Workshop: Data science with R

ZEW - Session #1

Obryan Poyser
2019/02/19

What's R?

- R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing.
- Is an implementation of the S programming language combined with lexical scoping semantics, inspired by Scheme.
- Created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and currently developed by the R Development Core Team. [1]
- The most common integrated development environment (IDE) is RStudio. The one we are going to use

[1] Wikipedia

Installing R and RStudio IDE

Install R base

1. Go to The R Project for Statistical Computing official website
2. Search for the link
3. Select the closest Comprehensive R Archive Network (CRAN) mirror (it won't make much difference if you choose another one)
4. Select the file according to your OS

Install IDE

1. Go to RStudio
2. Choose the RStudio Desktop Open Source License
3. Then again, the option for your specific OS
4. Execute both

R Console

Here it is possible to execute all kind of build-in functions. Give it a try

RStudio interface

One can change the layout by following (among other things):

Creating a project

RStudio projects make it straightforward to divide your work into multiple contexts, each with their own working directory, workspace, history, and source documents.[1]

[1] Rstudio

Basic commands

Getting help

Install a package

Load a package

Load a build-in dataset

Current working directory

Change current WD

Assigning values

- R is an object oriented programming language. That is, simple things such a number can be a object when is assigned to . can also be an image, model, dataset, function, etc.
- There are three types of assiggment operadors
  - which is the same as
  - and mostly used inside functions to assign values to objects in the environment.

It is possible to visualize the object either with the function or only naming it.

Classes

- Each object is identified with a class. The basic classes are:
  - Logical (True or False)
  - Numeric
  - Character or string
  - Factor
  - Complex

Then using `class()` on every object above:

Note: There are rules for naming, for example, one can not create an object with the name `c`. Try it!

Classes: Missing values

Missing values are denote by for general use, or for undefined numerical operations.

Testing could be performed with the following functions

Classes: Coercion

Objects such as `` are vectors of length 1. Vectors, as we will see further can only be composed by one type of class. Explicit coercion is achieved as:

Functions

Functions have named arguments, some can have default values. Other unnamed arguments such as can be passed to the function too. Arguments can be obtained with

Creating a function is easy, one has to create arguments then pass them to the statement.

If is not specified explicitly, then the default value is passed to the function

Functions

Note we can change the position of the arguments as long as they are specified.

However, if they are not denoted explicitly, R uses the value according to the order of the arguments

Scoping rules

In R functions is possible to define local variables. For instance, `` does not exist outside the function `` .

- If a variable at a given run-time:
  - Avoid renaming of the variables in the environment
  - All potentical confussions
- Scoping rules treats about how R manage variables, that is, variables that are not defined inside the function.

The `` does not find `` inside, looks outside, and neither is there. Hence, if we intentionally create a variable `` , then:

Scoping rules

Pro-tip 😎

Using the special assigning symbol inside a function makes any variable scale to the environment. For example, does not exist in the current environment.

But if we purposely try a second into the following function:

Vectors

Creating vectors is easy with . Also, the function stands for concatenate, and is used for aggregating elements into a single object.

[redacted]

Vectors . If a given vector is initialy composed by a mixture of elements, R automatically apply coercion rules to homogenize the vector.

[redacted]

[redacted]

Mathematical operations

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

Selecting elements (subsetting) vectors

By position

By coincidence

For this, we have to take into account logical conditions

Matrices

Vectors are one dimensional array, while matrices are two-dimensional. A matrix can be created as:

[redacted]

[redacted]

[redacted]

[redacted]

As any matrix, in R a matrix has a size $m \times n$, where m is the number of rows and n the number of columns. Subsetting a matrix is fairly simple, one can only has to index a second element of interest divided by a comma inside the squared bracket.

Lists

A list is a set of elements which can be of different types.

Empty list

[redacted]

List of one element

[redacted]

List of several elements (unnamed)

[redacted]

Lists

Since lists admit various types of types or even objects, it is possible to name them.

Subsetting of list can be performed by several ways. It will depend on the structure, for instance, list can be nested into other lists. The function shows the structure of the objects.

Lists

Understanding lists can be a pain in the arse. Mainly because complex nested configurations and their first order internal structure.

For instance, a list can allocate a from the past slide and a dataset

Subsetting lists

Option #1 with [redacted]

[redacted]

Option #2 with a [redacted] (💡 lists need to be named in order to use this alternative)

[redacted]

Subsetting lists

Option #3

It is possible to subset a list partially

For example, in order to access to the first column of the dataset one could write

Dataframes

Dataframes are a special type of list in which each column represents a vector of elements associated with a row, they resemble the typical dataset that we use in Stata.

Dataframe and matrix share several features. Depending on the task (graphics or regression) we have to covert one to the other.

Useful functions

Environment

Numeric

Character or factor

Next session

1. Read and write data frames 1. CSV

    1. Excel
    2. Stata, SPSS, etc.

2. Control structures and loops

    1. If statements
    2. While loop
    3. For loop