Introduction to R
Data Science Skills Day 2022

Anjali Silva, PhD

Summer Undergraduate Data Science Research Program
University of Toronto
03 June 2022

a.silva@utoronto.ca anjalisilva.github.io @Silva_Anjali

Material

- Lesson Material:
  - http://swcarpentry.github.io/r-novice-inflammation/
- Slides:
  - https://github.com/anjalisilva/DSI_IntroductionToR
  - SlideIntroR2022.pdf
- R Script:
  - https://github.com/anjalisilva/DSI_IntroductionToR
  - Script.R

Welcome!

- Instructor: Anjali Silva, PhD
  - Researcher and Lecturer, Department of Cell & Systems Biology, U of T
  - Assessment Data Analyst, University of Toronto Libraries
  - Pronouns: she/her
  - Name Phonetic: Un-j-li Sil-va
  - Hear Name Pronunciation: https://namedrop.io/anjalisilva

Course

- Introduction to R
  Data Science Skills Day

  - The vast amount of data produced by evolving information technology requires tools and skills. Among the many tools, R is a free, open-source language for data sciences. R is a programming language that can aid in the process of data analysis. This course is a beginner level, introductory course for R for data analysis. We will learn about R, RStudio (the environment use to work in R), including installation, and apply R for beginner-level data modeling and visualization. By the end of the course, you'll have a introduction to the flexibility of R, different functionalities, and understand how to apply it for basic data exploration.

  - Friday 10:00 am – 4 pm EST; online - synchronous.

Course

- Introduction to R
  Data Science Skills Day

  - Learning objectives:

  - Install R and RStudio
  - Navigate the RStudio environment
  - Discover how to use RStudio to apply R to your analysis.
  - Importing data from a spreadsheet
  - View attributes of a dataset
  - Understand differences in varying data types and structure
  - Write and test functions
  - Generate simple visualizations
  - Be aware of sources for getting help in R
  - Be aware of sources for expanding skills in R

Outline

- Introduction
- Setup
- Introduction to RStudio
- Analyzing Patient Data
- Data Types and Structures
- Creating Functions
- Next Steps and Final Remarks

Course Expectations

- Be respectful.
- Keep yourself muted, unless you need to speak or ask a question.
- You may save your questions to 'Any questions?' section.
- If you have a question, raise hand. Before speaking, say your name.
- If you have a question, you may type it to chat.

Any questions?

Let's begin the lecture...

Any questions?

R

What is R?

- A language and environment for statistical computing and graphics.
- R was initially written by Ross Ihaka and Robert Gentleman.
- Since mid-1997, the R Core Team modify the R source.
- R runs on a wide variety of UNIX platforms, Windows and MacOS.

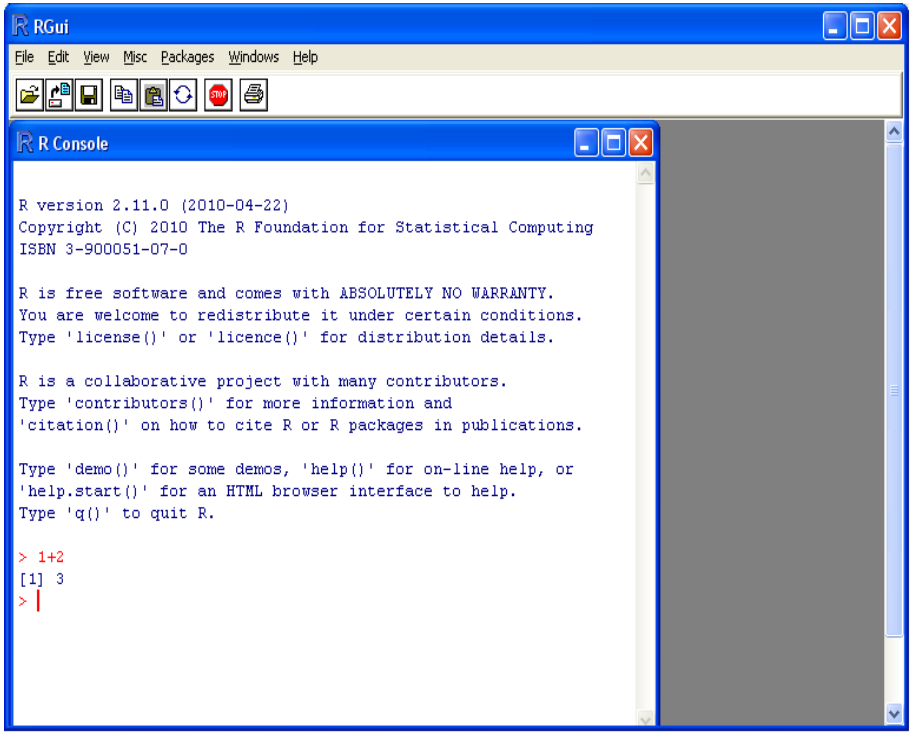
R continue...

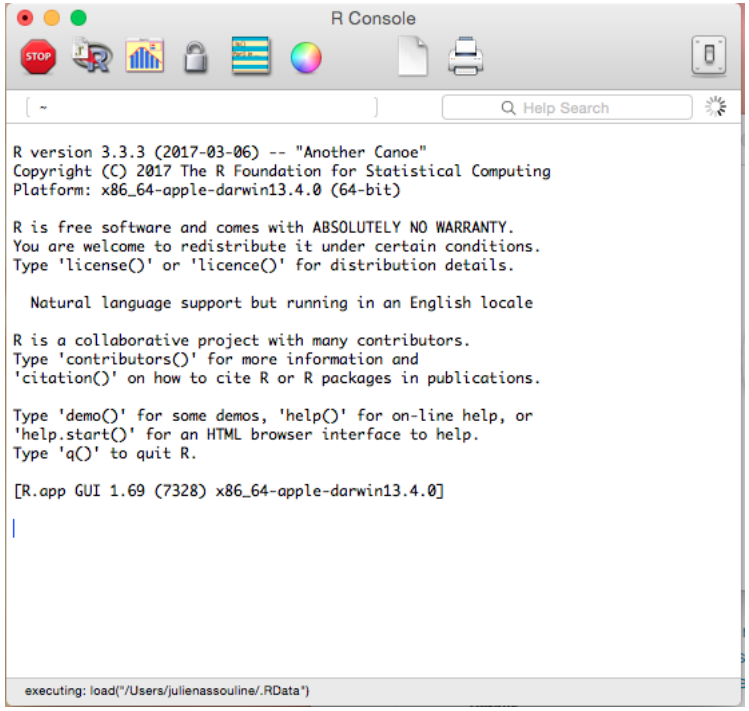
- R is a scripting language, thus an interpreter executes commands one line at a time.
- A Free software under the terms of the GNU General Public License.
- R home page: https://www.R-project.org/
- How can R be obtained?
  - Via CRAN, the "Comprehensive R Archive Network".
  - https://cran.r-project.org/

R continue...

- How can R be installed?
  - Unix
  - https://cran.r-project.org/doc/FAQ/R-FAQ.html#How-can-R-be-installed-_0028Unix_002dlike_0029
  - Windows
  - https://cran.r-project.org/bin/windows/base/
  - Mac
  - https://cran.r-project.org/bin/macosx/

R continue...


RGui
File Edit View Misc Packages Windows Help
R Console
R version 2.11.0 (2010-04-22)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
> 1+2
[1] 3
>


R Console
Help Search
R version 3.3.3 (2017-03-06) -- "Another Canoe"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[R.app GUI 1.69 (7328) x86_64-apple-darwin13.4.0]
executing: load("/Users/julienassouline/.RData")

R continue...

- R can be used interactively or non-interactively.
- Interactively, with or without an integrated development environment (IDE): RStudio.
- Non-interactively via scripts.
- R is designed with interactive data exploration in mind.
- A version of R is released each year. Current release is 4.0.2.

Documentation for R

- Online documentation for functions and variables in R exists.
- Obtained by typing *help(FunctionName)* or *?FunctionName* at the R prompt, where FunctionName is name of function.
- E.g., if 'sum' is the function then:

```
> help(sum)
> ?sum
```

R packages

- Mechanism for extending the basic functionality of R.
- It is natural to put together many functions together into a package achieving a specific goal.
  - Function for preprocessing data.
  - Function for clustering data.
  - Function for selecting best cluster.
  - Function to visualize the clustering results.
  - Put together = Package for Clustering.
- Provide a defined interface, with inputs (arguments) and outputs (return values).

R packages

- Building R packages requires tools that must be in place before process of development can start.
- Mainly R and RStudio (recommended).
- Mac OS
  - Xcode development environment
  - https://apps.apple.com/us/app/xcode/id497799835?mt=12
- Windows
  - Rtools
  - https://cran.r-project.org/bin/windows/Rtools/

R packages: Mac OS

- For more information: https://r-pkgs.org/setup.html
- Mac OS
  - Xcode development environment
  - https://apps.apple.com/us/app/xcode/id497799835?mt=12
- Then, in the shell, do:

```
xcode-select --install
```

R packages: Windows

- Windows:
  - Rtools
  - https://cran.r-project.org/bin/windows/Rtools/
- For more information: https://r-pkgs.org/setup.html
- During the Rtools installation you may see a window asking you to "Select Additional Tasks".
  - Do not select the box for "Edit the system PATH". devtools and RStudio should put Rtools on the PATH automatically when it is needed.
  - Do select the box for "Save version information to registry". It should be selected by default.

R packages: Linux

- For more information: https://r-pkgs.org/setup.html
- Install R, but also the R development tools. For example, on Ubuntu (and Debian) you need to install the r-base-dev package.

What R packages are available?

- CRAN
  - >16K packages [as of 2022]
  - https://cran.r-project.org/web/packages/
- Bioconductor
  - >1900 packages [as of 2022]
  - https://bioconductor.org/packages/release/bioc/
- GitHub
  - > 63K results [as of 2022]
  - https://github.com/search?q=r+packages&type=Repositories

RStudio

- RStudio is not required to build R packages.
- However, it contains many features that make the development process easier and faster.

RStudio

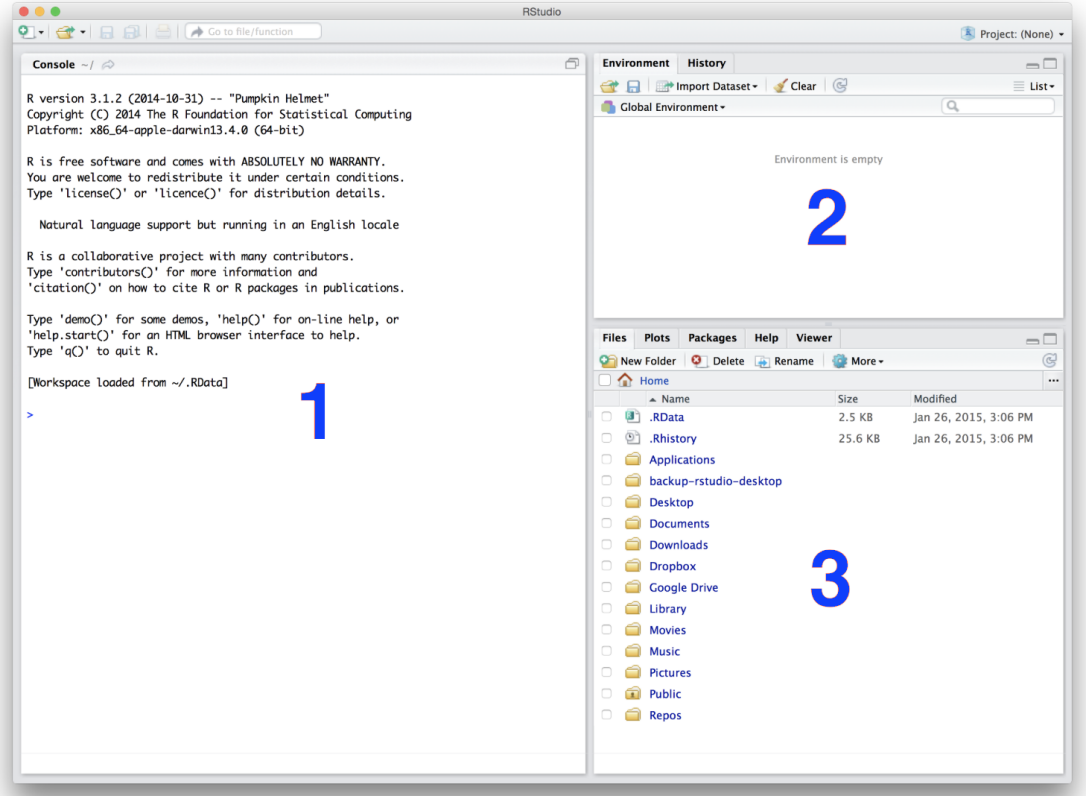


Figure: Anatomy of RStudio. 1. This is the Console. 2. Environment and History. 3. Files, Plots, Packages, Help and Viewer.

RStudio

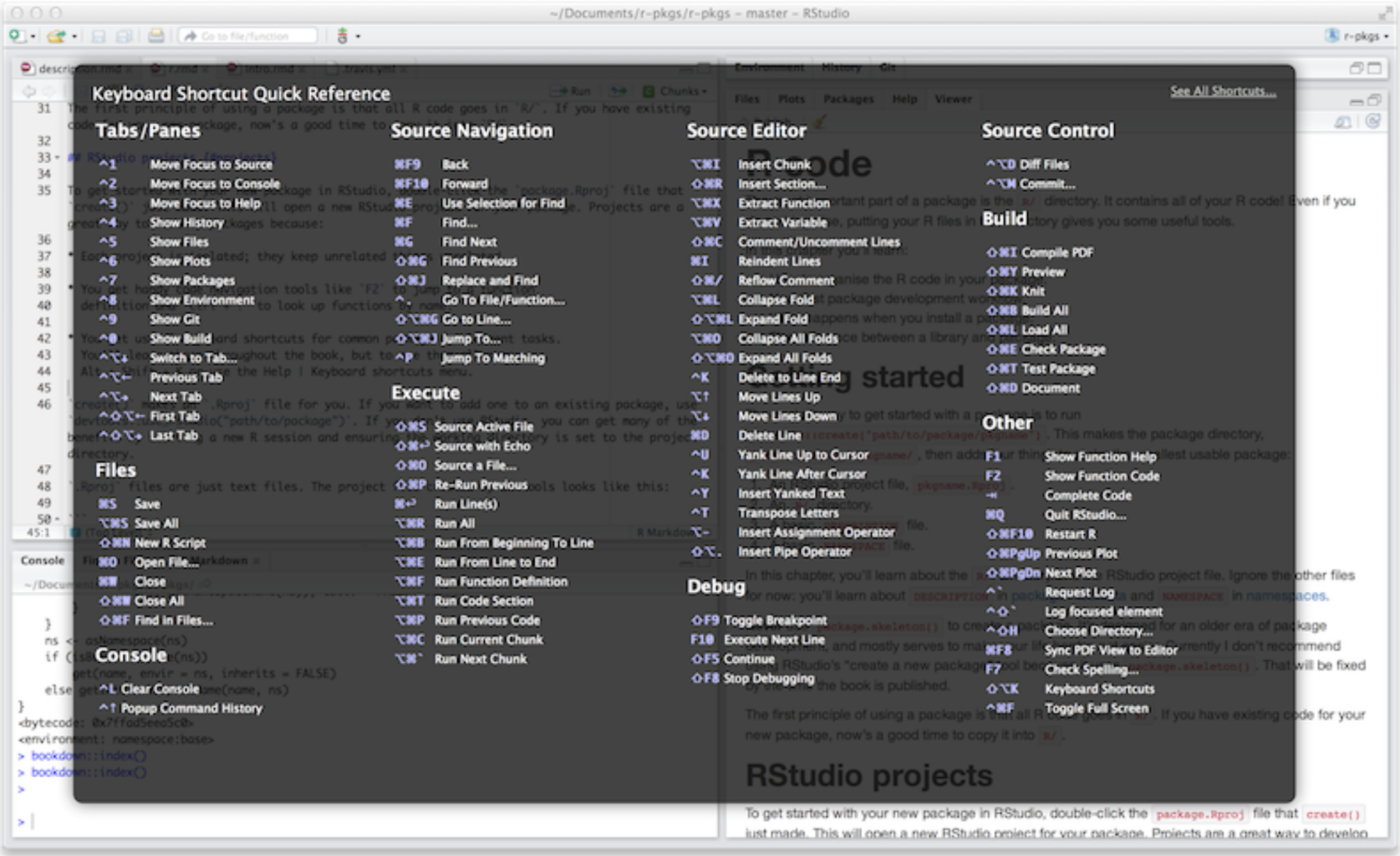


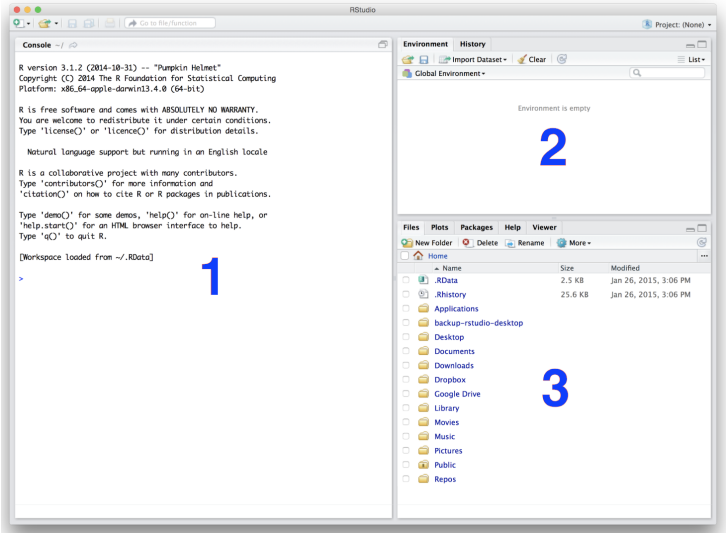
Figure: Tools → Keyboard Shortcuts Help.

Any questions?

Practical

RStudio

- Let's open up RStudio.



- On Console, get working directory:

  > `getwd()`

- To set to desired directory
  Session → Set Working Directory → Choose Directory...

RStudio

- Alternatively, you may use:

  ```
  > setwd("/../../..")
  ```

- To open a new script:
  File → New File → R Script

- Save this:
  File → Save → Practical_StudentName.R

- Practical_StudentName.R is called a script.

R Features

- In R, the indexing begins from 1.
- R is case sensitive ("X" is not the same as "x").
- R uses dynamic variable typing, so variables can be used over and over again.

Assignment and Commenting

- The ← symbol is the assignment operator.
- To assign a value to a variable called 'test1'

```
test1 <- 123
test1
```

- Comment using # character

```
test1 <- 123  # This is a comment
test1 # This is called auto-printing
```

Over-writing

- From previous slide we had:

```
test1 <- 123
test1
```

- Over-write previous value of the 'test1' variable with a new value:

```
test1 <- test1 + 2
test1 # 125
```

- Over-write previous value of the 'test1' variable with a new value:

```
test1 <- 5 + 2
test1 # 7
```

Version

- To obtain session information

  `sessionInfo()`

- Version information:

  `R.Version()`

- Show objects in workspace

  `ls()`

R Built-in Functions

- There are many built-in functions. You will learn these as you go.
- The "argument" of the function is provided inside the brackets.
- The "return value" of the function is the value provided back.
- We will cover some basic functions:

```
x <- 5
x # auto-printing
print(x) # explicit printing
class(x) # "numeric"
typeof(x) # "double"
length(x) # 1
```

R Built-in Functions

- Return value from functions can be assigned to a variable or printed:

```
x <- 5
x # auto-printing

y <- x + 5
y # 10

z <- tyepof(y) # return value  assigned to variable
z # "double"
```

R Help Function

- Getting help:

```
?"<-"  # help on assignment operator
help("<-") # help on assignment operator

?typeof # help on typeof function

?class # help on class function

?print # help on print function
```

Any questions?

R Data Types

- Numeric: floating types (double precision).
- Logicals: booleans = TRUE/FALSE or T/F.
- Character strings.
- Examples:

```
xValue <- 100
xValue

yVariable <- FALSE
yVariable

zVariable <- "hello"
zVariable
```

R Class

- Numbers in R are usually treated as numeric objects (i.e. double precision real numbers).
- To explicitly assign an integer, need to specify the L suffix.

```
x <- 1L
x
class(x) # "integer"
```

R Class

- Complex class:

```
x <- c(2 + 0i, 5 + 4i)
class(x) # "complex"
```

- Inf represents infinity:

```
Inf
1 / Inf # 0
```

- NaN represents an undefined value/missing value:

```
NaN # not a number
0 / 0 # NaN
```

Concatenating

- c() function concatenating elements together:

```
x <- c(0.5, 0.6)
class(x) # "numeric"

x <- c("a", "b", "c")
class(x) # "character"

x <- c(TRUE, FALSE)
class(x) # "logical"
```

Character Strings

- Character strings are collections of characters.
- Provided as values in single or double quotes.

```
xVariable <- ‘hello’
class(xVariable) # "character"

zVariable <- "hello"
class(zVariable) # "character"
```

- “paste” converts inputs to strings, concatenate and return:

```
paste(xVariable)
```

Character Strings

- "cat" concatenates and prints the arguments to the screen:

```
cat("\n", xVariable, zVariable) # "\n" adds new line
```

- "print" prints the argument:

```
print(c(zVariable, xVariable))
```

Missing Values

- Missing values are denoted by NA (Not Available) or NaN (Not a Number).

```
x <- c(1, 3, NA, 4, 5)
class(x) # "numeric"

y <- c(1, 3, NaN, 4, 5)
class(y) # "numeric"

# is.na() is used to test objects if they are NA
# is.nan() is used to test for NaN

is.na(x) # FALSE FALSE  TRUE FALSE FALSE
is.nan(x) # FALSE FALSE FALSE FALSE FALSE
```

Question: What is the difference between NA and NaN in R?

Any questions?

- To do: Journal Entry 1 (Note, may need a distribution of Latex installed).
- Take a look at 'Initial submission + Presentation of R package'.

Practical

- Today we looked at the following topics.
  - Assignment and Commenting
  - Over-writing
  - Built-in Functions
  - Help
  - Classes
  - Concatenating
  - Character Strings
  - Missing Values

Practical - Tips for Solving Issues

- Copy and paste the entire **exact** error message into Google.
  - Someone else may have gotten this same error and has asked a question.
- Copy and paste the entire error message into Google, followed by 'r'.
- Google the name of the function with term 'tutorial r' to see tutorials.
- If struggling with code for a plot, Google 'r plot plotname', then click on Images.
- If errors with reading files, ensure path is correct. Check using getwd().