

Some of the exercises contain more than what was covered by the lecture to show you some of the common and nice features of R. If you can't find the answer in the lecture slides or the help menus (or Google) I have also put out the solutions on the course web page.

Exercise 1

Getting started:

- Calculate 10 times 3
- Make a variable x equal to 5
- Calculate 10 times x

Matrix operations

- Make the matrix A equal to

```
1 2 4 4
3 5 3 4
```

- Give A columns name by using the function `colnames()`
- Add 3 to each element in the matrix and call this matrix B
- Add A and B. What do you get?
- Include a 5th column equal to [1,2] to your matrix A and call the new matrix D. Use the command `cbind()`
- Get the third column of D.

Dataframe

- There are built-in data frames examples in R. One of them are called `mtcars`
- What does `mtcars` look like? If you write `head(mtcars)` will you only see the first 6 rows.
- How big is the data frame? Use the function `dim()` or the functions `nrow()` and `ncol()`.
- What is the cell value from the first row, second column of `mtcars`?
- Could you get the same value by using row and column names instead? Which names?

Exercise 2

One of the big strengths of R is the easy and nice plotting functions. Here you can try some of them:

- Make a histogram of mpg (miles per gallon) in mtcars, use `hist()` (hint: use the “\$” sign to access mpg)
- Make a histogram of mpg in mtcars with more breaks. (hint: see the available options for the `hist()` function with `f1` or `?hist()`. Hint2: “breaks = “).
- Make a boxplot of mpg in mtcars, use `boxplot()`. (To make a boxplot for each number of cylinders use: `boxplot(mtcars$mpg ~ mtcars$cyl)`)
- Make a scatterplot of mpg in mtcars using `plot()`
- Plot horsepower (hp) versus mpg using `plot` (hint: `plot(x, y)`)

Exercise 3

It is very important that you know how to read text files from your computer into R. First, get your current working directory by typing `getwd()` and download and save *mydataframe.txt* in this location (on the wiki page).

- Read the file into the object `mydataframe` using `read.table(file = “mydataframe.txt”)`
- Inspect the dataframe by typing `mydataframe`. How does it look like? Can you spot what has gone wrong? (Everything has been put into a single column named `V1`)
- The values are separated by a comma. Try importing with this in mind. How does it look like now? Is it still something wrong? (The numbers are separated into columns, but there are `V1`, `V2` etc above the columns).
- We want to have “`treated1`”, “`treated2`” etc. as the column names. How can you fix this?
- There is still one thing missing. How many columns are there in your dataframe? Use `ncol()`. We want to have 6 columns and use the “gene” columns as the names of the rows. You can tell R which column to use as row names during import with `row.names = 1`. Try this.