Remember to change your working directory before you start (and after the breaks)! Note that in the context of these exercises, printing means displaying values in the console of RStudio.

1 Strings

Text is stored in character vectors (character arrays) in R. Note that each element of a character vector is a whole string, rather than just an individual character.

1. Create a string vector where the first element is "red" and the second is "yellow".
2. Combine the string vector into a single string. Use the *paste()* function to combine an arbitrary string vector into a single string. Try different separators.
3. The opposite of the *paste* function is the *strsplit* function. Try the *strsplit* function for the strings created in exercise (2).
4. Print a string to the console without the quotation marks.
5. Create a vector p, that has the elements (π, π^2, π^3).
6. Create a string with special characters: tab and newline.
7. Use the function *sprintf()* to print three lines where each of the lines have the following format: "Pi to the power i is p_1", where i is the number of the line and p_i is the ith element of p with three decimals.

2 Factors

8. Create a data frame that has 8 observations of height. Simulate the heights from the uniform distribution between 140 and 200. Furthermore, simulate a gender for each of the observations such that the probability of being female is 60%. Name the variables height and gender. The data frame should then have the height (numeric) as the first column and the gender (factor) as the second.

9. Inspect the variable gender (from the data frame created above) and check the data type of it.

10. Try to replace the gender of the first observation with "unknown". What happens?

11. Create a new factor level to the data frame and then replace the gender of the first observation with "unknown".

12. Simulate a data set, where the factors are ordered. For example, the question "How happy are you?" could have the possible responses: "depressed", "grumpy", "cheery" and "ecstatic". The resulting factors have a natural ordering to them. In this case, you might want to store the responses as ordered factors.

13. Simulate 10^4 observations from the beta distribution with parameters 2 and 3. Separate your simulated data into 5 different groups, such that the intervals are of equal length.

14. How many observations are in the different groups?

3 Uploading Data, Installing Packages

15. Install the package *ggplot2* and download it into your workspace.

16. Download the file mtcars.xlsx from MyCourses and save it to your working directory. Import the data into your workspace. Use the *read.xlsx()* function. Note that the function *read.xlsx()* does not work in Linux and usually it is more safe to save your Excel spreadsheets as comma-separated-values (csv) before importing them to R.

17. Download the file mtcars.txt from MyCourses. Import the data into your workspace using the *read.table()* function. Compare the imported data set of this exercise and the one imported in exercise (16).

18. The data format you currently have is so called data frame. Transform the data data into a matrix C and perform the matrix multiplication C^TC. Data frames can be thought of as matrices where each column can store a different type of data (think of lists and vectors).

19. Create a new variable that only contains the observations with value 6 in the cyl (cylinder) column.

4 Dates and Times

Dates and times are relatively common in statistics (e.g. time series analysis). R has a wide range of capabilities for handling times and dates. The two standard date-time classes in R are POSIXct and POSIXlt. The difference between the data types is related to storing of the data. Usually, the class of the data-time object has no difference in R-programming.

20. Save the current time to a variable. Use the function *Sys.time()* and check the class of the created variable.

21. Make a different variable of the current time and calculate the time difference between the new variable and the one created in exercise (20).

22. In real life applications, the data is rarely stored in a format that is the most convenient for us. Read the data set moon.txt into your workspace and use the function *strptime()* to convert it to R time format.

23. What are the week days of the different moon landings?

5 Graphical tools

24. We continue using the mtcars data set. Make a scatter plot with the variables wt and mpg.

25. Pairwise scatter plot all the variables.

26. Create a pie chart for the number of cylinders in the different cars.

27. Rename the axis and give the plot a title.

28. Rescale the axis such that x takes values from 0 to 6 and y takes values from 10 to 40.

29. Use the function *identify* to label the points.

30. Make a heat map of the sample correlation matrix of the data set cars. If you use ggplot, the package *reshape2* might be useful.

31. Make the same plot as in exercise (24), using the command *ggplot()*. Try different shapes and sizes.

32. Download the data HW.txt from MyCourses into your workspace. Plot the variables heightIn and ageYear such that the males and females have a different color and shape in the plot.

33. Plot the variables heightIn and ageYear such that the males and females have a different shape in the plot. Furthermore, fill the points that have a weightGroup greater or equal to 100.

34. Add a linear regression line and the 95% confidence region to the heightIn and ageYear plot.

35. Add the row names (id numbers) to the previous plot.

36. Download the data FT.txt from MyCourses and make a histogram. First plot the histogram such that you have binwidth=5. After that, plot the histogram such that you have 15 bins.

37. There is something wrong with the dataset BP2.txt. Find and replace the incorrect entries in the dataset. What are the indices of the problematic elements?