

Math 141 Coding Cheat Sheet

```
# Don't delete this code! It should exist at the start of every Rmd file.  
# It makes sure that your images knit correctly, and that your code chunks by default  
# show up in the PDF, and don't get unhelpful warnings/messages.  
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
```


Using R

The file pane (what does the ~ mean? importing data?)

- The file pane is on the lower right of your RStudio screen, and has all of the files you've saved onto the online RStudio portal saved there! By default, the files will open up to the *home* screen (unless you have a project from a previous class loaded!). If not, you can click the “home” button at the top near the house to go back to the home screen.
- Additionally, you could click the three little dots on the top left and type in the tilde (this guy: ~, press shift then the button above tab on the keyboard) to go back to your home screen.
- As a shortcut to load data files from your file pane into an RStudio code chunk:
 - Click on the name of the file
 - From the pop-up menu, hit “Import Dataset...”



Then, select the first two lines of code (with `library()` and the `read_csv()` functions. `View()` is useful, but only to pull up your dataframe in a separate tab in RStudio). Hit cancel, then go back to your code chunk and paste the code. Now you have a variable that contains a dataframe for the data that you uploaded into RStudio :]

```
Code Preview:   
library(readr)  
X116_cong <- read_csv("Downloads/Math 241  
/116_cong.csv")  
View(X116_cong]
```

The help menu

Anytime you start to use a new function and want to understand what to put inside it, you can either type it in slowly and R will pull up a short yellow pop-up of optional parameters (see below for an example of the `read_csv()` function!).

```
read_csv(file, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"),
  quoted_na = TRUE, quote = "\"", comment = "#",
  trim_ws = TRUE, skip = 0, n_max = Inf, guess_max =
  min(1000, n_max), progress = show_progress(),
  skip_empty_rows = TRUE)
```

`read_csv()` and `read_tsv()` are special cases of the general `read_delim()`. They're useful for reading the most common types
 Press F1 for additional help

Alternatively, go to the “Help” pane (where the Files pane is, but three tabs over!). Here, in the search bar type in any function and hit enter to pull up a handy menu of all inputs and what they mean for function. You can also run in a code chunk or the console `?name_of_any_function_here` to pull up the same help menu!

How to use RMarkdowns—Robin’s tips & tricks!

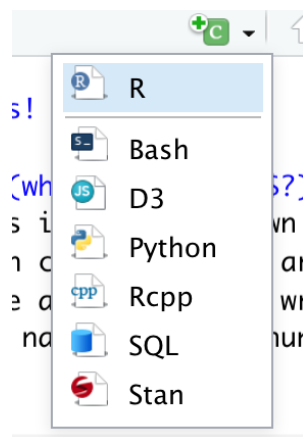
Explanation of code chunks v. plain text (what are these \$?)

- Code chunks: these are the shaded gray areas in the RMarkdown files, distinguished by three backticks (“”) at the beginning and end, with curly brackets around a lower case r {r} on the first line. See below for an example! In these areas, you can write any code you’d like to run! To the right of the lowercase r, you can name the code chunk anything you’d like, as I’ve done.

```
1 + 1
```

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

- Shortcuts for making code chunks: you can either write out all of the backticks and curly brackets everytime you want a new code chunk *or* you can hit the plus C button at the top of the screen. Then, hit the “R” option (see below)



- What else can they include?
 - Code chunks can also include messages such as `echo = FALSE`, or `message = FALSE`. These refer to the outcomes you want with your code chunk. When `echo = FALSE`, we are telling R to **not** print out the code chunk when we knit the document. `message = FALSE` tells R to not print out any long messages (such as when you load in packages). View other resources here about other things you can change! These are not required in this class, though.

How to change size of graphs To change the size of a graph, include the arguments `fig.width = _` and `fig.height = _`. There isn’t a one-size-fits-all measurement here, so I would recommend playing around with these values until you like how your graph looks. Generally, I use `fig.width = 10`, `fig.height = 8`, if you want to start there!

Running Code

- In the console: always feel free to run code in the console (lower left portion of your screen). However, this code doesn't save between sessions, and the graphs pull up on the "Plots" panel, to the right of the "Files" tab.
- In code chunks! Click the regular green play button to run all of the code in that chunk, and the button directly to the left of it to run all code chunks before it (this is helpful if you've just reloaded RStudio and need to load your libraries again!)
- By selecting some text in your code chunk, and hitting command + enter (or control + enter, if you have a Windows machine)! This allows you to run code line by line, or just quickly check a variable's content without writing it in a new line!

Distinction between RMarkdowns and RScripting

Scripting is a helpful tool we talked about briefly in the first class! This type of coding will be useful if we ever want to just create code without interpreting it (as the only text you can include in scripts is code or comments). For instance, scripts do not handle code chunks, with separate areas for text—it's all one big code chunk. That's why hitting the run button runs everything in the script file, not just one part of the code!

Explanation of YAML

The code at the *very beginning* of an RMarkdown document is called the YAML, and is noted by three dashes at the top (—) and at the bottom (—). Here is where you can specify info such as the title, what you want to document to output as when you knit it, and any formatting pieces when you knit! For an example of what Alex wants the YAML to look like, go here on the website. If you're interested in learning more, view this resource on YAMLS for PDFs, and this resource for HTML output :]

Accessing Packages

Installing Packages Before you can load in a package, make sure it's installed! There are a few ways to do this: 1. Type in `library(package_name)`. If the package is not loaded in R, a yellow bar will pop up at the top of your screen, and prompt you to install the package. Click "Install." 2. In the console, write `install.packages("your_package_name_here")` and click enter. 3. Run the `install.packages("your_package_name_here")` in your RMarkdown. Just be sure to delete the code or comment it out before knitting!

Loading (when/how often?) Load packages *once a session*, at the top of your RMarkdown (after the setup chunk). This is because when you knit your document, it runs the code from top to bottom, and you want to make sure that all of the libraries are loaded before it runs code that may need those packages! To load a package, use the `library()` function like so:

```
library(tidyverse)
```

Variables

To assign any information/data you have *changed* to be saved to a variable (essentially a shortcut you can use to grab that changed data), use the `<-` key. For instance, if I want to rename my mariokart data to something shorter, I would write:

```
m <- mariokart
```

It is important to note that you can name your variables just about anything! Make sure it makes logical sense, though, and that if you needed to grab it later you would know what you're referring to.

However: - Your variable cannot start with a number. For instance, `2_species` is not a valid name, but `two_species` or `species_2` would be. - Don't write over another variable when naming things! It's likely that

you'll want to refer back to old data at some point, and writing over it makes things complicated. - If you want to read more about naming conventions, feel free to read through this short document!

Manipulating Data (Base R)

What are these brackets?

- Using these brackets `[]` after a variable name allows you to pull out a certain value(s) from how you've stored it. If you have a vector, say `nums <- c(1, 2, 3, 4, 5)`, you can grab the second value by saying `nums[2]`. If you want to grab sequential values, you can write `nums[2:4]`. This grabs the numbers in the vector from the second value to the fourth value (2, 3, and 4).
- For a dataframe, which stores data in rows and columns, we can still use brackets, but now we have to consider not just one list of vectors, but instead asking for a specific *column number* and then *row number*.
 - This is distinguished in R by using **commas** in your brackets! The first entry `[,]` on the left side of the brackets is where you specify which *rows* you would like to grab. You can grab just one row, or can specify multiple rows using the `:` symbol like above! The second entry corresponds to the *columns* you would like to grab. So, if you want to grab rows 1:15 from column two, write: `data_set_name[1:15, 2]`.
 - * For context, the dataframe I'm using below is world record data for each track in different MarioKart games!

```
mariokart[1:15, 2]
```

```
## # A tibble: 15 x 1
##   Duration
##   <dbl>
## 1      115
## 2       79
## 3      128
## 4       79
## 5      131
## 6      220
## 7      466
## 8      481
## 9       71
## 10     757
## 11     353
## 12     103
## 13     536
## 14       18
## 15     162
```

- Commas can also do the really exciting work of allowing us to grab *all* rows and *all* columns if we so wish. What if we want the first 11 rows of a dataframe and all of the columns? Just write: `data_set_name[1:11,]`. To grab all of the rows but just column one, write: `data_set_name[,1]`.

```
mariokart[1:11,]
```

```
## # A tibble: 11 x 7
##   Date           Duration Character Vehicle Tires      Glider      Game
##   <date>         <dbl> <chr>    <chr>    <chr>    <chr>    <chr>
## 1 2020-12-28      115 Metal Mario B Dasher Red Monster Flower Glider Mariok~
## 2 2021-02-02       79 Metal Mario Bruiser Slick      Swooper    Mariok~
## 3 2020-12-15      128 Honey Queen B Dasher Slick      Beast Glider Mariok~
## 4 2021-02-02       79 Metal Mario Bruiser Red Monster Super Glider Mariok~
```

```
## 5 2020-12-12      131 Honey Queen Bruiser      Slick      Peach Parasol Mariok~
## 6 2020-09-14      220 Honey Queen B Dasher      Red Monster Peach Parasol Mariok~
## 7 2020-01-12      466 Metal Mario B Dasher      Slick      Gold Glider   Mariok~
## 8 2019-12-28      481 Honey Queen Bruiser      Slick      Beast Glider  Mariok~
## 9 2021-02-10        71 Honey Queen B Dasher      Slick      Beast Glider  Mariok~
## 10 2019-03-27     757 Yoshi      Egg 1      Wood      Super Glider  Mariok~
## 11 2020-05-04     353 Metal Mario Koopa Clown Red Monster Swooper      Mariok~
```

```
mariokart[,1]
```

```
## # A tibble: 181 x 1
##   Date
##   <date>
## 1 2020-12-28
## 2 2021-02-02
## 3 2020-12-15
## 4 2021-02-02
## 5 2020-12-12
## 6 2020-09-14
## 7 2020-01-12
## 8 2019-12-28
## 9 2021-02-10
## 10 2019-03-27
## # ... with 171 more rows
```

What does the dollar sign mean?

In R, you can use the dollar sign in the same way that we would grab all of the rows for a specific column. Above we talk about how to grab all of the rows for columns two through four using brackets (writing `data_set[2:4,]`). The dollar sign allows us to identify the name of a column we want to grab from the data. To call on all the rows in the third column in the `mariokart` dataframe, `Character`, we would write:

```
mariokart$Character
```

```
## [1] "Metal Mario"      "Metal Mario"      "Honey Queen"
## [4] "Metal Mario"      "Honey Queen"      "Honey Queen"
## [7] "Metal Mario"      "Honey Queen"      "Honey Queen"
## [10] "Yoshi"            "Metal Mario"      "Wario"
## [13] "Bowser"           "Metal Mario"      "Bowser"
## [16] "Metal Mario"      "Metal Mario"      "Wario"
## [19] "Wario"            "Wario"            "Metal Mario"
## [22] "Wario"            "Metal Mario"      "Metal Mario"
## [25] "Metal Mario"      "Metal Mario"      "Metal Mario"
## [28] "Bowser"           "Metal Mario"      "Metal Mario"
## [31] "Metal Mario"      "Honey Queen"      NA
## [34] "Heavy Mii"        "Morton"           "Morton"
## [37] "Heavy Mii CptFal" "Morton"           "Dry Bowser"
## [40] "Morton"           "Heavy Mii"        "Morton"
## [43] "Morton"           "Morton"           "Heavy Mii"
## [46] "Morton"           "Morton"           "Morton"
## [49] "Morton"           "Morton"           NA
## [52] "Dry Bowser"       "Bowser"           "Morton"
## [55] "Morton"           "Heavy Mii"        "Morton"
## [58] "Morton"           "Morton"           "Dry Bowser"
## [61] "Wario"            "Morton"           "Heavy Mii Toad"
## [64] "Heavy Mii"        "Morton"           "Morton"
```

```
## [67] "Bowser"           "Morton"           "Morton"
## [70] "Dry Bowser"       "Wario"            "Morton"
## [73] "Wario"            "Heavy Mii"        "Heavy Mii Sonic"
## [76] "Dry Bowser"       "Bowser"           "Dry Bowser"
## [79] "Morton"           "Morton"           "Morton"
## [82] "Morton"           "Morton"           NA
## [85] "Wario"            "Waluigi"          "Baby Daisy"
## [88] "Waluigi"          "Waluigi"          "Waluigi"
## [91] "Lemmy"            "Waluigi"          "Waluigi"
## [94] "Rosalina"         "Roy"              "Roy"
## [97] "Waluigi"          "Baby Daisy"        "Roy"
## [100] "Baby Daisy"        "Waluigi"          "Waluigi"
## [103] "Roy"              "Wario"            "Baby Daisy"
## [106] "Baby Peach"       "Waluigi"          "Wario"
## [109] "Wario"            "Baby Peach"       "Lemmy"
## [112] "Lemmy"            "Baby Peach"       "Donkey Kong"
## [115] "Donkey Kong"      "Dry Bowser"       "Waluigi"
## [118] "Baby Daisy"        "Baby Daisy"        "Waluigi"
## [121] "Wario"            "Donkey Kong"      "Baby Peach"
## [124] "Waluigi"          "Waluigi"          "Baby Daisy"
## [127] "Waluigi"          "Baby Daisy"        "Baby Daisy"
## [130] "Waluigi"          "Baby Daisy"        "Wario"
## [133] NA                  "Wario"            "Heavy Mii Mega Man"
## [136] "Wario"            "Wario"            "Heavy Mii"
## [139] "Wario"            "Wario"            "Waluigi"
## [142] "Waluigi"          "Waluigi"          "Roy"
## [145] "Wario"            "Wario"            "Rosalina"
## [148] "Dry Bowser"       "Yoshi"            "Wario"
## [151] "Waluigi"          "Dry Bowser"       "Heavy Mii Mario"
## [154] "King Boo"         "Baby Daisy"        "Waluigi"
## [157] "Donkey Kong"      "Wario"            "King Boo"
## [160] "Roy"              "Waluigi"          "Roy"
## [163] "Dry Bowser"       "Donkey Kong"      "Dry Bowser"
## [166] "Baby Peach"       "Waluigi"          "Baby Daisy"
## [169] "Heavy Mii Cpt Falcon" "Wario"            "Roy"
## [172] "Wario"            "Roy"              "Baby Daisy"
## [175] "Baby Peach"       "Waluigi"          "Heavy Mii Crossing"
## [178] "Baby Daisy"        "Roy"              "Waluigi"
## [181] "Dry Bowser"
```

This prints out all of the characters ever raced with to achieve a world record time! (Gotta go fast).

However, what do we do when we want to call on multiple different columns, without using brackets? To do this, we turn to the `c()` function.

Using the combine function!

This function will probably be the one you use the most, and love the most! It can combine together all sorts of different values, functions you've written, etc. all into one nice variable! - You may remember using this to combine multiple y values from your linear equation function in lab one. If not, never fear, I'll show some examples!

```
# Adding together multiple columns
multiple_cols <- c(mariokart$Character, mariokart$Vehicle, mariokart$Tires)
multiple_cols
```

##	[1]	"Metal Mario"	"Metal Mario"	"Honey Queen"
##	[4]	"Metal Mario"	"Honey Queen"	"Honey Queen"
##	[7]	"Metal Mario"	"Honey Queen"	"Honey Queen"
##	[10]	"Yoshi"	"Metal Mario"	"Wario"
##	[13]	"Bowser"	"Metal Mario"	"Bowser"
##	[16]	"Metal Mario"	"Metal Mario"	"Wario"
##	[19]	"Wario"	"Wario"	"Metal Mario"
##	[22]	"Wario"	"Metal Mario"	"Metal Mario"
##	[25]	"Metal Mario"	"Metal Mario"	"Metal Mario"
##	[28]	"Bowser"	"Metal Mario"	"Metal Mario"
##	[31]	"Metal Mario"	"Honey Queen"	NA
##	[34]	"Heavy Mii"	"Morton"	"Morton"
##	[37]	"Heavy Mii CptFal"	"Morton"	"Dry Bowser"
##	[40]	"Morton"	"Heavy Mii"	"Morton"
##	[43]	"Morton"	"Morton"	"Heavy Mii"
##	[46]	"Morton"	"Morton"	"Morton"
##	[49]	"Morton"	"Morton"	NA
##	[52]	"Dry Bowser"	"Bowser"	"Morton"
##	[55]	"Morton"	"Heavy Mii"	"Morton"
##	[58]	"Morton"	"Morton"	"Dry Bowser"
##	[61]	"Wario"	"Morton"	"Heavy Mii Toad"
##	[64]	"Heavy Mii"	"Morton"	"Morton"
##	[67]	"Bowser"	"Morton"	"Morton"
##	[70]	"Dry Bowser"	"Wario"	"Morton"
##	[73]	"Wario"	"Heavy Mii"	"Heavy Mii Sonic"
##	[76]	"Dry Bowser"	"Bowser"	"Dry Bowser"
##	[79]	"Morton"	"Morton"	"Morton"
##	[82]	"Morton"	"Morton"	NA
##	[85]	"Wario"	"Waluigi"	"Baby Daisy"
##	[88]	"Waluigi"	"Waluigi"	"Waluigi"
##	[91]	"Lemmy"	"Waluigi"	"Waluigi"
##	[94]	"Rosalina"	"Roy"	"Roy"
##	[97]	"Waluigi"	"Baby Daisy"	"Roy"
##	[100]	"Baby Daisy"	"Waluigi"	"Waluigi"
##	[103]	"Roy"	"Wario"	"Baby Daisy"
##	[106]	"Baby Peach"	"Waluigi"	"Wario"
##	[109]	"Wario"	"Baby Peach"	"Lemmy"
##	[112]	"Lemmy"	"Baby Peach"	"Donkey Kong"
##	[115]	"Donkey Kong"	"Dry Bowser"	"Waluigi"
##	[118]	"Baby Daisy"	"Baby Daisy"	"Waluigi"
##	[121]	"Wario"	"Donkey Kong"	"Baby Peach"
##	[124]	"Waluigi"	"Waluigi"	"Baby Daisy"
##	[127]	"Waluigi"	"Baby Daisy"	"Baby Daisy"
##	[130]	"Waluigi"	"Baby Daisy"	"Wario"
##	[133]	NA	"Wario"	"Heavy Mii Mega Man"
##	[136]	"Wario"	"Wario"	"Heavy Mii"
##	[139]	"Wario"	"Wario"	"Waluigi"
##	[142]	"Waluigi"	"Waluigi"	"Roy"
##	[145]	"Wario"	"Wario"	"Rosalina"
##	[148]	"Dry Bowser"	"Yoshi"	"Wario"
##	[151]	"Waluigi"	"Dry Bowser"	"Heavy Mii Mario"
##	[154]	"King Boo"	"Baby Daisy"	"Waluigi"
##	[157]	"Donkey Kong"	"Wario"	"King Boo"
##	[160]	"Roy"	"Waluigi"	"Roy"

## [163]	"Dry Bowser"	"Donkey Kong"	"Dry Bowser"
## [166]	"Baby Peach"	"Waluigi"	"Baby Daisy"
## [169]	"Heavy Mii Cpt Falcon"	"Wario"	"Roy"
## [172]	"Wario"	"Roy"	"Baby Daisy"
## [175]	"Baby Peach"	"Waluigi"	"Heavy Mii Crossing"
## [178]	"Baby Daisy"	"Roy"	"Waluigi"
## [181]	"Dry Bowser"	"B Dasher"	"Bruiser"
## [184]	"B Dasher"	"Bruiser"	"Bruiser"
## [187]	"B Dasher"	"B Dasher"	"Bruiser"
## [190]	"B Dasher"	"Egg 1"	"Koopla Clown"
## [193]	"B Dasher"	"B Dasher"	"B Dasher"
## [196]	"B Dasher"	"B Dasher"	"B Dasher"
## [199]	"B Dasher"	"B Dasher"	"B Dasher"
## [202]	"B Dasher"	"B Dasher"	"B Dasher"
## [205]	"B Dasher"	"Zucchini"	"Blue 7"
## [208]	"B Dasher"	"B Dasher"	"Blue 7"
## [211]	"B Dasher"	"B Dasher"	"B Dasher"
## [214]	NA	"Gold Standard"	"Streetle"
## [217]	"Streetle"	"Streetle"	"Streetle"
## [220]	"Mach 8"	"Streetle"	"Streetle"
## [223]	"Blue Falcon"	"Blue Falcon"	"Streetle"
## [226]	"Blue Falcon"	"Pipe Frame"	"Streetle"
## [229]	"Streetle"	"Blue Falcon"	"Blue Falcon"
## [232]	NA	"Streetle"	"Streetle"
## [235]	"Blue Falcon"	"Blue Falcon"	"Blue Falcon"
## [238]	"Streetle"	"Blue Falcon"	"Streetle"
## [241]	"Streetle"	"Tanooki Kart"	"Mach 8"
## [244]	"Streetle"	"Mach 8"	"Blue Falcon"
## [247]	"Streetle"	"Pipe Frame"	"Mach 8"
## [250]	"Blue Falcon"	"Blue Falcon"	"Gold Standard"
## [253]	"Pipe Frame"	"Pipe Frame"	"W 25 Silver Arrow"
## [256]	"Blue Falcon"	"Gold Standard"	"Streetle"
## [259]	"Mach 8"	"Streetle"	"Mach 8"
## [262]	"Streetle"	"Streetle"	"Streetle"
## [265]	NA	"Wild Wiggler"	"Mr. Scooty"
## [268]	"Biddybuggy"	"Mr. Scooty"	"Biddybuggy"
## [271]	"Mr. Scooty"	"Biddybuggy"	"Mr. Scooty"
## [274]	"Biddybuggy"	"Streetle"	"Biddybuggy"
## [277]	"Biddybuggy"	"Biddybuggy"	"Biddybuggy"
## [280]	"Biddybuggy"	"Biddybuggy"	"Biddybuggy"
## [283]	"Biddybuggy"	"Streetle"	"Sport Bike"
## [286]	"Biddybuggy"	"Biddybuggy"	"Biddybuggy"
## [289]	"Sports Coupe"	"Inkstriker"	"Biddybuggy"
## [292]	"Biddybuggy"	"Biddybuggy"	"Biddybuggy"
## [295]	"Biddybuggy"	"Biddybuggy"	"Mach 8"
## [298]	"Biddybuggy"	"Biddybuggy"	"Biddybuggy"
## [301]	"Mr. Scooty"	"Splat Buggy"	"Biddybuggy"
## [304]	"Biddybuggy"	"Biddybuggy"	"Mr. Scooty"
## [307]	"Biddybuggy"	"Biddybuggy"	"Biddybuggy"
## [310]	"Biddybuggy"	"Biddybuggy"	"Biddybuggy"
## [313]	"Mach 8"	NA	"Wild Wiggler"
## [316]	"Biddybuggy"	"Biddybuggy"	"W 25 Silver Arrow"
## [319]	"W 25 Silver Arrow"	"Blue Falcon"	"Yoshi Bike"
## [322]	"Wild Wiggler"	"Wild Wiggler"	"Streetle"

## [325]	"Mach 8"	"Wild Wiggler"	"Teddy Buggy"
## [328]	"Biddybuggy"	"Wild Wiggler"	"Biddybuggy"
## [331]	"Sneeker"	"Wild Wiggler"	"Landship"
## [334]	"Sport Bike"	"Landship"	"Biddybuggy"
## [337]	"Teddy Buggy"	"Biddybuggy"	"Biddybuggy"
## [340]	"Streettle"	"W 25 Silver Arrow"	"Wild Wiggler"
## [343]	"Biddybuggy"	"W 25 Silver Arrow"	"Biddybuggy"
## [346]	"Mach 8"	"Biddybuggy"	"Biddybuggy"
## [349]	"Biddybuggy"	"Mach 8"	"Mach 8"
## [352]	"Biddybuggy"	"Wild Wiggler"	"Mr. Scooty"
## [355]	"Mr. Scooty"	"Mr. Scooty"	"Biddybuggy"
## [358]	"Teddy Buggy"	"Biddybuggy"	"Biddybuggy"
## [361]	"Biddybuggy"	"Mach 8"	"Red Monster"
## [364]	"Slick"	"Slick"	"Red Monster"
## [367]	"Slick"	"Red Monster"	"Slick"
## [370]	"Slick"	"Slick"	"Wood"
## [373]	"Red Monster"	"Slick"	"Slick"
## [376]	"Red Monster"	"Slick"	"Slick"
## [379]	"Red Monster"	"Slick"	"Slick"
## [382]	"Red Monster"	"Slick"	"Mushroom"
## [385]	"Slick"	"Slick"	"Red Monster"
## [388]	"Red Monster"	"Slick"	"Slick"
## [391]	"Red Monster"	"Slick"	"Mushroom"
## [394]	"Slick"	NA	"Slick"
## [397]	"Leaf Tires"	"Slick"	"Cyber Slick"
## [400]	"Slick"	"Slick"	"Cyber Slick"
## [403]	"Slick"	"Cyber Slick"	"Triforce Tires"
## [406]	"Slick"	"Slick"	"Slick"
## [409]	"Slick"	"Cyber Slick"	"Cyber Slick"
## [412]	"Slick"	NA	"Cyber Slick"
## [415]	"Slick"	"Slick"	"Cyber Slick"
## [418]	"Cyber Slick"	"Slick"	"Cyber Slick"
## [421]	"Slick"	"Cyber Slick"	"Leaf Tires"
## [424]	"Slick"	"Cyber Slick"	"Slick"
## [427]	"Cyber Slick"	"Cyber Slick"	"Slim"
## [430]	"Slick"	"Slick"	"Slick"
## [433]	"Slick"	"Slim"	"Slim"
## [436]	"Slick"	"Slick"	"Slick"
## [439]	"Standard"	"Slick"	"Slick"
## [442]	"Slick"	"Slick"	"Cyber Slick"
## [445]	"Cyber Slick"	NA	"Roller"
## [448]	"Azure Roller"	"Azure Roller"	"Azure Roller"
## [451]	"Leaf Tires"	"Azure Roller"	"Azure Roller"
## [454]	"Azure Roller"	"Azure Roller"	"Azure Roller"
## [457]	"Azure Roller"	"Azure Roller"	"Azure Roller"
## [460]	"Azure Roller"	"Azure Roller"	"Azure Roller"
## [463]	"Roller"	"Leaf Tires"	"Azure Roller"
## [466]	"Slim"	"Azure Roller"	"Azure Roller"
## [469]	"Roller"	"Azure Roller"	"Azure Roller"
## [472]	"Azure Roller"	"Azure Roller"	"Azure Roller"
## [475]	"Azure Roller"	"Azure Roller"	"Azure Roller"
## [478]	"Leaf Tires"	"Azure Roller"	"Azure Roller"
## [481]	"Azure Roller"	"Leaf Tires"	"Slim"
## [484]	"Azure Roller"	"Azure Roller"	"Azure Roller"

```
## [487] "Leaf Tires"           "Roller"           "Leaf Tires"
## [490] "Azure Roller"         "Azure Roller"     "Azure Roller"
## [493] "Azure Roller"         "Leaf Tires"       NA
## [496] "Azure Roller"         "Azure Roller"     "Button"
## [499] "Azure Roller"         "Leaf Tires"       "Crimson Slim"
## [502] "Button"               "Leaf Tires"       "Azure Roller"
## [505] "Azure Roller"         "Leaf Tires"       "Azure Roller"
## [508] "Azure Roller"         "Azure Roller"     "Leaf Tires"
## [511] "Roller"               "Crimson Slim"    "Leaf Tires"
## [514] "Azure Roller"         "Metal"            "Azure Roller"
## [517] "Azure Roller"         "Roller"           "Azure Roller"
## [520] "Azure Roller"         "Azure Roller"     "Azure Roller"
## [523] "Azure Roller"         "Azure Roller"     "Leaf Tires"
## [526] "Azure Roller"         "Leaf Tires"       "Azure Roller"
## [529] "Azure Roller"         "Leaf Tires"       "Wood"
## [532] "Crimson Slim"         "Azure Roller"     "Azure Roller"
## [535] "Roller"               "Button"           "Azure Roller"
## [538] "Leaf Tires"           "Azure Roller"     "Azure Roller"
## [541] "Azure Roller"         "Azure Roller"     "Leaf Tires"
```

```
# Adding together the results of multiple functions:
FUNctions <- c(seq(0,10, length.out = 4), rep("boo", 8), rep("halloween", 13))
FUNctions
```

```
## [1] "0"                "3.333333333333333" "6.666666666666667" "10"
## [5] "boo"              "boo"                "boo"                "boo"
## [9] "boo"              "boo"                "boo"                "boo"
## [13] "halloween"        "halloween"          "halloween"          "halloween"
## [17] "halloween"        "halloween"          "halloween"          "halloween"
## [21] "halloween"        "halloween"          "halloween"          "halloween"
## [25] "halloween"
```

Subsetting Data (dplyr)

Loading the package

- First things first, you have to *load* in your package! This means run the `library()` command to grab the dplyr package from where it usually lives in the background of R to the forefront!

The pipe, %>%

- There's lots of different ways to think about what the pipe does! I usually think about it in my code as indicating "and then," pointing from the line we have just written to the next line. As an example:

```
# Taking my mariokart data, and renaming it to mariokart_measures. AND THEN
mariokart_measures <- mariokart %>%
  # Creating a new column called hello AND THEN
  mutate(hello = "hello")
print(mariokart_measures)
```

```
## # A tibble: 181 x 8
##   Date           Duration Character Vehicle Tires      Glider      Game      hello
##   <date>         <dbl> <chr>      <chr>    <chr>    <chr>      <chr>    <chr>
## 1 2020-12-28     115 Metal Mario B Dasher Red Monster Flower Gl~ Mariok~ hello
## 2 2021-02-02      79 Metal Mario Bruiser Slick      Swooper    Mariok~ hello
## 3 2020-12-15     128 Honey Queen B Dasher Slick      Beast Gli~ Mariok~ hello
## 4 2021-02-02      79 Metal Mario Bruiser Red Monster Super Gli~ Mariok~ hello
```

```
## 5 2020-12-12      131 Honey Queen Bruiser Slick      Peach Par~ Mariok~ hello
## 6 2020-09-14      220 Honey Queen B Dasher Red Monster Peach Par~ Mariok~ hello
## 7 2020-01-12      466 Metal Mario B Dasher Slick      Gold Glid~ Mariok~ hello
## 8 2019-12-28      481 Honey Queen Bruiser Slick      Beast Gli~ Mariok~ hello
## 9 2021-02-10        71 Honey Queen B Dasher Slick      Beast Gli~ Mariok~ hello
## 10 2019-03-27     757 Yoshi      Egg 1    Wood      Super Gli~ Mariok~ hello
## # ... with 171 more rows
```

Boolean arguments, ==, |, and &, numerical < >

- It's important to note when working in dplyr and *filtering* for certain values that we are running true/false, or boolean arguments. What this means is when we are telling R we want some rows and not others, it will run through every row and ask if that value is equal to what we've asked (TRUE) or not equal (FALSE).
 - Some important boolean characters to know (as commas are not specific enough) is the == sign. Use this to tell R that you want your row to *equal* a value.

```
mariokart %>%
  filter(Character == "Metal Mario")

## # A tibble: 17 x 7
##   Date      Duration Character Vehicle Tires      Glider      Game
##   <date>    <dbl> <chr>      <chr>  <chr>  <chr>      <chr>
## 1 2020-12-28    115 Metal Mario B Dasher Red Monster Flower Glider Mariok~
## 2 2021-02-02     79 Metal Mario Bruiser Slick      Swooper      Mariok~
## 3 2021-02-02     79 Metal Mario Bruiser Red Monster Super Glider Mariok~
## 4 2020-01-12    466 Metal Mario B Dasher Slick      Gold Glider Mariok~
## 5 2020-05-04   353 Metal Mario Koopa Clown Red Monster Swooper      Mariok~
## 6 2021-04-04     18 Metal Mario B Dasher Red Monster Flower Glider Mariok~
## 7 2020-02-08   439 Metal Mario B Dasher Slick      Gold Glider Mariok~
## 8 2020-11-05   168 Metal Mario B Dasher Red Monster Flower Glider Mariok~
## 9 2020-10-20   184 Metal Mario B Dasher Slick      Super Glider Mariok~
## 10 2020-11-19   154 Metal Mario B Dasher Slick      Super Glider Mariok~
## 11 2021-02-04     77 Metal Mario B Dasher Slick      Flower Glider Mariok~
## 12 2020-12-29   114 Metal Mario Zucchini Red Monster Super Glider Mariok~
## 13 2019-11-26   513 Metal Mario Blue 7 Red Monster Super Glider Mariok~
## 14 2021-02-21     60 Metal Mario B Dasher Slick      Swooper      Mariok~
## 15 2019-12-06   503 Metal Mario Blue 7 Red Monster Swooper      Mariok~
## 16 2021-04-17     5 Metal Mario B Dasher Slick      Beast Glider Mariok~
## 17 2020-08-22   243 Metal Mario B Dasher Mushroom Swooper      Mariok~
```

```
mariokart %>%
  filter(Duration == 466)

## # A tibble: 1 x 7
##   Date      Duration Character Vehicle Tires Glider      Game
##   <date>    <dbl> <chr>      <chr>  <chr> <chr>      <chr>
## 1 2020-01-12    466 Metal Mario B Dasher Slick Gold Glider Mariokart 7
```

- What if you want all of the rows where the cell *does not* contain a value? That's simple, use the not equal sign: !=

```
mariokart %>%
  filter(Duration != 154)

## # A tibble: 176 x 7
##   Date      Duration Character Vehicle Tires      Glider      Game
```

```
##   <date>         <dbl> <chr>      <chr>    <chr>      <chr>      <chr>
## 1 2020-12-28      115 Metal Mario B Dasher Red Monster Flower Glider Mariokart~
## 2 2021-02-02       79 Metal Mario Bruiser Slick      Swooper      Mariokart~
## 3 2020-12-15     128 Honey Queen B Dasher Slick      Beast Glider Mariokart~
## 4 2021-02-02       79 Metal Mario Bruiser Red Monster Super Glider Mariokart~
## 5 2020-12-12     131 Honey Queen Bruiser Slick      Peach Parasol Mariokart~
## 6 2020-09-14     220 Honey Queen B Dasher Red Monster Peach Parasol Mariokart~
## 7 2020-01-12     466 Metal Mario B Dasher Slick      Gold Glider  Mariokart~
## 8 2019-12-28     481 Honey Queen Bruiser Slick      Beast Glider Mariokart~
## 9 2021-02-10       71 Honey Queen B Dasher Slick      Beast Glider Mariokart~
## 10 2019-03-27    757 Yoshi      Egg 1    Wood      Super Glider Mariokart~
## # ... with 166 more rows
```

```
mariokart %>%
  filter(Character != "Metal Mario")
```

```
## # A tibble: 160 x 7
##   Date          Duration Character Vehicle Tires      Glider      Game
##   <date>         <dbl> <chr>      <chr>    <chr>      <chr>      <chr>
## 1 2020-12-15     128 Honey Queen B Dasher Slick      Beast Glider Mariokart~
## 2 2020-12-12     131 Honey Queen Bruiser Slick      Peach Parasol Mariokart~
## 3 2020-09-14     220 Honey Queen B Dasher Red Monster Peach Parasol Mariokart~
## 4 2019-12-28     481 Honey Queen Bruiser Slick      Beast Glider Mariokart~
## 5 2021-02-10       71 Honey Queen B Dasher Slick      Beast Glider Mariokart~
## 6 2019-03-27    757 Yoshi      Egg 1    Wood      Super Glider Mariokart~
## 7 2021-01-09     103 Wario      B Dasher Slick      Super Glider Mariokart~
## 8 2019-11-03     536 Bowser    B Dasher Slick      Beast Glider Mariokart~
## 9 2020-11-11     162 Bowser    B Dasher Slick      Beast Glider Mariokart~
## 10 2021-04-13        9 Wario      B Dasher Slick      Beast Glider Mariokart~
## # ... with 150 more rows
```

- For numerical values only, you can use the greater than (>), less than (<) or greater than/equal to signs (>=, <=)!

```
mariokart %>%
  filter(Duration > 400)
```

```
## # A tibble: 35 x 7
##   Date          Duration Character Vehicle Tires      Glider      Game
##   <date>         <dbl> <chr>      <chr>    <chr>      <chr>      <chr>
## 1 2020-01-12     466 Metal Mario B Dasher Slick      Gold Glid~ Marioka~
## 2 2019-12-28     481 Honey Queen Bruiser Slick      Beast Gli~ Marioka~
## 3 2019-03-27    757 Yoshi      Egg 1    Wood      Super Gli~ Marioka~
## 4 2019-11-03     536 Bowser    B Dasher Slick      Beast Gli~ Marioka~
## 5 2020-02-08     439 Metal Mario B Dasher Slick      Gold Glid~ Marioka~
## 6 2019-07-16     646 Wario      B Dasher Mushroom Flower Gl~ Marioka~
## 7 2019-11-26     513 Metal Mario Blue 7    Red Monster Super Gli~ Marioka~
## 8 2019-12-06     503 Metal Mario Blue 7    Red Monster Swooper  Marioka~
## 9 2017-04-02    1481 Heavy Mii Gold Standard Slick      Cloud Gli~ Marioka~
## 10 2017-05-25    1428 Morton    Streetle Leaf Tires Cloud Gli~ Marioka~
## # ... with 25 more rows
```

```
mariokart %>%
  filter(Duration <= 100)
```

```
## # A tibble: 74 x 7
##   Date          Duration Character Vehicle Tires      Glider      Game
```

```
##   <date>         <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 2021-02-02      79 Metal Mario Bruiser Slick      Swooper     Mariokart~
## 2 2021-02-02      79 Metal Mario Bruiser Red Monster Super Glider Mariokart~
## 3 2021-02-10      71 Honey Queen B Dasher Slick      Beast Glider Mariokart~
## 4 2021-04-04      18 Metal Mario B Dasher Red Monster Flower Glider Mariokart~
## 5 2021-04-13       9 Wario      B Dasher Slick      Beast Glider Mariokart~
## 6 2021-02-14      67 Wario      B Dasher Slick      Peach Parasol Mariokart~
## 7 2021-02-20      61 Wario      B Dasher Red Monster Super Glider Mariokart~
## 8 2021-02-04      77 Metal Mario B Dasher Slick      Flower Glider Mariokart~
## 9 2021-02-21      60 Metal Mario B Dasher Slick      Swooper     Mariokart~
## 10 2021-04-17     5 Metal Mario B Dasher Slick      Beast Glider Mariokart~
## # ... with 64 more rows
```

- Lastly, there is the or | symbol, along with the and & symbol. When grabbing multiple results in one column, you'll use the or sign. When grabbing combinations of results between columns, use the and sign!
 - If you have lots of values you want to pull from and don't want to type out the *or* sign and `column_name ==` for each occurrence, you can use the `%in%` to refer to a vector with all your values of interest. See below for an example!

```
mariokart %>%
  filter(Character == "Yoshi" | Character == "Heavy Mii")
```

```
## # A tibble: 9 x 7
##   Date      Duration Character Vehicle      Tires      Glider      Game
##   <date>    <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 2019-03-27    757 Yoshi      Egg 1      Wood      Super Gl~ Marioka~
## 2 2017-04-02   1481 Heavy Mii Gold Standard Slick      Cloud Gl~ Marioka~
## 3 2016-06-11   1776 Heavy Mii Streetle    Slick      Gold Gli~ Marioka~
## 4 2020-10-04    200 Heavy Mii Blue Falcon Slick      Parafoil  Marioka~
## 5 2017-02-04   1538 Heavy Mii Blue Falcon Cyber Slick MKTV Par~ Marioka~
## 6 2017-04-27   1456 Heavy Mii Mach 8      Slick      Super Gl~ Marioka~
## 7 2021-01-20    92 Heavy Mii W 25 Silver Arrow Slick      MKTV Par~ Marioka~
## 8 2021-03-12    41 Heavy Mii W 25 Silver Arrow Leaf Tires Paper Gl~ Marioka~
## 9 2021-02-01    80 Yoshi      Bidbuggy    Roller     Paper Gl~ Marioka~
```

```
mariokart %>%
  filter(Character == "Yoshi" & Vehicle == "Egg 1")
```

```
## # A tibble: 1 x 7
##   Date      Duration Character Vehicle Tires Glider      Game
##   <date>    <dbl> <chr>      <chr>      <chr> <chr>      <chr>
## 1 2019-03-27    757 Yoshi      Egg 1      Wood Super Glider Mariokart 7
```

```
mariokart %>%
  filter(Character %in% c("Yoshi", "Wario", "Morton"))
```

```
## # A tibble: 55 x 7
##   Date      Duration Character Vehicle Tires      Glider      Game
##   <date>    <dbl> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 2019-03-27    757 Yoshi      Egg 1      Wood      Super Glider Mariokart 7
## 2 2021-01-09    103 Wario      B Dasher Slick      Super Glider Mariokart 7
## 3 2021-04-13     9 Wario      B Dasher Slick      Beast Glider Mariokart 7
## 4 2021-02-14    67 Wario      B Dasher Slick      Peach Parasol Mariokart 7
## 5 2021-02-20    61 Wario      B Dasher Red Monster Super Glider Mariokart 7
## 6 2019-07-16   646 Wario      B Dasher Mushroom Flower Glider Mariokart 7
## 7 2017-05-25  1428 Morton    Streetle Leaf Tires Cloud Glider Mariokart 8
```

```
## 8 2020-06-26      300 Morton   Streetle Slick      Cloud Glider  Mariokart 8
## 9 2020-07-03      293 Morton   Streetle Slick      Cloud Glider  Mariokart 8
## 10 2020-09-14     220 Morton   Streetle Cyber Slick MKTV Parafoil Mariokart 8
## # ... with 45 more rows
```

Each Verb's Abilities

From the website:

- Rows:
 - `filter()` chooses rows based on column values.
 - `slice()` chooses rows based on location.
 - `arrange()` changes the order of the rows.
- Columns:
 - `select()` changes whether or not a column is included.
 - `rename()` changes the name of columns.
 - `mutate()` changes the values of columns and creates new columns.
 - `relocate()` changes the order of the columns.
- Groups of rows:
 - `summarise()` collapses a group into a single row.

Dplyr Cheat Sheet

Visit this link to look at a cheat sheet for dplyr made by the creators of the package! It has some pretty neat visualizations, if you're more of a visual learner :)

Dataframes

To print out a dataframe (or any vector/object you're working with in R), you can just write its name.

```
mariokart

## # A tibble: 181 x 7
##   Date          Duration Character  Vehicle  Tires      Glider      Game
##   <date>         <dbl> <chr>     <chr>   <chr>     <chr>     <chr>
## 1 2020-12-28      115 Metal Mario B Dasher Red Monster Flower Glider Mariokart~
## 2 2021-02-02       79 Metal Mario Bruiser Slick      Swooper      Mariokart~
## 3 2020-12-15     128 Honey Queen B Dasher Slick      Beast Glider Mariokart~
## 4 2021-02-02       79 Metal Mario Bruiser Red Monster Super Glider Mariokart~
## 5 2020-12-12     131 Honey Queen Bruiser Slick      Peach Parasol Mariokart~
## 6 2020-09-14     220 Honey Queen B Dasher Red Monster Peach Parasol Mariokart~
## 7 2020-01-12     466 Metal Mario B Dasher Slick      Gold Glider  Mariokart~
## 8 2019-12-28     481 Honey Queen Bruiser Slick      Beast Glider Mariokart~
## 9 2021-02-10       71 Honey Queen B Dasher Slick      Beast Glider Mariokart~
## 10 2019-03-27     757 Yoshi      Egg 1    Wood      Super Glider Mariokart~
## # ... with 171 more rows
```

A shortcut I use to view objects as I'm coding it to select just the variable name (whether or not it is on a line with other code or not), and hit Command + Return if a Mac and Control + Enter if a Windows computer. This will run **just the code that is selected**, allowing you to view the object.

Graphs! (ggplot)

I'm going to give an example of what the code looks like for a `ggplot` graph!

Syntax

```
ggplot(data = your_data_here, aes(x = your_x_here,
                                   y = your_y_here_if_applicable,
                                   color = your_color_here_if_applicable,
                                   etc...)) +
  geom_your_geom_here()
```

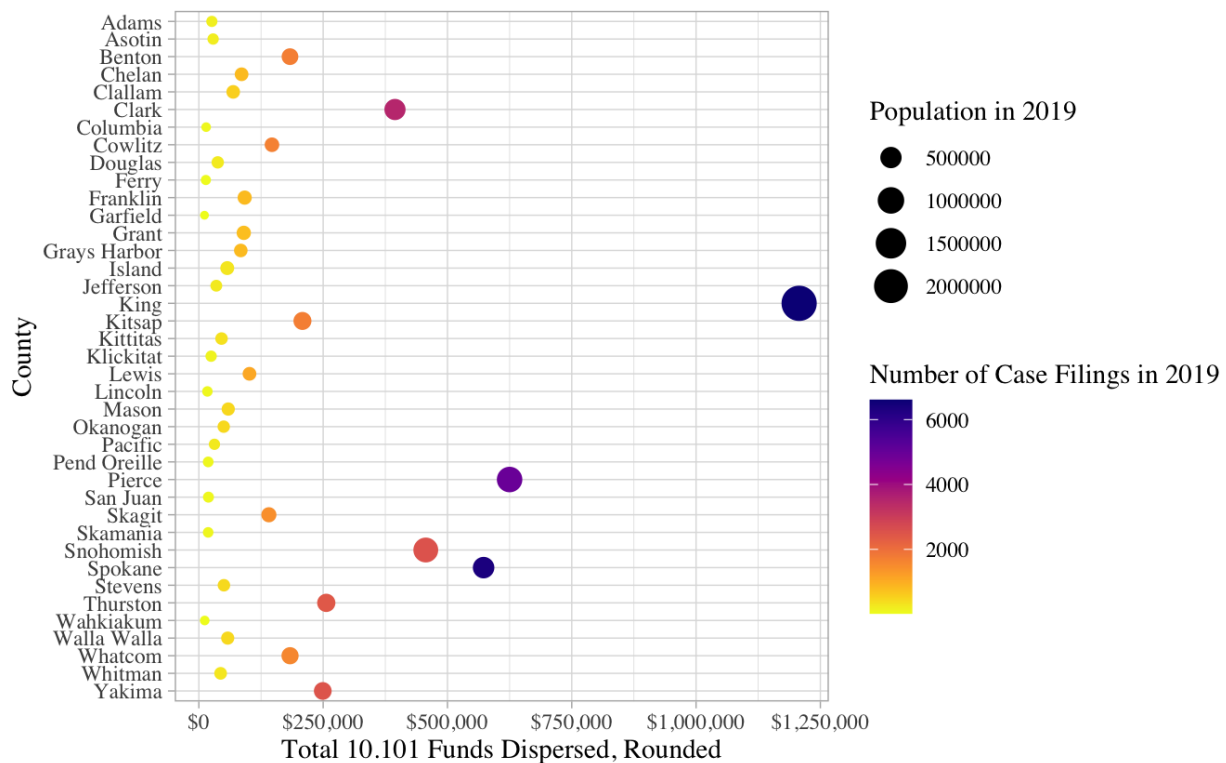
If you fill in all of the spots with your data and geoms/aesthetics of interest, you will get a `ggplot()`™!

What are these aesthetic things?

Aesthetics refer to all parts of your graph that you can use to show a variable! For instance, your x and y axes are aesthetics for the variables you display on them, as is things like color and size. See the below graph for an example that uses all of these aesthetics!

Factors Considered in the RCW 10.101 Model

Data from Katrin Johnson with OPD: 10.101 2020 county disbursement estimates.xlsx



© Robin Hardwick | Washington Defender Association

Geoms?

Geometries refer to the type of graph you want to create, or the geometry that the graph will take on. This will include points (scatterplots), bars (for histograms or bar plots), and so on and so forth. To see a comprehensive list of all the aesthetics and geometries you can manipulate in a `ggplot`, visit their cheat sheet here!

Functions

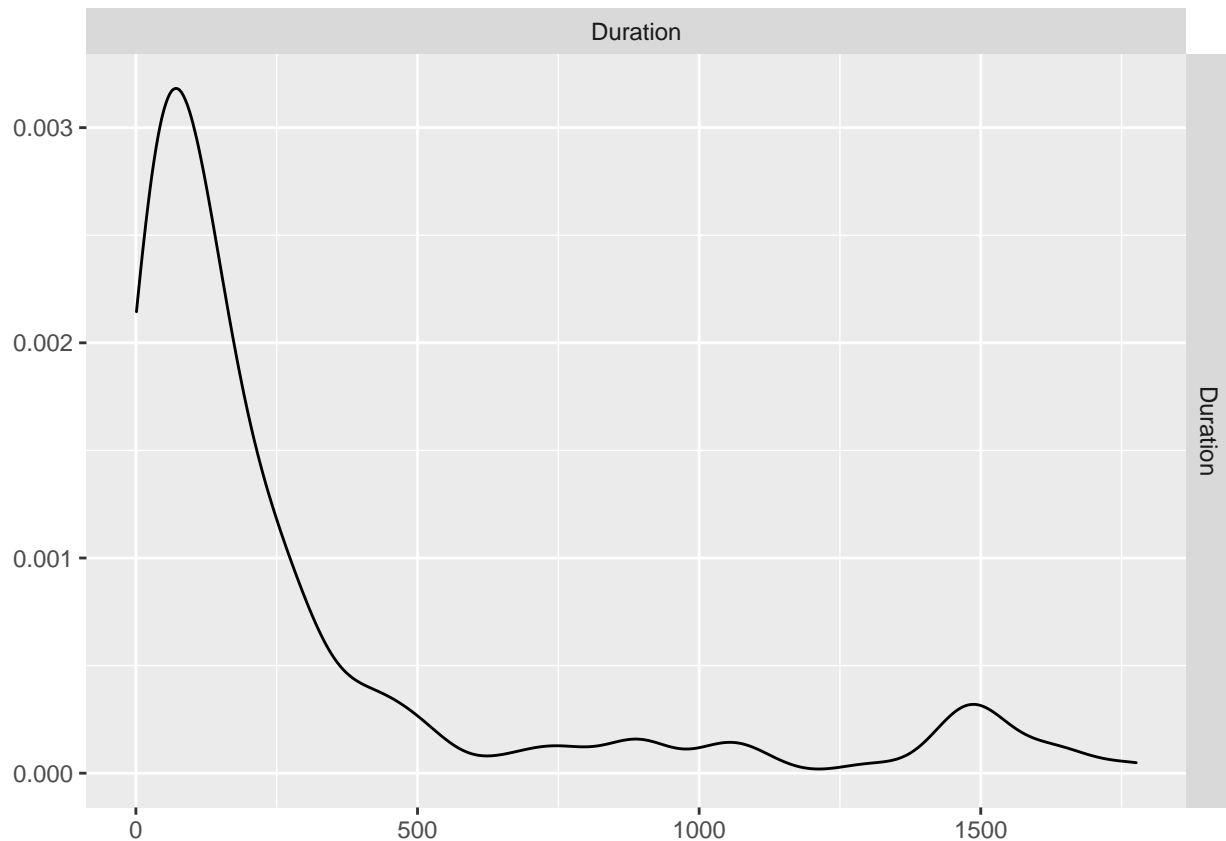
To make a function in R, you're essentially setting aside a bit of code, so you don't have to run the same code over and over again. Thus, you give the function an input and using the written code it will transform it into some sort of an output. See below for an example!

```
your_function_name <- function(input1, input2, ...) {  
  # Put all of the body of your code in these brackets!  
  y <- m * x + b  
  
  # To print your output, run the variable name  
  y  
  
  # Additionally, if you're just working with a couple of lines and do not  
  # need to save your data, you could just write  
  m * x + b  
  # for it to print your output.  
}
```

Correlations

Find correlations by running the `cor()` function and feeding in your dataframe! Additionally, you can feed in a data frame with all *numerical values* to the `pairs()` function to create a bunch of scatterplots. Also, the GGally library (made by Reed alum Grayson White), you can create scatterplots in ggplot!

```
library(GGally)  
mariokart_numeric <- mariokart %>%  
  select_if(is.numeric)  
  
ggpairs(mariokart_numeric)
```

(The Mariokart data isn't the best example here, as there is only one numeric variable, but for other datasets such as iris and county that have lots of numeric data, this will be super useful!)

Linear models

Below, I will walk through the syntax of a linear model!

```
# The lm stands for "linear model"
name_of_your_model <- lm(your_response_variable ~ your_explanatory_variable,
                          data = your_data_here)

# To view info on your model, use the summary command
summary(name_of_your_model)
```

This model will store the information required to make a line fit your data as best as possible, with the least distance from all of the points!

Another way to make a linear model (though without all the details you'd get above) is to make a scatterplot and fit a `geom_smooth` line to it. See below!

```
ggplot(data = your_data, aes(x = your_x,
                             y = your_y)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)

# This same method will also create three separate linear models if you've added in
# another aesthetics for a new variable!
```