**RRA R Orientation**

Has your mentor and/or advisor told you to learn R? Do you need to learn R and need to learn it now? Well, you’re in luck!

RA R Orientation is dedicated to students who know nothing about R, but need to get started. The curriculum for this short course will get you going with the basics as well as how to use R Studio to manage projects.

**Table of Contents**

Page 3: What is R?

Page 4-5: Working with RStudio

Page 6: Creating a GitHub Account

Page 7: Descriptive Statistics Commands List

Page 8: Plotting and Graphing Commands List

Page 9-10: Working With Data

Page 11-12: Markdown Files

Page 13: Downloading Packages

Page 14: Using the Help Function

Page 15: Practice Exercise

**What is R?**

R is a coding language and environment for statistical computing and graphic. R provides a large variety of statistical and graphical functions. One of the biggest strengths of the R software is that it is very well-designed which makes it easy to produce quality plots, statistics, and formulae. R is a free software that is available under the Free Software Foundation’s GNU General Public License. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

***Functions of R***

* Handling data and storage.
* Can provide a large variety of calculations.
* Collection of tools for data analysis.
* Many graphical functions for data analysis.
* Programming language

***Downloading R***

Windows: <https://cran.r-project.org/bin/windows/base/>

MacOS: <https://cran.r-project.org/bin/macosx/>

**Working with RStudio**

RStudio is a must-know application for anyone wanting to work with the R programming language. R studio helps to import, access, transform, plot, and make predictions on data analysis. RStudio integrates with R as an integrated development environment to provide further functionality and organization. RStudio provides the ability to write and save reusable scripts, import data, create objects, preview plots, share your work with collaborators, track your history, code, and receive many sources of help on any topic.

***Downloading RStudio***

1. First download the R programming language
2. Visit the download page of the [RStudio Website](https://posit.co/download/rstudio-desktop/).
3. Scroll down and download the correct version of RStudio that matches your computer software.

***How to Use RStudio***

Three working windows:

* Full left side: Console, Terminal, and Background Jobs
* Top Right: Environment, History, Connections, and Tutorial
* Bottom Right: Files, Plots, Packages, Help, Viewer, and Presentation

***Full Left Side:***

* Console: Install packages, perform mathematical operations, import data, explore data, statistical analysis, build data visualizations
* Terminal: Helps run commands
* Background Jobs: R script that runs in a separate, dedicated R session.

***Top Right Side:***

* Environment: Provides meta-project information such as what values you have stored in variables and your custom functions.
* History: Tracks all history during the current session.
* Connections: Helps to connect to existing data sources.
* Tutorial: Set of tutorials that help with the basics of R and teaches you how to use all the tools.

***Bottom Right Side:***

* Files: See the structure of the working folder and navigate between other folders.
* Plots: Preview and export data visualizations.
* Packages: Access to packages intended to help you navigate RStudio easier.
* Help: Provide help for R functions, data sets, and other objects.
* Viewer: Used to view local web content
* Presentation: Enable easy authoring of HTML5 presentations using a combination of Markdown and R.

For Further Help:

[R Studio Reference Guide](https://www2.stat.duke.edu/courses/Spring14/sta101.001/UsersGuide.pdf)

**Creating a GitHub Account**

**What is GitHub?**

GitHub is an online software platform that is useful for tracking, storing, and collaborating on different software projects. GitHub makes it easy to share code files and collaborate with other developers. GitHub users create accounts, upload files, and create coding projects. You can create coding files in RStudio and export these files into GitHub where you can collaborate and edit your files.

**How to create an account?**

1. Visit [https://github.com/.](https://github.com/)
2. Click the Sign-up button in the top right corner.

A screenshot of a computer

Description automatically generated

1. Follow the following prompts to sign up.
2. For further help: [GitHub Reference Guide](https://www.freecodecamp.org/news/introduction-to-git-and-github/)

**Descriptive Statistics Commands List**

|  |  |
| --- | --- |
| summary(x) | computes the 5-number summary and the mean of the variable x |
| mean(x) | computes the mean of the variable x |
| sd(x) | computes the standard deviation of the variable x |
| var(x) | computes the sample variance of variable x |
| min(x) | returns the minimum value in some vector x |
| max(x) | returns the maximum value in some vector x |
| length(x) | used to get the length of the vector x |
| median(x) | computes the median of the variable x |
| range(x) | the highest value minus the lowest value |
| t.test(x) | get a one sample t-test |
| t.test(x,y) | get a two-sample t-test |
| cor(x,y) | computes the correlation coefficient |

**Plotting and Graphing Commands List**

|  |  |
| --- | --- |
| plot (x, y) | Scatterplot |
| hist () | Histogram |
| boxplot () | Boxplot |
| barplot () | Bar Diagram |
| dotplot () | Dot Diagram |
| piechart () | Pie Chart |
|  |  |
| Plotting Elements: |  |
|  |  |
| lines () | Lines |
| abline () | Regression Line |
| points () | Points |
| arrows () | Arrows |
| box() | Frame around plot |
| title() | Title above plot |
| text() | Text in plot |
| mtext() | Text in margin |
| legend() | List of symbols |
| xlim, ylim | Plot limits |

**Working With Data**

***Importing***

R gives you the ability to load data from outside sources\ into the software. Most common data types used in R come from a CSV or TXT file. These steps will help you import data into R using a CSV file:

1. Find a CSV file in which you would like to analyze.
2. First, set the directory where the file is by using the setwd ("/…") function. If you’re a windows user, replace every "\" in the file path with "\\".
3. To import the CSV file, we will use the readr package’s `read.csv` function.
4. Example of the code:

A close-up of a computer code

Description automatically generated

1. **file.choose():** Lets you choose a CSV file from the desktop.

***Exporting***

You can work with data sets in R and export them to different programs. These few steps will help you export data files:

1. Set the directory where the file will go by using setwd("/…").
2. For .csv files, you can just use the write.csv function. That will look like this:

write.csv( dat,"data1" ) #file path can be added also.

***Saving***

As you work with data in R, it is important to save it. This will enable you to work with the data later. It will also help you in the process of sharing your data with other analysts. Following this sentence are the few steps to saving files in R:

1. Set the directory where the file will go by using setwd("/…"). Then:
2. To save the workspace use the save.image() function.
3. These functions will guide you through the saving process:

save.image( "dat.RData" ) #file path can be added also.

rm( list=ls() ) # clear the workspace.

ls() # check the contents of the workspace.

load( "dat.RData" ) #load the workspace that was previously saved.

**Markdown Files**

An R Markdown file allows you to write all the text, syntax for the analysis, and keep it in a document. This helps keep the file in the same program. Before you work with Markdown, you must download the markdown package using “rmarkdown.”

Ex: install.packages( "rmarkdown" )

After this, open a markdown file using the drop-down menu:

A screenshot of a computer

Description automatically generated

Once you click on the markdown file, you can then use the options to select what type of file you want to create. Then, you click on the knit function in the .Rmd pane:

A screenshot of a computer

Description automatically generated

**Some helpful guides:**

* [R Markdown Cheat Sheet](https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf)
* [R Markdown Reference Guide](https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf)

**Downloading Packages**

R has a numerous amount of different packages available. R has many preprogrammed functions that are automatically loaded when you open the program. Functions are stored in “packages”. Although there are many preprogrammed functions, there are even more functions that you can install on your own. A package in R is a collection of functions, usually written for a specific purpose. We can see the packages available from CRAN. Packages can be found at <https://cran.r-project.org/web/packages/index.html>.

**Steps:**

1. If there is a particular package you want to add, you simply use the install.packages() function like this: install.packages("package name").
2. Load the package using the library() function like this: library(package name).
3. when you use the install.packages() function the package name needs to be in "", but not for the library() function.
4. install.packages( "sna" )
5. library( sna )
6. install.packages( "tnet" )
7. library( tnet )
8. Each time you open R you have to load any packages that you manually loaded using the install.packages() function.
9. Note that we do not have to re-install the package using install.packages(), we just have to load the library. You will only need to re-install when there is a major update to R and you have to download the new version.

**Using the Help Function**

One of the most useful features of R is the access to help that is provides. To access the main R help, archive, type: help.start(). The help() function, or a simple ?, can be used to get help about a specific function. For example: help(c) or ?c returns the help page for the c() function.

Extra Help: [Getting Help With R](https://www.r-project.org/help.html)

**Practice Exercise**