1 R Basics

Exercise 1 (Data structures and subsetting)

(a) Create a `list` named `list1` with the following components

`x` : a numeric vector $35, 34, 33, \dots, 7, 6, 5, 6, 7, \dots, 33, 34, 35$ (don't type the single numbers!)

`y` : a factor with the first five letters of the Latin alphabet as levels, each repeated 12 times, except the last one which should be repeated 13 times

`mat` : a numeric 4×4 matrix whose entries are randomly drawn from an exponential distribution with $\lambda = 5$. Use the `set.seed()` function to make your result reproducible.

`list2` : a list with the components

`t` : a numeric variable with the value 35

`d` : a data.frame with the variables

`gender` : a factor with elements male, male, male, female, female, female, male, male, male, female, female, female

`age` : a numeric vector with elements $23, 48, 37, 37, 19, 54, 21, 20, 41, 26, 35, 32$

(b) Use subsetting operators to access the following information

(a) the 4th and the 7th element of the vector `x`

(b) the entry on position $(3, 4)$ of the matrix `mat`

(c) the first six elements of the first column of the data frame `d`

(d) the age of the female individuals

What is the difference between the usage of `[]`, `[[]]` and $?

(c) Modify the created objects in the following ways

(a) redefine the levels of the factor y from lowest to highest as 'c', 'd', 'b', 'a', 'e'

(b) eliminate the first row and the third column of `mat`

(c) add a column `age2` to `d` with a factor taking the value 'old', if the individual's age is above the threshold `t`, and the value 'young' otherwise

(d) exclude the individuals that are at over (or exactly) 50 and strictly younger than 21 from the study

Exercise 2 (Loops and data manipulation)

(a) Create a matrix X of dimension 2500×12 containing random numbers $x_{ij} \sim \mathcal{N}(\mu = 0, \sigma^2 = 3)$.

(b) Create a new matrix $(y_{ij})_{ij}$ based on the one defined above, in which each column is standardized, i.e. $y_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}$, where $\bar{x}_j$ is the sample mean of the j-th column, and σ_j its sample standard deviation.

(c) From X keep only the rows for which at least 6 out of the 12 values are greater than 0. Hint: Remember that a logical vector can be converted to a numeric vector where `FALSE` takes the value 0, and `TRUE` the value 1.

(d) Check, if the elements of the 3rd column of mat from Exercise 1 are within the range defined by the elements in the first two columns. The output should be a logical vector.

(e) In each row of X replace the maximum value with the minimum value.

(f) Write a for-loop to do the following calculation for $i = 1, \dots, 2500$:

- if the i-th element of the first column X is positive, then add to the subsequent element $x_{i+1,j}$ a random number $u \sim \mathcal{U}(-1, 1)$
- otherwise, if x_{ij} is negative, add to the subsequent element $x_{i+1,j}$ a random number $u \sim \mathcal{U}(-2, 2)$.

(g) Consider the data.frame resulting from

```
set.seed(2015)
w <- runif(10)
x <- runif(10)
y <- runif(10)
z <- runif(10)
dat <- data.frame(a = w, b = x, c = y, d = z).
> dat
            a          b         c          d
1  0.06111892 0.70285557 0.6919100 0.75185674
2  0.83915986 0.39172125 0.4067718 0.25684487
3  0.29861322 0.03306277 0.2109400 0.38967137
4  0.03143242 0.40940319 0.6652073 0.88448520
5  0.13857171 0.74234713 0.7377556 0.57390551
6  0.35318471 0.88301877 0.9190050 0.18367673
7  0.49995552 0.26623321 0.8734601 0.11168811
8  0.07707116 0.07427093 0.8012774 0.32047459
9  0.65134483 0.81368426 0.5243978 0.09095567
10 0.51172371 0.38194719 0.1272213 0.47959896
```

Now imagine in each row the entries define two intervals $[a, b]$ and $[c, d]$. Add a column to `DF` with entry `TRUE`, if the intervals overlap and `FALSE`, if they don't.
Find ways to avoid for-loops in (a)–(c). Useful functions might be

`rnorm(), runif(), apply(), mean(), sd(), scale(), rowSums(), min(), max().`

If you are not familiar with some of these functions, use `?function` to check the documentation. Compare the `system.time()` of the solutions with and without for-loops.

Exercise 3 (Basic graphical tools)

(a) Consider again the matrix X from Exercise 2 (a) and create the following plots:

- A scaled histogram of the first column. Also add a dashed red line representing the estimated density.
- A quantile-quantile plot comparing the sample quantiles of the first and second column. Also add a line with intercept 0 and slope 1 to improve comparability. Try different `pch`-values and colours.
- Boxplots of the first, fourth and seventh row in one plotting window. Rename the group labels to `blue, green and yellow.`

2 Advanced Graphics

Exercise 1 (From basic plots to complex graphics)
Read the data set `school_math.raw`. It describes the mathematical achievements (`mathach`) of male and female (`Sex`) students at 4 different schools (`school`). Additionally there is information about the socio-economic status (`ses`) of the students family and the sector (`Sector`) of the school (public or catholic).

(a) Get familiar with the data set, e.g. via the functions `summary()` or `head()`.

(b) Create the following four plots in only one plotting window:

a scatterplot of the achievement vs. the socio-economic status. Add a line representing the estimated linear relationship between those variables.

a histogram of `mathach`.

a scaled histogram of `ses`. Add a line with the estimated density.

boxplots of the mathematical achievement for each school separately.
First, don't modify any options. In a second step include the following variations:

- The color of the numbers on the axes should be red.
- Data points should be represented by filled squares (■) instead of circles (○).
- Use squared plotting regions for each plot.
- All occurring lines should be dashed.

Modify your plot by changing the settings within the single plot functions and in the overall graphics options using `par()`.
Hint: Make a backup of the default par() settings, such that you can reset the settings to default afterwards.
Hint: The documentations of plot, points, lines, abline, par, hist, boxplot might be helpful.

(c) Reset the graphic settings to the default values.

Exercise 2 (The layout function)
Add additional information to the scatterplot from Exercise 1 in form of boxplots of the data at the axes. For this purpose, get familiar with the function `layout()` and its options. Also use `layout.show()` to visualize different settings.

Exercise 3 (Lattice plots and grouped data)

(a) Estimate regression lines of the form

$$\texttt{mathach} = \beta_0 + \beta_1 \cdot \texttt{ses} + \epsilon$$

for the different schools separately. Plot your point estimates against the school. Use one plotting window for the intercepts and another one for the slopes. The y-axes should be labeled β_0 and β_1, respectively.
Hint: Look at the function `lmList()` from the package `nlme`. Use `expression()` for the labeling.

(b) Plot the mathematical achievement (in dependence of the `ses`) of boys as green triangles and of girls as red squares. Add a legend to the plot indicating this grouping.

(c) Load the packages lattice and `nlme`. For the school data set, we want to compare the data in the different groups. For that purpose create a groupedData object with response `mathach`, covariate `ses` and school as grouping variable. Now

- Plot the new object. What is the difference between the results from plot(`data`) and plot(`data_new`), where data is the original data set without grouping structure and `data_new` is the `groupedData` object?
- Create box-whiskers plots (`bwplot()`) of the residuals from an overall linear model according to `mathach` $\sim$ `ses` grouped by the different schools.
- Introduce a random intercept for each school and estimate a mixed model using the function `lme()`. Compare the results with your findings from (a). Try `compareFits()` and `comparePred()`.
- Plot the confidence intervals of the estimated intercepts and slopes for the different schools.