Introduction to R

Sean Hellingman

Data Visualization and Manipulation through Scripting (ADSC1010)

shellingman@tru.ca

Fall 2024

THOMPSON RIVERS UNIVERSITY

Topics

2. Introduction
3. User Interface
4. R packages
5. Operations in R
6. Workspace and Important Base Functions
7. Getting Help
8. Exercises and References

Introduction

- R is a *programming language for statistical computing and graphics*
  - R is software is open source unlike some other statistical software.
  - R can be downloaded at `https://www.r-project.org/`.
- RStudio *is an integrated development environment for R*
  - RStudio will be used in this course.
  - Can be downloaded at `https://posit.co/download/rstudio-desktop/`.
  - Please install R before RStudio, otherwise RStudio will not work.

User Interface I

- R Console
  - Can give R direct instructions via the R console.
- R script
  - A text file containing a set of commands and comments.
  - Can create an R script, with the file extension '.R'.
  - Used to run a sequence of commands in R.
  - Other types of scripts such as R markdown files can be created.
  - Good idea to **save** your script or R markdown file periodically.
- Environment
  - Where variables are created and stored.
  - Serves as a *Workspace*.

User Interface II

- History
  - Stores the code that has been ran.
  - Next to the Environment tab.
- Working Directory
  - R scripts are stored here by default.
  - Can be important when importing data.
  - Permanently change working directory:
    Tools ⇒ Global Options ⇒ General ⇒ Default working directory
  - Temporarily change working directory:
    Session ⇒ Set working directory ⇒ Choose Directory
  - The working directory resets to the default when RStudio restarts.

Base R & R packages

- Base R
  - Contains some basic functions and visualization tools.
  - The base R functions contain only a tiny fraction of what is actually available.
- R packages
  - A package is essentially a suite of related functions that were created by one or more R users.
  - Currently over 19000 R packages avalibe at `https://cran.r-project.org/`.
- Some packages we will use:
  - *dplyr* for data manipulation.
  - *tidyr* for reshaping data.
  - *ggplot2* for creating visualisations.

Initializing R packages

- Use `install.packages("package_name")` in the console to install R packages on your own machine.
  - Example: `install.packages("ggplot2")` to install ggplot2.
- Use `library(package_name)` to initializing already installed packages.
  - Example: `library(ggplot2)` to load ggplot2.
- To install the package if it isn't already installed and then initialize it after installation:

```
if(!require(package_name)) install.packages("package_name")
library(package_name)
```

Arithmetic Operations

- Order of arithmetic operations (BEDMAS) applies in R.
- Basic operations:

Operation	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
∧	Exponent
%%	Modulus (Remainder from division)
%/%	Integer Division

- # is used to add comments to R scripts.

Relational Operators

- Relational operators are used to compare between values:

Operation	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Assigning Variables

- The operators =, <-, and -> can be used to assign values to variables
  - If we want to assign `x = 2`; `x=2`, `x<-2`, and `2->x` are equivalent.
  - Common convention is to use <- but feel free to use =.
- Variable naming conventions
  - Try to use meaningful names.
  - Periods and underscores are allowed in variable names.

Displaying Output

- R will display the outputs of unassigned calculations
  - `exp(3)` would compute e^3 (approximately) and display the result.
- R suppresses the output of assigned calculations
  - `x<-exp(3)` would compute e^3 and store the result in `x` but would not display anything.
  - To see that value of `x` we would need to run `print(x)` or just `x`.

Workspace (Environment)

- Use `ls()` or `objects()` to generate a list of objects in the workspace.
- Use `rm()` to remove/delete objects from the current workspace
    - The call `rm(x,y,z)` will remove the variables `x`, `y`, and `z` from the workspace.
    - The call `rm(list=ls())` will remove all objects from the workspace.
    - Can also click the *broom* symbol at the top of the workspace to remove all of the objects.
    - The workspace is automatically cleared when RStudio is exited.

Some Important Functions

Function	Description
`exp(x)`	Computes the exponential function e^x.
`log(x,base=exp(1))`	Computes the logarithm of `x`, by default natural logarithm (e).
`sqrt(x)`	Computes the square root of a numeric data object.
`factorial(x)`	Computes the factorial of a positive integer `x`.
`sin(x)`	Computes the sine of `x` in radians. (similar for `cosine(x)` ect.).
`abs(x)`	Computes the absolute value of `x`.

Useful R Resources

- Use `?` to learn more about how a function works.
  - For example, `?sqrt` would produce a brief explanation of the `sqrt` function in a separate window.
  - You can also go to the Help tab and search for the function or command you need help with.
- It may be more useful to use Google
  - `https://stackoverflow.com/`
  - `https://www.r-bloggers.com/`

Exercise 1

- What are the final values of `x` and `y` after following this sequence of commands:
  - `x <- 6`
  - `x^2`
  - `x <- 2*x-1`
  - `x + 3`
  - `x <- -3*x`
  - `y <- -x/2`
- Next, use R to confirm your answers.

References & Resources

- Douglas, A., Roos, D., Mancini, F., Couto, A., & Lusseau, D. (2023). *An introduction to R*. Retrieved from https://intro2r.com/