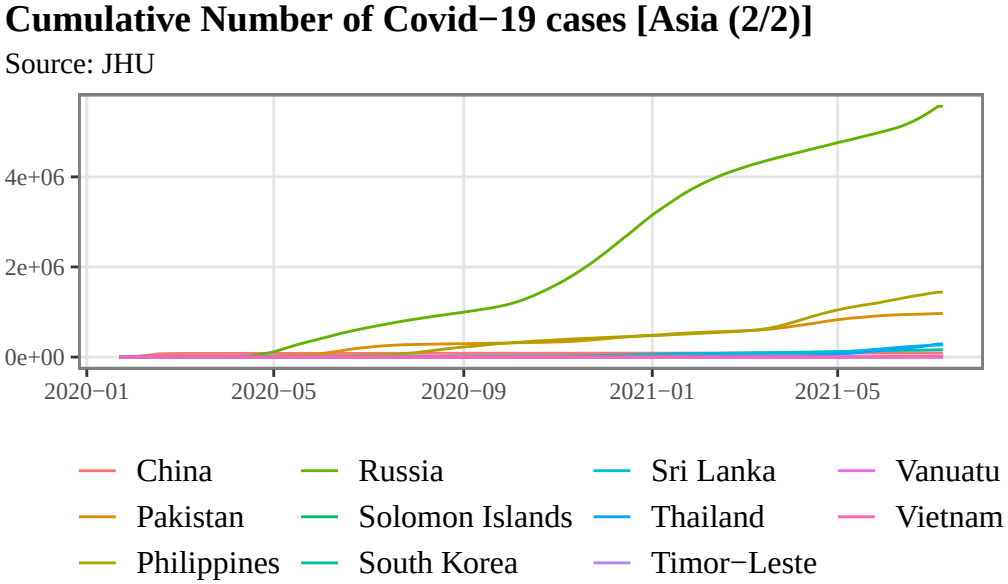




For Asia:

```
g <- split_chunks(1:length(asia), n = 2)

for (i in 1:length(g)) {
  p <- ggplot(
      data = confirmed_df |> filter(country %in% asia[g[[i]]]),
      mapping = aes(x = date, y = value, colour = country)
  ) +
    geom_line() +
    scale_colour_discrete(NULL) +
    scale_x_date(
      breaks = ymd(pretty_dates(confirmed_df$date, n = 5)),
      date_labels = "%Y-%m"
    ) +
    labs(
      x = NULL, y = NULL,
      title = str_c(
        "Cumulative Number of Covid-19 cases [Asia (", i, "/", length(g), ")]"
      ),
      subtitle = "Source: JHU") +
    theme_paper()
  print(p)
}
```

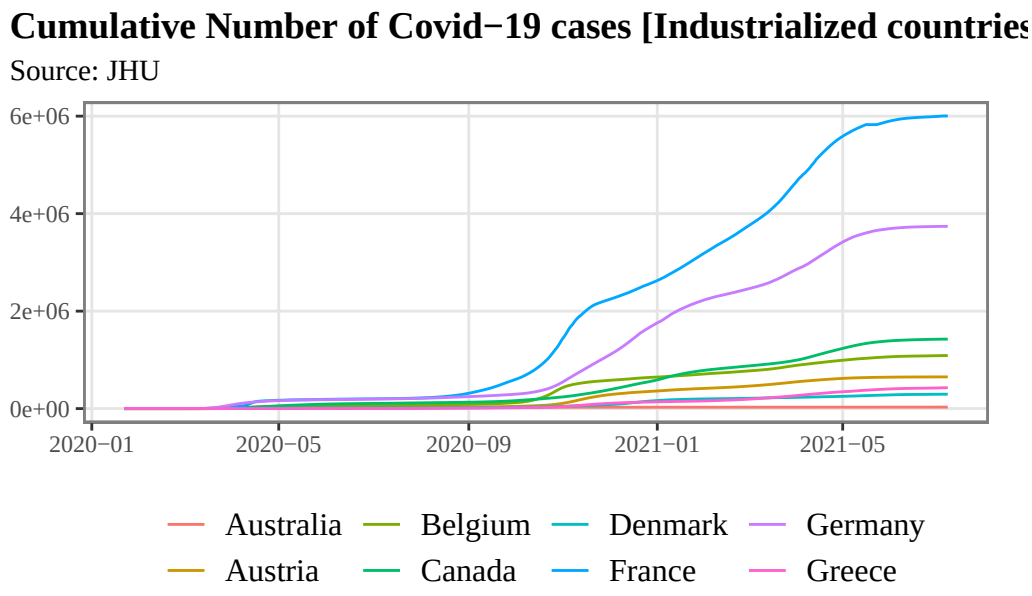


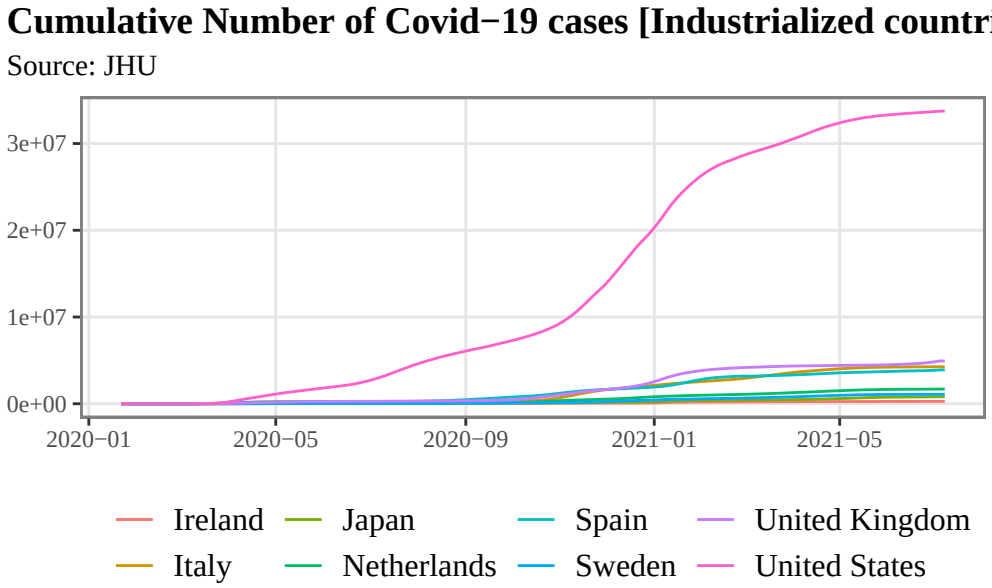


And for industrialized countries:

```
g <- split_chunks(1:length(indus), n = 2)

for (i in 1:length(g)) {
  p <- ggplot(
    data = confirmed_df |> filter(country %in% indus[g[[i]]]),
```

```
    mapping = aes(x = date, y = value, colour = country)
  ) +
    geom_line() +
    scale_colour_discrete(NULL) +
    scale_x_date(
      breaks = ymd(pretty_dates(confirmed_df$date, n = 5)),
      date_labels = "%Y-%m"
    ) +
    labs(
      x = NULL, y = NULL,
      title = str_c(
        "Cumulative Number of Covid-19 cases [Industrialized countries (",
        i, "/", length(g), ")]"
      ),
      subtitle = "Source: JHU"
    ) +
    theme_paper()
  print(p)
}
```





Let us save the dataset:

```
covid_cases <- confirmed_df |>
  select(country, date, country_code, value, stringency_index)
save(covid_cases, file = "./data/output//covid_cases.rda")
```

1.1.2 Socio-Economic Data

The socio-economic data come from five sources:

- World Development Indicators
- Global Footprint Network National Footprint and Biocapacity Accounts, 2021 Edition
- Size of the Shadow Economy, from Medina and Schneider (2019)
- Gini index from the CIA
- Policy variables from Hale et al. (2021)

1.1.2.1 Creation of List of Variables to Keep

The variables that are kept in the study are divided in 6 categories, 5 of which that concern the WDI database. The 6th (Policy variables and governance) is extracted from the Oxford COVID-19Government Response Tracker.

```r
# S1: Healthcare infrastructure
variables_to_keep_S1 <- c(
  "Physicians (per 1,000 people)",
  "Hospital beds (per 1,000 people)",
  "Nurses and midwives (per 1,000 people)",
  "Domestic general government health expenditure (% of general government expenditure)",
  "Domestic private health expenditure per capita (current US$)",
  "Domestic private health expenditure per capita, PPP (current international $)"
)

# S2: Vulnerability to comorbidities
variables_to_keep_S2 <- c(
  "Incidence of malaria (per 1,000 population at risk)",
  "Incidence of HIV, all (per 1,000 uninfected population)",
  "Incidence of tuberculosis (per 100,000 people)",
  "Diabetes prevalence (% of population ages 20 to 79)",
  "Mortality from CVD, cancer, diabetes or CRD between exact ages 30 and 70 (%)"
)

# S3: Vulnerability to natural environment
variables_to_keep_S3 <- c(
  "Mortality rate attributed to household and ambient air pollution, age-standardized (per
  "PM2.5 air pollution, mean annual exposure (micrograms per cubic meter)",
  "PM2.5 pollution, population exposed to levels exceeding WHO Interim Target-1 value (% o
  "PM2.5 pollution, population exposed to levels exceeding WHO Interim Target-2 value (% o
  "PM2.5 pollution, population exposed to levels exceeding WHO Interim Target-3 value (% o
  "Air transport, passengers carried",
  "International tourism, number of arrivals",
  "Ecological Footprint"
)

# S4: Living conditions
variables_to_keep_S4 <- c(
  # "Multidimensional poverty index (scale 0-1)",# No data...
  "People using at least basic drinking water services (% of population)",
  "People using at least basic sanitation services (% of population)",
  "Prevalence of undernourishment (% of population)",
  "Prevalence of anemia among women of reproductive age (% of women ages 15-49)",
  "Poverty headcount ratio at $3.65 a day (2017 PPP) (% of population)",
  "Poverty headcount ratio at $6.85 a day (2017 PPP) (% of population)",
  "Poverty headcount ratio at national poverty lines (% of population)",
```

```
  "GDP per capita (current US$)",
  "GDP per capita, PPP (current international $)",
  "Urban population", # Will not be kept in PCA
  "Urban land area (sq. km)", # Will not be kept in PCA
  "Urban density"
)

#S5: Economic and societal characteristics
variables_to_keep_S5 <- c(
  "Population, total",
  "GDP per capita growth (annual %)",
  "International migrant stock (% of population)",
  "Population ages 65 and above (% of total population)",
  "Individuals using the Internet (% of population)",
  "Mobile cellular subscriptions (per 100 people)",
  "Gini index (World Bank estimate)", # Will no be kept in PCA
  "Gini index (CIA estimate)",
  "Shadow size Economy"
)

# c("Shadow size Economy", "Ecological Footprint", "Gini index (CIA estimate)")


#S6: Policy variables and governance
variables_to_keep_S6 <- c(
  "GovernmentResponseIndex",
  "ContainmentHealthIndex",
  "StringencyIndex",
  "EconomicSupportIndex"
)

variables_to_keep <- c(
  variables_to_keep_S1, variables_to_keep_S2, variables_to_keep_S3,
  variables_to_keep_S4, variables_to_keep_S5, variables_to_keep_S6
)
```

For convenience, a table that contains the list of variable and their type can be created:

```
variables_to_keep_df <-
  tibble(variable = variables_to_keep_S1, type = "S1") |>
  bind_rows(tibble(variable = variables_to_keep_S2, type = "S2")) |>
```

```
    bind_rows(tibble(variable = variables_to_keep_S3, type = "S3")) |>
    bind_rows(tibble(variable = variables_to_keep_S4, type = "S4")) |>
    bind_rows(tibble(variable = variables_to_keep_S5, type = "S5")) |>
    bind_rows(tibble(variable = variables_to_keep_S6, type = "S6"))
  variables_to_keep_df
```

```
# A tibble: 44 x 2
   variable                                                                type
   <chr>                                                                   <chr>
 1 Physicians (per 1,000 people)                                           S1
 2 Hospital beds (per 1,000 people)                                        S1
 3 Nurses and midwives (per 1,000 people)                                  S1
 4 Domestic general government health expenditure (% of general governmen~ S1
 5 Domestic private health expenditure per capita (current US$)            S1
 6 Domestic private health expenditure per capita, PPP (current internati~ S1
 7 Incidence of malaria (per 1,000 population at risk)                     S2
 8 Incidence of HIV, all (per 1,000 uninfected population)                 S2
 9 Incidence of tuberculosis (per 100,000 people)                          S2
10 Diabetes prevalence (% of population ages 20 to 79)                     S2
# i 34 more rows
```

1.1.2.2 WDI data

The study relies on socio-economic data. Most variables are extracted from the World Development Indicators (WDI) from the World Bank.

We downloaded a bulk CSV file version of the database. The data can then easily be loaded into R:

```
  wdi_df <- read_csv("./data/raw/WDI_csv.zip")
```

```
Multiple files in zip: reading 'WDIData.csv'
New names:
Rows: 392882 Columns: 68
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (4): Country Name, Country Code, Indicator Name, Indicator Code
dbl (63): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...
lgl  (1): ...68
```

```
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Since the last column contains only `NA` values, it should be removed:

```
last_col_name <- last(colnames(wdi_df))
wdi_df <-
  wdi_df |> select(-!!last_col_name)
```

Let us extract the variable names:

```
variable_names <- wdi_df$`Indicator Name` |>
  unique() |>
  sort()
head(variable_names)
```

```
[1] "Access to clean fuels and technologies for cooking (% of population)"
[2] "Access to clean fuels and technologies for cooking, rural (% of rural population)"
[3] "Access to clean fuels and technologies for cooking, urban (% of urban population)"
[4] "Access to electricity (% of population)"
[5] "Access to electricity, rural (% of rural population)"
[6] "Access to electricity, urban (% of urban population)"
```

We have selected a few variables from the database. To be able to correctly identify the names of the variables we are interested in, we used the following piece of code (here is an example to identify variables that contain the word '*poverty*'):

```
variable_names[str_detect(variable_names, regex("poverty", ignore_case = TRUE))]
```

Let us rename some countries in the WDI dataset so that their names match those that can be found in the Oxford dataset.

```
wdi_df <-
  wdi_df |>
  mutate(
    `Country Name` = recode(
      `Country Name`,
      # old = new
      "Brunei Darussalam" = "Brunei",
      "Cabo Verde" = "Cape Verde",
      "Congo, Dem. Rep." = "Democratic Republic of Congo",
```

```
      "Congo, Rep." = "Congo",
      "Egypt, Arab Rep." = "Egypt",
      "Gambia, The" = "Gambia",
      "Hong Kong SAR, China" = "Hong Kong",
      "Iran, Islamic Rep." = "Iran",
      "Korea, Rep." = "South Korea",
      "Lao PDR" = "Laos",
      "Russian Federation" = "Russia",
      "Syrian Arab Republic"  = "Syria",
      "Venezuela, RB" = "Venezuela",
      "Yemen, Rep." = "Yemen"
    )
  )
```

1.1.2.3 Ecological Footprint

We extract the ecological footprint data from the Global Footprint Network National Footprint and Biocapacity Accounts. It measures, in gobal hectares, 'how much area of biologically productive land and water an individual, population, or activity requires to produce all the resources it consumes and to absorb the waste it generates, using prevailing technology and resource management practices.'

To get the data, we made some queries to the API, after creating an account (for free). This notebook will present the way to get the data, but to avoid making unecessary requests to the API, the data are loaded from a previous call to the API.

```
library(rvest)
library(httr)
```

The username and API key need to be sent at each call.

```
username <- "replace_with_your_username"
key <- "replace_with_your_key"
```

We created a simple function, as explained in the R vignette of {httr}, to make a call to the API:

```
#' Request made to footprintnetwork.org API
#'
#' @param url endpoint
#' url <- "http://api.footprintnetwork.org/v1/types"
```

```
request_api <- function(url) {
  resp <- GET(url, authenticate(username, key))

  if (http_type(resp) != "application/json") {
    stop("API did not return json", call. = FALSE)
  }

  jsonlite::fromJSON(content(resp, "text"), simplifyVector = TRUE)
}# End of request_api()
```

We defined a function that prepares the URL depending on the country code, the year and the type of data:

```
#' Fetch data from footprintnetwork.org API
#' @param country_code code of the country (e.g., 68 for France, or all for all countries)
#' @param year year
#' @param data_type Record type taken from types list or 'all' (e.g., pop)
get_data_country <- function(country_code, year, data_type) {
  url <- str_c(
    "http://api.footprintnetwork.org/v1/data/",
    country_code, "/", year, "/", data_type
  )
  request_api(url)
}
```

Then, we make a simple call to the API to list of variables that can be asked:

```
types_footprint <- request_api("http://api.footprintnetwork.org/v1/types")
save(types_footprint, file = "./data/raw/footprint/types_footprint.rda")
```

```
load("./data/raw/footprint/types_footprint.rda")
if (knitr::is_html_output()) {
 types_footprint |>
  DT::datatable(options = list(
    searching = TRUE,
    pageLength = 10,
    lengthMenu = c(5, 10, 15, 20)
  ))
}
```

```
knitr::kable(types_footprint)
```

id	version	code	record	note	enabled	name
3	NA	BCpc	BiocapPerCap	Biocapacity divided by population in global hectares (gha)	1	Biocapacity per person
4	NA	BCtot	BiocapTotGHA	Total Biocapacity in global hectares (gha)	1	Biocapacity
5	NA	EFCpc	EFConsPerCap	Ecological Footprint of consumption in global hectares (gha) divided by population	1	Ecological Footprint per person
6	NA	EFCtot	EFConsTotGHA	Total Ecological Footprint of consumption in global hectares (gha)	1	Ecological Footprint
13	NA	earth	BiocapPerCap / EFConsPerCap	BiocapPerCap of Country / EFConsPerCap of World	1	Earths
14	NA	pop	Population	Population	1	Population
15	NA	hdi	HDI	Human Development Index; Source: Trends in the Human Development Index, 1990-2017, downloaded 04/05/2019 from http://hdr.undp.org/en/data	1	Human Development Index
16	NA	gdp	GDP	GDP per capita (constant 2010 US$); Source: World Bank, downloaded 03/13/2019 from http://data.worldbank.org/indicator/NY.GDP.PCAP.KD	1	Gross Domestic Product

Another call is made to get the list of countries:

```
list_countries_footprint <- request_api("http://api.footprintnetwork.org/v1/countries")
save(
  list_countries_footprint,
  file = "./data/raw/footprint/list_countries_footprint.rda"
)
```

```
load("./data/raw/footprint/list_countries_footprint.rda")
if (knitr::is_html_output()) {
  list_countries_footprint |>
  DT::datatable(options = list(
    searching = TRUE,
    pageLength = 10,
    lengthMenu = c(5, 10, 15, 20)
  ))
}
```

Please see the HTML version for the whole table.

```
head(list_countries_footprint)
```

```
  id version countryCode         countryName           shortName isoa2 score
1  1      NA           1             Armenia             Armenia    AM    3A
2  2      NA           2         Afghanistan         Afghanistan    AF    3A
3  3      NA           3             Albania             Albania    AL    3A
4  4      NA           4             Algeria             Algeria    DZ    2A
5  5      NA           7              Angola              Angola    AO    3A
6  6      NA           8 Antigua and Barbuda Antigua and Barbuda    AG    2B
```

Then, we get the data from 2009, 2010 (for later use with the Influenza H1N1 epidemic), as well as the 2017 data (latest available).

```
footprint_2009 <- get_data_country(
  country_code = "all", year = 2009, data_type = "EFCpc"
)
footprint_2010 <- get_data_country(
  country_code = "all", year = 2010, data_type = "EFCpc"
)
footprint_2009_2010 <-
  footprint_2009 |>
  bind_rows(footprint_2010) |>
  tibble()

footprint_2017 <- get_data_country(
  country_code = "all", year = 2017, data_type = "EFCpc"
)
footprint_2017 <- tibble(footprint_2017)

save(footprint_2009_2010, file = "./data/raw/footprint/footprint_2009_2010.rda")
save(footprint_2017, file = "./data/raw/footprint/footprint_2017.rda")
```

```
load("./data/raw/footprint/footprint_2017.rda")
if (knitr::is_html_output()) {
  footprint_2017 |>
    DT::datatable(options = list(
      searching = TRUE,
      pageLength = 10,
      lengthMenu = c(5, 10, 15, 20)
    ))
}
```

Please see the HTML version for the whole table.

```r
head(footprint_2017)
```

```
# A tibble: 6 x 16
     id version  year countryCode countryName    shortName isoa2 record cropLand
  <int> <lgl>   <int>       <int> <chr>          <chr>     <chr> <chr>     <dbl>
1   305 NA       2017           1 Armenia        Armenia   AM    EFCon~    0.424
2   989 NA       2017           2 Afghanistan    Afghanis~ AF    EFCon~    0.252
3  1673 NA       2017           3 Albania        Albania   AL    EFCon~    0.552
4  2357 NA       2017           4 Algeria        Algeria   DZ    EFCon~    0.567
5  3041 NA       2017           7 Angola         Angola    AO    EFCon~    0.289
6  3497 NA       2017           8 Antigua and B~ Antigua ~ AG    EFCon~   NA
# i 7 more variables: grazingLand <dbl>, forestLand <dbl>, fishingGround <dbl>,
#   builtupLand <dbl>, carbon <dbl>, value <dbl>, score <chr>
```

Let us only keep the following variables:

```r
df_footprint <-
  footprint_2017 |>
  select(countryName, year, value)
```

Some countries need to be renamed to match those of the Oxford dataset:

```r
df_footprint <-
  df_footprint |>
  mutate(
    country = countryName |>
      recode(
        "Brunei Darussalam" = "Brunei",
        "Côte d'Ivoire" = "Cote d'Ivoire",
        "Congo, Democratic Republic of" = "Democratic Republic of Congo",
        "Cabo Verde" = "Cape Verde",
        "Iran, Islamic Republic of" = "Iran",
        "Korea, Republic of" = "South Korea",
        "Lao People's Democratic Republic" = "Laos",
        "Russian Federation" = "Russia",
        # "" = "South Sudan",# Missing
        # "" = "Seychelles",#Missing
        "Syrian Arab Republic" = "Syria",
        "Tanzania, United Republic of" = "Tanzania",
        # "" = "Uruguay",# Missing
        "United States of America" = "United States",
```

```
        "Venezuela, Bolivarian Republic of" = "Venezuela",
        "Viet Nam" = "Vietnam"
        # "" = "Vanuatu"# Missing
      )
  )
```

1.1.2.4 Size of the Shadow Economy

We rely on the estimates of the size of the Shadow Economy reported in Medina and Schneider (2019). The data reported in table A.1 from the Appendix of their paper have been manually extracted and saved in a CSV file.

```
df_shadow_economy <- read_csv2("./data/raw/shadow_economy.csv", skip = 3)
```

```
i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
Rows: 157 Columns: 2
-- Column specification --------------------------------------------------------
Delimiter: ";"
chr (1): ISO
num (1): shadow_size

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
if (knitr::is_html_output()) {
  df_shadow_economy |>
  DT::datatable(options = list(
    searching = TRUE,
    pageLength = 10,
    lengthMenu = c(5, 10, 15, 20)
  ))
}
```

Please see the HTML version for the whole table.

```
head(df_shadow_economy)
```

```
# A tibble: 6 x 2
  ISO   shadow_size
  <chr>       <dbl>
1 ALB           270
2 DZA           320
3 AGO           391
4 ARG           209
5 ARM           345
6 AUS           116
```

There will be some missing values:

```
unique(confirmed_df$country_code)[!unique(confirmed_df$country_code) %in% df_shadow_econom
```

```
 [1] "AFG" "DJI" "IRQ" "PAN" "SDN" "SOM" "SSD" "SYC" "TLS" "VUT"
```

1.1.2.5 Gini Index

The WDI dataset does not give estimates for the Gini Index for many countries. We turn to data from the World Factbook, by the CIA.

```
df_cia <-
  read_csv("./data/raw/gini_index_cia.csv") |>
  select(name, date_of_information, value)
```

```
Rows: 173 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (4): name, slug, date_of_information, region
dbl (2): value, ranking

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Some countries need to be renamed to match those of the Oxford dataset:

```
df_cia <- df_cia |>
  mutate(
    country = name |>
      recode(
```

```
        # "" = "Bahrain",#Missing
        # "" = "Brunei",#Missing
        "Congo, Democratic Republic of the" = "Democratic Republic of Congo",
        "Congo, Republic of the" = "Congo",
        "Cabo Verde" = "Cape Verde",
        # "" = "Eritrea",#Missing
        "Gambia, The" = "Gambia",
        "Korea, South" = "South Korea"
        # "" = "Kuwait",#Missing
        # "" = "Myanmar",# Mssing
        # "" = "Oman",#Missin
        # "" = "Somalia",#Missing
        # "" = "Syria"# Missing
      )
  )
```

```
if (knitr::is_html_output()) {
  df_cia |>
    DT::datatable(
      options = list(
        searching = TRUE,
        pageLength = 10,
        lengthMenu = c(5, 10, 15, 20)
      )
    )
}
```

Please see the HTML version for the whole table.

```
head(df_cia)
```

```
# A tibble: 6 x 4
  name                   date_of_information value country
  <chr>                  <chr>               <dbl> <chr>
1 South Africa           2014 est.            63   South Africa
2 Namibia                2015 est.            59.1 Namibia
3 Zambia                 2015 est.            57.1 Zambia
4 Sao Tome and Principe  2017 est.            56.3 Sao Tome and Principe
5 Eswatini               2016 est.            54.6 Eswatini
6 Mozambique             2014 est.            54   Mozambique
```

1.1.2.6 Policy Variables

We extract policy variables from Hale et al. (2021).

```
df_oxford <- read_csv("./data/raw/OxCGRT_latest.csv") |>
  select(country = CountryName,`Date`,all_of(variables_to_keep_S6))
```

```
Rows: 177528 Columns: 51
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (5): CountryName, CountryCode, RegionName, RegionCode, Jurisdiction
dbl (45): Date, C1_School closing, C1_Flag, C2_Workplace closing, C2_Flag, C...
lgl  (1): M1_Wildcard

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_oxford
```

```
# A tibble: 177,528 x 6
   country    Date GovernmentResponseIn~1 ContainmentHealthIndex StringencyIndex
   <chr>     <dbl>                  <dbl>                  <dbl>           <dbl>
 1 Aruba    2.02e7                      0                      0               0
 2 Aruba    2.02e7                      0                      0               0
 3 Aruba    2.02e7                      0                      0               0
 4 Aruba    2.02e7                      0                      0               0
 5 Aruba    2.02e7                      0                      0               0
 6 Aruba    2.02e7                      0                      0               0
 7 Aruba    2.02e7                      0                      0               0
 8 Aruba    2.02e7                      0                      0               0
 9 Aruba    2.02e7                      0                      0               0
10 Aruba    2.02e7                      0                      0               0
# i 177,518 more rows
# i abbreviated name: 1: GovernmentResponseIndex
# i 1 more variable: EconomicSupportIndex <dbl>
```

We filter the data to keep the period from January 2020 to January 2021 and compute an average of the variables of interes (`variables_to_keep_S6`) over the period:

```r
# Aggregation by year
df_oxford_year <-
  df_oxford |>
  mutate(Date = ymd(Date)) |>
  filter(Date >= "2020-01-01" & Date <"2021-01-01") |>
  pivot_longer(cols = variables_to_keep_S6, names_to = "Indicator Name") |>
  group_by(country, `Indicator Name`) |>
  summarise(value = mean(value, na.rm = T)) |>
  mutate(type = "2020") |>
  pivot_wider(names_from = "Indicator Name", values_from = c("value","type")) |>
  select(
    country,
    str_c(
      rep(c("value_","type_"), length(variables_to_keep_S6)),
      rep(variables_to_keep_S6, each = 2)
    ),
  )
```

```
Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(variables_to_keep_S6)

  # Now:
  data %>% select(all_of(variables_to_keep_S6))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.

`summarise()` has grouped output by 'country'. You can override using the
`.groups` argument.
```

The resulting averages can be shown as follows:

```r
if (knitr::is_html_output()) {
 df_oxford_year |>
  DT::datatable(options = list(
    searching = TRUE,
    pageLength = 10,
    lengthMenu = c(5, 10, 15, 20)
  ))
}
```

Please see the HTML version for the whole table.

```
head(df_oxford_year)
```

```
# A tibble: 6 x 9
# Groups:   country [6]
  country   value_GovernmentResp~1 type_GovernmentRespo~2 value_ContainmentHea~3
  <chr>                      <dbl> <chr>                                    <dbl>
1 Afghanis~                   34.8 2020                                     39.4
2 Albania                     51.8 2020                                     52.8
3 Algeria                     48.5 2020                                     50.4
4 Andorra                     44.7 2020                                     40.7
5 Angola                      42.8 2020                                     45.9
6 Argentina                   62.1 2020                                     62.7
# i abbreviated names: 1: value_GovernmentResponseIndex,
#   2: type_GovernmentResponseIndex, 3: value_ContainmentHealthIndex
# i 5 more variables: type_ContainmentHealthIndex <chr>,
#   value_StringencyIndex <dbl>, type_StringencyIndex <chr>,
#   value_EconomicSupportIndex <dbl>, type_EconomicSupportIndex <chr>
```

1.1.2.7 Creation of the Datasets

We create a function that allows us to create and export a database for each group of countries. This database contains the 2020 value for each variable. When the 2020 value was missing in the data, we took the average value over the last 5 years. For the remaining missing values, we took the 10 years average, or the 15 years average. When there was no data available in the selected years, we replaced the missing value by the average of the region.

```
#'@param countries group of countries
#'@param file export database
get_out_countries <- function(countries, file) {

  df_countries_wdi <-
    wdi_df |>
    # Keeping countries of interest
    filter(`Country Name` %in% !!countries) |>
    # And variables of interest
    filter(`Indicator Name` %in% !!variables_to_keep) |>
    pivot_longer(cols = `1960`:`2020`, names_to = "year") |>
    select(ISO = `Country Code`, country = `Country Name`, variable = `Indicator Name`, ye
```

```r
df_countries_wdi <-
  df_countries_wdi |>
  mutate(year = as.numeric(year)) |>
  filter(year >= 2005) |>
  mutate(
    value_2020    = ifelse(year == 2020, yes = value, no = NA),
    value_last_5  = ifelse(year %in% seq(2015,2020), yes = value, no = NA),
    value_last_10 = ifelse(year %in% seq(2010,2020), yes = value, no = NA),
    value_last_15 = ifelse(year %in% seq(2005,2020), yes = value, no = NA)
  ) |>
  pivot_longer(
    cols = value_2020:value_last_15,
    names_to = "var",
    values_to = "replacement_val"
  ) |>
  group_by(ISO, country, variable, var) |>
  summarise(
    # nb_val = sum(!is.na(replacement_val)),
    replacement_val = mean(replacement_val, na.rm = TRUE)
  ) |>
  pivot_wider(names_from = var, values_from = replacement_val) |>
  mutate(
    # If missing value: average of the 5 previous years
    type = ifelse(
      is.na(value_2020) | is.nan(value_2020),
      yes = "last 5", no = "value 2020"
    ),
    value = ifelse(
      is.na(value_2020) | is.nan(value_2020),
      yes = value_last_5, no = value_2020
    )
  ) |>
  mutate(
    # If missing value: average of the 10 previous years
    type = ifelse(
      is.na(value) | is.nan(value),
      yes = "last 10", no = type
    ),
    value = ifelse(
      is.na(value) | is.nan(value),
      yes = value_last_10, no = value
```

```
    )
  ) |>
  mutate(
    # If missing value: average of the 15 previous years
    type = ifelse(
      is.na(value) | is.nan(value),
      yes = "last 15", no = type
    ),
    value = ifelse(
      is.na(value) | is.nan(value),
      yes = value_last_15, no = value
    )
  ) |>
  mutate(
    # If missing value: NA
    type = ifelse(is.na(value) | is.nan(value), yes = "flag", no = type),
    value = ifelse(is.na(value) | is.nan(value), yes = NA, no = value)
  ) |>
  select(-value_2020, -value_last_5, -value_last_10, -value_last_15)

# Malaria set to 0 when it is NA, in 2020
df_countries_wdi <-
  df_countries_wdi |>
  mutate(
    value = ifelse(
      variable == "Incidence of malaria (per 1,000 population at risk)" &
        is.na(value),
      yes = 0, no = value)
  )

# Ecological Footprint
df_countries_footprint <-
  # corresp_iso |>
  # filter(country %in% countries) |>
  # left_join(df_footprint, by = c("ISO", "country")) |>
  df_footprint |>
  filter(country %in% !!countries) |>
  mutate(variable = "Ecological Footprint") |>
  mutate(type = str_c(year, " est.")) |>
  select(-year, -countryName)
```

```r
# Gini
df_countries_gini <-
  # corresp_iso |>
  # filter(country %in% countries) |>
  # left_join(df_cia, by = c("ISO", "country")) |>
  df_cia |>
  filter(country %in% !!countries) |>
  rename(type = date_of_information) |>
  select(country, value, type) |>
  mutate(variable = "Gini index (CIA estimate)")


# Shadow Size of the epidemic
df_shadow_size <-
  confirmed_df |>
  select(country, country_code) |>
  unique() |>
  filter(country %in% countries) |>
  left_join(
    df_shadow_economy, by = c("country_code" = "ISO")
  ) |>
  rename(value = shadow_size) |>
  mutate(
    variable = "Shadow size Economy",
    type = "2015 est."
  ) |>
  select(country, value, type, variable)


df_countries <-
  df_countries_wdi |> ungroup() |> select(-ISO) |>
  bind_rows(df_countries_footprint) |>
  bind_rows(df_countries_gini) |>
  bind_rows(df_shadow_size) |>
  ungroup()

df_countries <-
  df_countries |>
  complete(country, variable)

df_countries <-
```

```
  df_countries |>
  mutate(type = ifelse(is.na(type), yes = "flag", no = type))

# Average within the zone
group_average <-
  df_countries |>
  group_by(variable) |>
  summarise(value_group = mean(value, na.rm=TRUE))


# Replacing missing values with group average
df_countries <-
  df_countries |>
  left_join(group_average, by = "variable") |>
  mutate(
    # If missing value: group average
    type = ifelse(
      is.na(value) | is.nan(value), yes = "group average", no = type
    ),
    value = ifelse(
      is.na(value) | is.nan(value), yes = value_group, no = value
    )
  ) |>
  select(-value_group)

# Variables in columns
df_countries <-
  df_countries |>
  pivot_wider(names_from = variable, values_from = c(type, value))
# select(
#   ISO, country, str_c(c("value_", "type_"), rep(variables_to_keep, each = 2))
# )

df_countries <-
  df_countries |>
  # Computing Urban density
  mutate(`value_Urban density` = `value_Urban population` / `value_Urban land area (sq.
         `type_Urban density` = `type_Urban population`)

# Adding Oxford indicators
df_countries_excel <-
```

```
    df_countries |>
    left_join(df_oxford_year, by = "country")

  # Table with summary of missing data
  df_missing <-
    df_countries_excel |>
    select(country, starts_with("type_")) |>
    pivot_longer(cols = -c(country), names_to = "variable") |>
    group_by(variable, value) |>
    summarise(n = n()) |>
    group_by(variable) |>
    mutate(freq = scales::percent(n / sum(n), accuracy = 0.1)) |>
    select(-n) |>
    pivot_wider(names_from = value, values_from = freq, values_fill = "") |>
    mutate(variable = str_remove(variable, "^type_"),
           variable = factor(variable, levels = variables_to_keep)) |>
    arrange(variable)

  # reordering the columns
  vn <- c("variable", "2020", "last 5", "last 10", "last 15", "group average")
  vn <- vn[vn %in% colnames(df_missing)]
  vn2 <- sort(colnames(df_missing)[!colnames(df_missing)%in%vn])

  df_missing <-
    df_missing |>
    select(!!vn, !!vn2)

  write_excel_csv(df_countries_excel, file = file)

  list(df_countries_excel = df_countries_excel, df_missing = df_missing)
} # End of get_out_countries()
```

1.1.2.7.1 Mena

The data for MENA:

```
mena
```

```
 [1] "Algeria"      "Bahrain"      "Egypt"        "Iran"         "Iraq"
 [6] "Israel"       "Jordan"       "Kuwait"       "Lebanon"      "Morocco"
[11] "Oman"         "Qatar"        "Saudi Arabia" "Syria"        "Tunisia"
```

```
[16] "Yemen"
```

```
file <- "./data/output/covid/df_mena.csv"

res_mena <- get_out_countries(countries=mena, file=file)
```

```
`summarise()` has grouped output by 'ISO', 'country', 'variable'. You can
override using the `.groups` argument.
`summarise()` has grouped output by 'variable'. You can override using the
`.groups` argument.
```

```
df_mena_missing <- res_mena$df_missing
df_mena <- res_mena$df_countries_excel
```

```
if (knitr::is_html_output()) {
  df_mena_missing |>
  DT::datatable(options = list(
    searching = TRUE,
    pageLength = 10,
    lengthMenu = c(5, 10, 15, 20)
  ))
}
```

Please see the HTML version for the whole table.

```
head(df_mena_missing)
```

```
# A tibble: 6 x 17
# Groups:   variable [6]
  variable            `2020` `last 5` `last 10` `last 15` `group average` `2007`
  <fct>               <chr>  <chr>    <chr>     <chr>     <chr>           <chr>
1 Physicians (per 1,~ ""     62.5%    "6.2%"    ""        ""              ""
2 Hospital beds (per~ ""     100.0%   ""        ""        ""              ""
3 Nurses and midwive~ ""     87.5%    ""        ""        ""              ""
4 Domestic general g~ ""     12.5%    "6.2%"    ""        ""              ""
5 Domestic private h~ ""     12.5%    "6.2%"    ""        ""              ""
6 Domestic private h~ ""     12.5%    "6.2%"    ""        ""              ""
# i 10 more variables: `2010 est.` <chr>, `2011 est.` <chr>, `2012 est.` <chr>,
#   `2013 est.` <chr>, `2014 est.` <chr>, `2015 est.` <chr>, `2016 est.` <chr>,
#   `2017 est.` <chr>, flag <chr>, `value 2020` <chr>
```

1.1.2.7.2 Sub-Saharan Africa

```
ssa
```

```
 [1] "Angola"                         "Benin"
 [3] "Burkina Faso"                   "Burundi"
 [5] "Cape Verde"                     "Cameroon"
 [7] "Central African Republic"       "Chad"
 [9] "Democratic Republic of Congo"   "Congo"
[11] "Cote d'Ivoire"                  "Djibouti"
[13] "Ethiopia"                       "Gabon"
[15] "Gambia"                         "Ghana"
[17] "Guinea"                         "Kenya"
[19] "Lesotho"                        "Liberia"
[21] "Madagascar"                     "Malawi"
[23] "Mali"                           "Mauritania"
[25] "Mauritius"                      "Mozambique"
[27] "Namibia"                        "Niger"
[29] "Nigeria"                        "Rwanda"
[31] "Senegal"                        "Seychelles"
[33] "Sierra Leone"                   "Somalia"
[35] "South Africa"                   "South Sudan"
[37] "Sudan"                          "Eswatini"
[39] "Tanzania"                       "Togo"
[41] "Uganda"                         "Zimbabwe"
```

```
file <- "./data/output/covid/df_sub_saharian_africa.csv"

res_ssa <- get_out_countries(countries=ssa, file=file)
```

```
`summarise()` has grouped output by 'ISO', 'country', 'variable'. You can
override using the `.groups` argument.
`summarise()` has grouped output by 'variable'. You can override using the
`.groups` argument.
```

```
df_ssa_missing <- res_ssa$df_missing
df_ssa <- res_ssa$df_countries_excel
```

```
if (knitr::is_html_output()) {
  df_ssa_missing |>
    DT::datatable(options = list(
      searching = TRUE,
      pageLength = 10,
      lengthMenu = c(5, 10, 15, 20)
    ))
}
```

Please see the HTML version for the whole table.

```
head(df_ssa_missing)
```

```
# A tibble: 6 x 19
# Groups:   variable [6]
  variable            `2020` `last 5` `last 10` `last 15` `group average` `2001`
  <fct>               <chr>  <chr>    <chr>     <chr>     <chr>           <chr>
1 Physicians (per 1,~ ""     "64.3%"  "4.8%"    ""        ""              ""
2 Hospital beds (per~ ""     "11.9%"  "57.1%"   "26.2%"   "4.8%"          ""
3 Nurses and midwive~ ""     "66.7%"  "4.8%"    ""        ""              ""
4 Domestic general g~ ""     ""       ""        ""        "2.4%"          ""
5 Domestic private h~ ""     ""       ""        ""        "2.4%"          ""
6 Domestic private h~ ""     ""       ""        ""        "2.4%"          ""
# i 12 more variables: `2003 est.` <chr>, `2010 est.` <chr>, `2011 est.` <chr>,
#   `2012 est.` <chr>, `2013 est.` <chr>, `2014 est.` <chr>, `2015 est.` <chr>,
#   `2016 est.` <chr>, `2017 est.` <chr>, `2018 est.` <chr>, flag <chr>,
#   `value 2020` <chr>
```

1.1.2.7.3 Latin America

```
latin_america
```

```
 [1] "Argentina"   "Bolivia"     "Brazil"      "Chile"       "Colombia"
 [6] "Ecuador"     "El Salvador" "Guatemala"   "Haiti"       "Honduras"
[11] "Jamaica"     "Mexico"      "Nicaragua"   "Panama"      "Paraguay"
[16] "Peru"        "Uruguay"     "Venezuela"
```

```
file <- "./data/output/covid/df_latin_america.csv"
```

```
res_la <- get_out_countries(countries=latin_america, file=file)
```

```
`summarise()` has grouped output by 'ISO', 'country', 'variable'. You can
override using the `.groups` argument.
`summarise()` has grouped output by 'variable'. You can override using the
`.groups` argument.
```

```
df_la_missing <- res_la$df_missing
df_la <- res_la$df_countries_excel
```

```
if (knitr::is_html_output()) {
  df_la_missing |>
    DT::datatable(options = list(
      searching = TRUE,
      pageLength = 10,
      lengthMenu = c(5, 10, 15, 20)
    ))
}
```

Please see the HTML version for the whole table.

```
head(df_la_missing)
```

```
# A tibble: 6 x 15
# Groups:   variable [6]
  variable            `2020` `last 5` `last 10` `last 15` `group average` `2011`
  <fct>               <chr>  <chr>    <chr>     <chr>     <chr>           <chr>
1 Physicians (per 1,~ ""     "66.7%"  ""        ""        ""              ""
2 Hospital beds (per~ ""     "94.4%"  "5.6%"    ""        ""              ""
3 Nurses and midwive~ ""     "83.3%"  ""        ""        ""              ""
4 Domestic general g~ ""     ""       ""        ""        ""              ""
5 Domestic private h~ ""     ""       ""        ""        ""              ""
6 Domestic private h~ ""     ""       ""        ""        ""              ""
# i 8 more variables: `2012 est.` <chr>, `2014 est.` <chr>, `2015 est.` <chr>,
#   `2016` <chr>, `2017 est.` <chr>, `2018 est.` <chr>, flag <chr>,
#   `value 2020` <chr>
```

1.1.2.7.4 Asia

```
asia
```

```
 [1] "Afghanistan"     "Bangladesh"      "Bhutan"          "Brunei"
 [5] "Cambodia"        "India"           "Indonesia"       "Laos"
 [9] "Malaysia"        "Mongolia"        "Myanmar"         "Nepal"
[13] "Pakistan"        "Philippines"     "Russia"          "Solomon Islands"
[17] "South Korea"     "Sri Lanka"       "Thailand"        "Timor-Leste"
[21] "Vanuatu"         "Vietnam"         "China"
```

```
file <- "./data/output/covid/df_asia.csv"

res_asia <- get_out_countries(countries=asia, file=file)
```

```
`summarise()` has grouped output by 'ISO', 'country', 'variable'. You can
override using the `.groups` argument.
`summarise()` has grouped output by 'variable'. You can override using the
`.groups` argument.
```

```
df_asia_missing <- res_asia$df_missing
df_asia <- res_asia$df_countries_excel
```

```
if (knitr::is_html_output()) {
  df_asia_missing |>
    DT::datatable(options = list(
      searching = TRUE,
      pageLength = 10,
      lengthMenu = c(5, 10, 15, 20)
    ))
}
```

Please see the HTML version for the whole table.

```
head(df_asia_missing)
```

```
# A tibble: 6 x 19
# Groups:   variable [6]
  variable            `2020` `last 5` `last 10` `last 15` `group average` `2008`
  <fct>               <chr>  <chr>    <chr>     <chr>     <chr>           <chr>
```

```
1 Physicians (per 1,~ ""     "39.1%"  "4.3%"    ""        ""              ""
2 Hospital beds (per~ ""     "60.9%"  "34.8%"   "4.3%"    ""              ""
3 Nurses and midwive~ ""     "65.2%"  ""        ""        ""              ""
4 Domestic general g~ ""     ""       ""        ""        ""              ""
5 Domestic private h~ ""     ""       ""        ""        ""              ""
6 Domestic private h~ ""     ""       ""        ""        ""              ""
# i 12 more variables: `2008 est.` <chr>, `2010 est.` <chr>, `2011 est.` <chr>,
#   `2012 est.` <chr>, `2013 est.` <chr>, `2014 est.` <chr>, `2015 est.` <chr>,
#   `2016 est.` <chr>, `2017 est.` <chr>, `2018 est.` <chr>, flag <chr>,
#   `value 2020` <chr>
```

1.1.2.7.5 Industrialized Countries

```
indus
```

```
 [1] "Australia"      "Austria"        "Belgium"        "Canada"
 [5] "Denmark"        "France"         "Germany"        "Greece"
 [9] "Ireland"        "Italy"          "Japan"          "Netherlands"
[13] "Spain"          "Sweden"         "United Kingdom" "United States"
```

```
file <- "./data/output/df_industrialized_countries.csv"

res_indus <- get_out_countries(countries=indus, file=file)
```

```
`summarise()` has grouped output by 'ISO', 'country', 'variable'. You can
override using the `.groups` argument.
`summarise()` has grouped output by 'variable'. You can override using the
`.groups` argument.
```

```
df_indus_missing <- res_indus$df_missing
df_indus <- res_indus$df_countries_excel
```

```
if (knitr::is_html_output()) {
  df_indus_missing |>
    DT::datatable(options = list(
      searching = TRUE,
      pageLength = 10,
      lengthMenu = c(5, 10, 15, 20)
    ))
```

```
}
```

Please see the HTML version for the whole table.

```
head(df_indus_missing)
```

```
# A tibble: 6 x 12
# Groups:   variable [6]
  variable     `2020` `last 5` `last 10` `group average` `2013 est.` `2014 est.`
  <fct>        <chr>  <chr>    <chr>     <chr>           <chr>       <chr>
1 Physicians ~ ""     "56.2%"  ""        ""              ""          ""
2 Hospital be~ ""     "100.0%" ""        ""              ""          ""
3 Nurses and ~ ""     "75.0%"  ""        ""              ""          ""
4 Domestic ge~ ""     ""       ""        ""              ""          ""
5 Domestic pr~ ""     ""       ""        ""              ""          ""
6 Domestic pr~ ""     ""       ""        ""              ""          ""
# i 5 more variables: `2015 est.` <chr>, `2016 est.` <chr>, `2017 est.` <chr>,
#   flag <chr>, `value 2020` <chr>
```

1.1.2.7.6 All countries

Let us aggregate the results in a single table:

```
df_all_covid <-
  df_mena |> mutate(group = "MENA") |>
  bind_rows(df_ssa |> mutate(group = "SSA")) |>
  bind_rows(df_la |> mutate(group = "Latin America")) |>
  bind_rows(df_asia |> mutate(group = "Asia")) |>
  bind_rows(df_indus |> mutate(group = "Industrialized"))

save(df_all_covid, file = "./data/output/covid/df_all_countries.rda")
```

1.1.3 Socio-Economic Data

1.2 Population Data

For the epidemiological models, we need data on the population size of the countries. Let us extract this information from the WDI data

```
population <-
  wdi_df |>
  filter(`Indicator Name` == "Population, total") |>
  select(country = `Country Name`, ISO = `Country Code`,`2015`:`2019`) |>
  pivot_longer(cols = -c(country, ISO)) |>
  group_by(country, ISO) |>
  summarise(pop = mean(value, na.rm=TRUE)) |>
  ungroup()
```

```
`summarise()` has grouped output by 'country'. You can override using the
`.groups` argument.
```

```
save(population, file = "./data/output/population.rda")
```

```
if (knitr::is_html_output()) {
  population |>
    DT::datatable(options = list(
      searching = TRUE,
      pageLength = 10,
      lengthMenu = c(5, 10, 15, 20)
    ))
}
```

Please see the HTML version for the whole table.

```
head(population)
```

```
# A tibble: 6 x 3
  country                     ISO          pop
  <chr>                       <chr>      <dbl>
1 Afghanistan                 AFG    35697881.
2 Africa Eastern and Southern AFE   633226547.
3 Africa Western and Central  AFW   431312070.
4 Albania                     ALB     2870166.
5 Algeria                     DZA    41130281.
6 American Samoa              ASM       49405.
```

1.3 Summary

File	Description	Source
./data/output/covid_cases.rda	Daily cumulative number of Covid-19 cases	JHU
./data/output/covid_df_all_countries.csv	Socio-economic variables for countries for the Covid-19 period	World Development Indicators, Global Footprint Network National Footprint and Biocapacity Accounts, Medina and Schneider (2019), CIA, Hale et al. (2021)
./data/output/population.rda	Population	World Development Indicators

Part II

Estimation of a generalized Richards model of epidemic curve

2 Estimation of a generalized Richards model of epidemic curve

This section provides the codes used to estimate the models.

2.1 Load Data

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```
rm_NA_headtail <- function(x) {
  cumsum(!is.na(x)) != 0 & rev(cumsum(!is.na(rev(x)))) != 0
}
```

We divide the countries according to the following geo-political groups, as in the first part.

```
load("./data/output/list_countries.rda")
```

Please see the HTML version for a dynamic table.

```
head(list_countries)
```

```
# A tibble: 6 x 2
  country group
  <chr>   <chr>
1 Algeria MENA
2 Bahrain MENA
3 Egypt   MENA
4 Iran    MENA
5 Iraq    MENA
6 Israel  MENA
```

The names of the countries can be stored in a variable:

```
names_countries <- list_countries$country |> unique()
```

The data from the Covid-19 epidemic can be loaded as follows (see 01_Load_Data.html for more details):

```
load("./data/output/covid_cases.rda")
covid_cases
```

```
# A tibble: 61,410 x 5
   country     date       country_code value stringency_index
   <chr>       <date>     <chr>        <dbl>             <dbl>
 1 Afghanistan 2020-01-22 AFG              0                 0
 2 Afghanistan 2020-01-23 AFG              0                 0
 3 Afghanistan 2020-01-24 AFG              0                 0
 4 Afghanistan 2020-01-25 AFG              0                 0
 5 Afghanistan 2020-01-26 AFG              0                 0
 6 Afghanistan 2020-01-27 AFG              0                 0
 7 Afghanistan 2020-01-28 AFG              0                 0
 8 Afghanistan 2020-01-29 AFG              0                 0
 9 Afghanistan 2020-01-30 AFG              0                 0
10 Afghanistan 2020-01-31 AFG              0                 0
# i 61,400 more rows
```

The population data also need to be loaded:

```
load("./data/output/population.rda")
```

Some country names need to be changed to match those found in the Oxford data:

```r
population <-
  population |>
  mutate(
    country_wdi = country,
    country_wdi  = recode(
      country_wdi,
      "Brunei Darussalam" = "Brunei",
      "Cabo Verde" = "Cape Verde",
      "Congo, Dem. Rep." = "Democratic Republic of Congo",
      "Congo, Rep." = "Congo",
      "Egypt, Arab Rep." = "Egypt",
      "Gambia, The" = "Gambia",
      "Iran, Islamic Rep." = "Iran",
      "Lao PDR" = "Laos",
      "Russian Federation" = "Russia",
      "Syrian Arab Republic" = "Syria",
      "Venezuela, RB" = "Venezuela",
      "Yemen, Rep." = "Yemen"
    )
  )
```

2.2 Functions

Our approach is based on an empirical approach. We estimate the parameters of a generalized Richards equation which belongs to the family of exponential growth models, extensively used in the literature:

$$\Delta C_t = C_{t+1} - C_t = rC_t^P \left[1 - \left(\frac{C_t}{K}\right)^\alpha\right] + \varepsilon_t, \quad P \in [0, 1].$$

where ε_t is the model error and is distributed as a Gaussian noise process $N(0, \sigma_\varepsilon^2)$ $\forall t$.

The four parameters will be estimated using non linear last square (using `nls.lm()` from {`minpack.lm`}).

Let us load the required package:

```r
library(minpack.lm)
```

We need to define the function objective function, i.e., the one that will be optimized:

```
#' Generalized Richards
#'
#' @param theta list of named parameters (r, P, alpha, and K)
#' @param x numeric, cumulative number of infected
generalized_richards_f <- function(theta, x) {
  r <- theta[["r"]];
  P <- theta[["P"]];
  alpha <- theta[["alpha"]] ;
  K <- theta[["K"]] ;
  r * x^P *(1-(x/K)^alpha) + x
}
```

The function that will be minimized, the quadratic error:

```
#' Loss function
#' @param theta list of named parameters
#' @param fun function to be optimized
#' @param y observed number of new infected
#' @param x cumulative number of infected
loss_function <- function(theta,
                          fun,
                          y,
                          x) {
  (y - fun(theta = theta, x = x))^2
}
```

Let us also define a function to plot multiple graphs on a single figure, with a shared legend:

```
g_legend <- function(a_gplot) {
  tmp <- ggplot_gtable(ggplot_build(a_gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)
}
```

We can define a theme for the graphs, as in the data section:

```
#' Theme for ggplot2
#'
#' @param ... arguments passed to the theme function
#' @export
#' @importFrom ggplot2 element_rect element_text element_blank element_line unit
```

```
#'   rel
theme_paper <- function (...) {
  theme(
    text = element_text(family = "Times"),
    plot.background = element_rect(fill = "transparent", color = NA),
    panel.background = element_rect(fill = "transparent", color = NA),
    panel.border = element_rect(fill = NA, colour = "grey50", linewidth = 1),
    axis.text = element_text(),
    legend.text = element_text(size = rel(1.1)),
    legend.title = element_text(size = rel(1.1)),
    legend.background = element_rect(fill = "transparent", color = NULL),
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.box = "vertical",
    legend.key = element_blank(),
    panel.spacing = unit(1, "lines"),
    panel.grid.major = element_line(colour = "grey90"),
    panel.grid.minor = element_blank(),
    plot.title = element_text(hjust = 0, size = rel(1.3), face = "bold"),
    plot.title.position = "plot",
    plot.margin = unit(c(1, 1, 1, 1), "lines"),
    strip.background = element_rect(fill = NA, colour = NA),
    strip.text = element_text(size = rel(1.1))
  )
}

save(theme_paper, file="./assets/theme_paper.rda")
```

2.3 Estimations

2.3.1 For a Single Country

Before estimating the parameters for all countries, let us go through an example for a single country, to better understand the codes.

Let us consider the first country:

```
i <- 1
country <- names_countries[i]
country
```

```
[1] "Algeria"
```

The population of the country:

```
pop_country <- population |>
  filter(country_wdi == !!country) |>
  pull(pop)
pop_country
```

```
[1] 41130281
```

The Oxford data need to be filtered to keep only those of the current country:

```
dat <-
  covid_cases |>
  filter(country %in% !!country)
```

Let us create C_{t+1} (C) and C_t (C_m):

```
dat <- dat |>
  mutate(C = lead(value), C_m = value) |>
  filter(rm_NA_headtail(C))
```

2.3.1.1 Non Linear Leat Square

We need to define starting valued for the optimization algorithm:

```
# starting param values
start <- list(
  r = 1,
  P  = .5,
  alpha = .5,
  K = .6*pop_country
)
```

Let us estimate the four parameters using `nls.lm()` from {`minpack.lm`}:

```
out <- nls.lm(
  par = start,
  fn = loss_function,
```

```
  y = dat$C,
  x = dat$C_m,
  fun = generalized_richards_f,
  control = nls.lm.control(maxiter = 1024),
  lower = c(r=0,P=0,alpha=0,K=0),
  upper = c(r=100,P=1,alpha=Inf,K=pop_country)
)
```

The estimated parameters:

```
nls_estimates <- out$par
nls_estimates
```

```
$r
[1] 26.29748

$P
[1] 0.2884883

$alpha
[1] 0.1584229

$K
[1] 41130281
```

Those can be used to compute the fitted values:

```
fit <- generalized_richards_f(nls_estimates, dat$C_m)
head(fit, 100)
```

```
  [1]     0.00000     0.00000     0.00000     0.00000     0.00000     0.00000
  [7]     0.00000     0.00000     0.00000     0.00000     0.00000     0.00000
 [13]     0.00000     0.00000     0.00000     0.00000     0.00000     0.00000
 [19]     0.00000     0.00000     0.00000     0.00000     0.00000     0.00000
 [25]     0.00000     0.00000     0.00000     0.00000     0.00000     0.00000
 [31]     0.00000    14.45817    17.67269    19.90339    21.67474    23.17191
 [37]    24.48413    27.73068    31.13966    38.11410    43.47919    49.94947
 [43]    55.53948    61.19269    66.11604    70.21619    72.32788    75.41277
 [49]    77.66918    82.53435    89.29164    96.86893   105.42271   116.50872
 [55]   128.94718   141.17733   160.05001   187.35624   217.76008   252.08860
 [61]   289.64705   334.94328   385.74821   435.30017   483.60924   538.33976
```

```
 [67]  607.68943  690.68949  784.32650  898.88698 1018.06442 1138.50904
 [73] 1262.97097 1374.20694 1481.20744 1581.37989 1668.16490 1752.49283
 [79] 1839.66129 1921.75711 2009.92753 2095.97140 2183.99208 2279.97786
 [85] 2383.47793 2487.77367 2594.90935 2702.84616 2812.02437 2919.53768
 [91] 3022.63108 3127.56261 3236.94691 3352.95833 3474.57532 3610.64030
 [97] 3755.48371 3904.31661 4054.82115 4212.93494
```

We can compute the Root Mean Square Error:

```
(err_nls <- sqrt(sum((dat$C - fit)^2)))
```

```
[1] 4997.746
```

2.3.2 For all Countries

Let us wrap the code used for a single country in a function.

```
estimate_country <- function(i) {
  country <- names_countries[i]

  pop_country <- population |>
    filter(country == !!country) |>
    pull(pop)

  # Filtering the Oxford data to keep only those of the current country
  dat <-
    covid_cases |>
    filter(country %in% !!country) |>
    filter(rm_NA_headtail(value))

  if(nrow(dat) == 0){
    return(NULL)
  }

  dat <- dat |>
    mutate(C = lead(value), C_m = value) |>
    filter(rm_NA_headtail(C))

  # starting param values
  start <- list(
    r = 1,
```

```
  P  = .5,
  alpha = .5,
  K = .6*pop_country
)

# non linear least square estimation
out <- try(
  nls.lm(
    par = start,
    fn = loss_function,
    y = dat$C,
    x = dat$C_m,
    fun = generalized_richards_f,
    control = nls.lm.control(maxiter = 1024),
    lower = c(r=0, P=0, alpha=0, K=0),
    upper = c(r=100, P=1, alpha=Inf, K=pop_country)
  )
)

if (!inherits(out, "try-error")) {
  # fitted values
  fit <- generalized_richards_f(out$par, dat$C_m)

  # extract estimated model parameters
  nls_estimates <- out$par

  # RMSE
  err_nls = sqrt(sum((dat$C - fit)^2))
} else {
  fit <- nls_estimates <- err_nls <- NULL
}

resul_c <-
  list(
    nls = list(
      estimates = nls_estimates,
      fit = fit,
      rmse = err_nls
    ),
    dat = dat
  )
```

```
  resul_c
}# End of estimate_country()
```

Then, let us loop over the countries. As each run is independent from the others, we can perform the computation in parallel. Please note that in this notebook, we will only load the results obtained using the same codes and not actually run those here.

```
library(parallel)
ncl <- detectCores()-1
(cl <- makeCluster(ncl))
```

Some packages need to be run in each cluster:

```
clusterEvalQ(cl, {
    library(dplyr)
    library(stringr)
  library(minpack.lm)
  }) |>
    invisible()
```

Some data and functions also need to be sent to the clusters:

```
clusterExport(cl, c("names_countries", "population", "covid_cases", "rm_NA_headtail"))
clusterExport(cl, c("generalized_richards_f", "loss_function"))
```

Then we can loop over the index of `names_countries`, using `pblapply()` from {`pbapply`}:

```
resul_fit_countries <- pbapply::pblapply(
  1:length(names_countries), estimate_country, cl = cl
)
names(resul_fit_countries) <- names_countries
save(resul_fit_countries, file = "./estim/resul_fit_countries.rda")
stopCluster(cl)
```

Part III

Measuring the influence of socioeconomic variables on P and alpha

3 Principal Component Analysis

Some packages are needed:

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(FactoMineR)
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ggcorrplot)
library(stats)
library(knitr)
library(rmarkdown)
library(kableExtra)
```

```
Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows
```

```r
library(DT)
```

3.1 Load data

```r
load("./data/output/covid/df_all_countries.rda")
df_all_covid
```

```
# A tibble: 115 x 88
   country type_Air transport, p~1 type_Diabetes preval~2 type_Domestic genera~3
   <chr>   <chr>                   <chr>                  <chr>
 1 Algeria value 2020              last 10                value 2020
 2 Bahrain value 2020              last 10                value 2020
 3 Egypt   value 2020              last 10                value 2020
 4 Iran    value 2020              last 10                value 2020
 5 Iraq    value 2020              last 10                value 2020
 6 Israel  value 2020              last 10                value 2020
 7 Jordan  value 2020              last 10                value 2020
 8 Kuwait  value 2020              last 10                value 2020
 9 Lebanon value 2020              last 10                value 2020
10 Morocco value 2020              last 10                value 2020
# i 105 more rows
# i abbreviated names: 1: `type_Air transport, passengers carried`,
#   2: `type_Diabetes prevalence (% of population ages 20 to 79)`,
#   3: `type_Domestic general government health expenditure (% of general government expendi
# i 84 more variables:
#   `type_Domestic private health expenditure per capita (current US$)` <chr>,
#   `type_Domestic private health expenditure per capita, PPP (current international $)` <ch
```

Let us check if there are duplicate countries:

```r
sum(duplicated(df_all_covid))
```

```
[1] 0
```

3.2 Principal Component Analysis

Let us define a function to perform a principal component analysis. It allows us to obtain the coordinates weighted by the eigenvalues for each factor S1, S2, S3, S4, S5, S6.

```r
#' Performs PCA on a group of variables (S1, S2, S3, S4, S5, or S6)
#'
#' @param variables_to_keep group of variables concerned
#' @param type (string) name of group of variables
get_res_acp <- function(variables_to_keep, type) {

  # Keep only the variables considered in the database
  df_acp <-
    df_all_covid |>
    select(country, str_c("value_", variables_to_keep)) |>
    rename_with(~str_remove(., "^value_"))

  # Take Country name like row_name
  df_acp <- data.frame(df_acp, row.names = 1, check.names = F)

  # PCA
  pca_res <- PCA(
    df_acp,
    graph = FALSE,
    scale.unit = TRUE,
    ncp = length(variables_to_keep)
  )

  # Obtain the coordinate of individuals
  ind_data <- get_pca_ind(pca_res)
  ind_coord <- data.frame(ind_data$coord)

  # Obtain the proportion of eigenvalues relating to each factor
  eigen <- get_eigenvalue(pca_res)
  weights <- data.frame(eigen[,2]/100)

  # Coordinates
  df_coord <- data.frame(crossprod(t(ind_coord),weights[,1]))
  df_coord <- df_coord |>
    mutate(country = rownames(df_coord))
  rownames(df_coord) <- NULL

  names(df_coord) <- c(paste("coordinate",type,sep ="_"),"country")

  names(weights) <- "prop_var_expl"
  rownames(weights) <- NULL
```

```
  weights <- weights |> mutate(dim = row_number())

  list(
    coord = df_coord,
    weights = weights
  )


} # End of get_res_acp
```

3.2.1 S1: healthcare infrastructure

```
variables_to_keep <- c(
  "Physicians (per 1,000 people)",
  "Hospital beds (per 1,000 people)",
  "Nurses and midwives (per 1,000 people)",
  "Domestic general government health expenditure (% of general government expenditure)",
  "Domestic private health expenditure per capita (current US$)",
  "Domestic private health expenditure per capita, PPP (current international $)"
)

pca_s1 <- get_res_acp(
  variables_to_keep = variables_to_keep,
  type = "s1"
)

df_coordinate_s1 <- pca_s1$coord
prop_var_expl_s1 <- pca_s1$weights
```

3.2.2 S2: vulnerability to comorbidities

```
variables_to_keep <- c(
  "Incidence of malaria (per 1,000 population at risk)",
  "Incidence of HIV, all (per 1,000 uninfected population)",
  "Incidence of tuberculosis (per 100,000 people)",
  "Diabetes prevalence (% of population ages 20 to 79)",
  "Mortality from CVD, cancer, diabetes or CRD between exact ages 30 and 70 (%)"
  )
```

```r
pca_s2 <- get_res_acp(
  variables_to_keep = variables_to_keep,
  type = "s2"
)

df_coordinate_s2 <- pca_s2$coord
prop_var_expl_s2 <- pca_s2$weights
```

3.2.3 S3: Vulnerability to natural environment

```r
variables_to_keep <- c(
  "Mortality rate attributed to household and ambient air pollution, age-standardized (per
  "PM2.5 air pollution, mean annual exposure (micrograms per cubic meter)",
  "PM2.5 pollution, population exposed to levels exceeding WHO Interim Target-1 value (% o
  "PM2.5 pollution, population exposed to levels exceeding WHO Interim Target-2 value (% o
  "PM2.5 pollution, population exposed to levels exceeding WHO Interim Target-3 value (% o
  "Air transport, passengers carried",
  "International tourism, number of arrivals",
  "Ecological Footprint")


pca_s3 <- get_res_acp(
  variables_to_keep = variables_to_keep,
  type = "s3"
)

df_coordinate_s3 <- pca_s3$coord
prop_var_expl_s3 <- pca_s3$weights
```

3.2.4 S4: Living conditions

```r
variables_to_keep <- c(
  "People using at least basic drinking water services (% of population)",
  "People using at least basic sanitation services (% of population)",
  "Prevalence of undernourishment (% of population)",
  "Prevalence of anemia among women of reproductive age (% of women ages 15-49)",
  "Poverty headcount ratio at $3.65 a day (2017 PPP) (% of population)",
  "Poverty headcount ratio at $6.85 a day (2017 PPP) (% of population)",
```

```
  "Poverty headcount ratio at national poverty lines (% of population)",
  "GDP per capita (current US$)",
  "GDP per capita, PPP (current international $)",
  "Urban density"
  )

pca_s4 <- get_res_acp(
  variables_to_keep = variables_to_keep,
  type = "s4"
)

df_coordinate_s4 <- pca_s4$coord
prop_var_expl_s4 <- pca_s4$weights
```

3.2.5 S5: Economic and societal characteristics

```
variables_to_keep <- c(
  "Population, total",
  "GDP per capita growth (annual %)",
  "International migrant stock (% of population)",
  "Population ages 65 and above (% of total population)",
  "Individuals using the Internet (% of population)",
  "Mobile cellular subscriptions (per 100 people)",
  "Shadow size Economy",
  "Gini index (CIA estimate)"
)

pca_s5 <- get_res_acp(
  variables_to_keep = variables_to_keep,
  type = "s5"
)

df_coordinate_s5 <- pca_s5$coord
prop_var_expl_s5 <- pca_s5$weights
```

3.2.6 S6: Policy variables

```
variables_to_keep <- c(
  "GovernmentResponseIndex",
  "ContainmentHealthIndex",
  "StringencyIndex",
  "EconomicSupportIndex"
)

pca_s6 <- get_res_acp(
  variables_to_keep = variables_to_keep,
  type = "s6"
)

df_coordinate_s6 <- pca_s6$coord
prop_var_expl_s6 <- pca_s6$weights
```

3.2.7 Final table containing all the coordianetes

```
load("./data/output/list_countries.rda")
```

```
df_coordinates <-
  df_coordinate_s1 |>
  merge(df_coordinate_s2) |>
  merge(df_coordinate_s3) |>
  merge(df_coordinate_s4) |>
  merge(df_coordinate_s5) |>
  merge(df_coordinate_s6) |>
  left_join(list_countries, by = "country")

write_excel_csv(
  df_coordinates,
  file = "./estim/results_pca/df_coordinates.csv"
)
```

```
df_coordinates
```

```
                      country coordinate_s1 coordinate_s2 coordinate_s3
1                 Afghanistan   -1.21862387   0.585556190   1.468395696
```

2	Algeria	-0.21659429	-0.610474440	0.441892918
3	Angola	-1.18481459	0.695112791	0.356279120
4	Argentina	1.32352447	-0.593971671	-1.264056359
5	Australia	2.88084417	-0.948890170	-1.512026190
6	Austria	3.16206730	-0.854782998	-1.414217706
7	Bahrain	-0.02925388	-0.291208694	0.814800849
8	Bangladesh	-1.19133172	-0.007618773	1.225790522
9	Belgium	2.52610326	-0.893179082	-1.456893848
10	Benin	-1.32964814	0.228730643	0.864140926
11	Bhutan	-0.61470218	-0.219402561	0.456844652
12	Bolivia	-0.35290142	-0.333498120	-0.428982472
13	Brazil	0.85001831	-0.426444842	-1.172810347
14	Brunei	-0.02318024	-0.299477388	-1.564960601
15	Burkina Faso	-0.99780444	0.276220215	1.250252228
16	Burundi	-1.11003420	0.322473897	1.053216925
17	Cambodia	-0.98196013	0.107113208	0.002055174
18	Cameroon	-1.18369764	0.564011811	1.398047830
19	Canada	2.25500780	-0.848947543	-1.615274252
20	Cape Verde	-0.55209542	-0.548212538	0.554854717
21	Central African Republic	-1.24220390	1.886983211	1.481777623
22	Chad	-1.34706341	0.228819724	1.400944989
23	Chile	1.15047465	-0.737020429	-0.713203077
24	China	0.30308729	-0.460350982	1.735654891
25	Colombia	0.26863282	-0.711713502	-0.908090284
26	Congo	-0.90584615	1.410850823	0.986983041
27	Cote d'Ivoire	-1.16564784	0.303119594	0.542043480
28	Democratic Republic of Congo	-1.08474793	0.760757970	1.043918983
29	Denmark	1.99081221	-0.871072843	-1.481692227
30	Djibouti	-1.13794396	0.198000114	1.049881758
31	Ecuador	0.07366003	-0.733304723	-1.030438549
32	Egypt	-0.69668891	0.163103150	1.124555030
33	El Salvador	0.10110623	-0.670216668	-0.463673449
34	Eswatini	-0.56685921	2.978021294	-0.299418781
35	Ethiopia	-1.23508467	-0.322378032	0.732459929
36	France	2.66037328	-0.875786656	-1.554939742
37	Gabon	0.22503829	1.269753294	0.668726967
38	Gambia	-1.15585941	0.092117880	1.019831729
39	Germany	3.80212023	-0.836960913	-1.410873930
40	Ghana	-0.85738279	0.309142740	0.916442692
41	Greece	1.80856324	-0.823598220	-1.087022962
42	Guatemala	-0.24062878	-0.499572923	-0.309015149
43	Guinea	-1.27477590	0.665215950	0.554022821
44	Haiti	-1.29380792	0.460883859	-0.480918641

45	Honduras	-0.69867527	-0.435160405	-0.408769397
46	India	-1.10622365	0.085402926	1.469495893
47	Indonesia	-0.52420295	0.319112621	-0.704069192
48	Iran	0.28382192	-0.560077572	0.357599198
49	Iraq	-0.66896317	-0.209022684	0.863695235
50	Ireland	2.66461955	-0.919187619	-1.443407535
51	Israel	1.48686182	-0.923407264	-0.821948818
52	Italy	1.46842207	-0.970042038	-1.124883252
53	Jamaica	-0.42969015	-0.267339948	-1.245495733
54	Japan	3.56451464	-0.917888619	-1.310819039
55	Jordan	0.05219600	-0.548716511	0.005303699
56	Kenya	-0.92436206	0.295759495	0.061788483
57	Kuwait	0.27669178	-0.451639175	0.764844020
58	Laos	-1.00911260	0.043720502	0.052629053
59	Lebanon	0.57454989	-0.127775333	-0.065725895
60	Lesotho	-0.59499964	3.024188718	0.666433474
61	Liberia	-1.22125077	0.776562132	-0.017227459
62	Madagascar	-1.18909178	0.480425188	-0.143730684
63	Malawi	-0.99883391	0.544907953	-0.254481655
64	Malaysia	0.09397601	-0.219107057	-0.871377351
65	Mali	-1.34999751	0.213429340	1.054681508
66	Mauritania	-1.14039635	-0.456739466	1.179732513
67	Mauritius	0.57181405	0.009687679	-1.050236105
68	Mexico	0.05105243	-0.369969351	-0.650952921
69	Mongolia	1.04117250	1.032345478	0.620935940
70	Morocco	-0.77608409	-0.112806187	0.139183519
71	Mozambique	-1.18174367	1.360376633	-0.093608725
72	Myanmar	-1.02845848	0.426487717	0.620896170
73	Namibia	-0.16850028	1.370349367	-0.019062420
74	Nepal	-0.89140826	-0.027405485	1.554013334
75	Netherlands	3.08628392	-0.910929944	-1.487755168
76	Nicaragua	-0.26874709	-0.454698331	-0.689210150
77	Niger	-1.12463914	0.152470569	1.612044185
78	Nigeria	-1.00224120	0.368771740	1.261786625
79	Oman	-0.20051581	-0.264041926	0.737612306
80	Pakistan	-1.07258437	0.500544603	1.360572785
81	Panama	0.84731425	-0.695526889	-1.360748902
82	Paraguay	-0.27113810	-0.537758504	-1.334467669
83	Peru	-0.07332750	-0.658357330	-0.346023743
84	Philippines	-0.42277515	0.881241610	-0.293422125
85	Qatar	0.50511997	-0.457027863	0.994938380
86	Russia	2.12861097	-0.058570769	-0.846309256
87	Rwanda	-0.88616712	-0.028203215	0.962308276

```
88                  Saudi Arabia    0.72703455  -0.066280381   0.969485636
89                       Senegal   -1.26357911  -0.255767356   1.100739747
90                    Seychelles    0.63885892  -0.014355689  -0.771576379
91                  Sierra Leone   -1.26512653   0.801285524   0.264333743
92               Solomon Islands   -0.76265376   0.934950089  -0.641727470
93                       Somalia   -1.13811883   0.774193009   0.622413242
94                  South Africa    0.21052324   2.085008177  -0.258767421
95                   South Korea    3.18910163  -0.871220204  -0.391584077
96                   South Sudan   -1.34472667   0.519286474   0.872850966
97                         Spain    1.57996391  -0.890446000  -1.509234013
98                     Sri Lanka    0.01168562  -0.602700939  -0.640733623
99                         Sudan   -0.95027458  -0.063443118   1.075639966
100                       Sweden    2.10456255  -1.008366576  -1.591195727
101                        Syria   -0.81771758  -0.282453665   0.507701960
102                     Tanzania   -1.07979100   0.223795026   0.012844562
103                     Thailand   -0.18255675  -0.421787063  -0.231537275
104                  Timor-Leste   -0.07018220   0.458628791  -0.266206013
105                         Togo   -1.21908109   0.158139038   0.989155567
106                      Tunisia   -0.20420461  -0.517176134   0.356586030
107                       Uganda   -1.28691435   0.632719116   0.970965255
108                United Kingdom    1.68327867  -0.891418414  -1.452743602
109                 United States    6.10722432  -0.654581306  -0.759984116
110                       Uruguay    1.62895570  -0.520655825  -1.489792336
111                       Vanuatu   -0.76869596   0.687236467  -0.718336749
112                     Venezuela   -0.65083268  -0.429608837  -0.840063188
113                       Vietnam   -0.34485091  -0.155731081   0.152239524
114                         Yemen   -1.25095179   0.045044565   1.126789054
115                      Zimbabwe   -0.94407305   0.933417874  -0.190331594
    coordinate_s4 coordinate_s5 coordinate_s6              group
1     1.270616306  -0.956087817   -1.78989270               Asia
2    -1.099269663  -0.030008599    0.54343719               MENA
3     1.647035211  -0.881227673   -0.24910732 Sub-Saharan Africa
4    -1.115314454   0.644888636    2.44016976      Latin America
5    -2.383808110   0.745966967    1.14501180     Industrialized
6    -2.353472919   0.904544775    0.98979909     Industrialized
7    -1.146740332   0.649394318    1.65896777               MENA
8     0.312649160  -0.369105406    1.13743429               Asia
9    -2.365885257   0.678052413    1.00012446     Industrialized
10    1.684840654  -0.651092587   -1.05432610 Sub-Saharan Africa
11   -0.358015464   0.173132603    1.60272883               Asia
12   -0.232227870  -0.121553459    0.81515639      Latin America
13   -1.017579046   0.228588359    0.55889425      Latin America
14   -1.492464017   0.444580939   -0.78513346               Asia
```

15	2.193489495	-0.638172933	-2.23807144	Sub-Saharan Africa
16	2.264123144	-1.139977037	-4.53103218	Sub-Saharan Africa
17	0.072078349	0.037628251	-1.85818636	Asia
18	1.158591494	-0.472624402	-1.52087304	Sub-Saharan Africa
19	-2.228902185	0.617888729	0.72272782	Industrialized
20	0.158617858	0.090149618	1.39362807	Sub-Saharan Africa
21	3.680644147	-1.288581617	-1.60485963	Sub-Saharan Africa
22	2.532982648	-1.067827522	-0.05576503	Sub-Saharan Africa
23	-1.828215023	0.697298492	1.86045159	Latin America
24	-1.776040270	0.700821307	1.14166599	Asia
25	-0.239134657	0.452552046	1.77454870	Latin America
26	2.299780218	-0.568763109	-0.89574967	Sub-Saharan Africa
27	1.301393281	0.067436911	-0.98212331	Sub-Saharan Africa
28	3.549736293	-1.071427390	-1.16041293	Sub-Saharan Africa
29	-2.486155044	0.920896832	0.35853069	Industrialized
30	1.106655889	-0.681762414	-1.18915901	Sub-Saharan Africa
31	-0.472886893	0.020807688	1.31776975	Latin America
32	-0.565544017	-0.240703645	0.75673958	MENA
33	-0.974746300	0.463788440	1.72754181	Latin America
34	1.105038346	-0.150098011	-0.29977715	Sub-Saharan Africa
35	1.794676038	-1.218940592	-0.11301799	Sub-Saharan Africa
36	-2.240418995	0.867108732	1.12897877	Industrialized
37	0.478727364	0.241972438	1.91356881	Sub-Saharan Africa
38	1.633786787	-0.454456211	-0.92791004	Sub-Saharan Africa
39	-2.340559288	1.051280719	0.34189509	Industrialized
40	0.566707213	-0.013256653	-1.27039523	Sub-Saharan Africa
41	-1.751206288	0.779346194	1.09145584	Industrialized
42	0.204437579	-0.114211417	1.17711210	Latin America
43	1.760430092	-0.528130426	-0.16055964	Sub-Saharan Africa
44	2.697248573	-0.714936752	-0.98071342	Latin America
45	0.102386436	-0.496816623	2.29154077	Latin America
46	0.624536416	0.035788998	1.86728533	Asia
47	-0.679892066	0.200197683	-0.21010698	Asia
48	-1.060933514	0.452397899	-0.47086087	MENA
49	-0.802363699	-0.284842255	1.00736056	MENA
50	-2.721202484	0.482361539	1.59638669	Industrialized
51	-2.001743702	0.870787423	1.88794981	MENA
52	-2.025659701	0.964514216	1.65296235	Industrialized
53	-0.939041927	0.174933994	1.11666950	Latin America
54	-2.032987675	1.477237670	-0.48787626	Industrialized
55	-0.858374611	-0.114236795	0.25180997	MENA
56	1.752849789	-0.336352448	0.26512649	Sub-Saharan Africa
57	-1.377016418	1.177259814	0.85056157	MENA

58	-0.002582404	-0.593987851	-1.04678699	Asia
59	-1.019267466	0.392570120	0.64169943	MENA
60	1.632755477	-0.536044377	-0.99049266	Sub-Saharan Africa
61	2.320329624	-1.091453931	-1.00888416	Sub-Saharan Africa
62	3.661218992	-0.924884100	-0.59626965	Sub-Saharan Africa
63	1.890424263	-1.024593786	-0.09094998	Sub-Saharan Africa
64	-1.447965178	0.556810055	1.53164167	Asia
65	1.406402040	-0.319684064	-1.27816560	Sub-Saharan Africa
66	0.794704056	-0.007537122	-1.58269194	Sub-Saharan Africa
67	-1.582478225	0.715289041	-1.38293782	Sub-Saharan Africa
68	-0.780187287	0.122726541	-0.22062450	Latin America
69	-0.479011234	0.215896184	1.10315078	Asia
70	-0.788295982	0.499610544	1.32096869	MENA
71	2.588289017	-1.019745949	-1.01571891	Sub-Saharan Africa
72	0.125630020	0.059634631	0.50081621	Asia
73	0.256657222	0.044074568	-1.18889175	Sub-Saharan Africa
74	0.049925091	0.028402292	1.22309754	Asia
75	-2.389031052	0.898732240	0.68303725	Industrialized
76	-0.183019461	-0.311023566	-4.70602120	Latin America
77	2.464282831	-1.013991893	-3.11752960	Sub-Saharan Africa
78	1.543167751	-0.317338952	-0.10148203	Sub-Saharan Africa
79	-1.039518089	0.689336932	1.41983423	MENA
80	0.460443620	-0.696137644	0.70036472	Asia
81	-1.253938906	0.643978391	1.48594875	Latin America
82	-0.988941334	0.179450065	1.32007602	Latin America
83	-0.314590375	0.260196410	2.13402399	Latin America
84	-0.669668018	0.177593767	1.35706450	Asia
85	-1.732903821	0.892153709	0.99865026	MENA
86	-1.696535435	1.014755234	0.17776235	Asia
87	1.528193863	-0.629461476	1.54897884	Sub-Saharan Africa
88	-1.314446089	0.545867092	1.00917132	MENA
89	1.029867325	-0.215510766	-0.56693515	Sub-Saharan Africa
90	-0.916811982	0.901517082	-1.29205325	Sub-Saharan Africa
91	2.491478435	-0.731435221	-1.70525678	Sub-Saharan Africa
92	1.159273832	-0.661605582	-2.88865029	Asia
93	2.612177853	-1.188366653	-2.71760894	Sub-Saharan Africa
94	0.255176931	0.715394984	1.07177449	Sub-Saharan Africa
95	-1.986137486	0.826742645	0.28257479	Asia
96	3.112659781	-1.321141003	-0.79940066	Sub-Saharan Africa
97	-1.906171934	0.978931617	1.26428935	Industrialized
98	-0.835448064	0.175100543	-0.07661764	Asia
99	1.498538282	-0.641717560	-0.80246654	Sub-Saharan Africa
100	-2.319758014	0.948317454	-0.01954782	Industrialized

```
101  -0.631594008  -0.463102256   -1.66056589               MENA
102   1.804449352  -0.740921852   -3.94905060 Sub-Saharan Africa
103  -1.560188402   0.902739099    0.56720207               Asia
104   1.464144824  -0.816505654   -1.43974554               Asia
105   1.981719351  -0.706650965   -0.66091727 Sub-Saharan Africa
106  -1.216346554   0.324986281   -0.21314936               MENA
107   1.600659477  -1.079593815    0.49195854 Sub-Saharan Africa
108  -2.162610258   0.993382334    1.77154328     Industrialized
109  -2.499358160   0.757034626    0.52383291     Industrialized
110  -1.410288218   0.726538019   -0.24067229      Latin America
111   0.098100137  -0.388019678   -2.32734831               Asia
112  -0.247859892  -0.469485661    1.11395568      Latin America
113  -1.501627388   0.409768568    0.50977609               Asia
114   2.610979598  -0.862479280   -3.55365376               MENA
115   1.448674781  -0.533521259   -0.06218858 Sub-Saharan Africa
```

Let us normalise the scores:

```
df_coordinates <-
  df_coordinates |>
  pivot_longer(cols = coordinate_s1:coordinate_s6, values_to = "coord") |>
  group_by(name) |>
  mutate(coord_norm = (coord - min(coord)) / (max(coord) - min(coord))) |>
  mutate(name = str_remove(name, "coordinate_")) |>
  pivot_wider(values_from = c(coord, coord_norm), names_from = name)

save(df_coordinates, file = "./estim/results_pca/df_coordinates.rda")
```

```
prop_var_expl <-
  prop_var_expl_s1 |>
  mutate(type = "s1") |>
  bind_rows(
    prop_var_expl_s2 |> mutate(type = "s2")
  ) |>
  bind_rows(
    prop_var_expl_s3 |> mutate(type = "s3")
  ) |>
  bind_rows(
    prop_var_expl_s4 |> mutate(type = "s4")
  ) |>
  bind_rows(
```

Table 3.1: Proportion of variance explained by each component

dim	s1	s2	s3	s4	s5	s6
1	0.6896990	0.4654735	0.4762506	0.6480978	0.3671920	0.8101595
2	0.1335007	0.2299477	0.1906892	0.1069639	0.1631495	0.1753028
3	0.0852875	0.1470457	0.1132016	0.0824989	0.1343746	0.0145376
4	0.0556660	0.0891404	0.0755011	0.0560185	0.1075233	0.0000000
5	0.0331477	0.0683926	0.0570393	0.0378774	0.1032494	NA
6	0.0026991	NA	0.0451226	0.0330156	0.0637572	NA
7	NA	NA	0.0281633	0.0180959	0.0387484	NA
8	NA	NA	0.0140323	0.0097006	0.0220056	NA
9	NA	NA	NA	0.0047761	NA	NA
10	NA	NA	NA	0.0029553	NA	NA

```
    prop_var_expl_s5 |> mutate(type = "s5")
  ) |>
  bind_rows(
    prop_var_expl_s6 |> mutate(type = "s6")
  ) |>
  pivot_wider(names_from = "type", values_from = prop_var_expl)
```

```
knitr::kable(prop_var_expl)
```

3.3 Correlation plot

The correlations between the different factors can be obtained as follows:

```
corr <-
  df_coordinates |>
  select(coord_norm_s1:coord_norm_s6) |>
  rename_with(~str_remove(., "coord_norm_") |> str_to_upper()) |>
  cor()
corr
```

```
           S1         S2         S3         S4         S5         S6
S1  1.0000000 -0.5411434 -0.6800259 -0.7856012  0.7715171  0.3842682
S2 -0.5411434  1.0000000  0.4482727  0.6801212 -0.5717023 -0.3808154
S3 -0.6800259  0.4482727  1.0000000  0.6335275 -0.5869585 -0.2895693
S4 -0.7856012  0.6801212  0.6335275  1.0000000 -0.9077159 -0.5572667
```

```
S5  0.7715171 -0.5717023 -0.5869585 -0.9077159  1.0000000  0.5529047
S6  0.3842682 -0.3808154 -0.2895693 -0.5572667  0.5529047  1.0000000
```

It may be easier to use a correlation matrix visualization:

```
p_correl_synthetif_factors <-
  ggcorrplot(
    round(corr, 2),
    hc.order = FALSE,
    type = "lower",
    lab = TRUE
  ) +
  theme(axis.text.x = element_text(angle = 0)) +
  scale_fill_gradient2(
    guide = "none", low = "#005A8B", mid = "white", high = "#AA2F2F"
  )
p_correl_synthetif_factors
```

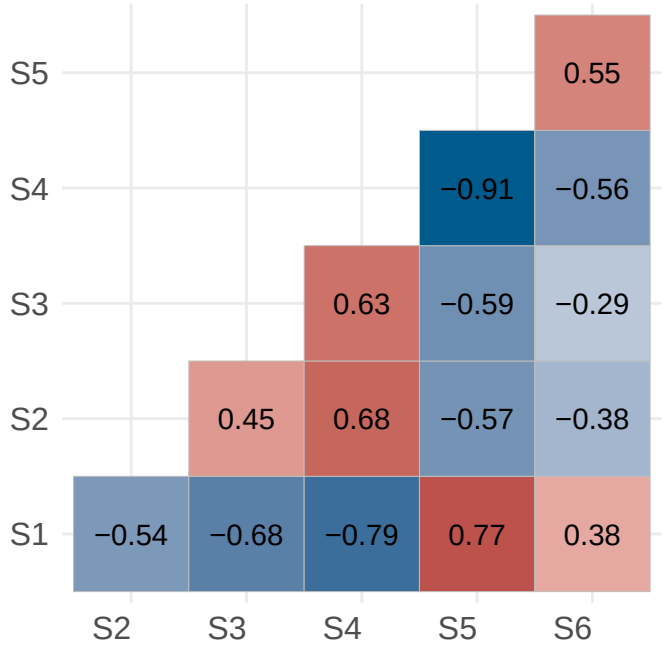


Figure 3.1: Correlation between the synthetic factors

3.4 Boxplots

```
library(grid)
load("./assets/theme_paper.rda")
```

```
boxplot_pca <-
  ggplot(
  data = df_coordinates |>
    pivot_longer(
      cols = c(coord_norm_s1:coord_norm_s6),
      values_to = "coordinates"
    ) |>
    mutate(
      group = factor(
        group,
        levels = rev(c(
          "MENA", "Sub-Saharan Africa", "Asia", "Latin America",
          "Industrialized"
        ))
      ),
      name = factor(
        name,
        labels = c(
          "coordinate_s1" = expression(S[1]*": Healthcare infrastructure"),
          "coordinate_s2" = expression(S[2]*": Vulnerability to comorbidites"),
          "coordinate_s3" = expression(S[3]*": Vulnerability to natural environment"),
          "coordinate_s4" = expression(S[4]*": Living conditions"),
          "coordinate_s5" = expression(S[5]*": Economic and societal characteristics"),
          "coordinate_s6" = expression(S[6]*": Policy variables")
        )
      )
    ),
  mapping = aes(x = coordinates, y = group)
) +
  geom_boxplot() +
  facet_wrap(~name, labeller = label_parsed) +
  theme_paper() +
  labs(x = "Factor Scores", y = NULL)

boxplot_pca
```

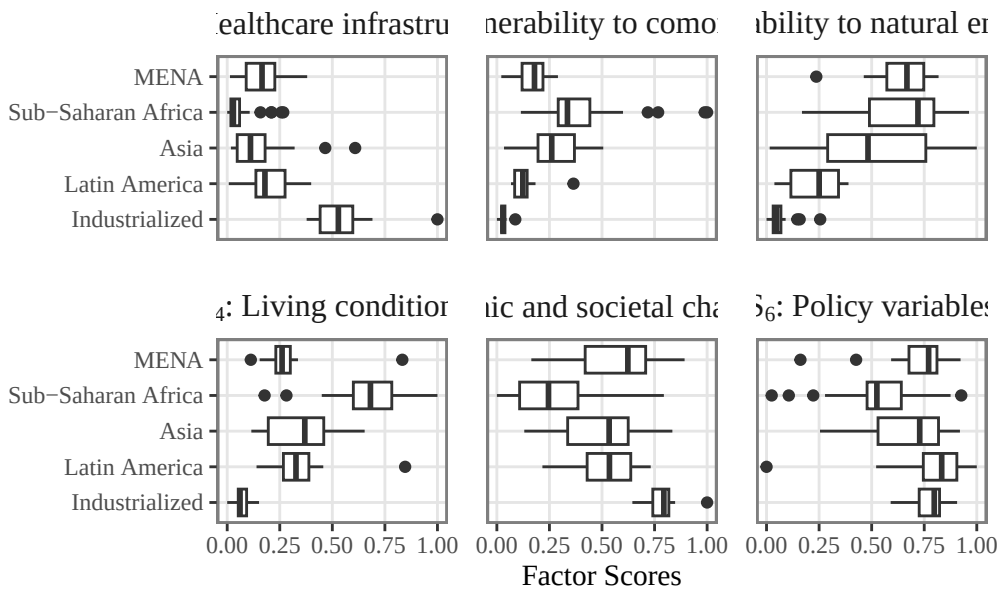


Figure 3.2: Distribution of factor scores estimated with the Principal Component Analysis in each region.

3.5 Maps

Let us define a theme for maps:

```
#' Theme for maps with ggplot2
#'
#' @param ... arguments passed to the theme function
#' @export
#' @importFrom ggplot2 element_rect element_text element_blank element_line unit
#'   rel
theme_map_paper <- function(...) {
  theme(
    text = element_text(family = "Times"),
    plot.background = element_rect(fill = "transparent", color = NA),
    panel.background = element_rect(fill = "transparent", color = NA),
    panel.border = element_blank(),
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank(), axis.line = element_blank(),
    plot.title.position = "plot",
    legend.text = element_text(size = rel(1.2)),
```

```
    legend.title = element_text(size = rel(1.2)),
    legend.background = element_rect(fill="transparent", color = NULL),
    legend.key = element_blank(),
    legend.key.height   = unit(2, "line"),
    legend.key.width    = unit(1.5, "line"),
    strip.background = element_rect(fill = NA),
    panel.spacing = unit(1, "lines"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    plot.margin = unit(c(1, 1, 1, 1), "lines"),
    strip.text = element_text(size = rel(1.2))
  )
}
```

A Shapefile that allows us to display the level 0 world administrative boundaries, freely available on the online open data platform "opendatasoft", can be loaded:

```
library(sf)
```

```
Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
world_map <- sf::read_sf(
  "./data/raw/world-administrative-boundaries/world-administrative-boundaries.shp"
)
```

There are some mismatching country names between the map file and the epidemic data. The names from the map file can be manually changed:

```
world_map <-
  world_map |>
  mutate(
    name = recode(
      name,
      # old = new
      "Brunei Darussalam" = "Brunei",
      "Côte d'Ivoire" = "Cote d'Ivoire",
      "Democratic Republic of the Congo" = "Democratic Republic of Congo",
      "Swaziland" = "Eswatini",
      "Iran (Islamic Republic of)" = "Iran",
      "Lao People's Democratic Republic" = "Laos",
      "Russian Federation" = "Russia",
```

```
      "Republic of Korea" = "South Korea",
      "Syrian Arab Republic" = "Syria",
      "United Republic of Tanzania" = "Tanzania",
      "U.K. of Great Britain and Northern Ireland" = "United Kingdom",
      "United States of America" = "United States"
    )
  )
```

```
df_coordinates_2 <-
  df_coordinates |> select(country, coord_norm_s1:coord_norm_s6) |>
  pivot_longer(
    cols = -country,
    names_to = "synthetic_factor",
    values_to = "coord_norm"
  )

world_map_coords <- NULL
# v <- str_c("coord_norm_s", 1:6)[1]
for (v in str_c("coord_norm_s", 1:6)) {
  map_current <-
    world_map |>
    left_join(
      df_coordinates_2 |>
        filter(synthetic_factor == v),
      by = c("name" = "country")
    )
  map_current$synthetic_factor <- v

  world_map_coords <-
    world_map_coords |> bind_rows(map_current)
}

world_map_coords <-
  world_map_coords |>
  mutate(
    synthetic_factor = factor(
      synthetic_factor,
      labels = c("coord_norm_s1" = "S1: Healthcare infrastructure",
                 "coord_norm_s2" = "S2: Vulnerability to comorbidites",
                 "coord_norm_s3" = "S3: Vulnerability to natural environment",
                 "coord_norm_s4" = "S4: Living conditions",
```

```
                "coord_norm_s5" = "S5: Economic and societal characteristics",
                "coord_norm_s6" = "S6: Policy variables")
    )
  )
```

```
p_map <-
  ggplot(data = world_map_coords) +
  geom_sf(mapping = aes(fill = coord_norm), colour = "gray95", size = .1) +
  scale_fill_gradient(NULL, low = "yellow", high = "red", na.value = "gray80") +
  theme_map_paper() +
  theme(legend.position = "bottom") +
  facet_wrap(~synthetic_factor, ncol = 2)

p_map
```

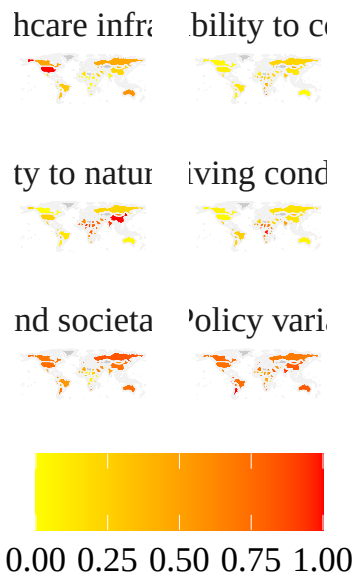


Figure 3.3: Normalised factor scores by countries.

4 Generalized Richards Model: estimates

This section provides the codes used to analyse the estimated values of P and α (see Chapter 2)

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

Let us load the results:

```
load("./estim/resul_fit_countries.rda")
```

4.1 Exploration of the Results

The complete set of estimated parameters (using NLS) can be accessed as follows:

```
estimates_nls <-
  map(resul_fit_countries, "nls") |>
  map_df("estimates", .id = "country") |>
  mutate(estim = "nls")
estimates_nls |>
  DT::datatable(options = list(
    searching = TRUE,
    pageLength = 10,
```

```
    lengthMenu = c(5, 10, 15, 20)
  ))
```

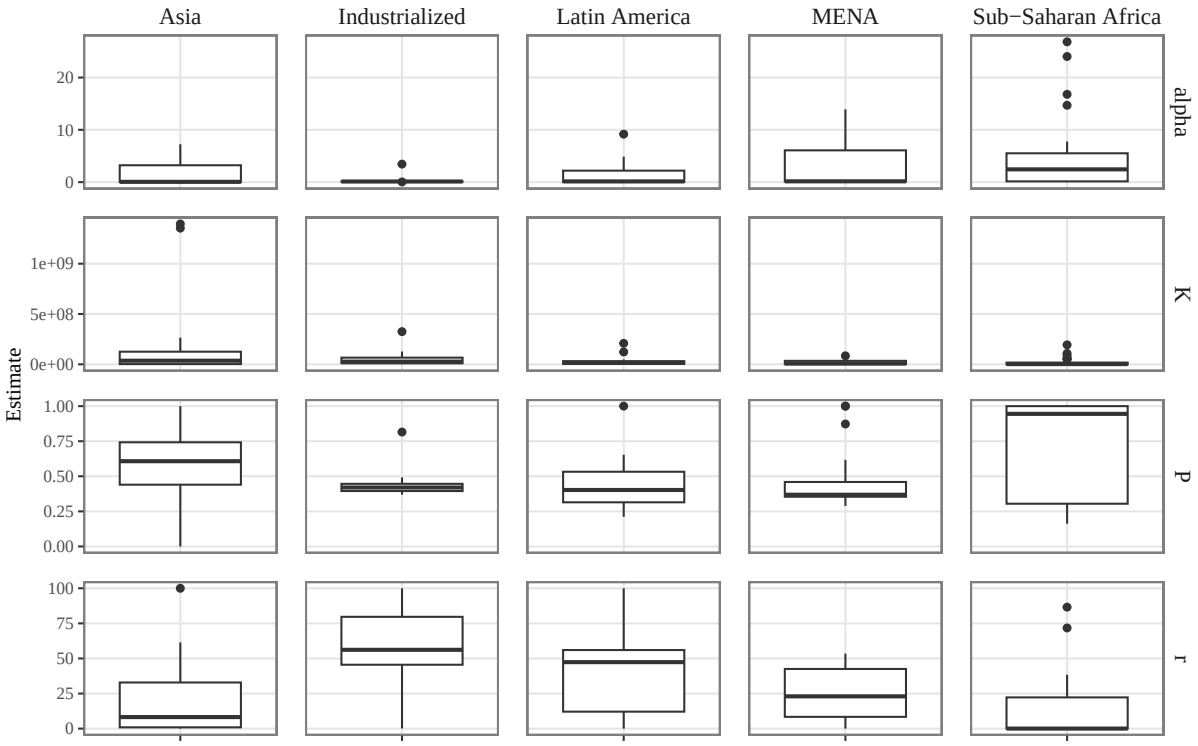
Let us visualize these estimates on boxplots. Let us first load the list of countries and the theme used for the graphs.

```
load("./data/output//list_countries.rda")
library(grid)
load("./assets/theme_paper.rda")
```

```
ggplot(
  data = estimates_nls |>
    left_join(list_countries) |>
    pivot_longer(cols = c(r, P, alpha, K)) |>
    group_by(name, group)
) +
  geom_boxplot(
    mapping = aes(x = estim, y = value),
    na.rm = TRUE
  ) +
  facet_grid(name ~ group, scales = "free") +
  labs(
    x = NULL, y = "Estimate",
    title = "Estimated coefficients for the Generalized Richards Model"
  ) +
  theme_paper() +
  theme(axis.text.x = element_blank())
```

```
Joining with `by = join_by(country)`
```

Estimated coefficients for the Generalized Richards Model



The same graph can be reproduced while removing outliers, to have a better idea of the variability between countries. We define a function to remove the outliers, as suggested by Andrew Baxter on Stack overflow:

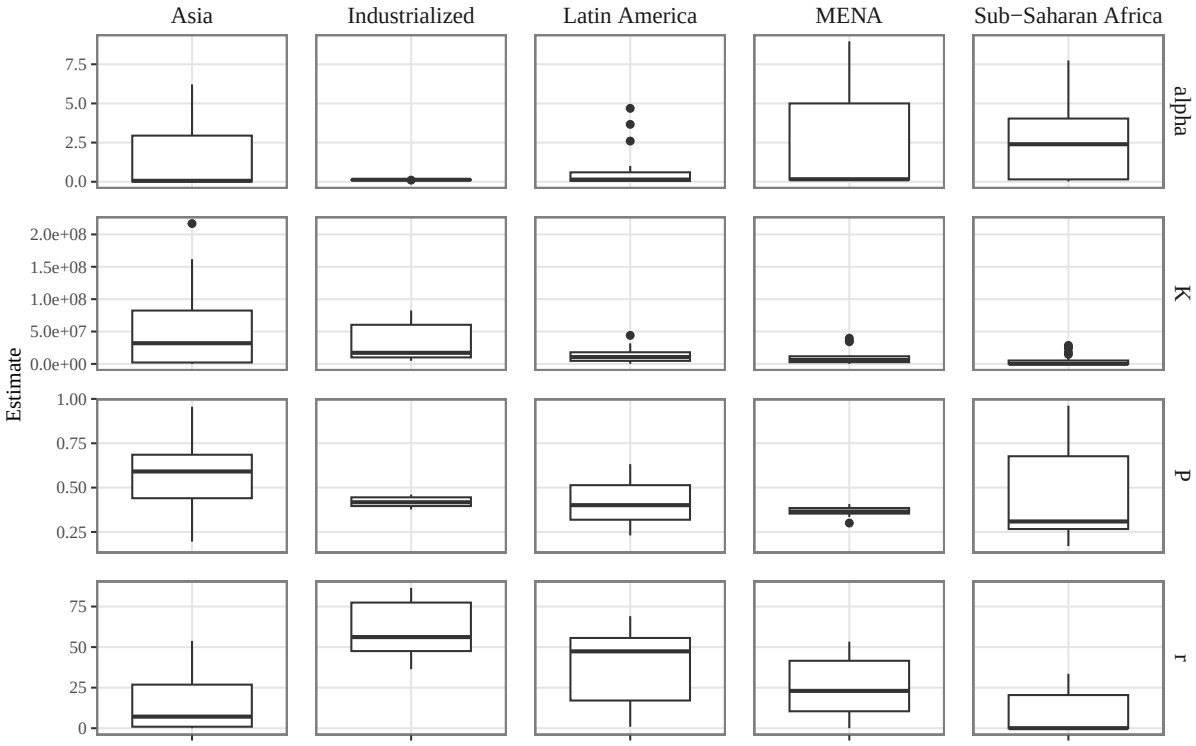
```
filter_lims <- function(x) {
  l <- boxplot.stats(x)$stats[1]
  u <- boxplot.stats(x)$stats[5]

  for (i in 1:length(x)) {
    x[i] <- ifelse(x[i]>l & x[i]<u, x[i], NA)
  }
  return(x)
}
```

```
ggplot(
  data = estimates_nls |>
    left_join(list_countries) |>
    pivot_longer(cols = c(r, P, alpha, K)) |>
```

```
    group_by(name, group) |>
    mutate(value2 = filter_lims(value))
) +
  geom_boxplot(
    mapping = aes(x = estim, y = value2),
    na.rm = TRUE
  ) +
  facet_grid(name ~ group, scales = "free") +
  labs(
    x = NULL, y = "Estimate",
    title = "Estimated coefficients for the Generalized Richards Model"
  ) +
  theme_paper() +
  theme(axis.text.x = element_blank())
```

```
Joining with `by = join_by(country)`
```

Estimated coefficients for the Generalized Richards Model



The Root Mean Squared Errors:

```
rmse_nls <-
  map(resul_fit_countries, "nls") |>
  map("rmse") |>
  map_df(as_tibble, .id = "country") |>
  mutate(estim = "nls")

rmse <-
  rmse_nls |>
  pivot_wider(names_from = estim, values_from = value)
```

Let us put these values in a table:

```
rmse |>
  DT::datatable(options = list(
    searching = TRUE,
    pageLength = 10,
    lengthMenu = c(5, 10, 15, 20)
  ))
```

The average values for all countries:

```
rmse |>
  summarise(
    rmse_nls = mean(nls),
    rmse_nls_med = median(nls)
  )
```

```
# A tibble: 1 x 2
  rmse_nls rmse_nls_med
     <dbl>        <dbl>
1   54056.        6809.
```

And by group of countries:

```
rmse |>
  left_join(list_countries, by = "country") |>
  group_by(group) |>
  summarise(
    rmse_nls = mean(nls),
    rmse_nls_med = median(nls)
  )
```

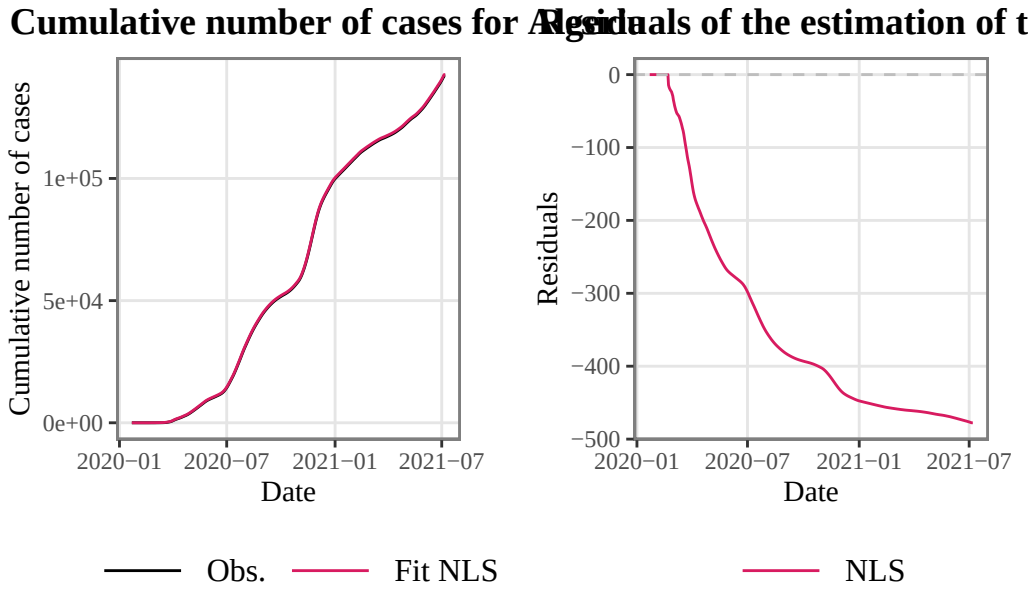
```
# A tibble: 5 x 3
  group              rmse_nls rmse_nls_med
  <chr>                 <dbl>        <dbl>
1 Asia                  98445.       10801.
2 Industrialized       167568.       51634.
3 Latin America         40172.       11237.
4 MENA                  21903.       13788.
5 Sub-Saharan Africa     4703.        1034.
```

```r
plot_fit <- function(country) {
  resul_country <-resul_fit_countries[[country]]
  dat_country <- resul_country$dat

  dat_country <-
    dat_country |>
    mutate(
      fit_nls = resul_country$nls$fit
    )

  # Obs and fitted values
  p_1 <-
    ggplot(
      data = dat_country |>
        select(date, C_m, fit_nls) |>
        pivot_longer(cols = c(C_m, fit_nls)) |>
        mutate(
          name = factor(
            name,
            levels = c("C_m", "fit_nls"),
            labels = c("Obs.", "Fit NLS")
          )
        ),
      mapping = aes(x = date, y = value, colour = name)
    ) +
    geom_line() +
    scale_colour_manual(
      NULL,
      values = c("Obs." = "black", "Fit NLS" = "#D81B60")
    ) +
    labs(
      x = "Date", y = "Cumulative number of cases",
      title = str_c("Cumulative number of cases for ", country)
```

```
  ) +
  theme_paper() +
  theme(
    legend.key.height = unit(1, "line"),
    legend.key.width  = unit(2.5, "line"),
    legend.box = "horizontal"
  ) +
  guides(colour = guide_legend(override.aes = list(size = 1.8)))

# Residuals
p_2 <-
  ggplot(
    data = dat_country |>
      select(date, C_m, fit_nls) |>
      pivot_longer(cols = c(fit_nls)) |>
      mutate(error = C_m - value) |>
      mutate(
        name = factor(
          name,
          levels = c("fit_nls"),
          labels = c("NLS")
        )
      ),
    mapping = aes(x = date, y = error, colour = name)
  ) +
  geom_line() +
  geom_hline(yintercept = 0, linetype = "dashed", colour = "grey") +
  scale_colour_manual(NULL, values = c("NLS" = "#D81B60")) +
  labs(
    x = "Date", y = "Residuals",
    title = str_c(
      "Residuals of the estimation of the cumulative number of cases for ",
      country
    )
  ) +
  theme_paper() +
  theme(
    legend.key.height = unit(1, "line"),
    legend.key.width  = unit(2.5, "line"),
    legend.box = "horizontal"
  ) +
```

```
    guides(colour = guide_legend(override.aes = list(size = 1.8)))

  cowplot::plot_grid(p_1, p_2)
}
```

```
plot_fit("Algeria")
```

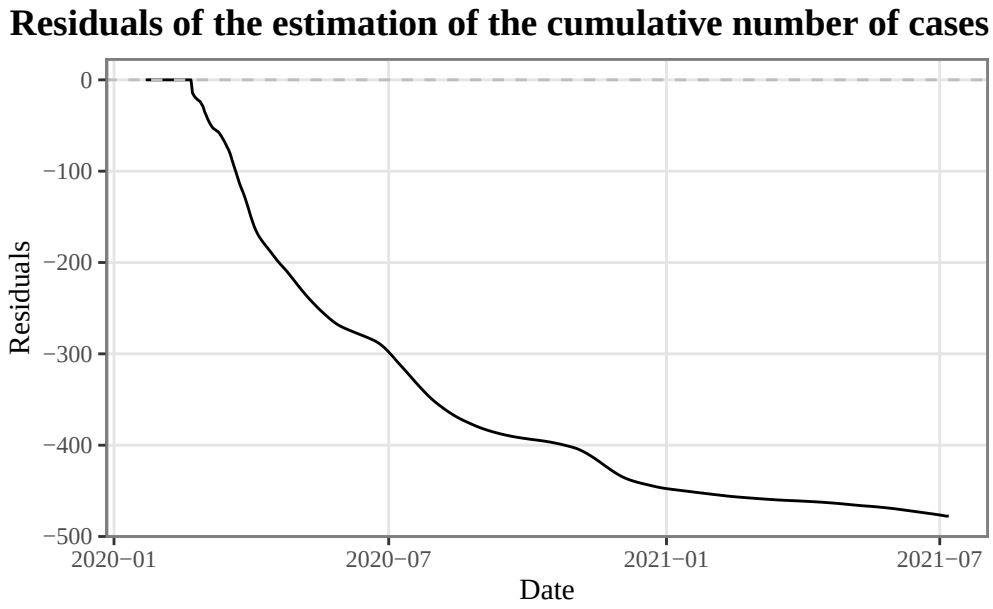


Let us focus on the residuals:

```
plot_fit_2 <- function(country) {
  resul_country <-resul_fit_countries[[country]]
  dat_country <- resul_country$dat

  dat_country <-
    dat_country |>
    mutate(fit_nls = resul_country$nls$fit)

  # Residuals
  p_2 <-
    ggplot(
      data = dat_country |>
        select(date, C_m, fit_nls) |>
        pivot_longer(cols = c(fit_nls)) |>
```

```
        mutate(error = C_m - value),
      mapping = aes(x = date, y = error)) +
    geom_line() +
    geom_hline(yintercept = 0, linetype = "dashed", colour = "grey") +
    scale_colour_manual(NULL, values = c("NLS" = "#D81B60")) +
    labs(
      x = "Date", y = "Residuals",
      title = str_c("Residuals of the estimation of the cumulative number of cases for ",c
    ) +
    theme_paper() +
    theme(
      legend.key.height = unit(1, "line"),
      legend.key.width  = unit(2.5, "line"),
      legend.box = "horizontal") +
    guides(colour = guide_legend(override.aes = list(size = 1.8)))

  p_2
}
```

```
plot_fit_2("Algeria")
```



4.2 Descriptive Statistics of the Estimates

```
estimates_all <-
  estimates_nls |>
  left_join(list_countries, by = "country")
```

```
library(kableExtra)
```

```
Attaching package: 'kableExtra'
```

```
The following object is masked from 'package:dplyr':

    group_rows
```

```
# All countries
results_desc_stat_all <-
  estimates_all |>
  mutate(estim = factor(estim, levels = c("nls"))) |>
  pivot_longer(cols = c(r, P, alpha, K), names_to = "coef_name") |>
  mutate(coef_name = factor(coef_name, levels = c("r", "P", "alpha", "K"))) |>
  mutate(group = "All countries") |>
  group_by(estim, group, coef_name) |>
  summarise(
    Min = min(value),
    Max = max(value),
    Q1 = quantile(value, probs = 0.25),
    Median = median(value),
    Q3 = quantile(value, probs = 0.75),
    Mean = mean(value),
    `Standard Error` = sd(value),
    n = n()
  ) |>
  pivot_longer(cols = Min:n, names_to = "stat_name") |>
  pivot_wider(names_from = c(estim, coef_name), values_from = value)
```

```
`summarise()` has grouped output by 'estim', 'group'. You can override using
the `.groups` argument.
```

```r
results_desc_stat_groups <-
  estimates_all |>
  mutate(
    estim = factor(estim, levels = c("nls")),
    group = factor(
      group,
      levels = c("MENA", "Sub-Saharan Africa", "Asia", "Latin America",
                 "Industrialized")
    )
  ) |>
  pivot_longer(cols = c(r, P, alpha, K), names_to = "coef_name") |>
  mutate(coef_name = factor(coef_name, levels = c("r", "P", "alpha", "K"))) |>
  group_by(estim, group, coef_name) |>
  summarise(
    Min = min(value),
    Max = max(value),
    Q1 = quantile(value, probs = 0.25),
    Median = median(value),
    Q3 = quantile(value, probs = 0.75),
    Mean = mean(value),
    `Standard Error` = sd(value),
    n = n()
  ) |>
  pivot_longer(cols = Min:n, names_to = "stat_name") |>
  pivot_wider(names_from = c(estim, coef_name), values_from = value)
```

```
`summarise()` has grouped output by 'estim', 'group'. You can override using
the `.groups` argument.
```

```r
results_desc_stat <-
  results_desc_stat_all |>
  bind_rows(results_desc_stat_groups) |>
  mutate(
    group = factor(
      group,
      levels = c("All countries", "MENA", "Sub-Saharan Africa", "Asia",
                 "Latin America", "Industrialized")
    )
  ) |>
  ungroup()
```

```r
names(results_desc_stat) <- str_replace(
  names(results_desc_stat), "(^nls_)", ""
)
```

```r
results_desc_stat <-
  results_desc_stat |>
  pivot_longer(cols = c(r, P, alpha, K)) |>
  pivot_wider(names_from = stat_name, values_from = value) |>
  mutate(name = factor(name, levels = c("r", "P", "alpha", "K"))) |>
  arrange(name, group)
```

```r
kableExtra::kable(
  results_desc_stat |> select(-name),
  caption = "Descriptive statistics of the estimates of the Parameters of Generalized Rich
  booktabs = TRUE,
  digits = 2
) |>
  pack_rows(index = table(fct_inorder(results_desc_stat$name))) |>
  kableExtra::kable_styling() |>
  kableExtra::scroll_box(width = "100%", box_css = "border: 0px;")
```

4.2.1 Comparison with and without Sub-Saharan Africa

```r
# All countries
results_desc_stat_all <-
  estimates_nls |>
  mutate(estim = factor(estim, levels = c("nls"))) |>
  pivot_longer(cols = c(r, P, alpha, K), names_to = "coef_name") |>
  mutate(coef_name = factor(coef_name, levels = c("r", "P", "alpha", "K"))) |>
  mutate(group = "All countries") |>
  group_by(estim, group, coef_name) |>
  summarise(
    Min = min(value),
    Max = max(value),
    Q1 = quantile(value, probs = 0.25),
    Median = median(value),
    Q3 = quantile(value, probs = 0.75),
    Mean = mean(value),
    `Standard Error` = sd(value),
```

Table 4.1: Descriptive statistics of the estimates of the Parameters of Generalized Richards model.

group	Min	Max	Q1	Median	Q3	Mean
r						
All countries	0.00	1.000000e+02	0.06	19.58	46.41	26.23
MENA	0.01	5.347000e+01	8.38	23.01	42.51	25.23
Sub-Saharan Africa	0.00	8.657000e+01	0.01	0.04	22.27	12.10
Asia	0.01	1.000000e+02	0.93	8.22	32.88	19.86
Latin America	0.00	1.000000e+02	12.09	47.37	55.99	38.84
Industrialized	0.12	1.000000e+02	45.53	56.17	79.68	59.33
P						
All countries	0.00	1.000000e+00	0.36	0.46	0.90	0.57
MENA	0.29	1.000000e+00	0.35	0.37	0.46	0.49
Sub-Saharan Africa	0.16	1.000000e+00	0.30	0.95	1.00	0.70
Asia	0.00	1.000000e+00	0.44	0.61	0.74	0.60
Latin America	0.21	1.000000e+00	0.31	0.40	0.53	0.45
Industrialized	0.37	8.200000e-01	0.39	0.42	0.45	0.45
alpha						
All countries	0.00	2.682000e+01	0.09	0.15	3.48	2.50
MENA	0.02	1.392000e+01	0.14	0.16	6.08	2.87
Sub-Saharan Africa	0.00	2.682000e+01	0.15	2.44	5.52	4.16
Asia	0.00	7.250000e+00	0.02	0.06	3.23	1.52
Latin America	0.00	9.170000e+00	0.04	0.14	2.20	1.50
Industrialized	0.04	3.450000e+00	0.11	0.12	0.14	0.32
K						
All countries	0.00	1.394874e+09	699453.96	10166566.20	42585176.30	57005096.26
MENA	6824.55	8.435678e+07	2354827.40	7407575.70	34558098.05	17824805.48
Sub-Saharan Africa	1865.80	1.935701e+08	15292.39	2164900.32	17107419.65	18074630.53
Asia	0.00	1.394874e+09	2170264.80	35697881.40	125541489.80	176102231.72
Latin America	61154.19	2.085004e+08	4660258.90	13760635.60	31303437.40	32291527.27
Industrialized	29362.36	3.248202e+08	9729979.20	26870197.60	66249338.55	54978492.49

```
    n = n()
  ) |>
  pivot_longer(cols = Min:n, names_to = "stat_name") |>
  pivot_wider(names_from = c(estim, coef_name), values_from = value)
```

```
`summarise()` has grouped output by 'estim', 'group'. You can override using
the `.groups` argument.
```

```
load("./data/output/list_countries.rda")

# All countries without SSA
results_desc_stat_wo_africa <-
  estimates_nls |>
  left_join(list_countries) |>
  mutate(estim = factor(estim, levels = c("nls"))) |>
  pivot_longer(cols = c(r, P, alpha, K), names_to = "coef_name") |>
  mutate(coef_name = factor(coef_name, levels = c("r", "P", "alpha", "K"))) |>
  filter(group != "Sub-Saharan Africa") |>
  mutate(group = "All countries without S.-S. Africa") |>
  group_by(estim, group, coef_name) |>
  summarise(
    Min = min(value),
    Max = max(value),
    Q1 = quantile(value, probs = 0.25),
    Median = median(value),
    Q3 = quantile(value, probs = 0.75),
    Mean = mean(value),
    `Standard Error` = sd(value),
    n = n()
  ) |>
  pivot_longer(cols = Min:n, names_to = "stat_name") |>
  pivot_wider(names_from = c(estim, coef_name), values_from = value)
```

```
Joining with `by = join_by(country)`
`summarise()` has grouped output by 'estim', 'group'. You can override using
the `.groups` argument.
```

```
#  SSA
results_desc_stat_africa <-
```

```
  estimates_nls |>
  left_join(list_countries) |>
  mutate(estim = factor(estim, levels = c("nls"))) |>
  pivot_longer(cols = c(r, P, alpha, K), names_to = "coef_name") |>
  mutate(coef_name = factor(coef_name, levels = c("r", "P", "alpha", "K"))) |>
  filter(group == "Sub-Saharan Africa") |>
  group_by(estim, group, coef_name) |>
  summarise(
    Min = min(value),
    Max = max(value),
    Q1 = quantile(value, probs = 0.25),
    Median = median(value),
    Q3 = quantile(value, probs = 0.75),
    Mean = mean(value),
    `Standard Error` = sd(value),
    n = n()
  ) |>
  pivot_longer(cols = Min:n, names_to = "stat_name") |>
  pivot_wider(names_from = c(estim, coef_name), values_from = value)
```

```
Joining with `by = join_by(country)`
`summarise()` has grouped output by 'estim', 'group'. You can override using
the `.groups` argument.
```

```
results_desc_stat_2 <-
  results_desc_stat_all |>
  bind_rows(results_desc_stat_wo_africa) |>
  bind_rows(results_desc_stat_africa) |>
  mutate(
  group = factor(
    group,
    levels = c("All countries", "All countries without S.-S. Africa", "Sub-Saharan Africa"
  )
) |>
  ungroup()

names(results_desc_stat_2) <- str_replace(
  names(results_desc_stat_2), "(^nls_)", ""
)

results_desc_stat_2 <-
```

```
  results_desc_stat_2 |>
  pivot_longer(cols = c(r, P, alpha, K)) |>
  pivot_wider(names_from = stat_name, values_from = value) |>
  mutate(name = factor(name, levels = c("r", "P", "alpha", "K"))) |>
  arrange(name, group)

results_desc_stat_latex_2 <-
  results_desc_stat_2 |>
  mutate(across(Min:`Standard Error`, ~ifelse(name == "K", yes = .x / 10^6, no = .x))) |>
  relocate(name, .before = group)
```

```
kableExtra::kable(
  format = "html",
  format.args = list(big.mark = ",", scientific = FALSE),
  results_desc_stat_latex_2[, -1],
  caption = "Descriptive statistics of the estimates of the Parameters of Generalized Rich
  booktabs = TRUE,
  digits = 2
) |>
  pack_rows(index = table(fct_inorder(results_desc_stat_latex_2$name))) |>
  kableExtra::kable_styling() |>
  kableExtra::scroll_box(width = "100%", box_css = "border: 0px;")
```

Table 4.2: Descriptive statistics of the estimates of the Parameters of Generalized Richards model.

group	Min	Max	Q1	Median	Q3	Mean	Standard Error	n
r								
All countries	0.00	100.00	0.06	19.58	46.41	26.23	27.90	115
All countries without S.-S. Africa	0.00	100.00	6.08	36.40	53.47	34.37	28.78	73
Sub-Saharan Africa	0.00	86.57	0.01	0.04	22.27	12.10	19.70	42
P								
All countries	0.00	1.00	0.36	0.46	0.90	0.57	0.29	115
All countries without S.-S. Africa	0.00	1.00	0.37	0.44	0.62	0.50	0.22	73
Sub-Saharan Africa	0.16	1.00	0.30	0.95	1.00	0.70	0.34	42
alpha								
All countries	0.00	26.82	0.09	0.15	3.48	2.50	4.49	115
All countries without S.-S. Africa	0.00	13.92	0.06	0.13	1.98	1.55	2.84	73
Sub-Saharan Africa	0.00	26.82	0.15	2.44	5.52	4.16	6.13	42
K								
All countries	0.00	1,394.87	0.70	10.17	42.59	57.01	184.88	115

group	Min	Max	Q1	Median	Q3	Mean	Standard Error	n
All countries without S.-S. Africa	0.00	1,394.87	4.17	16.74	52.27	79.40	227.92	73
Sub-Saharan Africa	0.00	193.57	0.02	2.16	17.11	18.07	36.98	42

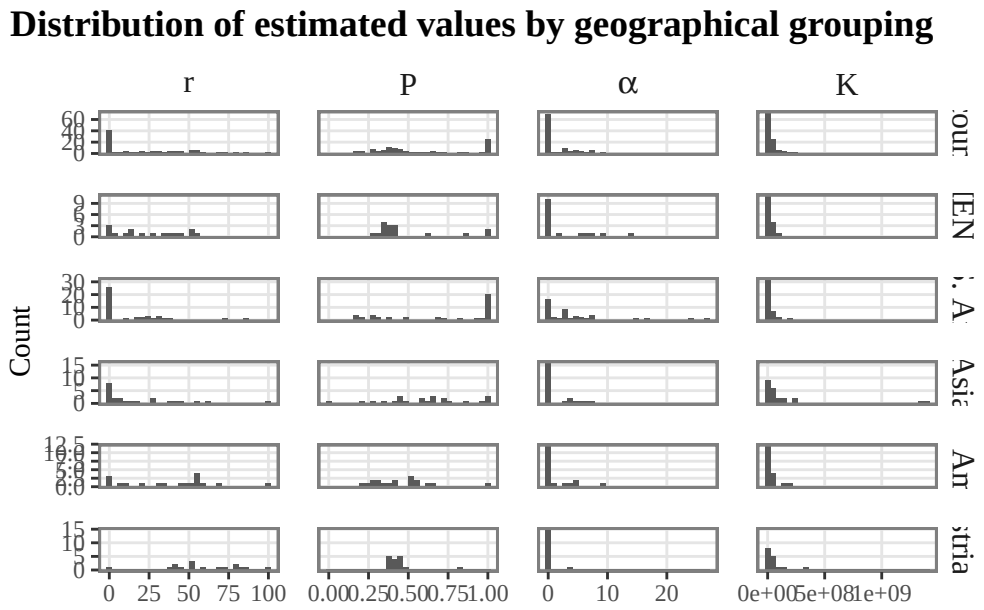
If we want to have a look at the distribution of the different estimates by geographical grouping, we can proceed as follows.

```
df_plot_density_estimates <-
  estimates_all |>
  pivot_longer(cols = c(r, P, alpha, K), names_to = "coef_name") |>
  mutate(group = "All countries") |>
  bind_rows(
    estimates_all |>
      pivot_longer(cols = c(r, P, alpha, K), names_to = "coef_name")
  ) |>
  mutate(
    coef_name = factor(
      coef_name,
      levels = c("r", "P", "alpha", "K"),
      labels = c("r", "P", "alpha", "K")),
    group = factor(
      group,
      levels = c("All countries", "MENA", "Sub-Saharan Africa", "Asia",
                 "Latin America", "Industrialized"),
      labels = c("All countries", "MENA", "S.-S. Africa", "Asia",
                 "Latin America", "Industrialized")
    )
  )
```

```
p_distrib_estimates_geo <- ggplot(
  data = df_plot_density_estimates,
  mapping = aes(x = value)
) +
  geom_histogram(mapping = aes(group = group)) +
  facet_grid(
    group~coef_name,
    scales = "free",
    labeller = labeller(coef_name = label_parsed)
  ) +
  theme_paper() +
```

```
    labs(
      x = NULL, y = "Count",
      title = "Distribution of estimated values by geographical grouping"
    )
  p_distrib_estimates_geo
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
  ggsave(
    p_distrib_estimates_geo + labs(title = NULL),
    file = "./figs/distrib_estimates_nls_geo.pdf", width = 10, height = 12
  )
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

4.3 Maps

Let us define a theme for maps:

```
#' Theme for maps with ggplot2
#'
#' @param ... arguments passed to the theme function
#' @export
#' @importFrom ggplot2 element_rect element_text element_blank element_line unit
#'   rel
theme_map_paper <- function(...) {
  theme(
    text = element_text(family = "Times"),
    plot.background = element_rect(fill = "transparent", color = NA),
    panel.background = element_rect(fill = "transparent", color = NA),
    panel.border = element_blank(),
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank(), axis.line = element_blank(),
    plot.title.position = "plot",
    legend.text = element_text(size = rel(1.2)),
    legend.title = element_text(size = rel(1.2)),
    legend.background = element_rect(fill="transparent", color = NULL),
    legend.key = element_blank(),
    legend.key.height   = unit(2, "line"),
    legend.key.width    = unit(1.5, "line"),
    strip.background = element_rect(fill = NA),
    panel.spacing = unit(1, "lines"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    plot.margin = unit(c(1, 1, 1, 1), "lines"),
    strip.text = element_text(size = rel(1.2))
  )
}
```

A Shapefile that allows us to display the level 0 world administrative boundaries, freely available on the online open data platform "opendatasoft", can be loaded:

```
library(sf)
```

```
Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
world_map <- sf::read_sf(
  "./data/raw/world-administrative-boundaries/world-administrative-boundaries.shp"
)
```

There are some mismatching country names between the map file and the epidemic data. The names from the map file can be manually changed:

```
world_map <-
  world_map |>
  mutate(
    name = recode(
      name,
      # old = new
      "Brunei Darussalam" = "Brunei",
      "Côte d'Ivoire" = "Cote d'Ivoire",
      "Democratic Republic of the Congo" = "Democratic Republic of Congo",
      "Swaziland" = "Eswatini",
      "Iran (Islamic Republic of)" = "Iran",
      "Lao People's Democratic Republic" = "Laos",
      "Russian Federation" = "Russia",
      "Republic of Korea" = "South Korea",
      "Syrian Arab Republic" = "Syria",
      "United Republic of Tanzania" = "Tanzania",
      "U.K. of Great Britain and Northern Ireland" = "United Kingdom",
      "United States of America" = "United States"
    )
  )
```

Let us add the estimated values for each country to the map data:

```
map_nls <-
  world_map |>
  left_join(
    estimates_all |>
      filter(estim == "nls"),
    by = c("name" = "country")
  )
```

```
ggplot(data = map_nls) +
  geom_sf(mapping = aes(fill = P), colour = "gray95", size = .1) +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "gray80") +
  theme_map_paper()
```

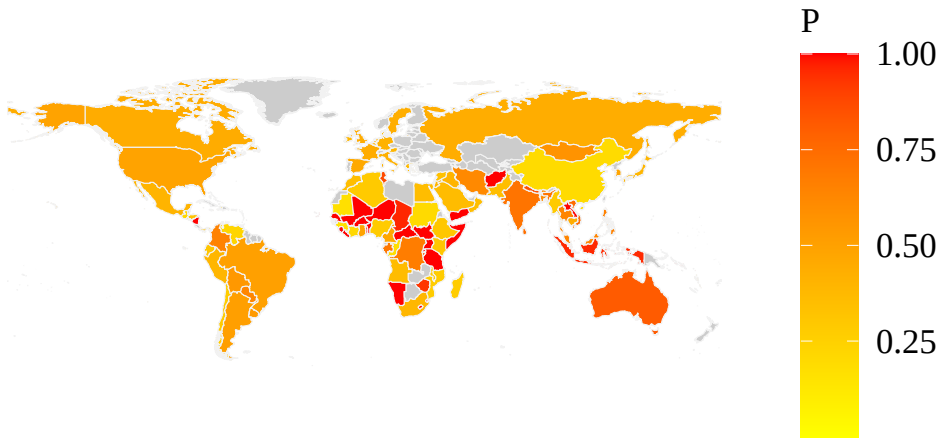


Figure 4.1: Spatial distribution of estimated values for P by countries (NLS).

```
map_nls_P <-
  ggplot(
    data = map_nls |>
      mutate(growth = ifelse(P < 1, "Sub-exponential", "Malthusian"))
  ) +
  geom_sf(mapping = aes(fill = growth), colour = "gray95", size = .1) +
  scale_fill_manual(
    expression(widehat(P)),
    values = c("Sub-exponential" = "#117733", "Malthusian" = "#88CCEE"),
    na.value = "gray80"
  ) +
  theme_map_paper() +
  theme(legend.position = "bottom")
map_nls_P
```

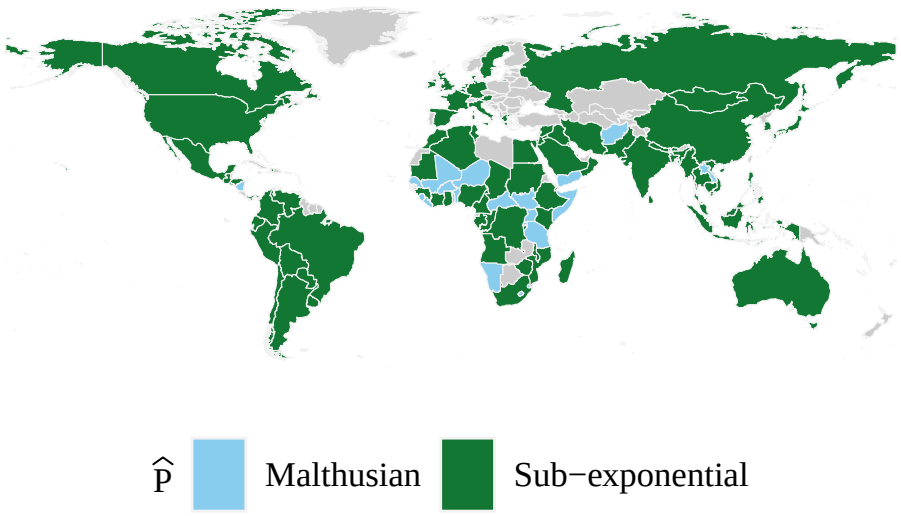


Figure 4.2: Spatial distribution of estimated values for ˆP by countries (NLS).

```
ggsave(map_nls_P, file = "./figs/map_nls_P.png", width = 12, height = 6)
```

```
map_nls_alpha <-
  ggplot(
    data = map_nls |>
      mutate(val = ifelse(alpha < 1, "<1", ">1"))
  ) +
  geom_sf(mapping = aes(fill = val), colour = "gray95", size = .1) +
  scale_fill_manual(
    expression(widehat(alpha)),
    values = c("<1" = "#CC6677", ">1" = "#882255"),
    na.value = "gray80"
  ) +
  theme_map_paper() +
  theme(legend.position = "bottom")
map_nls_alpha
```

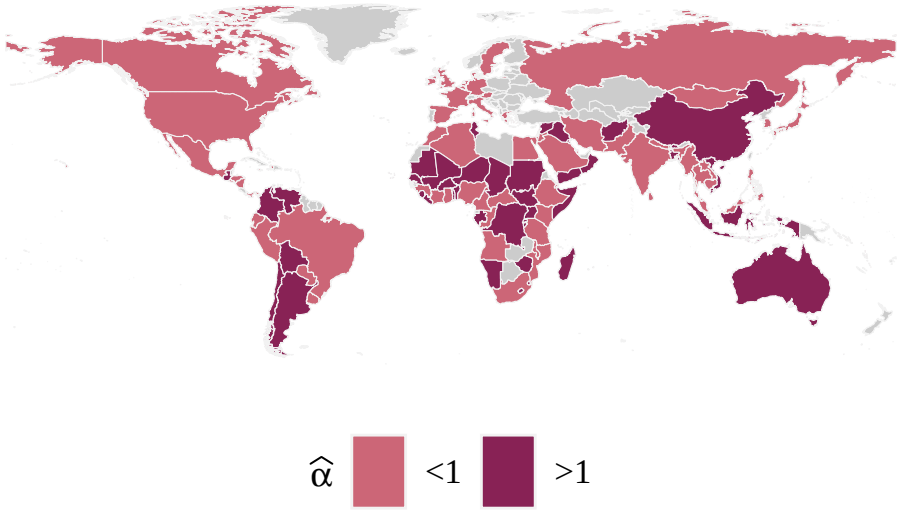


Figure 4.3: Spatial distribution of estimated values for ˆP by countries (NLS).

```
ggsave(map_nls_alpha, file = "./figs/map_nls_alpha.png", width = 12, height = 6)
```

```
estimates_nls |>
  left_join(list_countries) |>
  pivot_longer(cols = c(r, P, alpha, K)) |>
  group_by(name, group) |>
  mutate(
    estim = factor(estim, levels = c("nls"),
                   labels = c("Non linear Least Square"))
  ) |>
  filter(estim == "Non linear Least Square") |>
  filter(name %in% c("alpha", "P")) |>
  filter((name == "alpha" & value > 500))
```

```
Joining with `by = join_by(country)`
```

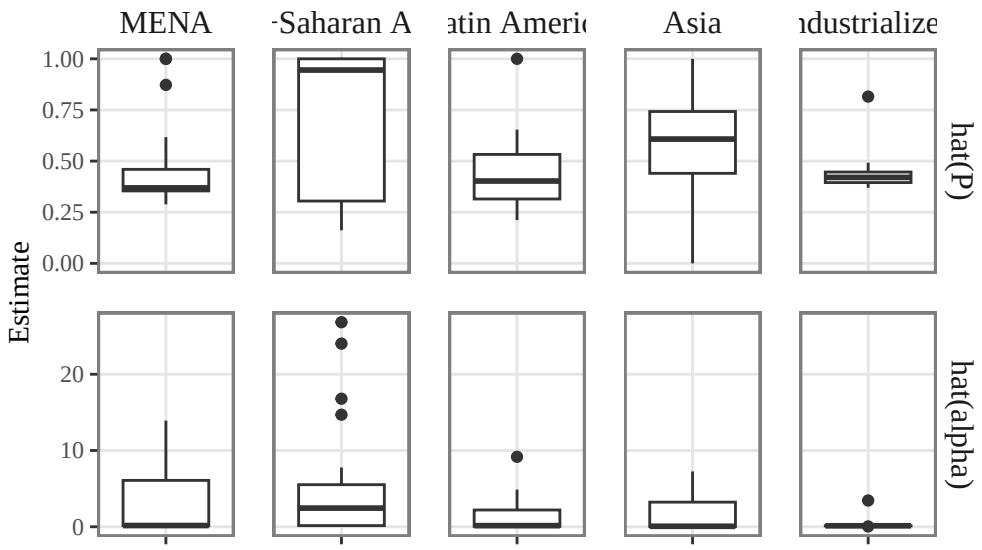
```
# A tibble: 0 x 5
# Groups:   name, group [0]
# i 5 variables: country <chr>, estim <fct>, group <chr>, name <chr>,
#   value <dbl>
```

```r
p_nls_alpha_P <-
  ggplot(
    data = estimates_nls |>
      left_join(list_countries) |>
      pivot_longer(cols = c(r, P, alpha, K)) |>
      group_by(name, group) |>
      mutate(
        estim = factor(estim, levels = c("nls"),
                       labels = c("Non linear Least Square"))
      ) |>
      filter(estim == "Non linear Least Square") |>
      filter(name %in% c("alpha", "P")) |>
      filter(!(name == "alpha" & value > 500)) |>
      mutate(
        name = factor(
          name,
          levels = c("P", "alpha"),
          labels = c(expression(hat(P)), expression(hat(alpha)))
        )
      ) |>
      mutate(
        group = factor(
          group,
          levels = c("MENA", "Sub-Saharan Africa", "Latin America", "Asia",
                     "Industrialized")
        )
      )
  ) +
  geom_boxplot(aes(x = estim, y = value), na.rm = TRUE) +
  facet_grid(name~group, scales = "free", labeller = labeller(type = label_parsed)) +
  labs(x = NULL, y = "Estimate",
       title = "Estimated coefficients for the Generalized Richards Model") +
  theme_paper() +
  theme(axis.text.x=element_blank())
```

```
Joining with `by = join_by(country)`
```

```r
p_nls_alpha_P
```

Estimated coefficients for the Generalized Richards Model




```
ggsave(
  p_nls_alpha_P + labs(title = NULL),
  file = "./figs/p_nls_alpha_P.pdf", width = 12, height = 6
)
```

5 Ordered Multinomial Models

```
# MEAN
formula_mean_v1 <- v1 ~ s_1 + s_2 + s_3 + s_4 + s_5 + s_6
formula_mean_v1_wo_ssa <- v1 ~ s_1 + s_2 + s_3 + s_4 + s_5 + s_6
formula_mean_v2 <- v2 ~ s_1 + s_2 + s_3 + s_4 + s_5 + s_6
formula_mean_v2_wo_ssa <- v2 ~ s_1 + s_2 + s_3 + s_4 + s_5 + s_6


# VARIANCE
formula_variance_v1 <- ~ s_1 + s_2 + s_4
formula_variance_v1_wo_ssa <- ~ s_1 + s_2 + s_4
formula_variance_v2 <- formula_variance_v2_wo_ssa <- ~ s_1 + s_2 + s_3 + s_4
```

Rather than looking at the effects of socioeconomic variables on the cumulative number of cases, we look at the effects during the different phases of the incidence curve, *i.e* on the parameters that identify the severity of the ascending and declining phases of the crises (slow or rapid) and the dynamics of the disease in the early stage of a pandemic. Our two variables of interest are $\hat{P}_i$ and $\hat{\alpha}_i$, where the sub-index i refers to a country (we have a sample of 103 countries).

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```
library(kableExtra)
```

```
Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows
```

Let us load the results obtained from the estimation of the generalized model:

```
load("./estim/resul_fit_countries.rda")
```

The coordinates obtained from the PCA:

```
load("./estim/results_pca/df_coordinates.rda")
```

```
estimates_nls <-
  map(resul_fit_countries, "nls") |>
  map_df("estimates", .id = "country") |>
  mutate(estim = "nls")


df <-
  estimates_nls |>
  left_join(
    df_coordinates,
    by = "country"
  ) |>
  rename(
    s_1 = coord_norm_s1,
    s_2 = coord_norm_s2,
    s_3 = coord_norm_s3,
    s_4 = coord_norm_s4,
    s_5 = coord_norm_s5,
    s_6 = coord_norm_s6
  )
```

For $\hat{P}_i$ we define a three-level multidimensional ordered variable to characterize the dynamics during the initial phase of an epidemic:

$$V_{1i} = \begin{cases} 1, & \text{if } 0 \leq \hat{P}_i \leq P_{.25}, \\ 2, & \text{if } P_{.25} < \hat{P}_i \leq P_{.75}, \\ 3, & \text{if } P_{.75} < \hat{P}_i \leq 1. \end{cases} \tag{5.1}$$

where $P_{.25}$ and $P_{.75}$ are chosen as the first and third quartiles of the observations of the vector $(\hat{P}_{i=1,\ldots,100})$. The values 1,2,3 mean respectively "slow", "medium" and "fast".

For $\hat{\alpha}_i$ we define a qualitative ordered variable with two characteristics: the speed of increase or decrease of an epidemic is either lower than the logistic decay ($\hat{\alpha}_i \leq 1$) or higher than the logistic decay ($\hat{\alpha}_i > 1$):

$$V_{2i} = \begin{cases} 1, & \text{if } \hat{\alpha}_i \leq 1, \\ 2, & \text{if } \hat{\alpha}_i > 1. \end{cases} \tag{5.2}$$

where 1 means "lower" and 2 means "higher".

Let us define those variables:

```
df <-
  df |>
  mutate(
    v2 = case_when(
      alpha <= 1 ~ 1,
      TRUE ~ 2
    ),
    v1 = case_when(
      P >= 0 & P <= quantile(P, probs = 0.25) ~ 1,
      P > quantile(P, probs = 0.25) & P <= quantile(P, probs = 0.75) ~ 2,
      P > quantile(P, probs = 0.75) & P <= 1 ~ 3,
      TRUE ~ 4
    )
  )
```

Sanity check:

```
table(df$v1)
```

```
 1  2  3
29 57 29
```

```
table(df$v2)
```

```
 1  2
69 46
```

We estimate ordered multinomial logit models, using the {oglmx} package. For each endogenous variable, we consider the following latent model:

$$V_{mi}^{*} = \sum_{k=1}^{6} \beta_k \; S_{ki} + \sigma_i \; \epsilon_i, \quad \epsilon_{\kappa t} \approx N(0,1), \; m = 1, 2.$$

The latent variable allows to define a partition of the endogenous variables as follows:

$$\hat{V}_{1i} = \begin{cases} 1, & \text{if } V_{1i}^{*} \in (-\infty, \lambda_1], \\ 2, & \text{if } V_{1i}^{*} \in (\lambda_1, \lambda_2] \\ 3, & \text{if } V_{1i}^{*} \in (\lambda_2, \infty). \end{cases} \tag{5.3}$$

and

$$\hat{V}_{2i} = \begin{cases} 1, & \text{if } V_{2i}^{*} \in (-\infty, \gamma_1], \\ 2, & \text{if } V_{2i}^{*} \in (\gamma_1, \infty). \end{cases} \tag{5.4}$$

The coefficient $\lambda_1 > \lambda_2$ and γ_1 are unknown parameters. Define $S_i = (S_{1i}, ..., S_{6i})$ the vector of explanatory variables and β the vector of parameters to be estimated. The probability of an outcome conditional on the explanatory variables is:

$$Pr\left(\hat{V}_{mi} = \nu \mid S_i, \; \beta\right) = \zeta\left(\kappa_{\nu+1} - S_i' \; \beta\right) - \zeta\left(\kappa_{\nu} - S_i' \; \beta\right).$$

where $\nu = 1, 2, 3$ or $\nu = 1, 2$ depending upon whether we consider the model (Equation 5.3) or (Equation 5.4). κ refers to the coefficients λ and γ. $\zeta(.)$ is the CDF of a logit model.\

5.1 Estimations

```
library(oglmx)
```

5.2 Impact on V1

Let us consider the estimation of the models that characterize the dynamics during the initial phase of the epidemic.

5.2.1 With all the Countries

In a first step, let us consider all the countries.

```r
res_v1_o_prob <- oglmx(
  formulaMEAN = formula_mean_v1,
  formulaSD = formula_variance_v1,
  data = df,
  link = "probit",
  constantMEAN = FALSE,
  constantSD = FALSE,
  delta = 0,
  threshparam = NULL
)
```

5.2.1.1 Summary

```r
summary(res_v1_o_prob)
```

```
Heteroskedastic Ordered Probit Regression
Log-Likelihood: -88.01135
No. Iterations: 11
McFadden's R2: 0.2660223
AIC: 196.0227
----- Mean Equation ------
    Estimate Std. error t value Pr(>|t|)
s_1   5.4686     3.4844  1.5695  0.11654
s_2   4.4385     3.6777  1.2069  0.22748
s_3  -2.1331     1.5418 -1.3835  0.16650
s_4   9.2440     6.9624  1.3277  0.18428
s_5   3.0956     3.6597  0.8459  0.39763
s_6  -4.9314     2.7665 -1.7825  0.07466 .
----- SD Equation ------
    Estimate Std. error t value  Pr(>|t|)
s_2 -1.61034    1.51725 -1.0614    0.2885
s_4  4.45907    0.99082  4.5004 6.783e-06 ***
----- Threshold Parameters -----
                 Estimate Std. error t value Pr(>|t|)
Threshold (1->2) -0.26948    3.93409 -0.0685   0.9454
Threshold (2->3)  5.78562    4.85658  1.1913   0.2335
```

5.2.1.2 Marginal Effects

Let us now compute the marginal effects, using the `margins.oglmx()` function from {oglmx}.

```
marginal_effects_v1 <- margins.oglmx(res_v1_o_prob, ascontinuous = TRUE)
```

Then, we can format the estimated values.

```
me_v1_table <-
  map(marginal_effects_v1, as_tibble, rownames = "variable") |>
  list_rbind(names_to = "state") |>
  mutate(
    variable = factor(
      variable,
      labels = c(
        "s_1" = "Healthcare infrastructure",
        "s_2" = "Vulnerability to comorbidites",
        "s_3" = "Vulnerability to natural environment",
        "s_4" = "Living conditions",
        "s_5" = "Economic and societal characteristics",
        "s_6" = "Policy variables and governance"
      )
    ),
    star = case_when(
      `Pr(>|t|)` < 0.01 ~ "***",
      `Pr(>|t|)` < 0.05 ~ "**",
      `Pr(>|t|)` < 0.1 ~ "*",
      TRUE ~ ""
    )
  ) |>
  mutate(prob_state = str_c(round(`Marg. Eff`, 2), star)) |>
  mutate(std_error = str_c("(", round(`Std. error`, 2), ")")) |>
  select(variable, state, prob_state, std_error) |>
  pivot_longer(c(prob_state, std_error)) |>
  pivot_wider(names_from = state, values_from = value) |>
  select(-name) |>
  mutate(variable = as.character(variable)) |>
  group_by(variable) |>
  mutate(variable = ifelse(row_number() == 1, variable, ""))
```

We can visualize the results as follows:

```
me_v1_table |>
  mutate(
    `1` = kableExtra::cell_spec(
      `1`,
      color = case_when(
        str_detect(`1`, "^\\(|-", negate = TRUE) ~ "#009D57",
        str_detect(`1`, "^-") ~ "#EE324E",
        TRUE ~ "black"
      )
    ),
    `2` = kableExtra::cell_spec(
      `2`,
      color = case_when(
        str_detect(`2`, "^\\(|-", negate = TRUE) ~ "#009D57",
        str_detect(`2`, "^-") ~ "#EE324E",
        TRUE ~ "black"
      )
    ),
    `3` = kableExtra::cell_spec(
      `3`,
      color = case_when(
        str_detect(`3`, "^\\(|-", negate = TRUE) ~ "#009D57",
        str_detect(`3`, "^-") ~ "#EE324E",
        TRUE ~ "black"
      )
    )
  ) |>
  ungroup() |>
  knitr::kable(
    escape = F, booktabs = T, format = "html",
    caption = str_c(
      "Marginal effects of the synthetic variables ",
      "on the probability of being in each regime for V1"
    ),
    col.names = c(
      "Variable", "\\(\\mathbb{P}(V_1 = 1)\\)<br>(Slow Start)",
      "\\(\\mathbb{P}(V_1 = 2)\\)<br>(Medium Start)",
      "\\(\\mathbb{P}(V_1 = 3)\\)<br>(Fast Start)"
    )
  ) |>
  kableExtra::add_header_above(c(" ", "Marginal effect on:" = 3)) |>
```

```
  kableExtra::kable_styling()
```

Table 5.1: Marginal effects of the synthetic variables on the probability of being in each regime for V1

Variable	Marginal effect on: \(\mathbb{P}(V_1 = 1)\) (Slow Start)	\(\mathbb{P}(V_1 = 2)\) (Medium Start)	\(\mathbb{P}(V_1 = 3)\) (Fast Start)
Healthcare infrastructure	\textcolor[HTML]{EE324E}{-0.36**} \textcolor{black}{(0.17)}	\textcolor[HTML]{EE324E}{-0.05} \textcolor{black}{(0.1)}	\textcolor[HTML]{009D57}{0.41*} \textcolor{black}{(0.23)}
Vulnerability to comorbidites	\textcolor[HTML]{EE324E}{-0.66} \textcolor{black}{(0.46)}	\textcolor[HTML]{009D57}{0.64} \textcolor{black}{(0.64)}	\textcolor[HTML]{009D57}{0.02} \textcolor{black}{(0.32)}
Living conditions	\textcolor[HTML]{009D57}{0.42} \textcolor{black}{(0.39)}	\textcolor[HTML]{EE324E}{-1.99***} \textcolor{black}{(0.52)}	\textcolor[HTML]{009D57}{1.57***} \textcolor{black}{(0.47)}
Vulnerability to natural environment	\textcolor[HTML]{009D57}{0.14} \textcolor{black}{(0.1)}	\textcolor[HTML]{009D57}{0.02} \textcolor{black}{(0.04)}	\textcolor[HTML]{EE324E}{-0.16} \textcolor{black}{(0.11)}
Economic and societal characteristics	\textcolor[HTML]{EE324E}{-0.2} \textcolor{black}{(0.23)}	\textcolor[HTML]{EE324E}{-0.03} \textcolor{black}{(0.06)}	\textcolor[HTML]{009D57}{0.23} \textcolor{black}{(0.26)}
Policy variables and governance	\textcolor[HTML]{009D57}{0.33*} \textcolor{black}{(0.18)}	\textcolor[HTML]{009D57}{0.05} \textcolor{black}{(0.08)}	\textcolor[HTML]{EE324E}{-0.37*} \textcolor{black}{(0.21)}

5.2.2 Without Sub-Saharan Africa

Let us reestimate the same model, but without Sub-Saharan Africa.

```
results_o_prob_wo_ssa <- oglmx(
  formulaMEAN = formula_mean_v1_wo_ssa,
  formulaSD = formula_variance_v1_wo_ssa,
  data = df |> filter(!group %in% "Sub-Saharan Africa"),
  link = "probit",
  constantMEAN = FALSE,
  constantSD = FALSE,
```

```
  delta = 0,
  threshparam = NULL
)
```

5.2.2.1 Summary

```
summary(results_o_prob_wo_ssa)
```

```
Heteroskedastic Ordered Probit Regression
Log-Likelihood: -47.34021
No. Iterations: 11
McFadden's R2: 0.220331
AIC: 114.6804
----- Mean Equation ------
    Estimate Std. error t value Pr(>|t|)
s_1   4.8333     4.2913  1.1263   0.2600
s_2   5.4116     6.5549  0.8256   0.4090
s_3  -1.7482     1.7799 -0.9822   0.3260
s_4   6.8768     7.2773  0.9450   0.3447
s_5   3.8568     3.8467  1.0026   0.3160
s_6  -5.0477     3.1967 -1.5790   0.1143
----- SD Equation ------
    Estimate Std. error t value Pr(>|t|)
s_2   1.3705     2.2431  0.6110   0.5412
s_4   2.4486     1.6835  1.4544   0.1458
----- Threshold Parameters -----
                 Estimate Std. error t value Pr(>|t|)
Threshold (1->2) -0.20451    4.24128 -0.0482   0.9615
Threshold (2->3)  5.99920    5.44750  1.1013   0.2708
```

5.2.2.2 Marginal Effects

```
marginal_effects_v1_wo_ssa <- margins.oglmx(
  results_o_prob_wo_ssa,
  ascontinuous = TRUE
)
```

Let us format the results.

```
me_v1_wo_ssa_table <-
  map(marginal_effects_v1_wo_ssa, as_tibble, rownames = "variable") |>
  list_rbind(names_to = "state") |>
  mutate(
    variable = factor(
      variable,
      labels = c(
        "s_1" = "Healthcare infrastructure",
        "s_2" = "Vulnerability to comorbidites",
        "s_3" = "Vulnerability to natural environment",
        "s_4" = "Living conditions",
        "s_5" = "Economic and societal characteristics",
        "s_6" = "Policy variables and governance"
      )
    ),
    star = case_when(
      `Pr(>|t|)` < 0.01 ~ "***",
      `Pr(>|t|)` < 0.05 ~ "**",
      `Pr(>|t|)` < 0.1 ~ "*",
      TRUE ~ ""
    )
  ) |>
  mutate(prob_state = str_c(round(`Marg. Eff`, 2), star)) |>
  mutate(std_error = str_c("(", round(`Std. error`, 2), ")")) |>
  select(variable, state, prob_state, std_error) |>
  pivot_longer(c(prob_state, std_error)) |>
  pivot_wider(names_from = state, values_from = value) |>
  select(-name) |>
  mutate(variable = as.character(variable)) |>
  group_by(variable) |>
  mutate(variable = ifelse(row_number() == 1, variable, ""))
```

They can then be displayed:

```
me_v1_wo_ssa_table |>
  mutate(
    `1` = kableExtra::cell_spec(
      `1`,
      color = case_when(
        str_detect(`1`, "^\\(|-", negate = TRUE) ~ "#009D57",
        str_detect(`1`, "^-") ~ "#EE324E",
        TRUE ~ "black"
```

```
    )
  ),
  `2` = kableExtra::cell_spec(
    `2`,
    color = case_when(
      str_detect(`2`, "^\\(|-", negate = TRUE) ~ "#009D57",
      str_detect(`2`, "^-") ~ "#EE324E",
      TRUE ~ "black"
    )
  ),
  `3` = kableExtra::cell_spec(
    `3`,
    color = case_when(
      str_detect(`3`, "^\\(|-", negate = TRUE) ~ "#009D57",
      str_detect(`3`, "^-") ~ "#EE324E",
      TRUE ~ "black"
    )
  )
) |>
ungroup() |>
knitr::kable(
  escape = F, booktabs = T, format = "html",
  caption = str_c(
    "Marginal effects of the synthetic variables ",
    "on the probability of being in each regime for V1, ",
    "without Sub-Saharan Africa"
  ),
  col.names = c(
    "Variable", "\\(\\mathbb{P}(V_1 = 1)\\)<br>(Slow Start)",
    "\\(\\mathbb{P}(V_1 = 2)\\)<br>(Medium Start)",
    "\\(\\mathbb{P}(V_1 = 3)\\)<br>(Fast Start)"
  )
) |>
kableExtra::add_header_above(c(" ", "Marginal effect on:" = 3)) |>
kableExtra::kable_styling()
```

Table 5.2: Marginal effects of the synthetic variables on the probability of being in each regime for V1, without Sub-Saharan Africa

	Marginal effect on:		
Variable	\(\mathbb{P}(V_1 = 1)\) (Slow Start)	\(\mathbb{P}(V_1 = 2)\) (Medium Start)	\(\mathbb{P}(V_1 = 3)\) (Fast Start)
Healthcare infrastructure	\textcolor[HTML]{EE324E}{-0.54} \textcolor{black}{(0.37)}	\textcolor[HTML]{009D57}{0.33} \textcolor{black}{(0.27)}	\textcolor[HTML]{009D57}{0.2} \textcolor{black}{(0.17)}
Vulnerability to comorbidites	\textcolor[HTML]{EE324E}{-0.27} \textcolor{black}{(0.87)}	\textcolor[HTML]{EE324E}{-0.19} \textcolor{black}{(1.09)}	\textcolor[HTML]{009D57}{0.46} \textcolor{black}{(0.52)}
Living conditions	\textcolor[HTML]{EE324E}{-0.18} \textcolor{black}{(0.84)}	\textcolor[HTML]{EE324E}{-0.53} \textcolor{black}{(0.91)}	\textcolor[HTML]{009D57}{0.71*} \textcolor{black}{(0.42)}
Vulnerability to natural environment	\textcolor[HTML]{009D57}{0.19} \textcolor{black}{(0.19)}	\textcolor[HTML]{EE324E}{-0.12} \textcolor{black}{(0.12)}	\textcolor[HTML]{EE324E}{-0.07} \textcolor{black}{(0.08)}
Economic and societal characteristics	\textcolor[HTML]{EE324E}{-0.43} \textcolor{black}{(0.4)}	\textcolor[HTML]{009D57}{0.26} \textcolor{black}{(0.28)}	\textcolor[HTML]{009D57}{0.16} \textcolor{black}{(0.16)}
Policy variables and governance	\textcolor[HTML]{009D57}{0.56*} \textcolor{black}{(0.34)}	\textcolor[HTML]{EE324E}{-0.35} \textcolor{black}{(0.26)}	\textcolor[HTML]{EE324E}{-0.21} \textcolor{black}{(0.15)}

5.2.3 Reproduction of Table 2

```
me_v1_both <-
  map(marginal_effects_v1, as_tibble, rownames = "variable") |>
  list_rbind(names_to = "state") |>
  mutate(model = "All") |>
  bind_rows(
    map(marginal_effects_v1_wo_ssa, as_tibble, rownames = "variable") |>
      list_rbind(names_to = "state") |>
      mutate(model = "Without SSA")
  ) |>
  mutate(
    variable = factor(
```

```
      variable,
      labels = c(
        "s_1" = "Healthcare infrastructure",
        "s_2" = "Vulnerability to comorbidites",
        "s_3" = "Vulnerability to natural environment",
        "s_4" = "Living conditions",
        "s_5" = "Economic and societal characteristics",
        "s_6" = "Policy variables and governance"
      )
    ),
    star = case_when(
      `Pr(>|t|)` < 0.01 ~ "***",
      `Pr(>|t|)` < 0.05 ~ "**",
      `Pr(>|t|)` < 0.1 ~ "*",
      TRUE ~ ""
    )
  ) |>
  mutate(prob_state = str_c(round(`Marg. Eff`, 2), star)) |>
  mutate(std_error = str_c("(", round(`Std. error`, 2), ")")) |>
  select(model, variable, state, prob_state, std_error) |>
  pivot_longer(c(prob_state, std_error)) |>
  pivot_wider(names_from = c(model, state), values_from = value) |>
  select(-name) |>
  mutate(variable = as.character(variable)) |>
  group_by(variable) |>
  mutate(variable = ifelse(row_number() == 1, variable, ""))
```

```
me_v1_both |>
  mutate(
    across(
      All_1:`Without SSA_3`,
      ~kableExtra::cell_spec(
        .x,
        color = case_when(
          str_detect(.x, "^\\(|-", negate = TRUE) ~ "#009D57",
          str_detect(.x, "^-") ~ "#EE324E",
          TRUE ~ "black"
        )
      )
    )
  ) |>
```

```
  ungroup() |>
  knitr::kable(
    escape = F, booktabs = T, format = "html",
    caption = str_c(
      "Marginal effects of the synthetic variables ",
      "on the probability of being in each regime for V1"
    ),
    col.names = c(
      "Variable",
      rep(
        c(
          "\\(\\mathbb{P}(V_1 = 1)\\)<br>(Slow Start)",
          "\\(\\mathbb{P}(V_1 = 2)\\)<br>(Medium Start)",
          "\\(\\mathbb{P}(V_1 = 3)\\)<br>(Fast Start)"
        ),
        2
      )
    )
  ) |>
  kableExtra::add_header_above(
    c(" ", "All countries" = 3, "Without Sub-Saharan Africa" = 3)
  ) |>
  kableExtra::kable_styling()
```

Table 5.3: Marginal effects of the synthetic variables on the probability of being in each regime for V1

	All countries			Without Sub-Saharan Africa		
Variable	\(\mathbb{P}(V_1 = 1)\) (Slow Start)	\(\mathbb{P}(V_1 = 2)\) (Medium Start)	\(\mathbb{P}(V_1 = 3)\) (Fast Start)	\(\mathbb{P}(V_1 = 1)\) (Slow Start)	\(\mathbb{P}(V_1 = 2)\) (Medium Start)	\(\mathbb{P}(V_1 = 3)\) (Fast Start)
Healthcare infrastructure	\textcolor[HTML]{009D57}{0.36**}	\textcolor[HTML]{009D57}{0.05}		\textcolor[HTML]{009D57}{0.54}		
	\textcolor{black}{(0.17)}	\textcolor{black}{(0.1)}		\textcolor{black}{(0.23)}	\textcolor{black}{(0.27)}	
Vulnerability to comorbidites	\textcolor[HTML]{009D57}{0.66}			\textcolor[HTML]{009D57}{0.27}	\textcolor[HTML]{009D57}{0.19}	
	\textcolor{black}{(0.46)}	\textcolor{black}{(0.64)}		\textcolor{black}{(0.32)}	\textcolor{black}{(0.37)}	\textcolor{black}{(0.52)}

	All countries			Without Sub-Saharan Africa		
Variable	\(\mathbb{P}(V_1 = 1)\) (Slow Start)	\(\mathbb{P}(V_1 = 2)\) (Medium Start)	\(\mathbb{P}(V_1 = 3)\) (Fast Start)	\(\mathbb{P}(V_1 = 1)\) (Slow Start)	\(\mathbb{P}(V_1 = 2)\) (Medium Start)	\(\mathbb{P}(V_1 = 3)\) (Fast Start)
Living conditions	\textcolor[HTML]{009D57}{1.99***}			\textcolor[HTML]{009D57}{0.18}	\textcolor[HTML]{009D57}{0.53}	
	\textcolor{black}{(0.39)}	\textcolor{black}{(0.52)}	\textcolor{black}{(0.47)}	\textcolor{black}{(0.84)}	\textcolor{black}{(0.91)}	\textcolor{black}{(0.42)}
Vulnerability to natural environment	\textcolor[HTML]{EE324E}{-0.16}		\textcolor[HTML]{009D57}{0.16}	\textcolor[HTML]{EE324E}{-0.12}	\textcolor[HTML]{009D57}{0.07}	
	\textcolor{black}{(0.11)}	\textcolor{black}{(0.04)}	\textcolor{black}{(0.11)}	\textcolor{black}{(0.19)}	\textcolor{black}{(0.12)}	\textcolor{black}{(0.08)}
Economic and societal characteristics	\textcolor[HTML]{009D57}{0.2}	\textcolor[HTML]{EE324E}{-0.03}		\textcolor[HTML]{009D57}{0.43}		
	\textcolor{black}{(0.23)}	\textcolor{black}{(0.06)}	\textcolor{black}{(0.26)}	\textcolor{black}{(0.4)}	\textcolor{black}{(0.28)}	\textcolor{black}{(0.16)}
Policy variables and governance	\textcolor[HTML]{EE324E}{-0.37}		\textcolor[HTML]{009D57}{0.37*}	\textcolor[HTML]{EE324E}{-0.35}	\textcolor[HTML]{009D57}{0.21}	
	\textcolor{black}{(0.18)}	\textcolor{black}{(0.08)}	\textcolor{black}{(0.21)}	\textcolor{black}{(0.34)}	\textcolor{black}{(0.26)}	\textcolor{black}{(0.15)}

5.3 Impact on V2

Let us now turn to the estimation of V_2.

5.3.1 With all the Countries

```
results_o_prob_v2 <- oglmx(
  formulaMEAN = formula_mean_v2,
  formulaSD = formula_variance_v2,
  data = df,
  link = "probit",
  constantMEAN = FALSE,
  constantSD = FALSE,
  delta = 0,
```

```
  threshparam = NULL
)
```

5.3.1.1 Summary

```
summary(results_o_prob_v2)
```

```
Heteroskedastic Probit Regression
Log-Likelihood: -65.90037
No. Iterations: 17
McFadden's R2: 0.1485338
AIC: 151.8007
----- Mean Equation ------
     Estimate Std. error t value Pr(>|t|)
s_1 -1.112638   1.167421 -0.9531   0.3406
s_2 -1.715519   1.763663 -0.9727   0.3307
s_3  0.644278   0.616647  1.0448   0.2961
s_4  1.713390   1.821409  0.9407   0.3469
s_5  0.835133   1.248853  0.6687   0.5037
s_6  0.072766   0.439697  0.1655   0.8686
----- SD Equation ------
    Estimate Std. error t value Pr(>|t|)
s_2  5.27218    3.80034  1.3873  0.16535
s_3 -0.19134    1.25145 -0.1529  0.87848
s_4 -3.43431    2.02201 -1.6985  0.08942 .
----- Threshold Parameters -----
                 Estimate Std. error t value Pr(>|t|)
Threshold (1->2)   1.2329     1.5020  0.8208   0.4118
```

5.3.1.2 Marginal Effects

```
marginal_effects_v2 <- margins.oglmx(results_o_prob_v2, ascontinuous = TRUE)
```

```
me_v2_table <-
  map(marginal_effects_v2, as_tibble, rownames = "variable") |>
  list_rbind(names_to = "state") |>
  mutate(
    variable = factor(
```

```
      variable,
      labels = c(
        "s_1" = "Healthcare infrastructure",
        "s_2" = "Vulnerability to comorbidites",
        "s_3" = "Vulnerability to natural environment",
        "s_4" = "Living conditions",
        "s_5" = "Economic and societal characteristics",
        "s_6" = "Policy variables and governance"
      )
    ),
    star = case_when(
      `Pr(>|t|)` < 0.01 ~ "***",
      `Pr(>|t|)` < 0.05 ~ "**",
      `Pr(>|t|)` < 0.1 ~ "*",
      TRUE ~ ""
    )
  ) |>
  mutate(prob_state = str_c(round(`Marg. Eff`, 2), star)) |>
  mutate(std_error = str_c("(", round(`Std. error`, 2), ")")) |>
  select(variable, state, prob_state, std_error) |>
  pivot_longer(c(prob_state, std_error)) |>
  pivot_wider(names_from = state, values_from = value) |>
  select(-name) |>
  mutate(variable = as.character(variable)) |>
  group_by(variable) |>
  mutate(variable = ifelse(row_number() == 1, variable, ""))
```

```
me_v2_table |>
  mutate(
    `1` = kableExtra::cell_spec(
      `1`,
      color = case_when(
        str_detect(`1`, "^\\(|-", negate = TRUE) ~ "#009D57",
        str_detect(`1`, "^-") ~ "#EE324E",
        TRUE ~ "black"
      )
    ),
    `2` = kableExtra::cell_spec(
      `2`,
      color = case_when(
        str_detect(`2`, "^\\(|-", negate = TRUE) ~ "#009D57",
```

```
        str_detect(`2`, "^-") ~ "#EE324E",
        TRUE ~ "black"
      )
    )
  ) |>
  ungroup() |>
  knitr::kable(
    escape = F, booktabs = T, format = "html",
    caption = str_c(
      "Marginal effects of the synthetic variables ",
      "on the probability of being in each regime for V2"
    ),
    col.names = c(
      "Variable", "\\(\\mathbb{P}(V_2 = 1)\\)<br>(Speed slower than the logistic decay)",
      "\\(\\mathbb{P}(V_2 = 2)\\)<br>(Speed faster than the logistic decay)"
    )
  ) |>
  kableExtra::add_header_above(c(" ", "Marginal effect on:" = 2)) |>
  kableExtra::kable_styling()
```

Table 5.4: Marginal effects of the synthetic variables on the probability of being in each regime for V2

	Marginal effect on:	
Variable	\(\mathbb{P}(V_2 = 1)\) (Speed slower than the logistic decay)	\(\mathbb{P}(V_2 = 2)\) (Speed faster than the logistic decay)
Healthcare infrastructure	\textcolor[HTML]{009D57}{0.5}	\textcolor[HTML]{EE324E}{-0.5}
	\textcolor{black}{(0.6)}	\textcolor{black}{(0.6)}
Vulnerability to comorbidites	\textcolor[HTML]{EE324E}{-0.13}	\textcolor[HTML]{009D57}{0.13}
	\textcolor{black}{(0.93)}	\textcolor{black}{(0.93)}
Vulnerability to natural environment	\textcolor[HTML]{EE324E}{-0.26}	\textcolor[HTML]{009D57}{0.26}
	\textcolor{black}{(0.29)}	\textcolor{black}{(0.29)}
Living conditions	\textcolor[HTML]{EE324E}{-0.18}	\textcolor[HTML]{009D57}{0.18}
	\textcolor{black}{(0.65)}	\textcolor{black}{(0.65)}
Economic and societal characteristics	\textcolor[HTML]{EE324E}{-0.37}	\textcolor[HTML]{009D57}{0.37}

Variable	Marginal effect on: \(\mathbb{P}(V_2 = 1)\) (Speed slower than the logistic decay)	\(\mathbb{P}(V_2 = 2)\) (Speed faster than the logistic decay)
Policy variables and governance	\textcolor{black}{(0.39)} \textcolor[HTML]{EE324E}{-0.03} \textcolor{black}{(0.19)}	\textcolor{black}{(0.39)} \textcolor[HTML]{009D57}{0.03} \textcolor{black}{(0.19)}

5.3.2 Without Sub-Saharan Africa

```
results_o_prob_v2_wo_ssa <- oglmx(
  formulaMEAN = formula_mean_v2_wo_ssa,
  formulaSD = formula_variance_v2_wo_ssa,
  data = df |> filter(!group %in% "Sub-Saharan Africa"),
  link = "probit",
  constantMEAN = FALSE,
  constantSD = FALSE,
  delta = 0,
  threshparam = NULL
)
```

5.3.2.1 Summary

All the models in a single object.

```
summary(results_o_prob_v2_wo_ssa)
```

```
Heteroskedastic Probit Regression
Log-Likelihood: -35.63926
No. Iterations: 13
McFadden's R2: 0.1685392
AIC: 91.27851
----- Mean Equation ------
    Estimate Std. error t value Pr(>|t|)
s_1 -2.77647    2.21493 -1.2535   0.2100
s_2 -0.94394    1.09380 -0.8630   0.3881
s_3  0.51990    0.34860  1.4914   0.1359
s_4 -1.05977    1.30355 -0.8130   0.4162
```

```
s_5  0.28903    0.51412  0.5622   0.5740
s_6 -0.75580    0.58748 -1.2865   0.1983
----- SD Equation ------
    Estimate Std. error t value Pr(>|t|)
s_2  -5.1107     7.3024 -0.6999  0.48401
s_3  -2.9968     1.3833 -2.1664  0.03028 *
s_4   3.7335     6.6236  0.5637  0.57298
----- Threshold Parameters -----
                Estimate Std. error t value Pr(>|t|)
Threshold (1->2) -0.88751    0.79300 -1.1192   0.2631
```

5.3.2.2 Marginal Effects

Again, we use the `margins.oglmx()` function to compute the marginal effects.

```r
marginal_effects_v2_wo_ssa <- margins.oglmx(
  results_o_prob_v2_wo_ssa,
  ascontinuous = TRUE
)
```

```r
me_v2_wo_ssa_table <-
  map(marginal_effects_v2_wo_ssa, as_tibble, rownames = "variable") |>
  list_rbind(names_to = "state") |>
  mutate(
    variable = factor(
      variable,
      labels = c(
        "s_1" = "Healthcare infrastructure",
        "s_2" = "Vulnerability to comorbidites",
        "s_3" = "Vulnerability to natural environment",
        "s_4" = "Living conditions",
        "s_5" = "Economic and societal characteristics",
        "s_6" = "Policy variables and governance"
      )
    ),
    star = case_when(
      `Pr(>|t|)` < 0.01 ~ "***",
      `Pr(>|t|)` < 0.05 ~ "**",
      `Pr(>|t|)` < 0.1 ~ "*",
      TRUE ~ ""
    )
```

```
  ) |>
  mutate(prob_state = str_c(round(`Marg. Eff`, 2), star)) |>
  mutate(std_error = str_c("(", round(`Std. error`, 2), ")")) |>
  select(variable, state, prob_state, std_error) |>
  pivot_longer(c(prob_state, std_error)) |>
  pivot_wider(names_from = state, values_from = value) |>
  select(-name) |>
  mutate(variable = as.character(variable)) |>
  group_by(variable) |>
  mutate(variable = ifelse(row_number() == 1, variable, ""))
```

Then, we can print the results.

```
me_v2_wo_ssa_table |>
  mutate(
    `1` = kableExtra::cell_spec(
      `1`,
      color = case_when(
        str_detect(`1`, "^\\(|-", negate = TRUE) ~ "#009D57",
        str_detect(`1`, "^-") ~ "#EE324E",
        TRUE ~ "black"
      )
    ),
    `2` = kableExtra::cell_spec(
      `2`,
      color = case_when(
        str_detect(`2`, "^\\(|-", negate = TRUE) ~ "#009D57",
        str_detect(`2`, "^-") ~ "#EE324E",
        TRUE ~ "black"
      )
    )
  ) |>
  ungroup() |>
  knitr::kable(
    escape = F, booktabs = T, format = "html",
    caption = str_c(
      "Marginal effects of the synthetic variables ",
      "on the probability of being in each regime for V2, ",
      "without Sub-Saharan Africa"
    ),
    col.names = c(
```

```
      "Variable", "\\(\\mathbb{P}(V_2 = 1)\\)<br>(Speed slower than the logistic decay)",
      "\\(\\mathbb{P}(V_2 = 2)\\)<br>(Speed faster than the logistic decay)"
    )
  ) |>
  kableExtra::add_header_above(c(" ", "Marginal effect on:" = 2)) |>
  kableExtra::kable_styling()
```

Table 5.5: Marginal effects of the synthetic variables on the probability of being in each regime for V2, without Sub-Saharan Africa

	Marginal effect on:	
Variable	\(\mathbb{P}(V_2 = 1)\) (Speed slower than the logistic decay)	\(\mathbb{P}(V_2 = 2)\) (Speed faster than the logistic decay)
Healthcare infrastructure	\textcolor[HTML]{009D57}{1.46***}	\textcolor[HTML]{EE324E}{-1.46***}
	\textcolor{black}{(0.45)}	\textcolor{black}{(0.45)}
Vulnerability to comorbidites	\textcolor[HTML]{009D57}{1.71}	\textcolor[HTML]{EE324E}{-1.71}
	\textcolor{black}{(2.26)}	\textcolor{black}{(2.26)}
Vulnerability to natural environment	\textcolor[HTML]{009D57}{0.44}	\textcolor[HTML]{EE324E}{-0.44}
	\textcolor{black}{(0.31)}	\textcolor{black}{(0.31)}
Living conditions	\textcolor[HTML]{EE324E}{-0.33}	\textcolor[HTML]{009D57}{0.33}
	\textcolor{black}{(1.89)}	\textcolor{black}{(1.89)}
Economic and societal characteristics	\textcolor[HTML]{EE324E}{-0.15}	\textcolor[HTML]{009D57}{0.15}
	\textcolor{black}{(0.22)}	\textcolor{black}{(0.22)}
Policy variables and governance	\textcolor[HTML]{009D57}{0.4**}	\textcolor[HTML]{EE324E}{-0.4**}
	\textcolor{black}{(0.19)}	\textcolor{black}{(0.19)}

5.3.3 Reproduction of Table 3

```
me_v2_both <-
  map(marginal_effects_v2, as_tibble, rownames = "variable") |>
  list_rbind(names_to = "state") |>
  mutate(model = "All") |>
```

```
  bind_rows(
    map(marginal_effects_v2_wo_ssa, as_tibble, rownames = "variable") |>
      list_rbind(names_to = "state") |>
      mutate(model = "Without SSA")
  ) |>
  mutate(
    variable = factor(
      variable,
      labels = c(
        "s_1" = "Healthcare infrastructure",
        "s_2" = "Vulnerability to comorbidites",
        "s_3" = "Vulnerability to natural environment",
        "s_4" = "Living conditions",
        "s_5" = "Economic and societal characteristics",
        "s_6" = "Policy variables and governance"
      )
    ),
    star = case_when(
      `Pr(>|t|)` < 0.01 ~ "***",
      `Pr(>|t|)` < 0.05 ~ "**",
      `Pr(>|t|)` < 0.1 ~ "*",
      TRUE ~ ""
    )
  ) |>
  mutate(prob_state = str_c(round(`Marg. Eff`, 2), star)) |>
  mutate(std_error = str_c("(", round(`Std. error`, 2), ")")) |>
  select(model, variable, state, prob_state, std_error) |>
  pivot_longer(c(prob_state, std_error)) |>
  pivot_wider(names_from = c(model, state), values_from = value) |>
  select(-name) |>
  mutate(variable = as.character(variable)) |>
  group_by(variable) |>
  mutate(variable = ifelse(row_number() == 1, variable, ""))
```

```
me_v2_both |>
  mutate(
    across(
      All_1:`Without SSA_2`,
      ~kableExtra::cell_spec(
        .x,
        color = case_when(
```

```
          str_detect(.x, "^\\(|-", negate = TRUE) ~ "#009D57",
          str_detect(.x, "^-") ~ "#EE324E",
          TRUE ~ "black"
        )
      )
    )
  ) |>
  ungroup() |>
  knitr::kable(
    escape = FALSE, booktabs = T, format = "html",
    caption = str_c(
      "Marginal effects of the synthetic variables ",
      "on the probability of being in each regime for V2"
    ),
    col.names = c(
      "Variable",
      rep(
        c(
          "\\(\\mathbb{P}(V_2 = 1)\\)<br>(Slower)",
          "\\(\\mathbb{P}(V_2 = 2)\\)<br>(Faster)"
        ),
        2
      )
    )
  ) |>
  kableExtra::add_header_above(
    c(" ", "All countries" = 2, "Without Sub-Saharan Africa" = 2)
  ) |>
  kableExtra::kable_styling()
```

Table 5.6: Marginal effects of the synthetic variables on the probability of being in each regime for V2

	All countries		Without Sub-Saharan Africa	
Variable	\(\mathbb{P}(V_2 = 1)\) (Slower)	\(\mathbb{P}(V_2 = 2)\) (Faster)	\(\mathbb{P}(V_2 = 1)\) (Slower)	\(\mathbb{P}(V_2 = 2)\) (Faster)
Healthcare infrastructure	\textcolor[HTML]{009D57}{0.5}	\textcolor[HTML]{EE324E}{-0.5}	\textcolor[HTML]{009D57}{1.46***}	\textcolor[HTML]{EE324E}{-1.46***}
	\textcolor{black}{(0.6)}	\textcolor{black}{(0.6)}	\textcolor{black}{(0.45)}	\textcolor{black}{(0.45)}

	All countries		Without Sub-Saharan Africa	
Variable	\(\mathbb{P}(V_2 = 1)\) (Slower)	\(\mathbb{P}(V_2 = 2)\) (Faster)	\(\mathbb{P}(V_2 = 1)\) (Slower)	\(\mathbb{P}(V_2 = 2)\) (Faster)
Vulnerability to comorbidites	\textcolor[HTML]{EE324E}{-0.13}	\textcolor[HTML]{009D57}{0.13}	\textcolor[HTML]{009D57}{0.13}	\textcolor[HTML]{EE324E}{-1.71}
	\textcolor{black}{(0.93)}	\textcolor{black}{(0.93)}	\textcolor{black}{(2.26)}	\textcolor{black}{(2.26)}
Vulnerability to natural environment	\textcolor[HTML]{EE324E}{-0.26}	\textcolor[HTML]{009D57}{0.26}	\textcolor[HTML]{009D57}{0.44}	\textcolor[HTML]{EE324E}{-0.44}
	\textcolor{black}{(0.29)}	\textcolor{black}{(0.29)}	\textcolor{black}{(0.31)}	\textcolor{black}{(0.31)}
Living conditions	\textcolor[HTML]{EE324E}{-0.18}	\textcolor[HTML]{009D57}{0.18}	\textcolor[HTML]{009D57}{0.33}	\textcolor[HTML]{EE324E}{-0.33}
	\textcolor{black}{(0.65)}	\textcolor{black}{(0.65)}	\textcolor{black}{(1.89)}	\textcolor{black}{(1.89)}
Economic and societal characteristics	\textcolor[HTML]{EE324E}{-0.37}	\textcolor[HTML]{009D57}{0.37}	\textcolor[HTML]{009D57}{0.15}	\textcolor[HTML]{EE324E}{-0.15}
	\textcolor{black}{(0.39)}	\textcolor{black}{(0.39)}	\textcolor{black}{(0.22)}	\textcolor{black}{(0.22)}
Policy variables and governance	\textcolor[HTML]{EE324E}{-0.03}	\textcolor[HTML]{009D57}{0.03}	\textcolor[HTML]{009D57}{0.4**}	\textcolor[HTML]{EE324E}{-0.4**}
	\textcolor{black}{(0.19)}	\textcolor{black}{(0.19)}	\textcolor{black}{(0.19)}	\textcolor{black}{(0.19)}

References

Hale, Thomas, Noam Angrist, Rafael Goldszmidt, Beatriz Kira, Anna Petherick, Toby Phillips, Samuel Webster, et al. 2021. "A Global Panel Database of Pandemic Policies (Oxford COVID-19 Government Response Tracker)." *Nature Human Behaviour*. https://doi.org/10.1038/s41562-021-01079-8.

Medina, Leandro, and Friedrich Schneider. 2019. "Shedding Light on the Shadow Economy: A Global Database and the Interaction with the Official One."