**Practical exercises**

1. Using the builtin rivers vector:

* Report minimum and maximum river lengths and their index in the vector.
* Plot a histogram of all integers between 1 and the maximum river length that do not appear in

the rivers vector.

* Compute mean and standard deviation of the rivers vector, and use them to create a new

vector called randomrivers (of the same length as rivers) of normally distributed points accord- ing to those parameters (set the random seed to 1 beforehand). Count how many elements in randomrivers are negative (answer: 12) and how many values in randomrivers are more than double than their corresponding element in rivers (answer: 39).

* Create a scatter plot of rivers and randomrivers and report the Pearson’s correlation coefficient between the two vectors at 3 significant digits (answer: 0.0892). What can be said about the distribution of rivers? In what other (simpler) way could we have reached the same conclusion?

2. Using the builtin islands vector:

* + Report names and sizes of islands with area in the first quartile, and compute the median area in this subset (answer: 14.5).
  + Report sizes of the 10th largest and of the 10th smallest island (answer: 280, 16), and count how many islands have an odd area (answer: 21).
  + Create a vector called islands3 which includes only islands with area divisible by 3 and report its median (answer: 36) and interquartile range (answer: 25.5, 244.5).
  + Report the smallest area in islands3 that is also present in the rivers vector (answer: 306), and for the areas in islands3 that are not present in rivers report the mean rounded to the first decimal place (answer: 1283.5).

3. Using the builtin attitude dataframe:

• For the “rating” variable, report median, range and interquartile range.  
• Report the median rating for observations that have above median values for the “raises” variable

(answer: 71).

* Compute the standard deviation for the “advance” variable and compare it to the one computed

after removing the extreme values (answer: 10.288706, 8.3864182).

* For each variable in the dataframe, produce histogram and box-plot (using function boxplot())

side by side: you will need to first specify par(mfrow=c(1,2)) to tell R that you want your image to contain one row and two columns. Assign correct axis labels as well as plot titles.

4. Using the builtin quakes dataframe:

* Do a scatter plot of longitude and latitude (set cex=0.5 to decrease the point size), then by using abline() add lines corresponding to median longitude and latitude. Using a different colour, also add lines corresponding to mean longitude and latitude.
* Create a dataframe called quakes.1sd which contains only points with longitude and latitude that are within one standard deviation from the mean or have earthquake magnitude at least 5.5. Add these observation to the previous plot using function points(), using yet another colour.
* Add a variable to quakes.1sd called “damage” according to the following equation: 􏰃 √4

A screenshot of a cell phone

Description automatically generated

Count the number of observations in quakes.1sd (answer: 585) and the range of variable “damage” rounded to 2 decimal places (answer: 23.17, 58.84). Report the correlation between “damage” and all other variables in quakes.1sd.

* Create a dataframe called quakes.40s which contains only points reported by more than 40 stations and count how many have a row name of length 3 (answer: 243), how many contain the character 7 (answer: 77), and how many contain the character 9 but not in the first position

(answer: 44).

5. Using the builtin nottem timeseries dataset:

• Convert to data table format. Note that the code for the conversion of time series data is provided for you using the lubridate package. Try writing your own code to convert the time series to a data table format.

**> library**(lubridate)  
**> library**(data.table)  
**>  
>** nottem.dt <- **data.table**(temp = **c**(nottem), year=(**c**(**time**(nottem))))[, **+** .(month=**format**(**date\_decimal**(year**+**.01), "%b"),

**+** Year=**format**(**date\_decimal**(year**+**.01), "%Y"), temp)]

* **>**nottem.dt <- **reshape**(nottem.dt, idvar="Year", timevar="month", direction="wide")
* **>  colnames**(nottem.dt) <- **gsub**("temp.","",**colnames**(nottem.dt))

• Using the assign by reference := assignment operator, add a column called summer.avg and add the mean temperature of the summer months June to September for each year. answer: check the sum of the summer average sum(nottem.dt$summer.avg) equals 1184.7

• Using the assign by reference := assignment operator, add a column called june.decade.avg which provides the average temperature in june from 1920 to 1929 and 1930 to 1939 answer: 56.99 for 20's and 59.09 for 30's