|  |
| --- |
| **Loading in Packages**  library(tidyverse)  Loads a package into the R workspace, so you can use the functions and data it contains |
| **Reading in Data**  IPEDS <- read\_csv(here::here("data",  "<NAME OF DATASET.csv>")  )  ***Note:*** The name of the dataset will change, but it will always need to have the .csv at the end of its name! |
| **Assignment Arrow**  penguins\_2007 <- filter(penguins, year == 2007)    Assigns a value (e.g., dataframe) to the name of a variable |
| **Preview a Dataset**  glimpse(<NAME OF DATASET>) |
| **Filtering a Dataset**  large\_adelie\_2008 <- filter(penguins,  species == "Adelie",  body\_mass\_g > 3000,  year == 2008)  Filters observations (rows) out of / into a dataframe, where the inputs (arguments) are the conditions to be satisfied in the data that are kept  ***Note:*** It makes your code more readable if you put each filter on a new line (hit enter after each comma)! |
| **Mutating a Dataset**  penguins\_large <- mutate(penguins,  body\_mass\_kg = body\_mass\_g / 1000)  Creates new variables or modifies existing variables |

|  |
| --- |
| **The Pipe Operator**  penguins %>%  select(species, island, body\_mass\_g, sex, year) %>%  filter(island == "Torgersen",  body\_mass\_g > 1000) %>%  summarize(mean\_mass = mean(body\_mass\_g),  median\_mass = median(body\_mass\_g),  observations = n()) %>%  arrange(desc(mean\_mass))  Sequences of data wrangling steps together, where each step is connected with “and then”. For example, in the above code, we take the penguins data **and then** select specific columns, **and then** filter out specific observations, **and then** summarize the body mass variable, **and then** arrange the body masses in descending order. |
| **Calculating Summary Statistics for Numerical Variables**  summarize(<NAME OF DATASET>,  <NAME OF STAT> = <STAT FUNCTION>(<NAME OF VARIABLE>)  )  For example, to calculate the mean and median of the dep\_delay variable from the nycflights dataset we have:  summarize(nycflights,  mean\_dep\_delay = mean(dep\_delay),  median\_dep\_delay = median(dep\_delay)  ) |
| **Histogram**  ggplot(data = <NAME OF DATASET>,  mapping = aes(x = <NAME OF VARIABLE>)) +  geom\_histogram(binwidth = <WIDTH OF BINS>) +  labs(x = "<TITLE FOR THE X-AXIS>")  ***Note:*** A histogram **must** have the variable on the x-axis! |
| **Boxplot**  ggplot(data = <NAME OF DATASET>,  mapping = aes(x = <NAME OF VARIABLE>)) +  geom\_boxplot() +  labs(x = "<TITLE FOR THE X-AXIS>")  ***Note:*** This boxplot is horizontal. If you want for your boxplot to be vertical, you use **y =** instead of **x =** . Keep in mind you will need to change the location of you axis label, too! |
| **Scatterplot**  ggplot(data = <NAME OF DATASET>,  mapping = aes(x = <NAME OF X-VARIABLE>,   y = <NAME OF Y-VARIABLE>)  ) +  geom\_point() +  labs(x = "<TITLE FOR THE X-AXIS>",  y = “<TITLE FOR THE Y-AXIS>”) |
| **Scatterplot with Regression Line**  ggplot(data = <NAME OF DATASET>,  mapping = aes(x = <NAME OF X-VARIABLE>,   y = <NAME OF Y-VARIABLE>)  ) +  geom\_point() +  geom\_smooth(method = “lm”) +  labs(x = "<TITLE FOR THE X-AXIS>",  y = “<TITLE FOR THE Y-AXIS>”) |