Lab 2: Intro to R Markdown `Rmd` Files

Stat 133, Spring 2022

Contents

Learning Objectives:

- Get started with `.Rmd` files
- Get to know markdown syntax

1) Introduction to R Markdown files

Most of the times you won't be working directly on the console. Instead, you will be typing your commands in some **source file**. The most basic type of source files are known as *R script files*. But there are more flavors of source files. A very convenient type of source file that allow you to mix R code with narrative is an **R markdown file** commonly referred to as `Rmd` file.

1.1) Get to know the `Rmd` files

In the menu bar of RStudio, click on **File**, then **New File**, and choose **R Markdown**. Select the default option (Document), and click **Ok**.

Rmd files are a special type of file, referred to as a *dynamic document*, that allows to combine narrative (text) with R code. Because you will be turning in most homework assignments as `Rmd` files, it is important that you quickly become familiar with this resource.

Locate the button **Knit HTML** (the one with a knitting icon) and click on it so you can see how `Rmd` files are renderer and displayed as HTML documents.

R markdown files use a special syntax called **markdown**. To be more precise, Rmd files let you type text using either: 1) R syntax for code that needs to be executed; 2) markdown syntax to write your *narrative*, and 3) latex syntax for math equations and symbols.

You will have time to learn the basics of this syntax during the semester, and we expect that feel comfortable with markdown at the end of the course.

1.2) Your turn: Pythagoras formula

Open a new `Rmd` file in the *source* pane, and use it to compute the following.

The pythagoras formula is used to compute the length of the hypotenuse, c, of a right triangle with legs of length a and b.

$$c = \sqrt{a^2 + b^2}$$

Calculate the hypotenuse of a right triangle with legs of length 3 and 4. Use the `sqrt()` function, and create variables `a = 3` and `b = 4`. If you don't know what's the symbol to calculate exponents, search for the help documentation of the arithmetic operators: `?Arithmetic`.

```
a <- 3
b <- 4
c <- sqrt(a^2+b^2)
c
```

```
## [1] 5
```

---

2) Binomial Formula

The formula for the binomial probability is:

$$Pr(k; n, p) = Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

where:

- n is the number of (fixed) trials
- p is the probability of success on each trial
- $1 - p$ is the probability of failure on each trial
- k is a variable that represents the number of successes out of n trials
- the first term in parenthesis is not a fraction, it is the number of combinations in which k success can occur in n trials

R provides the `choose()` function to compute the number of combinations:

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots 1}$$

For instance, the number of combinations in which $k = 2$ success can occur in $n = 5$ trials is:

```
choose(n = 5, k = 2)
```

```
## [1] 10
```

Combinations are typically expressed in terms of factorials as:

$$\frac{n!}{k!(n-k)!}$$

Conveniently, R also provides the function `factorial()` to calculate the factorial of an integer:

```
factorial(4)
```

```
## [1] 24
```

2.1) Your Turn: Binomial Formula

Let's consider a simple example. A fair coin is tossed 5 times. What is the probability of getting exactly 2 heads?

a) Create the objects `n`, `k`, and `p` for the number of trials, the number of success, and the probability of success, respectively.

```
n <- 5
k <- 2
p <- 0.5
n
```

```
## [1] 5
```

```
k
```

```
## [1] 2
```

```
p
```

```
## [1] 0.5
```

b) Use `factorial()` to compute the number of combinations "*n choose k*"

```
factorial(n)/(factorial(k)*factorial(n-k))
```

```
## [1] 10
```

c) Apply the binomial formula, using `factorial()`, to calculate the probability of getting exactly 2 heads out of 5 tosses.

```
(factorial(n)/(factorial(k)*factorial(n-k)))*p^k*(1-p)^(n-k)
```

```
## [1] 0.3125
```

d) Recalculate the same probability but now using `choose()` (instead of `factorial()`)

```r
choose(n=5, k=2)*p^k*(1-p)^(n-k)
```

```
## [1] 0.3125
```

e) Consider rolling a fair die 10 times. What is the probability of getting exactly 3 sixes?

```r
n <- 10
k <- 3
p <- 1/6
choose(n=10, k=3)*p^k*(1-p)^(n-k)
```

```
## [1] 0.1550454
```

f) Now look for help documentation (e.g. `help.search()` or `??`) using the keyword binomial: `binomial`. You should get a list of topics related with the searched term `binomial`. Choose the one related with the *Binomial Distribution*, which is part of the R package `stats` (i.e. `stats::Binomial`).

Read the documentation and figure out how to use the `dbinom()` function to obtain the above probabilities: 2 heads in 5 coin tosses, and 3 sixes in 3 rolls of a die.

```r
p_2h5t = dbinom(2, 5, 0.5, log = FALSE)
p_363r = dbinom(3, 3, (1/6), log = FALSE)
p_2h5t
```

```
## [1] 0.3125
```

```r
p_363r
```

```
## [1] 0.00462963
```

j) How would you modify the previous binomial function to calculate the same probability (2 heads in 5 tosses) of a **biased** coin with a chance of heads of 35%?

```r
biased_p_2h5t = dbinom(2, 5, 0.35, log = FALSE)
biased_p_2h5t
```

```
## [1] 0.3364156
```

k) Finally, obtain the probability of getting more than 3 heads in 5 tosses with a biased coin of 35% chance of heads.

```r
biased_p_3h5t = dbinom(3, 5, 0.35, log = FALSE)
answer = 1 - biased_p_3h5t
answer
```

```
## [1] 0.8188531
```

3) Getting Help

Because we work with functions all the time, it's important to know certain details about how to use them, what input(s) is required, and what is the returned output.

There are several ways to get help.

If you know the name of a function you are interested in knowing more, you can use the function `help()` and pass it the name of the function you are looking for:

```
# documentation about the 'abs' function
help(abs)

# documentation about the 'mean' function
help(mean)
```

Alternatively, you can use a shortcut using the question mark `?` followed by the name of the function:

```
# documentation about the 'abs' function
?abs

# documentation about the 'mean' function
?mean
```

- How to read the manual documentation
  - Title
  - Description
  - Usage of function
  - Arguments
  - Details
  - See Also
  - Examples!!!

`help()` only works if you know the name of the function your are looking for. Sometimes, however, you don't know the name but you may know some keywords. To look for related functions associated to a keyword, use double `help.search()` or simply `??`

```
# search for 'absolute'
help.search("absolute")

# alternatively you can also search like this:
??absolute
```

Notice the use of quotes surrounding the input name inside `help.search()`

3.1) Your turn

Type commands directly on the console:

a) There are several tabs in the pane `Files, Plots, Packages, Help, Viewer`. Find what does the tab **Files** is good for?

b) What about the tab **Help**?

c) In the tab **Help**, what happens when you click the button with a House icon?

d) Now go to the tab **History**. What is it good for? and what about the buttons of its associated menu bar?

e) Likewise, what can you say about the tab **Environment**?