Quick Introduction to R

All variables, data, functions, results etc. are stored as so-called *objects* which have their own *names* and other *attributes* such as length, dimensions (e.g. in the case of matrices), names of columns (e.g. in the case of data frame) etc.... In R we do not declare objects - they are created at the moment of their first appearance and they have a type accordingly to the type of the assigned value.

- **Types of data (modes):**

  `numeric`
  `character` - values: strings in single or double quotes
  `complex`
  `logical` - values: `FALSE, TRUE`

- **Some data structures (objects):**

  - `vector`

    Creating a vector:

    `c(...)` - combines its arguments to form a vector,
    `:` - colon operator, creates a regular sequence with an increment equal to 1, e.g. `1:4` is equal to `c(1,2,3,4)`,
    `seq()` - generates a regular sequence,
    `rep(`x`,`n`)` - creates a vector of n elements equal to x,
    `vector()` - creates a vector.

    Subseting a vector:

    `a[1]` - first element of vector `a`,
    `a[c(1,2,5)]` - first, second and fifth elements of vector `a`,
    `a[a>10]` - the elements of vector `a` that are greater than 10.

  - `factor` - categorical variable.

    Creating a factor: `factor()`.
    Function `levels()` provides information about all possible values (`levels`) of the factor.

  - `array` - table. Special case of table is matrix (`matrix`) - two-dimensional table.

    All elements of vectors, factors and arrays are of the same type.

  - `data frame` - table composed with one or several vectors and/or factors all of the same length but possibly of different modes.

    Useful functions:

    data`=data.frame(`*parameters*`)` - creates data frame named *data*,

    `data(`*data_name*`, package="`*package_name*`")` - loads data frame from `R` library,

    data`=read.table("`*file*`", header=FALSE, col.names=c(V1,V2,...), ...)` - reads a data frame from a file outside `R` and names it *data*. Here default values of some parameters are given.

    `dim(`*data_frame*`)` - dimensions of a data frame,

    Subseting data frame:

    `a[,1]` - vector made of elements contained in the first column of data frame `a`,

    `a$V1` - column (vector) named `V1` of a data frame `a`,

    `a[,'V1']` - data frame containing column `V1` of data frame `a`,

    `attach(`*data_frame*`)` - gives a direct access to names of variables in a data frame.

    `detach(`*data_frame*`)` - detaches data.

  - `list` - contains elements/objects of any type. Elements of a list may be for example: statistical models, functions, expressions...

    Creating a list: `list()`,

    Subseting a list: `a[[2]]` - second element of a list `a`,

    `a$b` - element named `b` of a list `a`.

Function `names()` gives names of the elements of a list.

Example: function `lm()` builds a linear model an returns an object which is a list containing elements such as: `coefficients, residuals, fitted.values, df.residual`, ...

- **Random numbers:**

  Normal distriubtion:

  `rnorm(`n`,`*mean*`,`*sd*`)` - random generation of n numbers

  `dnorm(`x`,`*mean*`,`*sd*`)` - density function

  `pnorm(`p`,`*mean*`,`*sd*`)` - distribution function,

  `qnorm(`q`,`*mean*`,`*sd*`)` - quantile function,

  Exponential distriubtion:

  `rexp(`n`,`*rate*`)` - random generation of n numbers

  `dexp(`x`,`*rate*`)` - density function

  `pexp(`p`,`*rate*`)` - distribution function,

  `qexp(`q`,`*rate*`)` - quantile function,

  And so on...

- **Numerical summaries:**

  `summary()` - provides basic numerical summaries of data such as minimal and maximal values, mean, median, quantiles,...,

  `mean()` - mean value of vector,

  `median()` - median,

  `range()` - returns a vector containing the minimum and maximum of all the given arguments,

  `quantile()` - sample quantiles corresponding to the given probabilities,

  `min()`, `max()` - minimal and maximal value of all the given arguments,

  `var(x)`, `sd(x)`, `cov(x,y)`, `cor(x,y)` - computes the variance of `x`, standard deviation of `x` and the covariance or correlation of `x` and `y` if these are vectors.

- **Graphics:**

  `hist()` - draws a histogram of the given data,

  `boxplot()` - produce box-and-whisker plot(s) of the given (grouped) values,

  `plot(x,y,type="p",...)` - scatterplot,

  `curve(`*function_name*`,xlim=c(a,b))` - draws a curve defined as a function in the range of argument from `a` to `b`,

  `points(x,...)` - draws a sequence of points at the specified coordinates,

  `lines(x,...)` - joins the corresponding points with line segments,

  `abline(a,b)` - adds a line with intercept `a` and slope `b` to the current plot,

  `qqnorm(x)` - produces a normal QQ plot of the values `x`,

  `par(mfrow=c(2,2))` - divides a graphics window into two rows and two colums (enables to produce several plots in one display),

  `get("windows")()` - opens a new graphics window (without closing the current one).

- **Useful commands**:

  `source("`*file_name*`")` - loads a file containing source code,

  `getwd(), setwd()` - gives the name of working directory, changes working directory,

  `library(`*package_name*`)` - loads a package,

  `?name, help(`*name*`), help("`*name*`")` - help about object named *name*

  `as.numeric(`*object_name*`)`, `as.logical(`*object_name*`)`, `as.character(`*object_name*`)`,...- conversion functions

`table(V1,V2)` - creates contingency table for variables `V1` and `V2`.
`sort()` - orders data,

`NA` - Not Available (missing data), `Inf` - Infinity, `NaN` - Not a Number