7316 - INTRODUCTION TO R

Module 2: Data manipulation in R
Correction of the assignment

Teacher: Mickaël Buffart (mickael.buffart@hhs.se)

In this assignment you will work on some tables from David Card and Alan Krueger's seminal 1994 paper on the minimum wage introduction in New Jersey. The paper is available on the course webpage.

1 BASIC SETUP: TIDYVERSE AND RIO

1. Install the packages "Tidyverse" and "Rio", if you have not yet installed them.

```
install.packages("rio")
install.packages("tidyverse")
```

2. Create a project for this assignment. Create a data folder in it.

- See module 1: to create a new project, you have to go to **File** > **New Project...** > **New Directory** > **New Project** and choose the name and location where you want to create the project on your computer.

3. Download the dataset `card_krueger_public.dta` (in module_2_assignement.zip) from the course website. Copy it into your project data folder. Load the data.

- Go to Canvas; download `module_2_assignement.zip`
- Create the `data` folder in your project directory, and copy the cotent of `module_2_assignement.zip` within.
- Then, load the data file from the data folder into a `data.frame`:

```
df <- rio::import("data/card_krueger_public.dta")
```

2 TAKING A FIRST LOOK

4. Describe your dataset: what is its structure: how many variables, types of variables, etc.

```
str(df)
```

- You can see the structure of your dataset with the `str()` command, or by looking at your environment tab, in RStudio:
  - The dataset contains 46 variables, and 410 observations.
  - The variables appear to be all of `numeric` type, named from V1 to V46.

You will notice that the data does not contain any variable names. We therefore refer to the codebook (on the course website) to find the necessary variables. I have prepared a csv file with the variable names and labels called `variable_names.csv`.

5. Load this list and assign each column the appropriate variable name.

- We can load the list using the same command as we used to load the dataset.

```
# Loading the vairable names
variable_names <- rio::import("data/variable_names.csv")
```

- The list of names contains 46 observations, one per variable. We assume the variable names appear in the order of the variables. We can assign names to the variables in the dataset with the following command:

```
# Assigning names from the variables_names file
names(df) <- variable_names$variable
```

- the `variable_names` file contains labels. It is possible to set labels to the variables in the `data.frame` with the package `sjlabelled`, and the `set_label` function:

```
df <- sjlabelled::set_label(df, variable_names$label)
```

- Now, looking at the structure of `df`, you will see the names and the labels.

The dataset is still a bit large, given that we only want to replicate a two tables.

6. Drop all variables except `SHEET`, `CHAIN`, `STATE`, `EMPFT`, `EMPPT`, `NMGRS`, `EMPFT2`, `EMPPT2`, `NMGRS2`, `STATUS2`

```
df <- df[, names(df) %in% c("SHEET", "CHAIN", "STATE", "EMPFT",
                            "EMPPT", "NMGRS", "EMPFT2", "EMPPT2",
                            "NMGRS2", "STATUS2")]
```

3 SUMMARIZING THE DATA

7. Check if `EMPFT` contains missing values.

```
table(is.na(df$EMPFT))
##
## FALSE
##   410
```

- There is no missing values in `EMPFT`.

We now want to get a feeling for the kind of observations we are dealing with. Card and Krueger sample restaurants of different US fast food chains (Burger King, KFC, Roy Rogers, Wendy's). We would like to know the distribution of the different chains across New Jersey and Pennsylvania (table 2 in the paper)

8. Create a separate dummy variable for each chain that equals 1 (or `TRUE`) if the store belongs to this chain and 0 (`FALSE`) otherwise

```
# Checking the number of chains
table(df$CHAIN)
##
##   1   2   3   4
## 171  80  99  60
```

- We have 4 chains. According to the labels, 1=bk; 2=kfc; 3=roys; 4=wendys.

```
# Creating dummies
df$chain_bk <- as.integer(df$CHAIN == 1)
df$chain_kfc <- as.integer(df$CHAIN == 2)
df$chain_roys <- as.integer(df$CHAIN == 3)
df$chain_wendys <- as.integer(df$CHAIN == 4)
```

9. Tabulate the mean of each of these 4 variables by State

```
# We recode state with their name, following instruction in the labels
df$STATE[df$STATE == 1] <- "New Jersey"
df$STATE[df$STATE == "0"] <- "Pennsylvania"

# We tabulate the means by state for the four variables
tmp_1 <- aggregate(chain_bk ~ STATE, df, mean)
tmp_2 <- aggregate(chain_kfc ~ STATE, df, mean)
tmp_3 <- aggregate(chain_roys ~ STATE, df, mean)
tmp_4 <- aggregate(chain_wendys ~ STATE, df, mean)
```

10. Save the tabulated values into a `matrix`

```
means_chain <- data.frame(bk = tmp_1$chain_bk,
                          kfc = tmp_2$chain_kfc,
                          roys = tmp_3$chain_roys,
                          wendys = tmp_4$chain_wendys)

# Adding rownames
rownames(means_chain) <- tmp_1$STATE

# Creating matrix
means_chain <- as.matrix(means_chain)
```

11. Remove the stata dummy and transpose the matrix, rename the columns such that it corresponds to the *Distributon of Store Types* section of Table 2 and turn it into a tibble.

```
# We did not include the State dummy. In case we add, we would
#   use
# means_chain$STATE <- NULL
#   to remove it

# transpose
means_chain <- t(means_chain)

# As tibble
means_chain <- tibble::as_tibble(means_chain)
```

12. Print the table

```
means_chain
## # A tibble: 4 x 2
##   `New Jersey` Pennsylvania
##          <dbl>        <dbl>
## 1        0.411        0.443
## 2        0.205        0.152
## 3        0.248        0.215
## 4        0.136        0.190
```

4 Tyding up the dataset

If you look at the data, you will realize that the values for a single store are spread across several columns. The number of full-time employees is recorded in the variable `EMPFT` for the first year and `EMPFT2` for the second year. This violates the tidy principle that each observation has its own row. To make the tyding easier, we first reduce the number of variables by aggregating full-time employment, part-time employment and managers into one variable for full-time equivalents (`FTE`).

13. Aggregate the employment for each store and period into two new variables called `FTE1` and `FTE2`. Follow the paper and use the formula $FTE = EMPFT + 0.5 * EMPPT + NMGRS$

```r
df$FTE1 <- df$EMPFT + 0.5 * df$EMPPT + df$NMGRS

df$FTE2 <- df$EMPFT2 + 0.5 * df$EMPPT2 + df$NMGRS2
```

14. Order the data in `FTE1` ascending order and `FTE2` descending order.

```r
df <- df[order(df$FTE1, -df$FTE2),]
```

15. Gather the data, such that for each store you have two observations of FTE, one for each year. Save this new dataset as `data_tidy`.

```r
data_tidy <- tidyr::gather(df,
                           key = "year",
                           value = "FTE", FTE1:FTE2)

data_tidy$year[data_tidy$year == "FTE1"] <- 1
data_tidy$year[data_tidy$year == "FTE2"] <- 2
data_tidy$year <- as.integer(data_tidy$year)
```