Introduction to R

First course - Answers

Jean-Baptiste Guiffard and Florence Lecuit

October 11, 2024

Load dataset

```
data_pollution <- read.csv2('DATA/owid-co2-data.csv', sep=",")
```

Exercise 1: Manipulating vectors

```
a=c("apple", "orange", "banana", "strawberry", "lemon")
b=c(1,2,3,4,5)
c=c(6,7,8)
```

1. What is the length of vector a?

```
length(a)
```

```
## [1] 5
```

2. Try doing a[1:3], what is the result?

```
a[1:3]
```

```
## [1] "apple"  "orange" "banana"
```

3. Create new vector *agrumes* with only values *orange* and *lemon*

```
agrumes <- a[c(2,5)]
agrumes
```

```
## [1] "orange" "lemon"
```

4. Try doing a[-1], what is the result?

```
a[-1]
```

```
## [1] "orange"     "banana"     "strawberry" "lemon"
```

5. Sort vector a alphabetically.

```
sort(a)
```

```
## [1] "apple"      "banana"     "lemon"      "orange"     "strawberry"
```

6. Combine vectors b and c into a data-frame. What is the problem?

```
#df_combined1 <- data.frame(b,c)
#print(df_combined1)
```

7. Combine vectors a and b into a data-frame. Why does this work?

```
df_combined2 <- data.frame(a,b)
print(df_combined2)
```

```
##            a b
## 1      apple 1
## 2     orange 2
## 3     banana 3
## 4 strawberry 4
## 5      lemon 5
```

Exercise 2: Describing a data-frame

1. What type of object is *data_pollution* ?

```
class(data_pollution)
```

```
## [1] "data.frame"
```

2. How many observations and variables does this dataset contain?

```
ncol(data_pollution) #number of variables
```

```
## [1] 60
```

```
nrow(data_pollution) #number of observations
```

```
## [1] 26008
```

3. How many missing values are there in this dataset?

```
#Whole dataset
sum(is.na(data_pollution))
```

```
## [1] 2130
```

```
#Specific column
sum(is.na(data_pollution$population))
```

```
## [1] 2130
```

```
sum(is.na(data_pollution$gdp))
```

```
## [1] 0
```

4. What type of variables are in this dataset?

```r
str(data_pollution) #whole description of data-frame
```

```r
class(data_pollution$population) #type of specific variable
```

```
## [1] "numeric"
```

Exercise 3: Subsetting, Selecting Columns, and Dropping Duplicates

1. Create a new data frame that contains only the variables: `country`, `iso_code`, `year`, `population`, gdp, and co2.

```r
library(tidyverse)
data_pollution <- read.csv2('DATA/owid-co2-data.csv', sep=",")
df1 <- data_pollution %>% select(country, iso_code, year, population, gdp, co2)
```

2. Filter the dataset to include only data for the country "France".

```r
df1_france <- df1 %>% filter(country == "France")
```

3. Subset the data to include only countries with a population greater than 50 million. Which variable should you use to do this?

```r
df1_pop50 <- df1 %>% filter(population > 50000000)
```

4. Check if there are any duplicate rows in the dataset and drop them if they exist.

```r
df1 <- df1 %>% distinct()
```

Exercise 4: Creating Variables

1. Create a new variable in the dataset called `gdp_per_capita`, which calculates GDP per capita (GDP divided by population).

```r
df1 <- df1 %>% mutate(gdp_per_capita = as.numeric(as.character(gdp)) / as.numeric(
```

2. Similarly, create a new variable called `co2_per_capita`, which calculates CO_2 emissions per capita.

```r
df1 <- df1 %>% mutate(co2_per_capita = as.numeric(as.character(co2)) / as.numeric(
```

3. Are there any missing values in the new variables you created? If so, filter out the rows where these values are missing.

```r
df1 <- df1 %>% filter(!is.na(gdp_per_capita) & !is.na(co2_per_capita))
df1 <- df1 %>% distinct(gdp_per_capita, .keep_all = TRUE)
```

4. Create a new variable that groups countries into quartiles based on GDP per capita.

```r
df1 <- df1 %>% mutate(quartiles = ntile(gdp_per_capita, 4))
```

Exercise 5: Basic Statistics

1. For the new dataset, calculate the mean, minimum, and maximum for the gdp_per_capita and co2_per_capita columns.

```
df1 %>% summarise(mean_gdp_per_capita = mean(gdp_per_capita, na.rm = TRUE),
                  min_gdp_per_capita = min(gdp_per_capita, na.rm = TRUE),
                  max_gdp_per_capita = max(gdp_per_capita, na.rm = TRUE),
                  mean_co2_per_capita = mean(co2_per_capita, na.rm = TRUE),
                  min_co2_per_capita = min(co2_per_capita, na.rm = TRUE),
                  max_co2_per_capita = max(co2_per_capita, na.rm = TRUE))
```

```
##   mean_gdp_per_capita min_gdp_per_capita max_gdp_per_capita mean_co2_per_capita
## 1            8695.532           363.2703           146405.5        3.824741e-06
##   min_co2_per_capita max_co2_per_capita
## 1        1.1993e-10       0.0001009685
```

2. Group the data by country and calculate the average co2_per_capita for each country.

```
df1 %>% group_by(country) %>% summarise(mean_co2_per_capita = mean(co2_per_capita,
```

```
## # A tibble: 165 x 2
##    country     mean_co2_per_capita
##    <chr>                     <dbl>
##  1 Afghanistan         0.000000129
##  2 Albania             0.00000147
##  3 Algeria             0.00000224
##  4 Angola              0.000000564
##  5 Argentina           0.00000239
##  6 Armenia             0.00000178
##  7 Australia           0.00000762
##  8 Austria             0.00000472
##  9 Azerbaijan          0.00000512
## 10 Bahrain             0.0000188
## # i 155 more rows
```

3. Group the data by quartiles and calculate the average co2_per_capita for each quartile.

```
df1 %>% group_by(quartiles) %>% summarise(mean_co2_per_capita = mean(co2_per_capita
```

```
## # A tibble: 4 x 2
##   quartiles mean_co2_per_capita
##       <int>               <dbl>
## 1         1         0.000000262
## 2         2         0.00000101
## 3         3         0.00000375
## 4         4         0.0000103
```

Exercise 6: Merging Datasets

1. Load a new dataset that contains additional information on countries.

```
df2 <- read.csv2('DATA/Metadata_Country.csv', sep=",")
```

2. Which variable(s) would you use as the key(s) to merge the two datasets (`df1` and `df2`)? Explain why.

`Country.Code` for df2 and `iso_code` for df1. They are the common variables between the two datasets.

3. Perform an inner join between `df1` and `df2` based on the common key(s).

```
df_merged <- inner_join(df1, df2, by = c("iso_code" = "Country.Code"))
```

4. After merging the datasets, check how many new columns were added. How many columns and rows does the new data frame now contain?

```
ncol(df_merged)
```

```
## [1] 14
```

5. Check for any missing values in the merged dataset after the join. Which countries or years might be missing from one of the datasets?

```
df_merged %>% filter(is.na(co2_per_capita))
```

```
##  [1] country       iso_code      year          population    gdp
##  [6] co2           gdp_per_capita co2_per_capita quartiles     Region
## [11] IncomeGroup   SpecialNotes  TableName     X
## <0 lignes> (ou 'row.names' de longueur nulle)
```

7. Group the data by `continents` and calculate the average `co2_per_capita` for each continent.

```
df_merged %>% group_by(Region) %>% summarise(mean_co2_per_capita = mean(co2_per_cap
```

```
## # A tibble: 7 x 2
##   Region                    mean_co2_per_capita
##   <chr>                                   <dbl>
## 1 East Asia & Pacific                0.00000309
## 2 Europe & Central Asia              0.00000534
## 3 Latin America & Caribbean          0.00000192
## 4 Middle East & North Africa         0.00000852
## 5 North America                      0.0000105
## 6 South Asia                         0.000000293
## 7 Sub-Saharan Africa                 0.000000712
```