Fork me on GitHub

Collect and use open access World Bank data to know your country

Bruno Thiago Tomio

R in Grenoble - Working Sessions

26 November 2020

Université Grenoble Alpes

Centre de Recherche en Economie de Grenoble

FURB
UNIVERSIDADE REGIONAL DE BLUMENAU

General view

World Bank data in R

The `WDI` package allows us to search and download data from the World Bank (several datasets, with thousands of variables)

World Bank data in R

The WDI package allows us to search and download data from the World Bank (several datasets, with thousands of variables)

Advantages

- Reproducible research

World Bank data in R

The `WDI` package allows us to search and download data from the World Bank (several datasets, with thousands of variables)

Advantages

- Reproducible research
- Annual, quarterly, and monthly data in long format

World Bank data in R

The `WDI` package allows us to search and download data from the World Bank (several datasets, with thousands of variables)

Advantages

- Reproducible research
- Annual, quarterly, and monthly data in long format
- Collect, plot, analyse, and estimate econometric models in the same free software

World Bank data in

The `WDI` package allows us to search and download data from the World Bank (several datasets, with thousands of variables)

Advantages

- Reproducible research
- Annual, quarterly, and monthly data in long format
- Collect, plot, analyse, and estimate econometric models in the same free software

Important links

- Package website, developed by Vincent Arel-Bundock https://vincentarelbundock.github.io/WDI/index.html

World Bank data in R

The `WDI` package allows us to search and download data from the World Bank (several datasets, with thousands of variables)

Advantages

- Reproducible research
- Annual, quarterly, and monthly data in long format
- Collect, plot, analyse, and estimate econometric models in the same free software

Important links

- Package website, developed by Vincent Arel-Bundock https://vincentarelbundock.github.io/WDI/index.html
- DataBank, to check variables details https://databank.worldbank.org/

World Bank data in R

The `WDI` package allows us to search and download data from the World Bank (several datasets, with thousands of variables)

Advantages

- Reproducible research
- Annual, quarterly, and monthly data in long format
- Collect, plot, analyse, and estimate econometric models in the same free software

Important links

- Package website, developed by Vincent Arel-Bundock https://vincentarelbundock.github.io/WDI/index.html
- DataBank, to check variables details https://databank.worldbank.org/
- `wbstats` (alternative package) http://nset-ornl.github.io/wbstats/index.html

Hands-on

Based on the package website
https://vincentarelbundock.github.io/WDI/index.html

Installation

```
install.packages('WDI')
```

Installation

```
install.packages('WDI')
```

Searching for data

You need to use **keywords** to search for data with the command `WDIsearch`

```
# Search for 'GDP'
WDIsearch('GDP')
```

Installation

```r
install.packages('WDI')
```

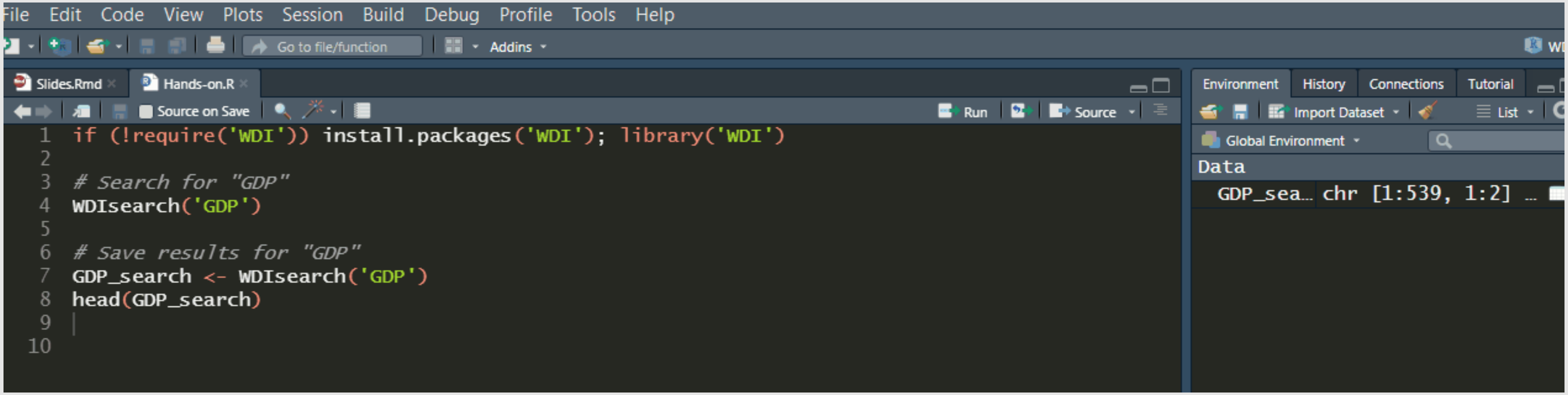
Searching for data

You need to use **keywords** to search for data with the command `WDIsearch`

```r
# Search for 'GDP'
WDIsearch('GDP')
```

You can create an object in the global environment to easily navigate all the variables with the keyword

```r
# Save results for 'GDP'
GDP_search <- WDIsearch('GDP')
```



```r
if (!require('WDI')) install.packages('WDI'); library('WDI')

# Search for "GDP"
WDIsearch('GDP')

# Save results for "GDP"
GDP_search <- WDIsearch('GDP')
head(GDP_search)
```

```
GDP_sea… chr [1:539, 1:2] …
```

Collect data

GDP per capita for France (FR) and Brazil (BR)

```
# indicator = NY.GDP.PCAP.KD / name = GDP per capita (constant 2010 US$)
indicator <- c("GDP per capita" = 'NY.GDP.PCAP.KD')
dat1 <- WDI(indicator, country=c('FR', 'BR'), end = 2019)
head(dat1)
```

```
##   iso2c country GDP per capita year
## 1    BR  Brazil       11121.74 2019
## 2    BR  Brazil       11079.71 2018
## 3    BR  Brazil       11021.72 2017
## 4    BR  Brazil       10965.97 2016
## 5    BR  Brazil       11431.15 2015
## 6    BR  Brazil       11951.21 2014
```

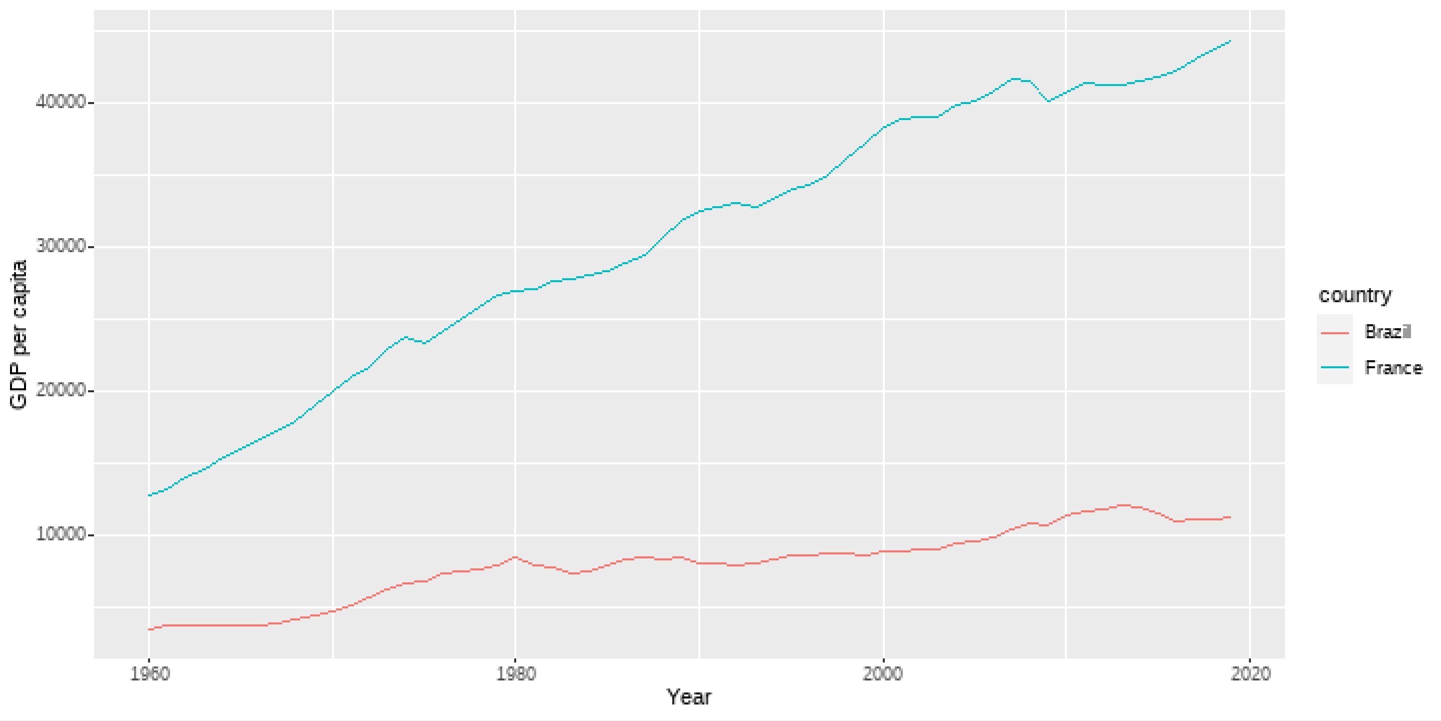
GDP per capita (US$ and local currency unity) for France (FR) and Brazil (BR)

```
# indicators = NY.GDP.PCAP.KD and NY.GDP.PCAP.KN / names = GDP per capita (constant 2010 US$
indicators <- c("GDP per capita (US$)" = 'NY.GDP.PCAP.KD', "GDP per capita (LCU)" = "NY.GDP.
dat2 <- WDI(indicators, country=c('FR', 'BR'), end = 2019)
head(dat2)
```

```
##   iso2c country year GDP per capita (US$) GDP per capita (LCU)
## 1    BR  Brazil 1960             3417.352             6011.806
## 2    BR  Brazil 1961             3660.391             6439.361
## 3    BR  Brazil 1962             3740.433             6580.170
## 4    BR  Brazil 1963             3664.978             6447.429
## 5    BR  Brazil 1964             3685.493             6483.519
## 6    BR  Brazil 1965             3692.846             6496.454
```

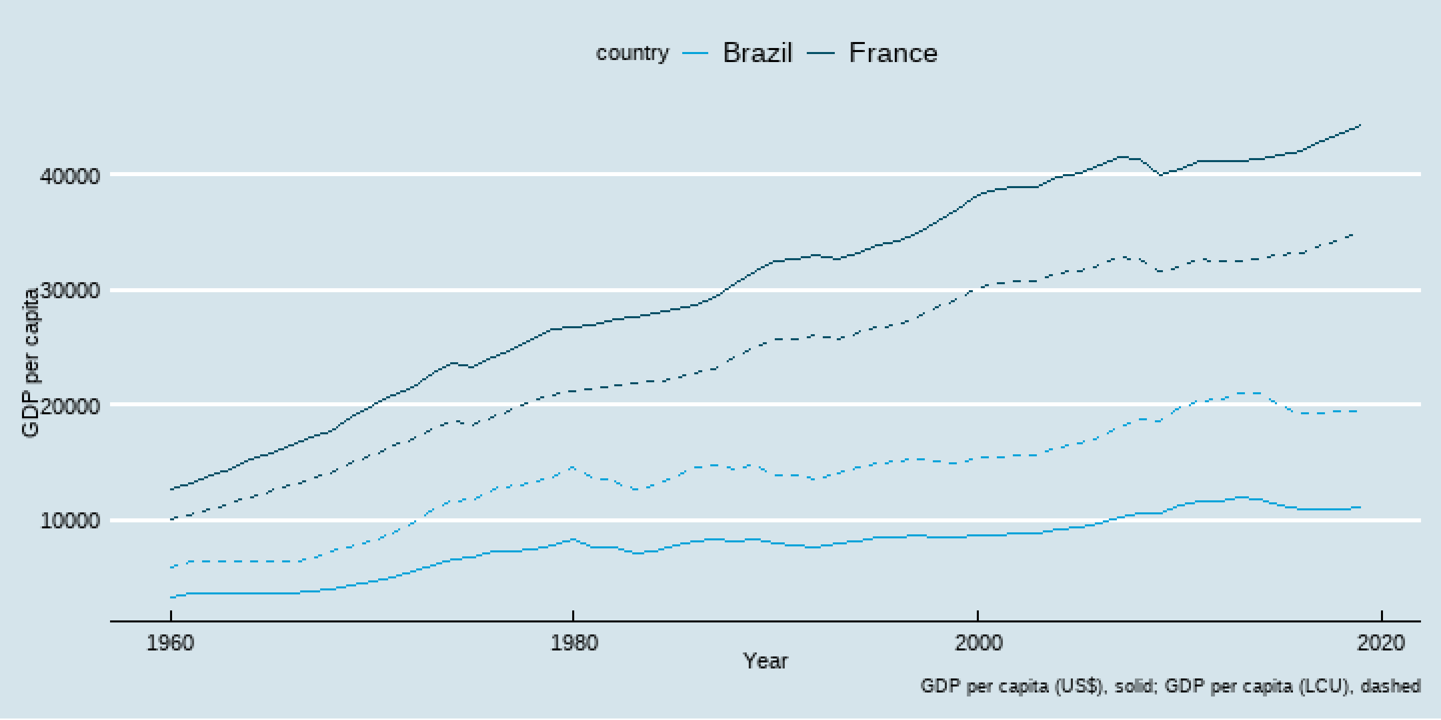
Plot data (1/2)

```r
library(ggplot2)
# GDP per capita for France and Brazil
ggplot(dat1, aes(year, `GDP per capita`, color=country)) + geom_line() +
    xlab('Year') + ylab('GDP per capita')
```



Plot data (2/2)

```
# GDP per capita (US$ and local currency unity) for France and Brazil
library(ggthemes)
ggplot(dat2, aes(year, color=country)) +
  geom_line(aes(year, `GDP per capita (US$)`)) +
  geom_line(aes(year, `GDP per capita (LCU)`), linetype = "dashed") +
  xlab('Year') + ylab('GDP per capita') +
  labs(caption = "GDP per capita (US$), solid; GDP per capita (LCU), dashed") +
  theme_economist() +
  scale_colour_economist()
```



More details on the indicators

```r
library(dplyr)
Data_info <- WDI_data
Data_series <- as.data.frame(Data_info$series) %>%
  filter(indicator == "NY.GDP.PCAP.KD")
colnames(Data_series)
```

```
## [1] "indicator"          "name"               "description"
## [4] "sourceDatabase"     "sourceOrganization"
```

```r
Data_series$description
```

[1] "GDP per capita is gross domestic product divided by midyear population. GDP is the sum of gross value added by all resident producers in the economy plus any product taxes and minus any subsidies not included in the value of the products. It is calculated without making deductions for depreciation of fabricated assets or for depletion and degradation of natural resources. Data are in constant 2010 U.S. dollars."

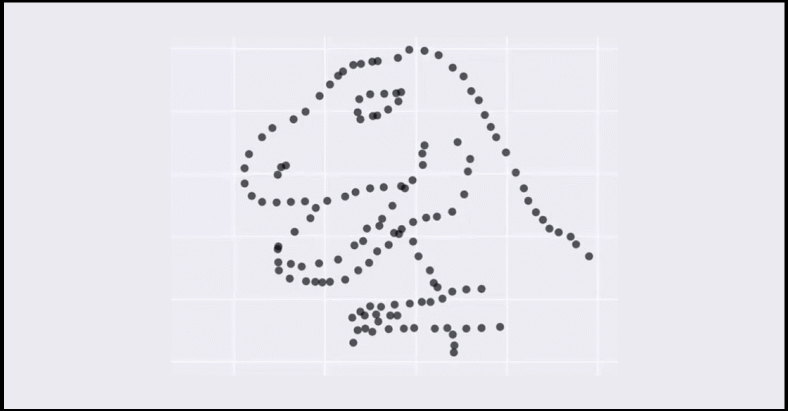
More details on the geographical selection

```r
library(knitr)
library(kableExtra)
Data_countries <- as.data.frame(Data_info$country)
Data_countries %>%
  kable("html") %>%
  kable_styling(font_size = 11) %>%
  scroll_box(width = "100%", height = "60%")
```

iso3c	iso2c	country	region	capital	longitude	latitude	income	lending
ABW	AW	Aruba	Latin America & Caribbean	Oranjestad	-70.0167	12.5167	High income	Not classified
AFG	AF	Afghanistan	South Asia	Kabul	69.1761	34.5228	Low income	IDA
AFR	A9	Africa	Aggregates				Aggregates	Aggregates
AGO	AO	Angola	Sub-Saharan Africa	Luanda	13.242	-8.81155	Lower middle income	IBRD
ALB	AL	Albania	Europe & Central Asia	Tirane	19.8172	41.3317	Upper middle income	IBRD
AND	AD	Andorra	Europe & Central Asia	Andorra la Vella	1.5218	42.5075	High income	Not classified
ANR	L5	Andean Region	Aggregates				Aggregates	Aggregates
ARB	1A	Arab World	Aggregates				Aggregates	Aggregates
ARE	AE	United Arab Emirates	Middle East & North Africa	Abu Dhabi	54.3705	24.4764	High income	Not classified
ARG	AR	Argentina	Latin America & Caribbean	Buenos Aires	-58.4173	-34.6118	Upper middle	IBRD

Some cool graphs

Based on the `webstats` website
http://nset-ornl.github.io/wbstats/index.html



Source

https://damassets.autodesk.net/content/dam/autodesk/www/autodesk-reasearch/Publications/images/same-stats-different-graphs-image-1920x1000.gif

World maps (1/2) 🌎

Code Map 1 Plot 1 Code Map 2 Plot 2 Code Map 3 Plot 3

```r
library(rnaturalearth)
library(tidyverse)

# Self-employed in 2019
indicator <- c("Self-employed" = 'SL.EMP.SELF.ZS')
datW <- WDI(indicator, country="all", start = 2019, end = 2019)

Data_info <- WDI_data

name_self_employed <- as.data.frame(Data_info$series) %>%
  filter(indicator == "SL.EMP.SELF.ZS") %>%
  select(name)

source_self_employed <- as.data.frame(Data_info$series) %>%
  filter(indicator == "SL.EMP.SELF.ZS") %>%
  select(sourceOrganization)

ne_countries(returnclass = "sf") %>%
  left_join(datW, c("iso_a2" = "iso2c")) %>%
  filter(iso_a2 != "ATA") %>% # remove Antarctica
  ggplot(aes(fill = `Self-employed`)) +
  geom_sf() +
  scale_fill_viridis_c(labels = scales::percent_format(scale = 1)) +
  theme(legend.position="bottom") +
  labs(
    title = name_self_employed,
    fill = NULL,
    caption = paste("Source:", source_self_employed)
  )
```

World maps (2/2) 🌍

Code Map 4 | Plot 4 | Code Map 5 | Plot 5 | Code Map 6 | Plot 6

```r
# GDP per capita (constant 2010 US$) in 2019
indicator <- c("GDP per capita" = 'NY.GDP.PCAP.KD')
datW <- WDI(indicator, country="all", start = 2019, end = 2019)

Data_info <- WDI_data

name_GDP_PC <- as.data.frame(Data_info$series) %>%
  filter(indicator == "NY.GDP.PCAP.KD") %>%
  select(name)

source_GDP_PC <- as.data.frame(Data_info$series) %>%
  filter(indicator == "NY.GDP.PCAP.KD") %>%
  select(sourceOrganization)

ne_countries(returnclass = "sf") %>%
  left_join(datW, c("iso_a2" = "iso2c")) %>%
  filter(iso_a2 != "ATA") %>% # remove Antarctica
  ggplot(aes(fill = `GDP per capita`)) +
  geom_sf() +
  scale_fill_viridis_c(labels = scales::dollar_format(scale = 1)) +
  theme(legend.position="bottom", legend.key.width = unit(2.5, "cm")) +
  labs(
    title = name_GDP_PC,
    fill = NULL,
    caption = paste("Source:", source_GDP_PC)
  )
```

Hans Rosling's Gapminder 🌐

Code HRG 1 Plot 1 Code HRG 2 Plot 2 Code Map 3 Plot 3

```r
indicators <- c(life_exp = "SP.DYN.LE00.IN",
                gdp_capita ="NY.GDP.PCAP.CD",
                pop = "SP.POP.TOTL")

hrg <- WDI(indicators, country="all", start = "2018", end = "2018")
Data_info <- WDI_data
Data_countries <- as.data.frame(Data_info$country)

hrg %>%
  left_join(Data_countries, "iso2c") %>%
  filter(region != "Aggregates") %>% # remove aggregates (groups of countries)
  ggplot() +
  geom_point(aes(x = gdp_capita, y = life_exp, size = pop, color = region)) +
  scale_x_continuous(
    labels = scales::dollar_format(),
    breaks = scales::log_breaks(n = 10)) +
  coord_trans(x = 'log10') +
  scale_size_continuous(
    labels = scales::number_format(scale = 1/1e6, suffix = "m"),
    breaks = seq(1e8,1e9, 2e8),
    range = c(1,20)) +
  theme_minimal() +
  labs(title = "An Example of Hans Rosling's Gapminder using WDI (Data for 2018)",
    x = "GDP per capita (log scale)",
    y = "Life expectancy at birth",
    size = "Population",
    color = NULL,
    caption = "Source: World Bank")
```

Thank you!

`bttomio@furb.br`

@bttomio

Slides created with `xaringan` and `xaringanthemer`.

UGA Université Grenoble Alpes

CREg Centre de Recherche en Economie de Grenoble

FURB UNIVERSIDADE REGIONAL DE BLUMENAU