R03: R Basics, part 2

Meike Niederhausen and Nicky Wakim

2024-10-07

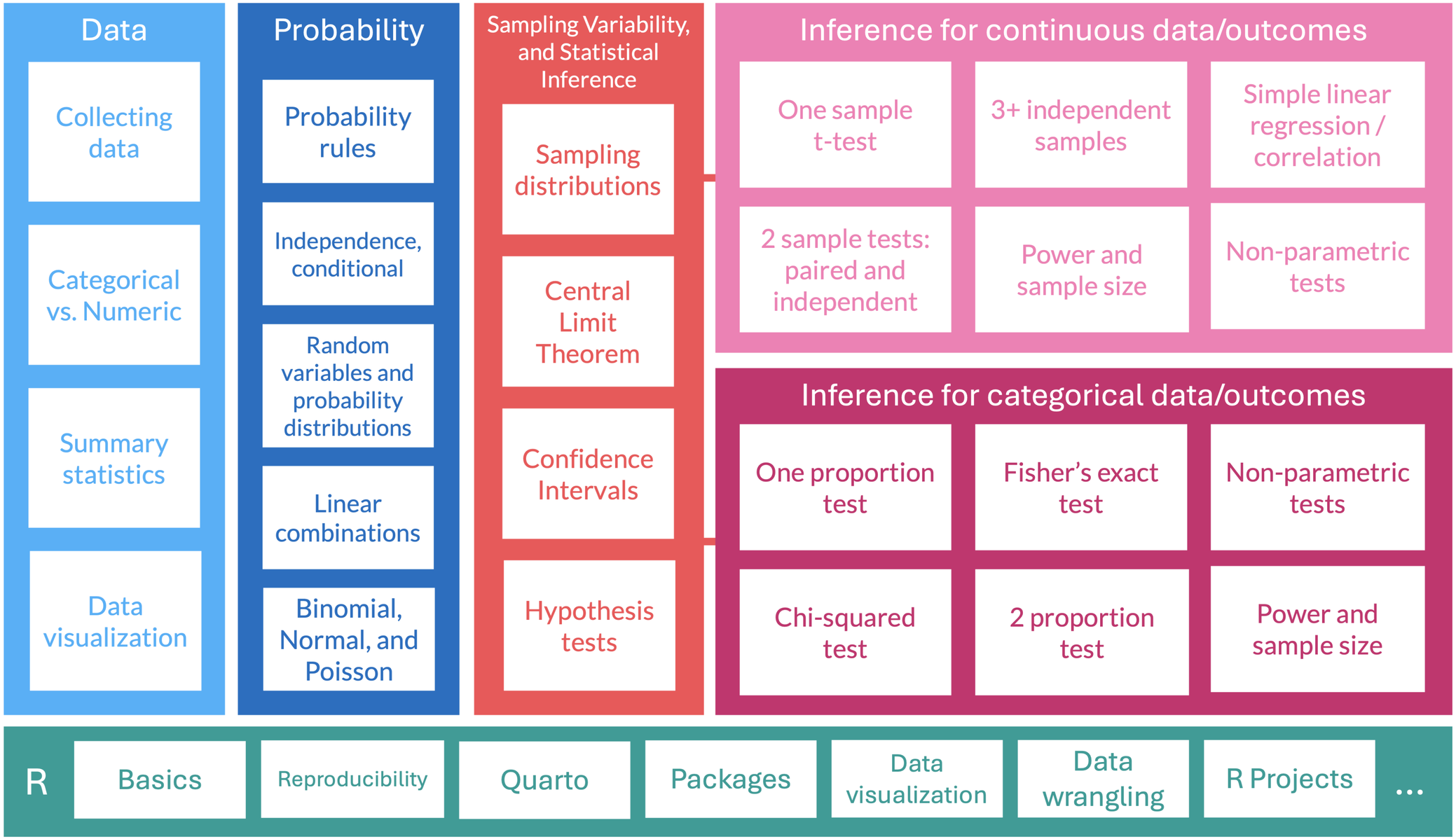
Last time

- Downloaded R and Rstudio
- If you are still having issues downloading, please come to my office hours!
- Became familiar with the console and a script file
- Did some math calculations in R!

Today, we're going to work on...

- Assigning things in R
- Using functions to calculate
- Familiarizing ourselves with common issues
- Troubleshooting with different tools
- Go over an example dataset

Where are we?


Data
Collecting data
Categorical vs. Numeric
Summary statistics
Data visualization
Probability
Probability rules
Independence, conditional
Random variables and probability distributions
Linear combinations
Binomial, Normal, and Poisson
Sampling Variability, and Statistical Inference
Sampling distributions
Central Limit Theorem
Confidence Intervals
Hypothesis tests
Inference for continuous data/outcomes
One sample t-test
3+ independent samples
Simple linear regression / correlation
2 sample tests: paired and independent
Power and sample size
Non-parametric tests
Inference for categorical data/outcomes
One proportion test
Fisher's exact test
Non-parametric tests
Chi-squared test
2 proportion test
Power and sample size
R
Basics
Reproducibility
Quarto
Packages
Data visualization
Data wrangling
R Projects
...

We will open RStudio on our computer (not R!)

1.1.2 Using R via RStudio

Recall our car analogy from earlier. Much as we don't drive a car by interacting directly with the engine but rather by interacting with elements on the car's dashboard, we won't be using R directly but rather we will use RStudio's interface. After you install R and RStudio on your computer, you'll have two new *programs* (also called *applications*) you can open. We'll always work in RStudio and not in the R application. Figure 1.2 shows what icon you should be clicking on your computer.

R: Do not open this	RStudio: Open this
	

FIGURE 1.2: Icons of R versus RStudio on your computer.

RStudio anatomy

BuzzR

RStudio anatomy

https://buzzrbeeline.blog/

Emma Rand

Script file

Write code here

To run code put your cursor on the line and click the run button

Edit to correct errors

⇒ record of commands that worked

Save scripts with the **.R** extension

⇒ syntax will be highlighted

⇒ good practice

<- is the assignment operator

⇒ puts what is on the right in to the object on the left

⇒ Assign results if you want to use them again

Console

When you click run, code is sent to the console and executed

> is the prompt

⇒ do not type it

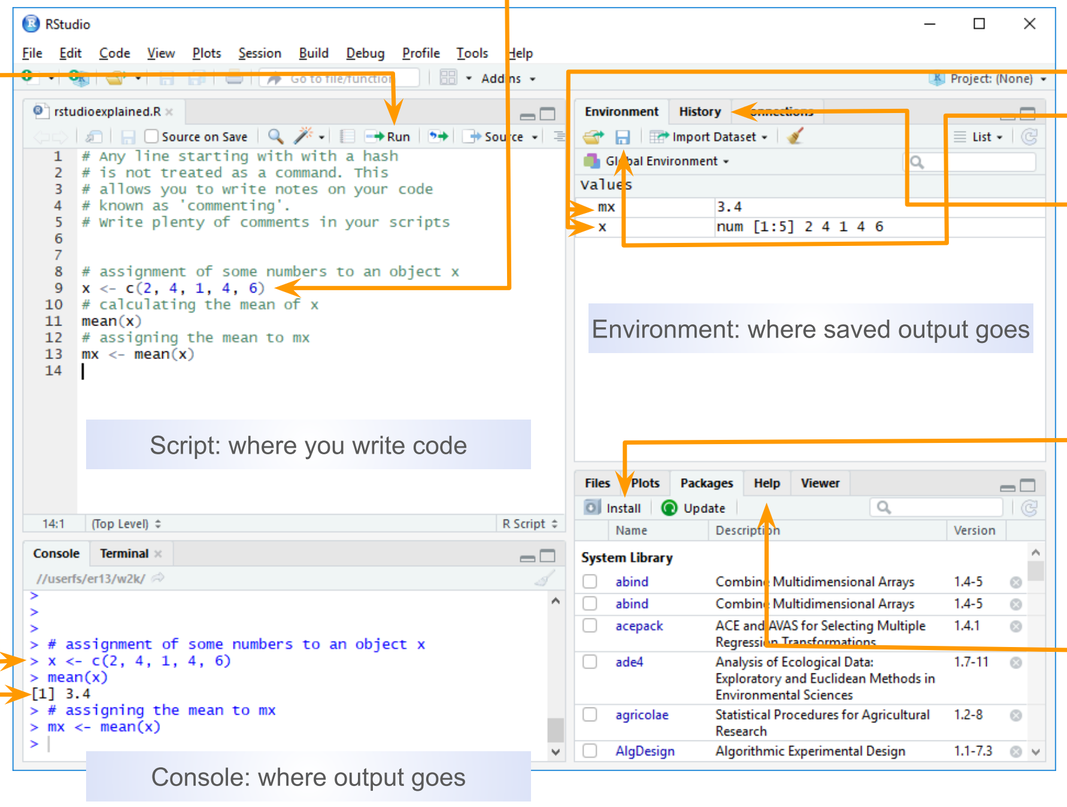
⇒ appears when R is ready for next command

Command output goes here by default

⇒ output is in a different colour

⇒ [1] indicates 3.4 is the first element of the output

⇒ many commands will not have output, the prompt just reappears



Environment

Name objects by assignment to use them again

All the objects you created in your session

Saving the environment saves all the objects, but not the code with a **.RData** extension

History

A history of every command you sent to the console, mistakes included.

File can be saved but usually you just need the script

Packages

Many functions come with R

A huge amount of extra functionality is available in packages

Packages can be installed by clicking the Install button

Help

Access to manual pages for all installed packages

Plots

Figure output appears here

Variables (saved R objects)

Variables are used to store data, figures, model output, etc.

- Can assign a variable using either = or <–
  - **Using <– is preferable** for certain occasions
  - I usually just use = because less typing hehe

Assign just one value:

```
1  x = 5
2  x
```

```
[1] 5
```

```
1  x <- 5
2  x
```

```
[1] 5
```

Assign a **vector** of values

- Consecutive integers using :

```
1  a <- 3:10
2  a
```

```
[1]  3  4  5  6  7  8  9 10
```

- **Concatenate** a string of numbers

```
1  b <- c(5, 12, 2, 100, 8)
2  b
```

```
[1]   5  12   2 100   8
```

Let’s try it out!

- Create a new variable `y` that is assigned the value of 8
- Create a new variable `c` that is assigned the vector of values 15 through 20
- Create a new variable `d` that is assigned the vector of values 16 through 19 and 22.

- Did you notice anything in the `Environment` section of Rstudio?

Doing math with variables

Math using variables with just one value

```
1 x <- 5
2 x
```

```
[1] 5
```

```
1 x + 3
```

```
[1] 8
```

```
1 y <- x^2
2 y
```

```
[1] 25
```

Math on vectors of values: **element-wise** computation

```
1 a <- 3:6
2 a
```

```
[1] 3 4 5 6
```

```
1 a+2; a*3
```

```
[1] 5 6 7 8
[1]  9 12 15 18
```

```
1 a*a
```

```
[1]  9 16 25 36
```

Let’s try it out!

- Use the variable name y to find the addition of y and 5
- Add 5 to the vector c

Variables can include text (characters)

```
1 hi <- "hello"
2 hi
```

```
[1] "hello"
```

```
1 greetings <- c("Guten Tag", "Hola", hi)
2 greetings
```

```
[1] "Guten Tag" "Hola"      "hello"
```

Using functions

- `mean()` is an example of a function
- functions have "arguments" that can be specified within the `()`
- `?mean` in console will show help file for `mean()`

Function arguments specified by name:

```
1 mean(x = 1:4)
```

```
[1] 2.5
```

```
1 seq(from = 1, to = 12, by = 3)
```

```
[1]  1  4  7 10
```

```
1 seq(by = 3, to = 12, from = 1)
```

```
[1]  1  4  7 10
```

Function arguments not specified, but listed in order:

```
1 mean(1:4)
```

```
[1] 2.5
```

```
1 seq(1, 12, 3)
```

```
[1]  1  4  7 10
```

Now let's use some functions for summary statistics

- We will calculate the mean for `c`
- Let's also calculate the standard deviation for `c`
  - Recall, our function is `sd()`
  - Use `?sd` in the console to identify the arguments for `c`

- If you have more time, you can try to calculate the median and IQR for `c`

Common console errors (1/2)

Incomplete commands

- When the console is waiting for a new command, the prompt line begins with >
  - If the console prompt is +, then a previous command is incomplete
  - You can finish typing the command in the console window

Example:

```
> 3 + (2*6
+ )
```

```
[1] 15
```

Common console errors (2/2)

Object is not found

- This happens when text is entered for a non-existent variable (object)

Example:

```
1 hello
```

```
Error in eval(expr, envir, enclos): object 'hello' not found
```

- Can be due to missing quotes

```
1 install.packages(dplyr)
```

```
Error in eval(expr, envir, enclos): object 'dplyr' not found
```

```
1 # correct code is: install.packages("dplyr")
```

Getting help with R

There are many ways to get help when you are stuck

1. Use the ? in front of the function name to get more information!
   - Usually if I need help with the arguments for a function
2. Google or go to stackoverflow.com
   - Often when I Google, I get redirected to something like stackoverflow
   - For example, let's say my `mean` function was outputting `NA`. I would Google something like "keep getting NA for mean in R" Then end up here
3. I can also go to my favorite AI tool to get help
   - This is most useful for getting code started if it's complicated (we're not really at that level yet)
   - I asked ChatGPT "can you give me the code for calculating the mean in R"
     - This is what I got
   - For code generation, it gives you WAY too much
   - I also asked ChatGPT "Why is the mean function in R giving me an NA?" (in above link)

More on AI usage

- In the syllabus
- If you cannot trace code back to the class notes, then do NOT use it!
  - There's different coding practices and functions out there
  - I'm giving you a specific set of tools that will serve as a good introduction
  - You should be able to explain all your code and work

Let’s try with an example dataset

Fisher's (or Anderson's) Iris data set

Data description:

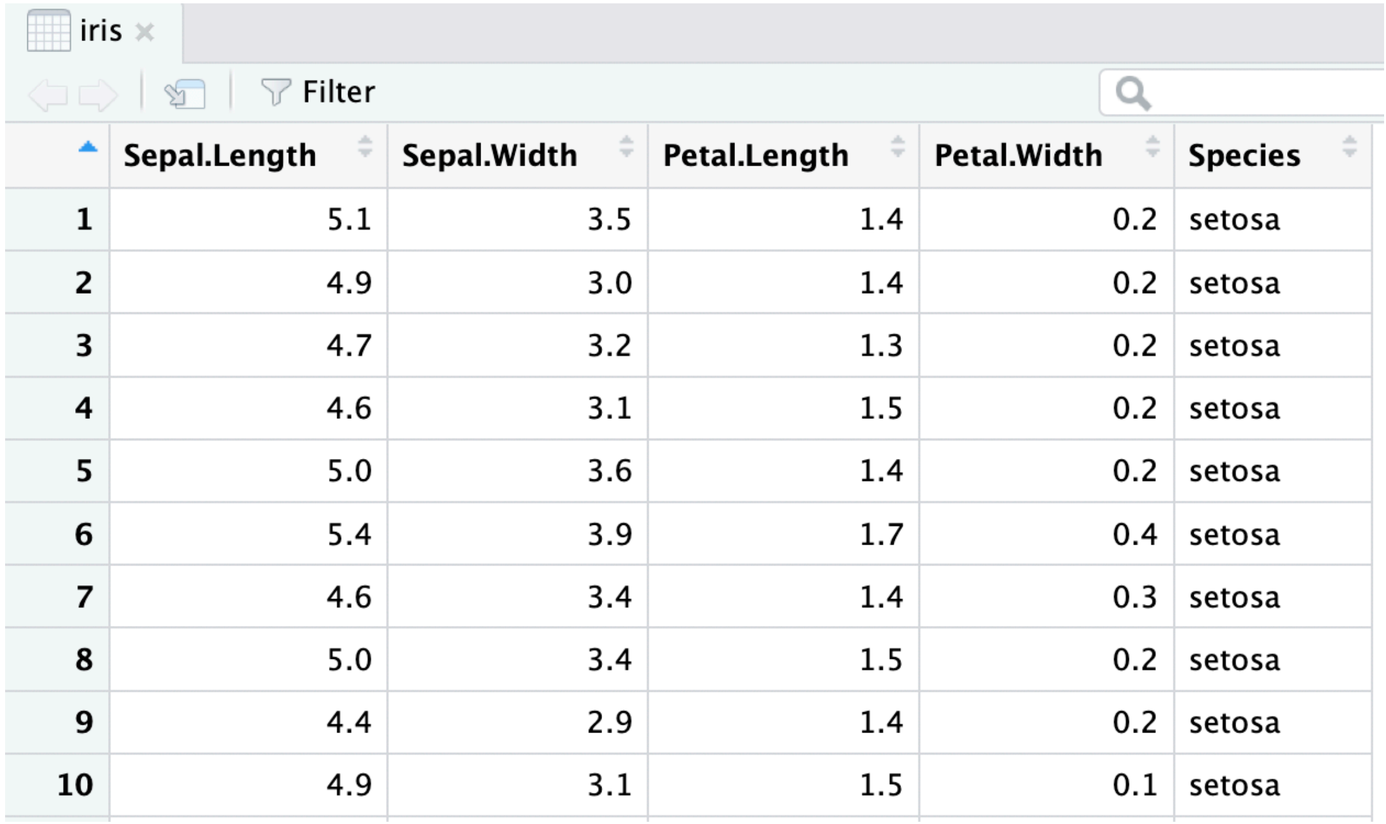
- n = 150
- 3 species of Iris flowers (Setosa, Virginica, and Versicolour)
  - 50 measurements of each type of Iris
- **Variables**:
  - sepal length, sepal width, petal length, petal width, and species

View the `iris` dataset

- The `iris` dataset is already pre-loaded in *base* R and ready to use.
- Type the following command in the console window
  - *Warning: this command cannot be rendered. It will give an error.*

```
1  View(iris)
```

A new tab in the scripting window should appear with the `iris` dataset.



iris

Filter

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

Data structure (1/2)

- What are the different **variable types** in this data set?
- We are going to use the `str` function
  - Can you use the console to tell me what we can input into `str`?

Data structure (2/2)

- What are the different **variable types** in this data set?
- We are going to use the `str` function
  - Can you use the console to tell me what we can input into `str`?

```
1 str(iris)   # structure of data
```

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Data set summary

- Can we quickly summarize all the data?

```
1  summary(iris)
```

```
 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
Median :5.800   Median :3.000   Median :4.350   Median :1.300
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
setosa    :50
versicolor:50
virginica :50
```

Data set info

- You can use different functions to find information on a data frame

```
dim(iris)
```

```
[1] 150   5
```

```
nrow(iris)
```

```
[1] 150
```

```
ncol(iris)
```

```
[1] 5
```

```
names(iris)
```

```
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

- We can also look at the `Environment` section

Take a moment to find the information on the iris data frame

- Go to environment section to see the `iris` data frame

View the beginning or end of a dataset

- These commands can be helpful if the data frame has a lot of rows

```
1 head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```
1 tail(iris)
```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

Specify how many rows to view at beginning or end of a dataset

```
1 head(iris, 3)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
```

```
1 tail(iris, 2)
```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

Some sources for useful base R commands

- https://sites.calvin.edu/scofield/courses/m143/materials/RcmdsFromClass.pdf
- https://www2.kenyon.edu/Depts/Math/hartlaub/Math206%20Spring2011/R.htm

R03 Slides