B.Sc. (Hons.) in Information Technology
Year 2 - Semester 2, 2023

Lab Exercise 1 (Introduction to R)

IT2110 - Probability & Statistics Week 01

The S language has been developed since the late 1970s by John Chambers and colleagues at Bell Labs as a language for programming with data. This provided an explicit and consistent structure for manipulating, analyzing statistically, and visualizing data.

R is an independent open source implementation similar to S, originally developed by Ross Ihaka and Robert Gentleman at the University of Auckland in the mid-1990s.

R is distributed under the GNU open software license and developed by the user community. It can be downloaded from https://cloud.r-project.org/.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Among other things it has,

- An effective data handling and storage facility,
- A suite of operators for calculations on arrays, in particular matrices,
- A large, coherent, integrated collection of intermediate tools for data analysis,
- Graphical facilities for data analysis and display either directly at the computer or on hardcopy.

R is a command line driven program and it is in itself a programming language. *"Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented."* says the developers.

Note: When installing R, **might** come across problems if there are spaces in the path name.

- **Latest Version**: 3.4.0 (You Stupid Darkness) - Released 21/04/2017.
- **Available On**: Windows, Linux, Mac (OS X).

- Terminal and GUI available.
- **IDEs for R**: R Studio, Rattle.

Basics

- The user should enter commands at the prompt (`>` by default) and each command is executed, one line at a time.
- Comments can be started with the hash sign (`#`).
- Documentation for objects & functions can be viewed by using the `help(object/function)` or by using `?object/function`.

The Workspace

The workspace is your current R working environment and includes any user-defined objects (vectors, matrices, data frames, lists, functions). At the end of an R session, the user can save an image of the current workspace that is automatically reloaded the next time R is started. **Up** and **down arrow keys** scroll through your command history.

Operators

Scaler Operators

Operator	Description
`+`	Addition
`-`	Subtraction
`*`	Multiplication
`/`	Division
`^` or `**`	Exponentiation
`%%`	Modulus (mod)
`%/%`	Integer Division

R's binary and logical operators will look very familiar to programmers. Note that binary operators work on vectors and matrices as well as scalars.

Logical Operators

Operator	Description
`<`	less than
`<=`	less than or equal to
`>`	greater than
`>=`	greater than or equal to
`==`	exactly equal to
`!=`	not equal to
`!`	NOT
`|`	OR
`&`	AND
`isTRUE(x)`	test if x is TRUE

Creating New Variables

There are three different assignment operators. The operators `<-` and `=` assign into the environment in which they are evaluated. The operator `<-` can be used anywhere, whereas the operator `=` is only allowed at the top level (e.g., in the complete expression typed at the command prompt) or as one of the sub expressions in a braced list of expressions.

Operator `<<-` is normally only used in functions, to access variables in the global environment, instead of the local environment.

<- and = use for local variables
<<- use for global variables

`<-` and `<<-` have leftwards and rightwards forms.

Packages

- Applications of R normally use a **package**; i.e., a library of special functions designed for a specific problem.
- Most predefined functions are in the base package.
- A user normally only loads a handful of packages for a particular analysis.
- To use a package, should be downloaded and installed once. Then `library(package)` loads the package to the current session.

Fundamental Data Objects

- `vector` - a sequence of numbers or characters, or higher-dimensional arrays like matrices.
- `list` - a collection of objects that may themselves be complicated.
- `factor` - a sequence assigning a category to each index.
- `data.frame` - a table-like structure (experimental results often collected in this form).
- `environment` - hash table. A collection of key-value pairs.

Classes of objects are built from these. Most commands in R involve applying a **function** to an object.

Unlike C variables do not need to be declared and typed. For example, assigning a sequence of numbers to x forces x to be a numeric vector.

Given `x`, executing `class(x)` reports the class. This indicates which functions can be used on `x`.

Vectors

In R the "base" type is a vector, not a scalar. A vector is known as an array in programming languages. A vector is an indexed set of values that are all of the same type. The type of the entries determines the class of the vector. The possible vectors are:

- numeric
- integer (subclass of numeric)
- character
- complex
- logical

Must use letter c when we are using vectors
w<-c(1,2,3,4)
In this values of vector will assign to w

can get data type of vector using class(w)

Vectors of different classes cannot be combined. Entries in a vector can be "named".

- Numeric vectors can be used in arithmetic expressions, in which case the operations are performed element by element to produce another vector.

- Missing values in a vector will be denoted by `NA`.

- Can subset with a logical vector.

Factor

Typically, in an experiment samples are classified into one of a set group of categories. In R such results are stored in a **factor**. A factor is a character vector augmented with information about the possible categories, called the **levels** of the factor. As an example,

use factor when assigning

Can get using class()

Lists

A list is an ordered collection of objects. Unlike vectors, lists can be used if we want to create collections of vectors or other data objects of mixed type. The objects in a list are known as its **components**.

Can add list of objects
can check type using class()

matrix(vector, nrow, ncol, byrow = TRUE/FALSE)
byrow can devide given data according to the number of rows and colomns

Matrix

Can be used to represent matrices in R. It is same as in vectors, all elements should be of the same type.

Data Frame

A `data.frame` object in R has similar dimensional properties to a matrix but it may contain different types of data in it. A list be made into a `data.frame` when,

- Components are vectors (numeric, character, logical) or factors.
- All vectors and factors must have the same lengths.
- Elements in a data frame is indexed like a matrix.
- Columns can be accessed using the column names as in lists (using `$` operator).
- Character vectors in a data frame are always stored as a factor.