Lista 7 - Introdução a Análise de Dados
Análise de Dados
Gabarito

Guilherme Masuko

May 2023

Para essa lista vamos analisar o PIB per capita de alguns países. Utilizaremos os dados do banco mundial para isso. Essa base de dados está vinculada ao R através do pacote `WDI`[1].

Para acessar os dados, precisamos instalar e chamar o pacote. Os dados que queremos estão armazenados pelo `indicador = 'NY.GDP.PCAP.KD'`. O parâmetro `country` recebe as siglas dos países que estamos interessados em analisar o PIB per capita. `extra = TRUE` adiciona algumas colunas nesse dataframe. `start` e `end` referenciam o intervalo temporal dos dados. A seguir o script.

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

help(WDI)

data = WDI(indicator='NY.GDP.PCAP.KD',
        country=c('CA','US','BR','AR','CL','CO','PY','PE',
               'UY','FR','DE','IT','ES','GB','DK','1W'),
        extra = TRUE,
        start=1960,
        end=2021)
View(data)
```

Questão 1

Para as três regiões (`region`) às quais temos dados nessa amostra, calcule as estatísticas, quantidade de países, PIB per capita médio, máximo e mínimo, referentes

[1] <https://www.r-project.org/nosvn/pandoc/WDI.html>

ao ano de 2021.

Solução

```
library(tidyverse)

# dados das regiões
regioes <- data %>%
  distinct(region)
regioes

# Estatísticas para região da América Latina
data %>%
  filter(region == regioes[1,1]) %>%
  filter(year == 2021) %>%
  summarise(quantidade = n(),
            media = mean(NY.GDP.PCAP.KD),
            maximo = max(NY.GDP.PCAP.KD),
            minimo = min(NY.GDP.PCAP.KD))

# Estatísticas para região da América do Norte
data %>%
  filter(region == regioes[2,1]) %>%
  filter(year == 2021) %>%
  summarise(quantidade = n(),
            media = mean(NY.GDP.PCAP.KD),
            maximo = max(NY.GDP.PCAP.KD),
            minimo = min(NY.GDP.PCAP.KD))

# Estatísticas para região da Europa
data %>%
  filter(region == regioes[3,1]) %>%
  filter(year == 2021) %>%
  summarise(quantidade = n(),
            media = mean(NY.GDP.PCAP.KD),
            maximo = max(NY.GDP.PCAP.KD),
            minimo = min(NY.GDP.PCAP.KD))

# Ou poderíamos usar o group by para obter as estatísticas para
   cada região diretamente
data %>%
```

```
  group_by(region) %>%
  filter(year == 2021) %>%
  summarise(quantidade = n(),
            media = mean(NY.GDP.PCAP.KD),
            maximo = max(NY.GDP.PCAP.KD),
            minimo = min(NY.GDP.PCAP.KD))
```

Questão 2

Compute as estatísticas média, mínimo e máximo, para cada país, durante os períodos:

a) Todo o período da amostra.

Solução

```
data %>%
  group_by(country) %>%
  summarise(media = mean(NY.GDP.PCAP.KD, na.rm = TRUE),
            maximo = max(NY.GDP.PCAP.KD, na.rm = TRUE),
            minimo = min(NY.GDP.PCAP.KD, na.rm = TRUE))
```

b) 1960-1980.

Solução

```
data %>%
  group_by(country) %>%
  filter(year %in% 1960:1980) %>%
  summarise(media = mean(NY.GDP.PCAP.KD, na.rm = TRUE),
            maximo = max(NY.GDP.PCAP.KD, na.rm = TRUE),
            minimo = min(NY.GDP.PCAP.KD, na.rm = TRUE))
```

c) 1980-2000.

Solução

```
data %>%
  group_by(country) %>%
  filter(year %in% 1980:2000) %>%
  summarise(media = mean(NY.GDP.PCAP.KD, na.rm = TRUE),
            maximo = max(NY.GDP.PCAP.KD, na.rm = TRUE),
            minimo = min(NY.GDP.PCAP.KD, na.rm = TRUE))
```

d) 2000-2021.

Solução

```
data %>%
  group_by(country) %>%
  filter(year %in% 2000:2021) %>%
  summarise(media = mean(NY.GDP.PCAP.KD, na.rm = TRUE),
            maximo = max(NY.GDP.PCAP.KD, na.rm = TRUE),
            minimo = min(NY.GDP.PCAP.KD, na.rm = TRUE))
```

Questão 3

Crie um `data.frame` para cada país (cada um com o nome do país, tudo em lower case), contendo apenas as colunas `country`, `year`, `NY.GDP.PCAP.KD`.

Solução

```
# países da amostra
paises <- data %>%
  distinct(country)
paises

# argentina
argentina <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[1,1])

# brazil
brazil <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[2,1])

# canada
canada <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[3,1])

# chile
chile <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[4,1])
```

```r
# colombia
colombia <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[5,1])

# denmark
denmark <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[6,1])

# france
france <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[7,1])

# germany
germany <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[8,1])

# italy
italy <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[9,1])

# paraguay
paraguay <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[10,1])

# peru
peru <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[11,1])

# spain
spain <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[12,1])
```

```r
# united kingdom
united_kingdom <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[13,1])

# united states
united_states <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[14,1])

# uruguay
uruguay <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[15,1])

# world
world <- data %>%
  select(country, year, NY.GDP.PCAP.KD) %>%
  filter(country == paises[16,1])
```

Questão 4

Crie uma função que recebe um dataframe (no padrão da questão 3) como parâmetro. Essa função deve fazer as seguintes manipulações nesse dataframe:

- Renomear a coluna `NY.GDP.PCAP.KD` para o nome do país do respectivo dataframe.
- Manter somente as colunas `year` e a (agora) do nome do país.

Use a função para alterar todos os dataframes dos países.

Solução

```r
# função
altera_df <- function(df) {
  # renomeando a coluna
  colnames(df)[3] <- df$country[1]

  # dropando a coluna country
  df$country <- NULL

  return(df)
}
```

```r
# nomes dos dataframes
df_names <- paises %>%
  mutate(country = tolower(country)) %>%
  mutate(country = str_replace(country, " ", "_"))

df_names

# alterando os dataframes
argentina <- altera_df(argentina)
brazil <- altera_df(brazil)
canada <- altera_df(canada)
chile <- altera_df(chile)
colombia <- altera_df(colombia)
denmark <- altera_df(denmark)
france <- altera_df(france)
germany <- altera_df(germany)
italy <- altera_df(italy)
paraguay <- altera_df(paraguay)
peru <- altera_df(peru)
spain <- altera_df(spain)
united_kingdom <- altera_df(united_kingdom)
united_states <- altera_df(united_states)
uruguay <- altera_df(uruguay)
world <- altera_df(world)
```

Questão 5

Crie um novo dataframe chamado `canada_` a partir do drop das linhas onde o PIB per capita do Canadá é `NA`. Crie um novo dataframe chamado `brazil_` a partir do drop das linhas onde o PIB per capita do Brasil é maior que 8000.

Solução

```r
# dropando os NA's de canada
canada_ <- canada %>%
  drop_na()

# mantendo somente os dados onde o PIBpc é menor que 8000
brazil_ <- brazil %>%
  filter(Brazil <= 8000)
```

Faça a união desses dois dataframes, `canada_` e `brazil_`, das seguintes maneiras:

a) Contendo somente as linhas onde os dois dataframes contenham dados.

Solução

```
# inner join
inner_join(canada_, brazil_, by = c("year" = "year"))
```

b) Contendo todas as linhas onde o dataframe `canada_` contenha dados.

Solução

```
# left join
left_join(canada_, brazil_, by = c("year" = "year"))
```

c) Contendo todas as linhas onde o dataframe `brazil_` contenha dados.

Solução

```
# right join
right_join(canada_, brazil_, by = c("year" = "year"))
```

d) Contendo todas as linhas onde pelo menos um dos dois dataframes contenham dados.

Solução

```
# full join
full_join(canada_, brazil_, by = c("year" = "year"))
```

Questão 6

Una todos dataframes. Renomeie as colunas dos países com nomes compostos, alterando o espaço entre os nomes por um underline "_".

Solução

```
# Una todos dataframes
df_names$country

# primeira maneira (mais direta)
df <- list(argentina,brazil,canada,chile,colombia,denmark,
        france,germany,italy,paraguay,peru,spain,united_kingdom,
        united_states,uruguay,world) %>%
```

```
  reduce(full_join, by='year')

# segunda maneira
df <- full_join(argentina, brazil, by='year')
df <- full_join(df, canada, by='year')
df <- full_join(df, chile, by='year')
df <- full_join(df, colombia, by='year')
df <- full_join(df, denmark, by='year')
df <- full_join(df, france, by='year')
df <- full_join(df, germany, by='year')
df <- full_join(df, italy, by='year')
df <- full_join(df, paraguay, by='year')
df <- full_join(df, peru, by='year')
df <- full_join(df, spain, by='year')
df <- full_join(df, united_kingdom, by='year')
df <- full_join(df, united_states, by='year')
df <- full_join(df, uruguay, by='year')
df <- full_join(df, world, by='year')

# renomeando as colunas
colnames(df)[14:15] <- c("United_Kingdom", "United_States" )
```

Faça um gráfico apresentando a série temporal do PIB per capita para cada país, um para cada região.

a) América Latina.

Solução

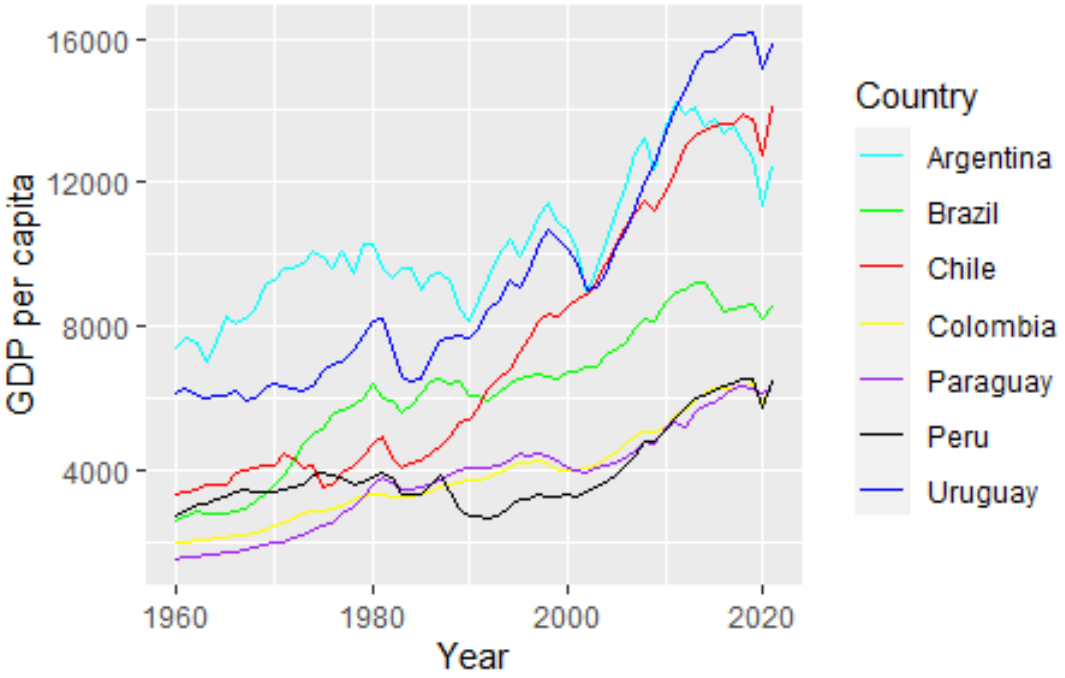
```
# Países por região
pais_regiao <- data %>%
  group_by(region) %>%
  distinct(country)

# América Latina
pais_regiao %>%
  filter(region == regioes[1,1])

# plot
ggplot(df, aes(year)) +
  geom_line(aes(y = Argentina, color = "Argentina")) +
```

```
  geom_line(aes(y = Brazil, color = "Brazil")) +
  geom_line(aes(y = Chile, color = "Chile")) +
  geom_line(aes(y = Colombia, color = "Colombia")) +
  geom_line(aes(y = Paraguay, color = "Paraguay")) +
  geom_line(aes(y = Peru, color = "Peru")) +
  geom_line(aes(y = Uruguay, color = "Uruguay")) +
  labs(y = "GDP per capita", x = "Year", color = "Country") +
  scale_color_manual(values = c("Argentina" = "cyan",
                                "Brazil" = "green",
                                "Chile" = "red",
                                "Colombia" = "yellow",
                                "Paraguay" = "purple",
                                "Peru" = "black",
                                "Uruguay" = "blue"))
```

Figure 1: América Latina



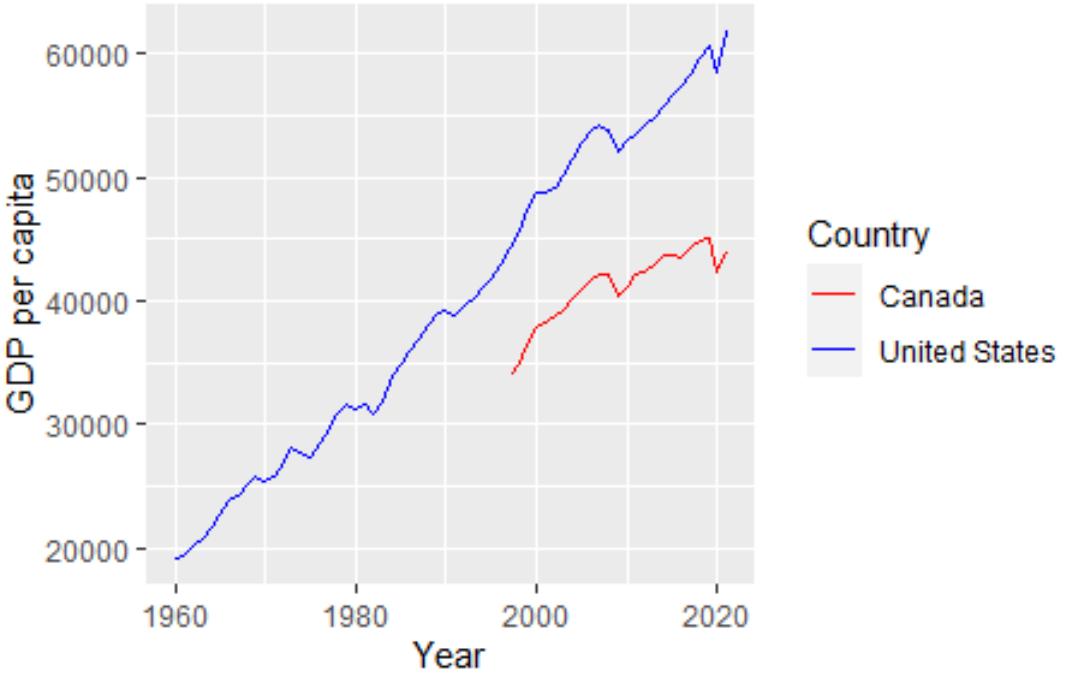
b) América do Norte.

Solução

```
# América do Norte
pais_regiao %>%
  filter(region == regioes[2,1])
```

```
# plot
ggplot(df, aes(year)) +
  geom_line(aes(y = Canada, color = "Canada")) +
  geom_line(aes(y = United_States, color = "United States")) +
  labs(y = "GDP per capita", x = "Year", color = "Country") +
  scale_color_manual(values=c("Canada"='red',
                              'United States'='blue'))
```

Figure 2: América do Norte



c) Europa.

Solução

```
# Europa
pais_regiao %>%
  filter(region == regioes[3,1])

# plot
ggplot(df, aes(year)) +
  geom_line(aes(y = Denmark, color = "Denmark")) +
  geom_line(aes(y = France, color = "France")) +
  geom_line(aes(y = Germany, color = "Germany")) +
```

```
geom_line(aes(y = Italy, color = "Italy")) +
geom_line(aes(y = Spain, color = "Spain")) +
geom_line(aes(y = United_Kingdom, color = "United Kingdom")) +
labs(y = "GDP per capita", x = "Year", color = "Country") +
scale_color_manual(values = c("Denmark" = "purple",
                              "France" = "blue",
                              "Germany" = "black",
                              "Italy" = "green",
                              "Spain" = "yellow",
                              "United Kingdom" = "cyan"))
```

Figure 3: Europa