Exercise set 3. Introduction to R

Data Visualization and Modelling

in Master in Modelling for Sciences and Engineering, UAB. September 2022.

3.0 The exercises 3.1–3.11 should be included in a single script:

(a) If possible send the solutions to the exercises in RMarkdown. You should create a file with name “yourname-3.Rmd”. As a comment, inside the script, write your name and your NIU.

3.1 Choose a data frame of at least 1000 rows and at least 4 numeric data and 2 factors. It should not be created by yourself using random functions.

(a) Use `pairs` to compare the numeric variables. Which are more correlated?

(b) Represent two numeric variables and two factors in the same graph. Use colors or size or plot characters for the factors. You should add two legends.

(c) Use the xyplot in the lattice package instead to do the representation of the same variables.

3.2 (a) Draw a random sample of size 100 from the interval $[0, 2]$, which contains 201 values. Sample without replacement.

(b) Use dt to evaluate the density function of the t distribution with 13 degrees of freedom at 21 values in the range -1 to 1.

(c) Find x such that $P[X \leq x] = 0.01$ for a t distribution with 9 degrees of freedom.

(d) IQ scores are known to have a normal distribution with mean 100 and standard deviation 15. What IQ would you have if you were in the 80th percentile?

(e) What IQ would you have if you were in the top 10 percent?

(f) What is the probability of having an IQ above 142?

(g) Set the seed to ”0” and create two samples of size 20 from the standard normal distribution with the same values. Repeat the process but set the seed to your ID number.

3.3 (a) Create a vector `x` of the values from 1 to 20.

(b) Create a vector `w <- 1 + sqrt(x)/2`.

(c) Create a data frame called `dummy`, with columns `x = x` and `y = x + rnorm(x)*w.` To ensure we all get the same values, set the seed to 0.

(d) Create a histogram and a boxplot of `y` and plot them side-by-side on the same graphing region. Label the axes accordingly. Save your results as a Jpeg file.

(e) Plot `y` versus `x` using an appropriate plotting command. Put a title on the graph and labels on the axes.

(f) Enter the command `fm <- lm(y~x, data=dummy)` to fit a linear regression model. Add the estimated regression line to the current plot and make it the colour blue.

(g) Extract the values of the residuals using `resids <- resid(fm).` Check that the residuals are normally distributed by creating a Q-Q plot.

3.4 (a) The `airquality` data set in the base library has columns `Ozone, Solar.R, Wind, Temp, Month` and `Day`. Plot `Ozone` against `Solar.R` for each of THREE temperature ranges and each of THREE wind ranges. (Hint: Use `coplot`.)

(b) Construct a histogram of `Wind` and overlay the density curve. (Hint: Need `hist, lines` and `density`.)

3.5 (a) Create a function that changes “,” into “.” in a string.

(b) Create a function that changes a string with numbers separeted by a "," into a numerical variable with the decimal mark ".". For instance the string "3,1416" should be changed into the number 3.1416.

(c) Using function `read.csv` read the file "countries of the word.csv" you will find in Moodle, and call it countries. It has comma as variable separator. But some numerical variables such as `Density (per sq. mi.)` are imported as characters or factors (they should be numerical but have a comma as a decimal separator). Use the function in (b) to fix all variables that are character type and should be numerical.

Note: to do it quickly in a single step you can use lapply sub data frame of `countries` that contains the variables you want to fix.

3.6 (a) Create a function that given a vector v1 of characters and another vector of characters v.na converts all the values in v1 contained in v.na to NA. Use this function to redo exercise 1.4(a)

(b) Create a function called `big.values` that given a vector and a value `tresh` returns the vector with the values bigger than `tresh` changed into NAs.

(c) Modify the previous function with an extra argument that gives the option of transforming the low values (lower than `tresh`) into NAs, instead. Make the default to change big values.

(d) Use the function to change in the `airquality` data frame all the values bigger than 320 into NA.

3.7 (a) Create a function called `append.f` that adds a new variable to a data frame.

(b) Modify the above function so that it checks whether the length of the vector is compatible with the dimensions of the data frame. And print out a message "the lenghth of the vector is wrong"

(c) Add a new argument that gives the name of the new variable. By default make this name to be "new.variable"

3.8 Create a function that, given a data frame, two of its numerical variables `x` and `y`, and two numbers `a` and `b` returns the data frame with a new v given by the formula

$$v_i = a * \sum_{k=1}^{i} x_k + b * \sum_{k=1}^{i} y_k$$

The name of the new variable should be `sumxy_a_b`. For instance, if the variables are the first and the third and the values are $a = 5$ and $b = 3.4$ the name should be `sum12_5_3.4`.

Be carefull with NA's values of variables x and y. They shoud be replaced by 0.

3.9 Write a function that when passed a numeric vector, prints the value of the median and interquartile range to the screen (Hint: use the `cat()` function in R.) and creates a boxplot of the data.

3.10 Add up all the numbers for 1 to 100 in two different ways: using a for loop and using sum.

3.11 Recover the list `bycountry` of dayly cases of monkeypox by country that you created in exercise 2.5 (exercise set 2). Use a function of the apply family to get a vector of the total number of monkeypox cases in each country. Sort it to get the 6 countries of highest incidence.