# Need to know:

Pipes: shift + cmd + m

# Lab 1 commands:

* Basic operations/ Arithmetic
* Median, mean, ? for help
  + **Note** for mean, be aware for NA. if you want to remove the NA’s
    - Mean(vect, na.rm = T)
* “<-“ gets

# Lab 2 commands:

* **OPENING LIBRARIES**:
* library(“tidyverse”)
* library("janitor")
* library("skimr")
* library("palmerpenguins")
* Concocted
* spring\_list\_data <- c(spring\_1, spring\_2, spring\_3, spring\_4, spring\_5, spring\_6, spring\_7, spring\_8)
* <- matrix(data, nrow, byrow?)
* Creates a simple data matrix
* Use “” for non objects
* Colnames
* Rownames
* rowMeans
* rowSums
* Cbind(data matrix, vector)
* Binds a new column
* Matrix[x,y]
* Shows value for x = row, and y = column

# Lab 3 commands:

* <- data.frame(data)
  + Creates data frame
* Dim(data)
* Summary(data)
* Nrow(data)
* Ncol(data)
* anyNA(data)
* table(data$variable)
  + gives how much of different things in a variable
* filter(data, variable with operation)
* to create vector from data frame:
  + data$variable

# Lab 4 commands:

* **TO READ A CSV FILE:**
  + <- readr::read\_csv(“data/name.csv”)
    - Remember to set right directory
* Glimpse(data)
  + Gives list of variables and their classes
* Summary(data)
  + Gives statistical data along with variables
* Class(data$variable)
* Data$variable <- as.factor(data$variable)
  + Want to do when there are two of the same “groups” within a variable
  + Ex. When you have like a ‘vore’ class and a list follows 1. Carniv 2. Herbiv. 3. Carniv 4. Carniv. 5. Herbiv.
    - So you can group them later on
* Levels(data$variable)
  + Displays what type of “groups” are in the variable
* Select(data, variable[s])
  + For columns
* Filter (data, variable with operation)
  + For rows
* Sort()
* Arrange()
  + Arrange(desc()) for descending

# Lab 5 commands

*With tidyverse*

* Names(data)
  + Names of variables, good for checking bad names
* Rename(data, [new name] = “old name”)
  + Janitor::clean\_names(data) [**MUCH BETTER**]
* Mutate\_if(class, new class)
* KNOW HOW TO PIPE
* Mutate\_all(tolower)
  + Lowercases everything
* Mutate(across(c(“var1”, “var2”), tolower))
  + Lowercases across the range of variables from var1 to var2
* **IF ELSE TO REMOVE -999 VALUES**
  + mammals %>%
    - select(genus, species, newborn) %>%
    - mutate(newborn\_new = ifelse(newborn == -999.00, NA, newborn)) %>%
    - arrange(newborn)
* General mutation
  + Data %>%
    - Mutate(new variable = var1 with operation)
* Tabyl(data, variable)
  + Gives # of cells associated with the groups in the variable
* Filter(!grepl(If a “string” is in any group, variable)
  + Ex. If you want to get any race associated with humans in a list of superheroes:
    - superhero\_info %>%
      * filter(!grepl("Human", race))
* If you want to get the all variables that satisfies a condition:
  + Ex. Trying to highlight only all of Doctor Doom’s powers:
    - superhero\_powers %>%
      * filter(hero\_names == "Doctor Doom") %>%
      * select\_if(all\_vars(.=="TRUE"))
  + This will only highlight only the cells that are “TRUE”

# Lab 6 commands

* Skim(data)
  + Gives a brief overview of the dataset
* Hist(data)
  + Gives histogram of data
* Summarize( new variable = operation on a variable)
  + Think like mutate where you can find the mean, min, max and total ‘n’ of observations
  + Summarise(n\_new variable = **n\_distinct(variable)**
    - Shows number of distinct observations and creates a new column
* Group\_by(variable)
  + Groups observations in a variable and you can connect it with summarize functions
* Count(variable, sort = T/F [T for descending order])
  + Counts how many observations within a variable.
  + Ex wanting to count how many penguins there are on each island.