

Class 1

Markdown

A simple markup language

Markdown allows us to format text. Markdown can be compiled directly into a PDF, a Word file or an HTML file.

Why Notebooks?

R Markdown Notebooks allow us to write text, write code and generate results within a single file. In this class all in-class assignments and problem sets will be submitted using Markdown Notebooks to generate PDF files.

How to use markdown

When you see a white background you are writing text. You can use Markdown (or LaTeX) here. However we will also be interested in writing code to explore and summarize data. Code is placed in a *chunk*. Chunks start and end with three grave accents (the same key as the tilde): “`”.

chunks are interactive and allow to you run code. When you execute code within the notebook, the results appear beneath the code.

Try executing the following chunk by clicking the *Run* button (play button) within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
# Take the square root of 49
# NOTE: we cannot use markdown in code chunks. Here # means that we are creating a comment (notes that
sqrt(49)
```

```
## [1] 7
```

Working with R

R is both a programming language and a programming environment. We can use R to compute simple calculations like a calculator.

1. Find the sum of 75, 83, 93, 34, 75 and 81. Searching the book and the internet will be very important in this class. You might find useful information here

Quick math refresher

1. Evaluate

$$\left| \frac{5^{25-2}}{6-4+1*2} \right|$$

2. Are the following chunks different? Why or why not?

```
7 * 5 / 8 * 3
```

```
## [1] 13.125
```

```
7 * (5 / 8) * 3
```

```
## [1] 13.125
```

3. Using R take the log of the number 14.

Working with objects

In R we can store *things* as objects. R is a weakly typed language (there are variable types, but you don't need to declare them upfront).

This lets us run a computation and store the result for later.

1. Save the results of

$$\left| \frac{5^{25-2}}{6 - 4 + 1 * 2} \right|$$

to a variable object named *output*.

2. Get the value of the variable *output*.

3. We have created a variable that contains a numeric value. If we look at the environment tab we will see the variable and its value.

We didn't tell R what kind of variable we wanted to create. R guessed that since we supplied a number, we wanted to create a numeric atomic vector.

Are the following two variables the same? Why or why not?

```
variable1 <- 27.4  
variable2 <- "27.4"
```

Working with functions

Functions are predefined blocks of code that perform a task on input (and generally) produce output. For example, *abs()* is a function that computes the absolute value of some input. With R functions input is provided between parentheses.

R has many built-in functions, but let's explore a few that are useful:

1. Use an R function to compute the length, maximum value, range, and mean for the variable **stress**

```
stress <- c(10, 50, 60, 12, 460.25, 4802, 40, 0, 0, 9.5)
```

2. For these functions we are only supplying one input (an argument): a vector (more on vectors in next class). Some functions are more complex and expect (require) multiple different inputs. A function you will need to master in this class is **round()**. **round()** rounds the numbers (and outputs of other functions) to a specified number of decimals

R can help us figure out what a function expects. We can load a help file for any function by adding a question mark and removing the parentheses. Try the following:

```
?round
```

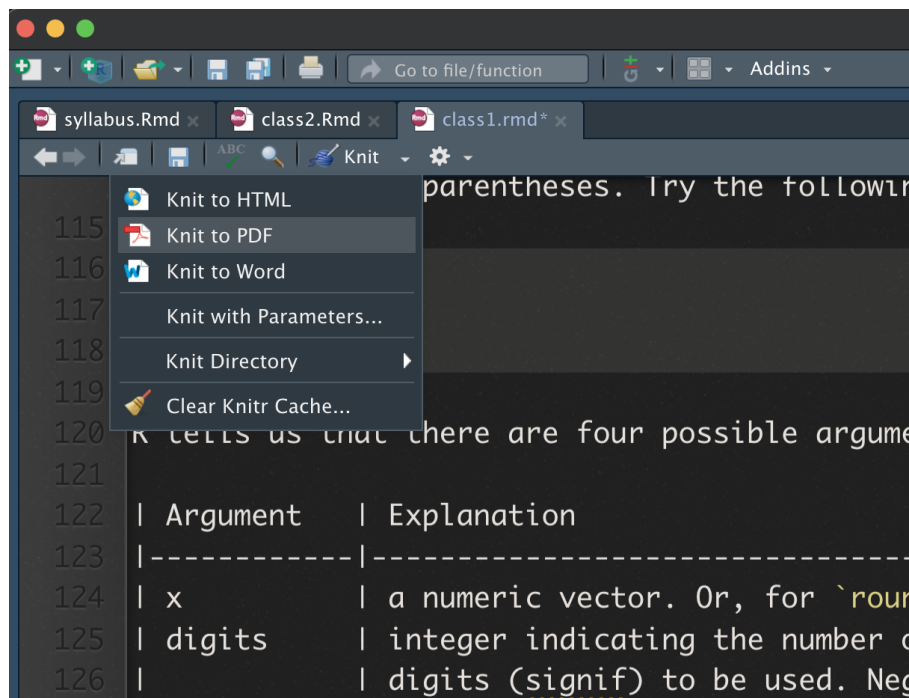
R tells us that there are four possible arguments:

Argument	Explanation
x	a numeric vector. Or, for round and signif , a complex vector.
digits	integer indicating the number of decimal places (round) or significant digits (signif) to be used. Negative values are allowed (see 'Details').

3. Round the number 10.198109 to two decimal places. Note that we would need to supply the x and $digits$ arguments.
4. Round the mean of `stress` to 1 decimal place.

Prepare for submission

You will submit all assignments and problem sets as PDFs. You create a PDF you will click the down arrow to the right of the *Knit* button and select *Knit to PDF*



This assignment is due at the end of class.

Evaluation

1. On a scale ranging between 1 (Too Hard) and 10 (Too Easy), how was today's class?
2. What was the easiest thing to understand?
3. What was the most difficult thing to understand?
4. How long did you spend on the assignment outside of class?