Lista 8 - Introdução a Análise de Dados
Análise de Dados
Gabarito

Guilherme Masuko

May 2023

Para essa lista vamos analisar a população (total e urbana) 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 = c("total_pop"="SP.POP.TOTL", "urban_pop"="SP.URB.TOTL")`. O parâmetro `country` recebe as siglas dos países que estamos interessados em analisar o PIB per capita. `start` e `end` referenciam o intervalo temporal dos dados. A seguir o script.

```
library(tidyverse)
library(WDI)

wdi = WDI(indicator = c("total_pop"="SP.POP.TOTL",
   "urban_pop"="SP.URB.TOTL"),
        country = c("CA","US","BR","AR","CL","CO","PY","PE",
                    "UY","FR","DE","IT","ES","GB","DK","1W"),
        start=1960,
        end=2020)
View(wdi)
```

Questão 1

Crie as seguintes colunas:

a) Taxa de população urbana.

Solução

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

```r
# coluna urban_rate
wdi <- wdi %>%
  mutate(urban_rate = urban_pop/total_pop)
```

b) Taxa de crescimento da população (total e urbana).

Solução

```r
# coluna total_pop_growth e urban_pop_growth
wdi <- wdi %>%
  group_by(country) %>%
  mutate(total_pop_growth = ( total_pop - lag(total_pop) )/
     lag(total_pop)) %>%
  mutate(urban_pop_growth = ( urban_pop - lag(urban_pop) )/
     lag(urban_pop))
```

Questão 2

Crie uma coluna contendo a região (continente) de cada país (crie utilizando código, sem utilizar o parâmetro `extra = TRUE`).

Solução

```r
# primeira maneira, usando ifelse
wdi <- wdi %>%
  mutate(region = ifelse(country %in% c("Argentina", "Brazil",
     "Chile", "Colombia", "Paraguay", "Peru", "Uruguay"), "South
     America", ifelse(country %in% c("Canada" ,"United States"),
     "North America", "Europe")))

# segunda maneira, usando o case_when
wdi <- wdi %>%
  mutate(region = case_when(
    country %in% c("Argentina", "Brazil", "Chile", "Colombia",
       "Paraguay", "Peru", "Uruguay") ~ "South America",
    country %in% c("Canada" ,"United States") ~ "North America",
    country %in% c("Denmark", "France", "Germany", "Italy", "Spain",
       "United Kingdom") ~ "Europe"
  ))
```

Questão 3

Calcule as estatísticas média, mínimo e máximo para as seguintes variáveis.

a) Taxa de população urbana agrupados por região para o ano de 2020.

Solução

```
# taxa de população urbana agrupados por região para o ano de
   2020
wdi %>%
  group_by(region) %>%
  filter(year == 2020) %>%
  summarise(media = mean(urban_rate),
            maximo = max(urban_rate),
            minimo = min(urban_rate))
```

b) Taxa de crescimento da população total agrupados por região para o ano de 2010.

Solução

```
# taxa de crescimento da população total agrupados por região
   para o ano de 2010
wdi %>%
  group_by(region) %>%
  filter(year == 2010) %>%
  summarise(media = mean(total_pop_growth),
            maximo = max(total_pop_growth),
            minimo = min(total_pop_growth))
```

c) Taxa de crescimento da população urbana agrupados por região para o ano de 2016.

Solução

```
# taxa de crescimento da população urbana agrupados por região
   para o ano de 2016
wdi %>%
  group_by(region) %>%
  filter(year == 2016) %>%
  summarise(media = mean(urban_pop_growth),
            maximo = max(urban_pop_growth),
            minimo = min(urban_pop_growth))
```

Questão 4

Calcule a média, mínimo e máximo da taxa de população urbana, taxa de crescimento da população total e taxa de crescimento da população urbana, para cada país durante todo o período que temos na amostra.

Solução

```
# taxa de população urbana
wdi %>%
  group_by(country) %>%
  summarise(media = mean(urban_rate),
            maximo = max(urban_rate),
            minimo = min(urban_rate))

# taxa de crescimento da população total
wdi %>%
  group_by(country) %>%
  summarise(media = mean(total_pop_growth, na.rm = TRUE),
            maximo = max(total_pop_growth, na.rm = TRUE),
            minimo = min(total_pop_growth, na.rm = TRUE))

# taxa de crescimento da população urbana
wdi %>%
  group_by(country) %>%
  summarise(media = mean(urban_pop_growth, na.rm = TRUE),
            maximo = max(urban_pop_growth, na.rm = TRUE),
            minimo = min(urban_pop_growth, na.rm = TRUE))
```

Questão 5

Calcule as médias de cada uma das variáveis abaixo agrupados por país (o resultado será uma média para cada país, assim como na questão anterior). A partir desse resultado, calcule o máximo e o mínimo dessas médias agrupados por região.

a) Taxa de população urbana.

Solução

```
# taxa de população urbana
wdi %>%
  group_by(country, region) %>%
  summarise(media = mean(urban_rate)) %>%
  group_by(region) %>%
  summarise(maximo = max(media),
            minimo = min(media))
```

b) Taxa de crescimento da população total.

Solução

```
# taxa de crescimento de população total
wdi %>%
  group_by(country, region) %>%
  summarise(media = mean(total_pop_growth, na.rm = TRUE)) %>%
  group_by(region) %>%
  summarise(maximo = max(media),
            minimo = min(media))
```

c) Taxa de crescimento da população urbana.

Solução

```
# taxa de crescimento de população urbana
wdi %>%
  group_by(country, region) %>%
  summarise(media = mean(urban_pop_growth, na.rm = TRUE)) %>%
  group_by(region) %>%
  summarise(maximo = max(media),
            minimo = min(media))
```

Questão 6

Crie um `data.frame` para cada país (cada um com o nome do país, tudo em lower case), contendo apenas as colunas `country`, `year` e a coluna contendo as informações sobre a taxa de população urbana.

Solução

```
# dataframe contendo os países
paises <- wdi %>%
  distinct(country)
paises

argentina <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[1,1])
```

```
brazil <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[2,1])

canada <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[3,1])

chile <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[4,1])

colombia <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[5,1])

denmark <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[6,1])

france <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[7,1])

germany <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[8,1])

italy <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[9,1])

paraguay <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[10,1])

peru <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[11,1])
```

```
spain <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[12,1])

united_kingdom <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[13,1])

united_states <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[14,1])

uruguay <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[15,1])

world <- wdi %>%
  select(country, year, urban_rate) %>%
  filter(country == paises[16,1])
```

Questão 7

Crie uma função que recebe um dataframe como parâmetro. Essa função deve fazer as seguintes manipulações nesse dataframe:

- Renomear a coluna contendo as informações sobre a taxa de população urbana 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

```
altera_df <- function(df) {
  # mudando a descrição da coluna
  attr(df$urban_rate, "label") <- "Urban Population Rate"

  # renomeando a coluna
  colnames(df)[3] <- df$country[1]

  # dropando a coluna country
  df$country <- NULL
```

```
  return(df)
}


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 8

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

Faça um gráfico apresentando a série temporal da taxa de população urbana para cada país, um para cada região.

Solução

```
# primeira maneira, juntando todos dataframes de uma vez
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, juntando um de cada vez
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" )
```

a) América Latina.

Solução

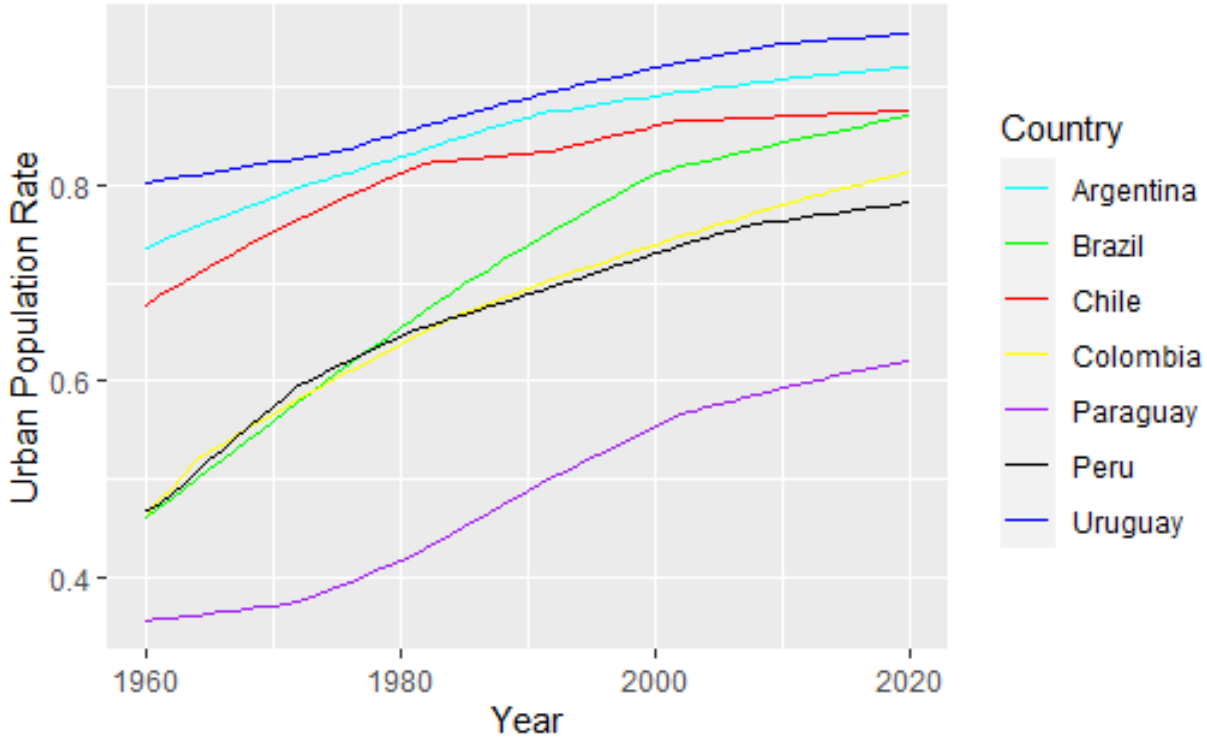
```
# países por região
pais_regiao <- wdi %>%
  group_by(region) %>%
  distinct(country)

# américa latina
pais_regiao %>%
  filter(region == "South America")

# 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 = "Urban Population Rate", 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



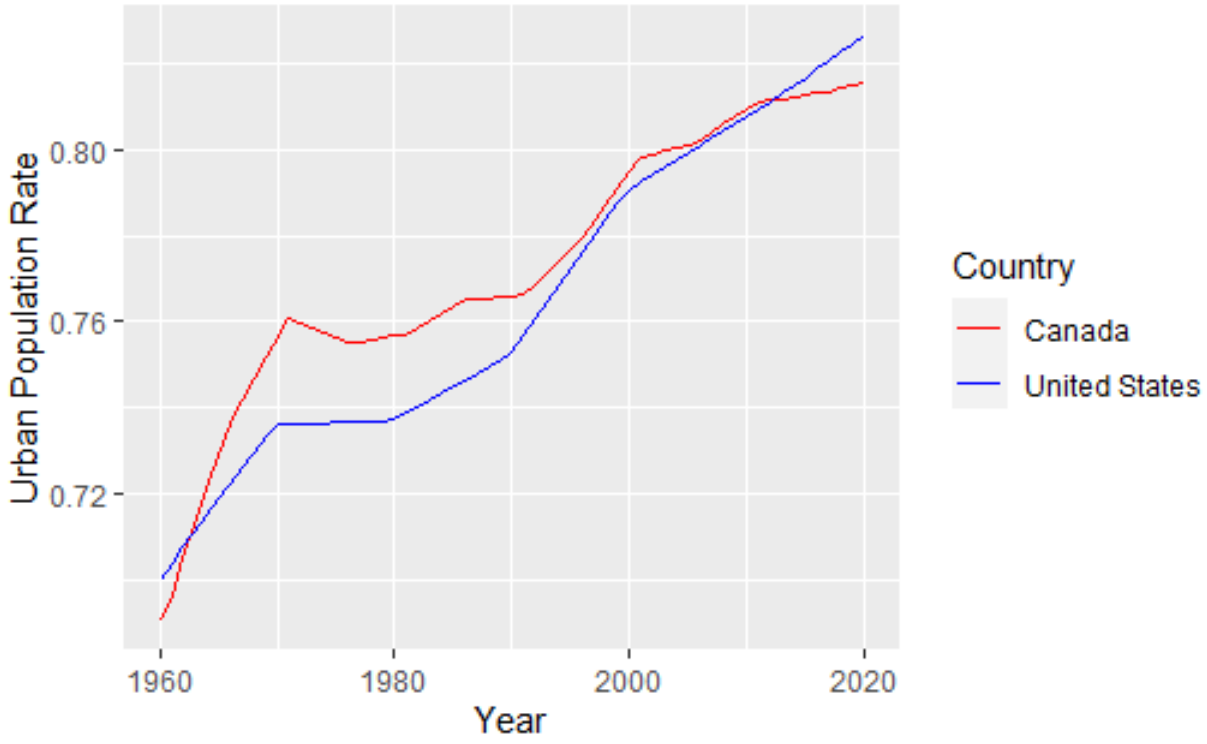
b) América do Norte.

Solução

```
# américa do norte
pais_regiao %>%
  filter(region == "North America")

# plot
ggplot(df, aes(year)) +
  geom_line(aes(y = Canada, color = "Canada")) +
  geom_line(aes(y = United_States, color = "United States")) +
  labs(y = "Urban Population Rate", 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 == "Europe")

# 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 = "Urban Population Rate", 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