Lab 01: -Basics and Graphs

**Handout date:** Thursday, 3, 2020

**Due date:** Thursday, September 17, 2020 submitted as Word document to eLearning’s ***SumitLab01*** link.

*This lab counts 8 % toward your total grade*

**Objectives:** In this lab   
[a] you will practice your  data handling skills.

[b] generate statistical graphs and maps

**Format of answer:** Your answers (graphs and verbal description) should be handed in as Word document. Add a running title into the header of the document with the following information: ***your name***, ***EPPS6316-Lab01*** and ***page numbers***. Label each answer properly starting with its task number. Maintain the sequence of questions. Format any code and computer output properly before inserting it into the document with your answer. -code and text output need to be in a ***monospaced*** font (i.e., fixed-pitch font) such as Courier New so proper spacing and alignments are preserved. Excessive and irrelevant output will lead to a deduction of your accumulated points.

# Task 1: Importing Data (0.5 pts)

Setup an  working directory and save the files **MyPower.RData** , **Concord1.sav** and **CPS1985.dbf**. into this directory.

1. Explore the **load( )** function and import **MyPower.RData** .
2. Use a function from the library **foreign** to import **Concord1.sav** and save it under the name **Concord**.
3. Use a function from the library **foreign** to import **CPS1985.dbf** and save it under the name **CPS1985**.
4. Explore the documentation of the data-frame **Mroz** in the library **carData** and link the data-frame to your  session with the **data( )** function.
5. To demonstrate that everything worked as intended show a screenshot of **Global Environment**, which displays all 4 data-frames.

# Task 2: Data-frame Basics (1.5 pts)

1. For the data-frame **MyPower** calculate the average daily power consumption by using the variables **kWhBill** and **DaysBill** and add the new variable to the data-frame **MyPower** with the variable name **DailykWh**. Show your  code for this calculation. (0.2 pts)
2. Apply the statements   
    **MyPowerNames <- names(MyPower)  
    length(MyPowerNames)  
    MyPowerNames[4:6]**  
   What are these statements doing? (0.2 pts)
3. Apply the statement **sapply(***data-frame***, is.factor)** on the data-frame **MyPower** to evaluate the which variables are factors. What is this statement doing? Show a copy of the Console with the output of this investigation. (0.1 pts)
4. Apply the **str( )** on the data-frame **MyPower**. What information about the data-frame does the **str( )** provide to you? (0.1 pts)
5. What are the following statements doing? Show the plot and elaborate on the syntax of the statements. Is the power consumption over time decreasing? (0.3 pts)  
    **plot(DailykWh~SeqID, data=MyPower)  
    abline( lm(DailykWh~SeqID, data=MyPower) )**
6. What is the following statement doing? Show the plot and elaborate on the syntax of the statements. Why does the power consumption fluctuate over the seasons? (0.2 pts)  
    **plot(DailykWh~Month, data=MyPower)**
7. Use the syntax **MyPower[***rows***,** *cols***]** to select all records with the three variables **c("MinTemp","AveTemp","MaxTemp")** (alternatively you could use **MyPowerNames[4:6]**) in the month of **MyPower$Month=="JAN"**. Show the code and its output. (0.2 pts)
8. Look and show at the header and the tail of the data-frame **MyPower** with the functions **head( )** and **tail( )**. (0.1 pts)
9. What class is the output **MetricPower** of the operation below? (0.1 pts)  
   **MetricPower <- MyPower[ , c("MinTemp","AveTemp","MaxTemp","DailykWh")]**

# Task 3: R Basics (1 pt)

1. Depending on the input object class type  functions behave differently. To see the different class-specific implementations of the generic **summary( )** function try the command **methods(summary)**.  
   Discuss the difference in the behavior of the **summary( )** function when applied to a **data.frame** or a **lm** object. (0.2 pts)
2. Explore the online help for  
    **?summary.lm  
    ?summary.data.frame**and discuss the optional parameters (0.2 pts)
3. What are the following statements below doing when processing the vector x of length 6? (0.3 pts)  
    **x <- c(1,3,5,7,9,NA)**  
    **x \* 2; x + 2**  
    **y <- seq(0,2, by=1); x \* y**  
    **z <- rep(c(1,2),3); x \* z**Note, the semicolon **;** allows to place several independent  commands into one line. Look the online help up for the functions **seq( )** and **rep( )**.
4. Define the function **myMean( )** and apply it on **x**  
    **myMean <- function(x){  
    x <- na.omit(x)  
    sum(x)/length(x)  
    }  
    myMean(x)**Compare its output to that of the standard  function **base::mean( )**. Look up the help for the functions **na.omit( )** and **mean( )**. (0.2 pts)
5. How does the statement **x[c(T,T,T,T,T,F)]** work? (0.1 pts)

# Task 4: Working with Data (1 pt)

For all steps below show your properly formatted code. You find the necessary code for the examples in Lander and the online help. Only if asked show also the output.

Import the SPSS data-file **Concord1.sav** as ***data-frame*** into the  environment by using a function from the library **foreign**. Make sure to name your data-frame properly.

1. Discuss the summary statistics for the water consumption: How did the ***average*** water consumption change from 1979 to 1981? (0.1 pts)
2. Discuss the summary statistics: Which variable has ***missing*** observations? (0.1 pts)
3. Discuss the summary statistics: Which variable is a factor? (0.1 pts)
4. List all ***case numbers*** (variable **case**), which have at least for one variable missing value in a variable. Show also the code. (0.2 pts)
5. Calculate the ***average*** water consumption over the 3 years for each household and save it the new variable **meanWater** into the data-frame. Caution: also include households, which have **NA**s in the water consumption. Hint: look at the documentation of the function **rowMeans( )**. (0.2 pts)
6. Use logical statements to identify those households (variable **case**), which have above average water consumption in 1981. Show the code and the household numbers (0.2 pts)
7. Draw a random sample of 10 households without repetitions. Show the household numbers (variable **case**) and the code. Hint: look at the documentation of the function **sample( )**. (0.1 pts)

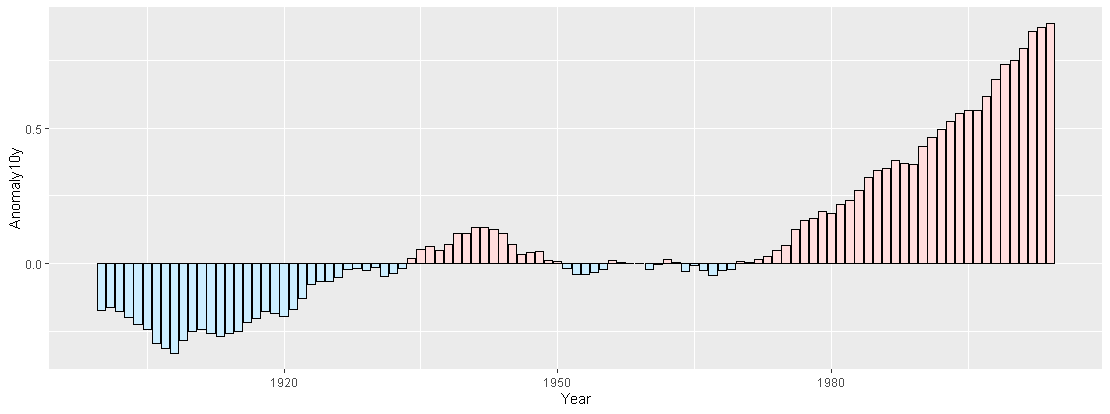
# Task 5: Statistical Graphs (2 pts)

The following graphs are reproductions from the [R Graphics Cookbook, 2e](https://r-graphics.org/index.html). One of the objectives of this task is that you visit the **R Graphics Cookbook** and explore the graphs on display there.

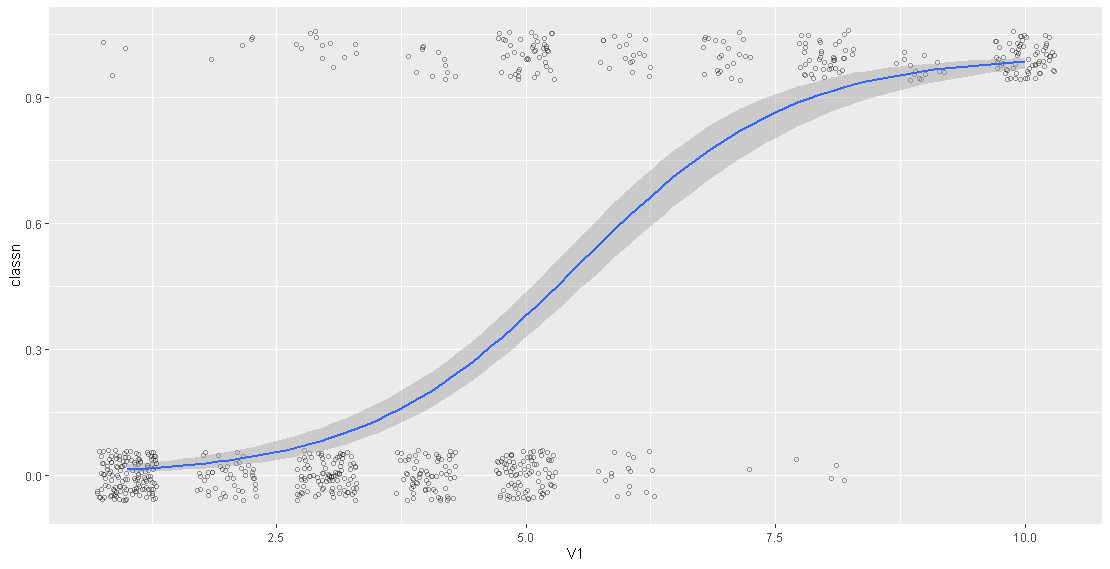
Study the sections associated with the graphs which are displayed below. Its sections document how these graphs were built. For educational purpose you may want to read the associated documentation of the employed functions and experiment with their options.

You job in this is to *reproduce* these graphs and *show the code*, which generated them. The data used in the **R Graphics Cookbook** are available in **library("gcookbook")**.

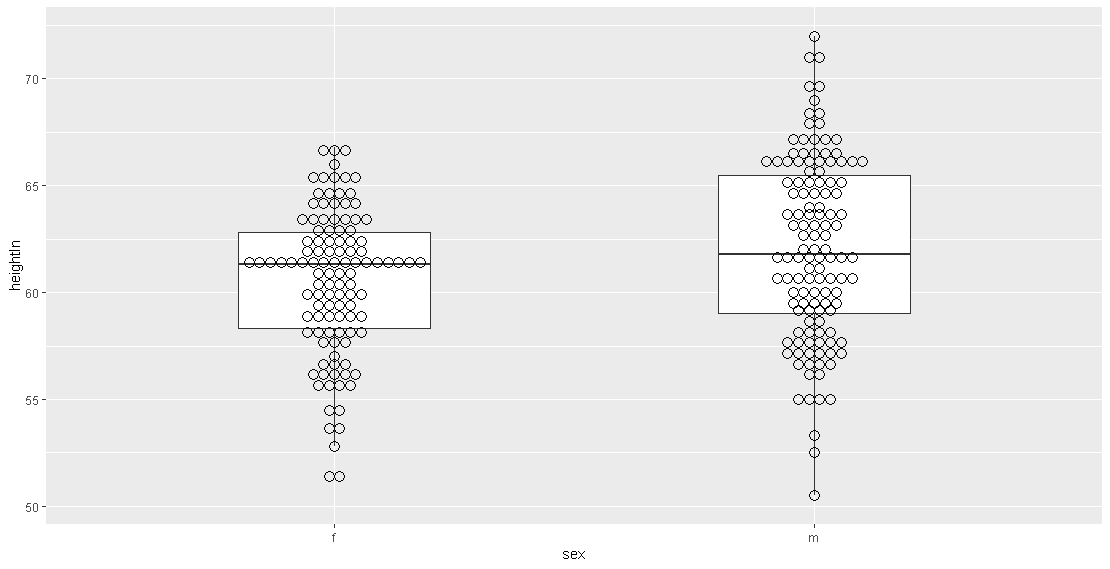
[a] Reproduce the graph below (0.5 pts)



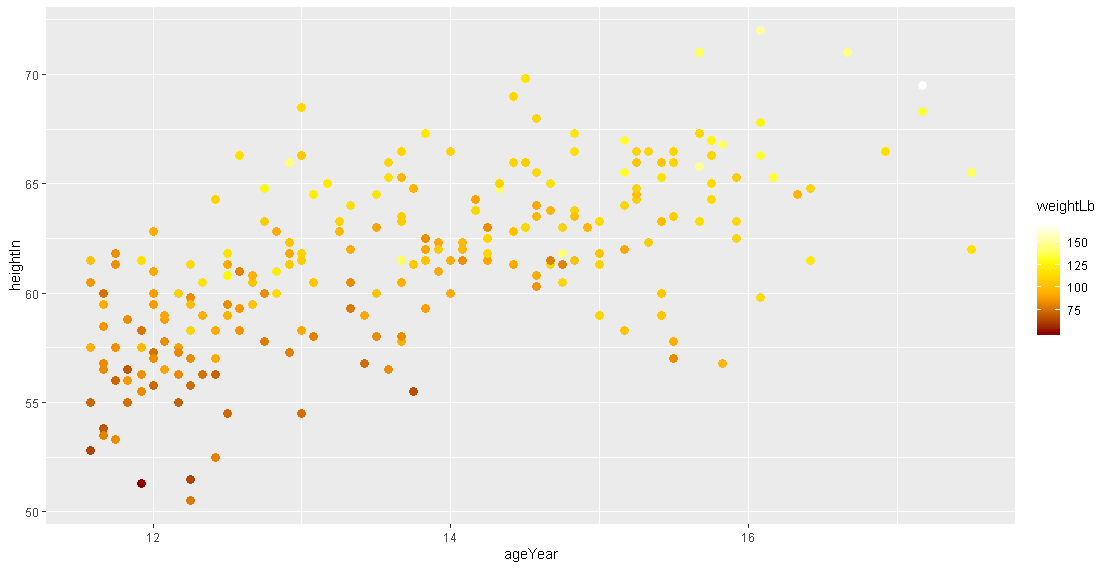
[b] Reproduce the graph below (0.5 pts)



[c] Reproduce the graph below (0.5 pts)



[d] Reproduce the graph below (0.5 pts)



# Task 6: Mapping (2 pts)

You are given tow ESRI shape files. These files are packed into the zipped file **Italy.zip**. The maps you will generate consist of two area layers: [a] the countries neighboring Italy (layer **Neighbors.shp**) and [b] added on top of it the 95 provinces of Italy (layer **Provinces.shp**). Make sure that the frame of the plot window is sized properly to embed Italy. The base map below shows these two layers in a properly sized window frame.

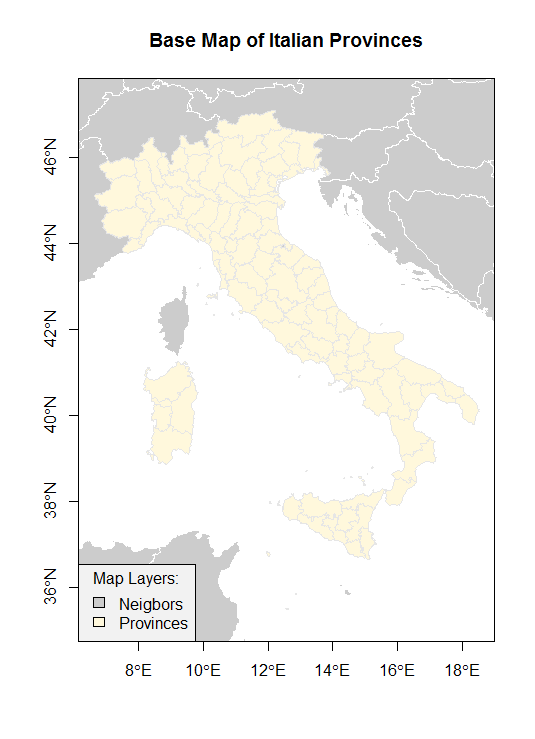


Figure 1: Base Map of the Italian Provinces

You will generate three color maps displaying different map themes. For your lab answers please show these maps in color. Each map should be properly framed, have a proper title and legend, show the neighboring countries as spatial reference frame. You can use the mapping functions in the package  **TexMix**. Just show the code for your maps.

[a] Generate a qualitative map showing the Italian regions, which are stored in the variable **REGION**. Show the relevant code used to generate the map. (0.6 pts)

[b] Generate a color ramp map showing the total fertility rate (number of births per woman) in the Italian provinces using 7 classes, which is stored in the variable **TOTFERTRAT**. Show the relevant code used to generate the map. (0.6 pts)

[c] Generate a bipolar map showing the gender ratio in the 95 provinces, which can be calculated with the transformation **logMigRatio <- log(shp$INFLOW/shp$OUTFLOW)**, where **shp** refers to the name of your imported shape-file. What is the neutral break-point for the variable **logMigRatio**. Use in total 8 classes but select the appropriate number of classes for the below and above breakpoint branches of the underlying distribution of **logMigRatio**. Justify your choice. Show the relevant code used to generate the map. (0.8 points)