Markdown Basics

Level 1 heading

Paragraph text. Note that comments in Rmd use html style <!- –> format instead of R # format. That's because the # symbols is used to define headings. Luckily you can use the RStudio comment/uncomment shortcut and it will comment text the right way.

Level 2 heading

Paragraph text. Use single asterisks for *italics*, double asterisks for **bold**. Notice that if you hit enter and create multiple lines, markdown interprets this as a continuous paragraph.

If you want a new paragraph, put a blank line in between. Github README files use these same conventions but are saved as .md (plain markdown instead of R Markdown). So these are good to know for documenting your project repo, not just for using in Rmd.

Level 3 heading

Lists are easy to create.

- Top level of an unordered list
    - 2nd level of list using tab
    - 2nd level again
- Back to top level

1. Numbered lists work the same way
2. But just keep using "1".
3. Rmd will figure it out for you
4. See!

Hyperlinks are easy to embded in text, like so: click here. Or if you want the URL to display you can do so like this https://padlab.ucr.edu.

Let's get some R in this markdown

Backticks (the funny looking apostrophes above tab) let us embded R code in the document. You can do this in a text sentence like so: 25. Why do we need to put 'r' in there? Let's try it without: `5*5`. You can use backticks to just format something as code (which is nice for sharing coding examples). Putting 'r' in the code tells Rmd to run it as R code. You need to specify R because Rmd can actually run other types of code, including python and shell scripts.

For more extended code, we need to insert a code chunk. You can use the insert menu in the upper right corner, or you can put three backticks on a line followed by {r} to specify that it is an R language chunk. Every line will be treated as R code until you put a line with another three backticks.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
5*5
```

```
## [1] 25
```

```r
x <- 5*5
```

Now we are back to markdown. Notice that by default, the R chunk prints the code, all warnings/messages, and the result. This might be desirable sometimes (such as when demonstrating code and its result), but at other times you might prefer to tailor the output of a chunk. You can do this with chunk options.

To simply display code without running it, use `eval = FALSE`:

```r
library(tidyverse)
5*5
z <- 'Hello'
```

To run code quietly without displaying anything, use `include = FALSE`:

To suppress the code but show the results, use `echo = FALSE`:

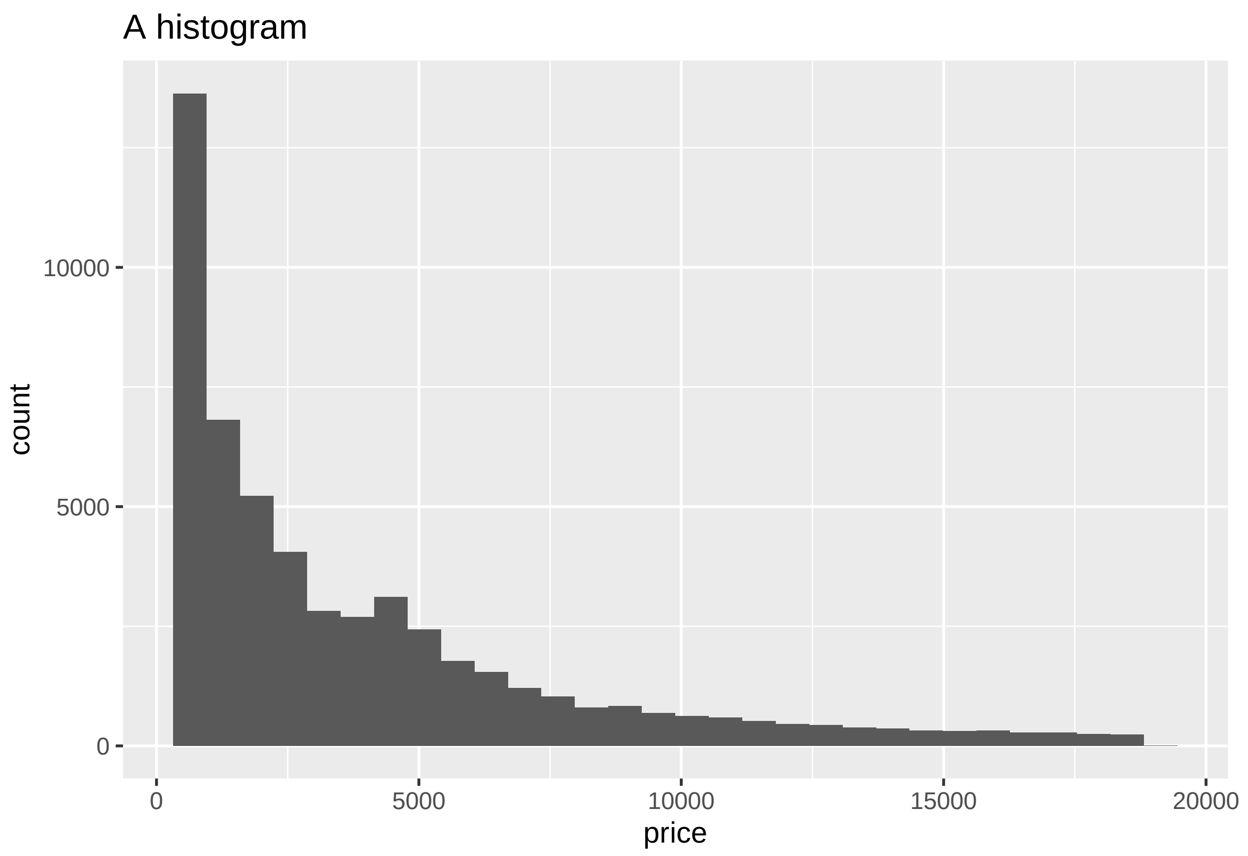
```
## [1] "Hello"
```

Other helpful options include `message = FALSE` and `warning = FALSE` to keep R warnings and messages from printing in your document.

Knitting will run all code chunks, but when working on a Rmd document you may want to test code chunks without knitting them (or see the results/warnings/messages if you chose to suppress them in the output). Using the little green play button in the upper-right corner will run a chunk and create a little output area beneath. This is similar to highlighting lines and running them from within a script (which you can also do in Rmd).

A code chunk that produces a figure will insert that figure into the output document.

```r
ds <- diamonds
ggplot(ds, aes(x = price)) + geom_histogram() + ggtitle("A histogram")
```



Tibbles can become tables really easily.

```
ds %>% select(price, cut) %>% head()
```

```
## # A tibble: 6 x 2
##   price cut
##   <int> <ord>
## 1   326 Ideal
## 2   326 Premium
## 3   327 Good
## 4   334 Premium
## 5   335 Good
## 6   336 Very Good
```

How to create a new Rmd file

File > New File > R Markdown. Pick a format, give it a title and author. But don't stress about your choices, because you can always change the YAML header later. RStudio will create a nice template depending on which format you choose to get you started.