Homework 2

Name: Arthur Viegas Eguia

I worked with:

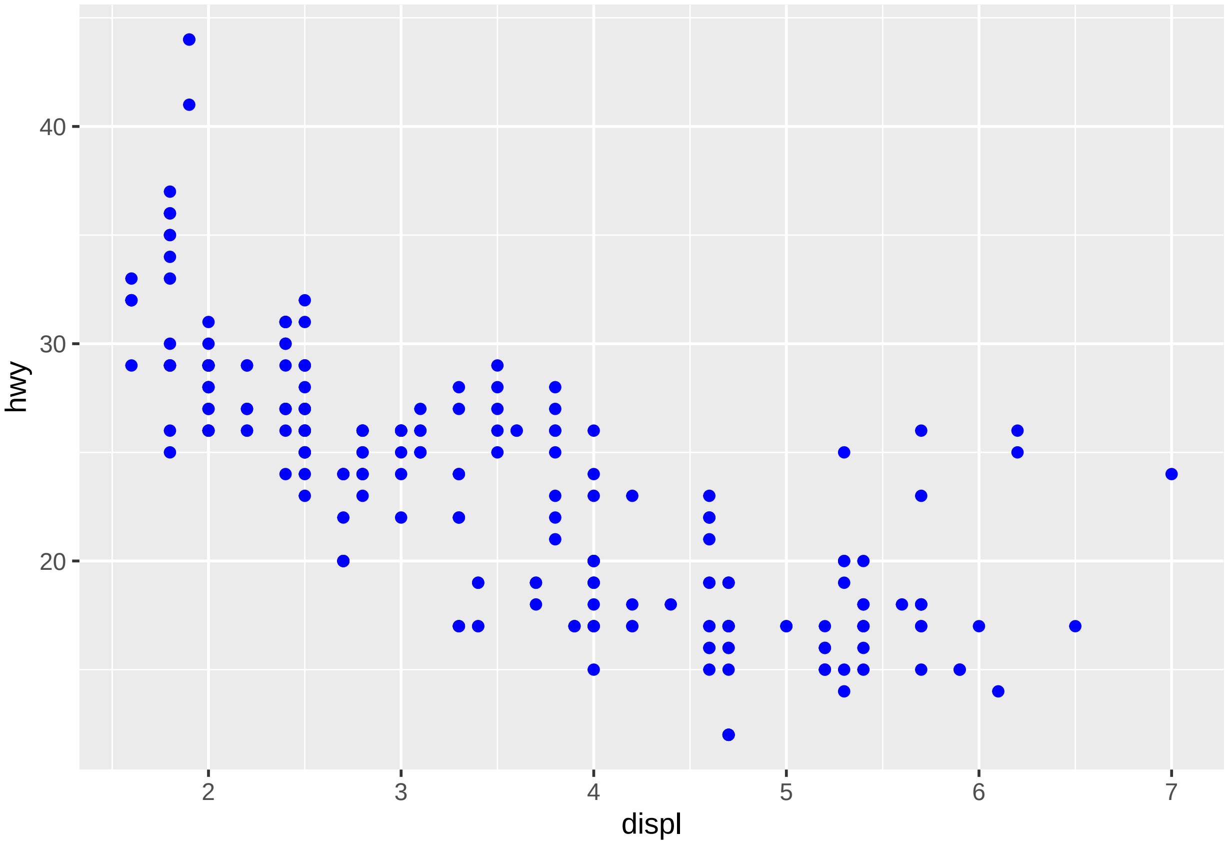
Click the “Knit” button in RStudio to knit this file to a pdf.

Problem 1: Spot the error

answer: When colot is inside aes, it is used as a third dimension attribute (adding an extra dimension to our data). It is something that can be used, for example, to outline points that are objects of different classes (in this case one might use color to outline the different years of the car model. This would print each different year as a different color (more precisely, a different shade of the same color)). When the color attribute is in aes, you should do something like color = column_of_your_dataframe. This happens because anything inside aes will be looking at the columns on data. You should not have something like color = “string” inside aes.

If we put color outside aes, it will work as expected

```
library(ggplot2)
head(mpg)
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color= "blue")
```



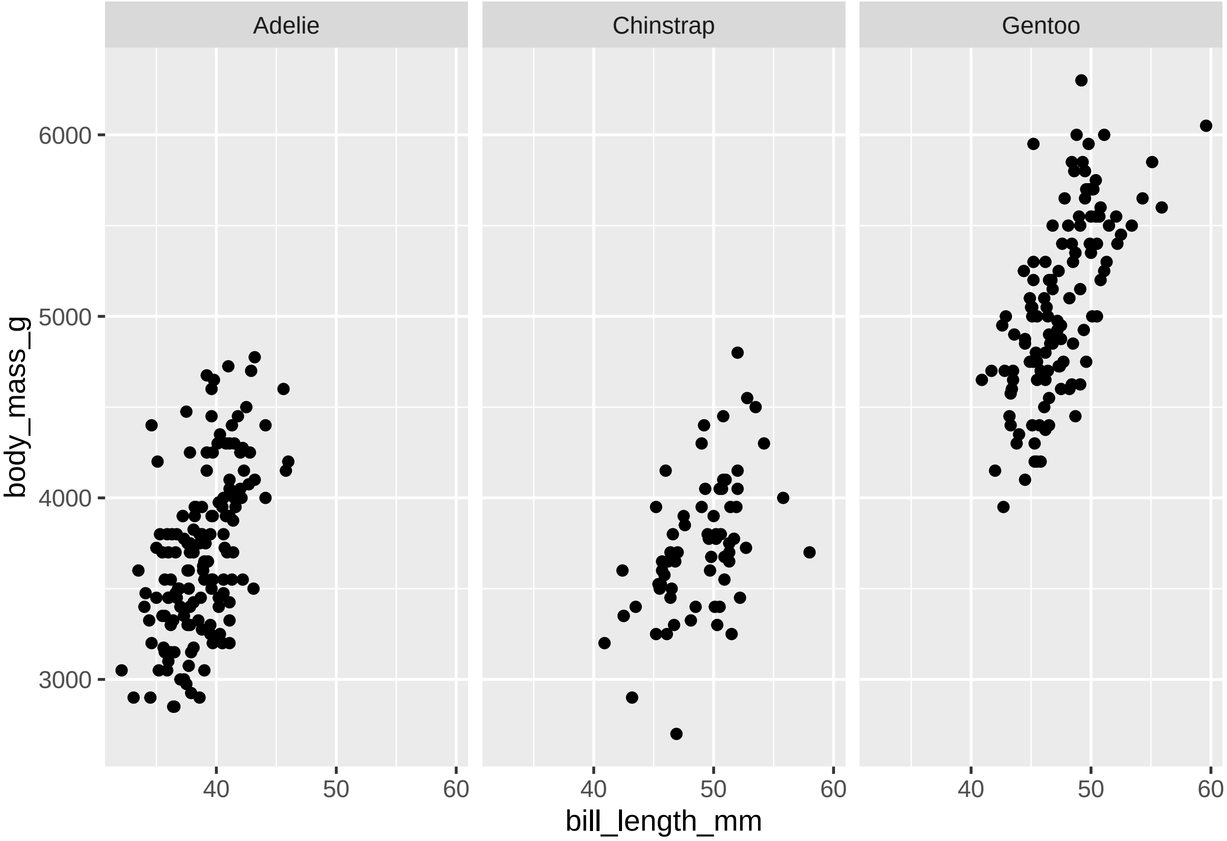
Problem 2: Penguins

```
data(penguins, package = 'palmerpenguins')
```

a.

answer: write your answer here

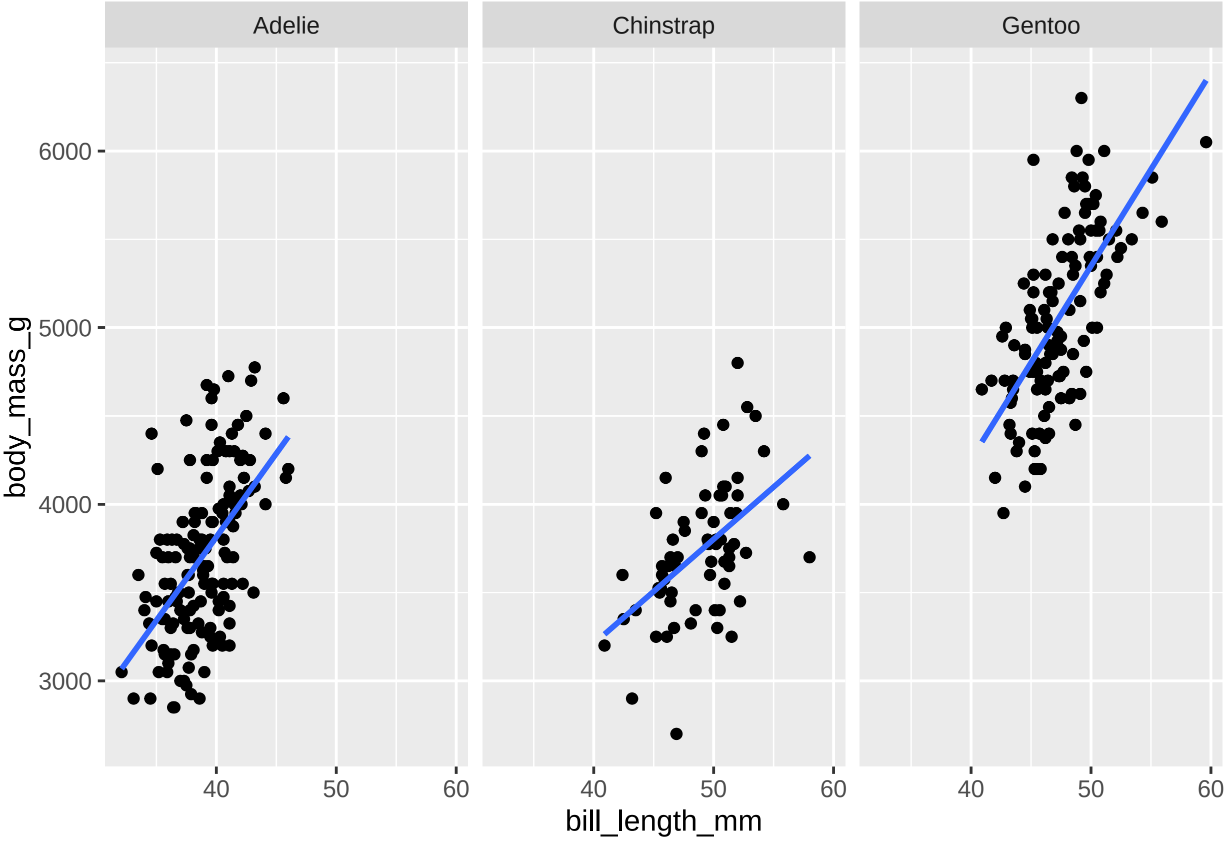
```
ggplot(penguins, aes(y = body_mass_g, x = bill_length_mm)) + geom_point() + facet_wrap(~species)
```



b.

answer: write your answer here

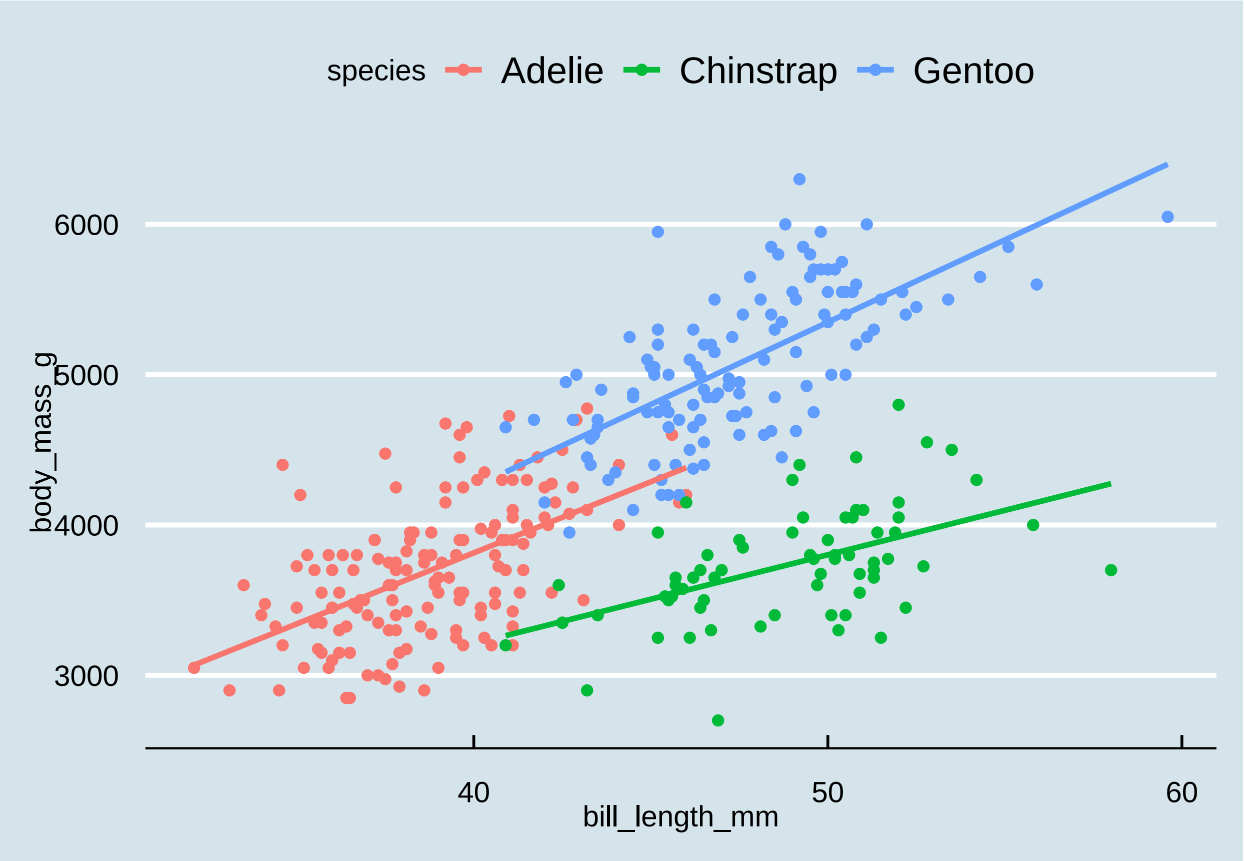
```
ggplot(penguins, aes(y = body_mass_g, x = bill_length_mm)) + geom_point() + facet_wrap(~species) + geom_
```



c.

answer: write your answer here

```
ggplot(penguins, aes(y = body_mass_g, x = bill_length_mm,
                     color = species)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE) +
  ggthemes::theme_economist()
```



d.

answer: Certainly c is easier to compare the slopes. All the lines are on the same plot, with the same scale, which makes it easy to visualize their slopes, and their differences. The colors also help make the plot be more intuitive and easier to interpret.

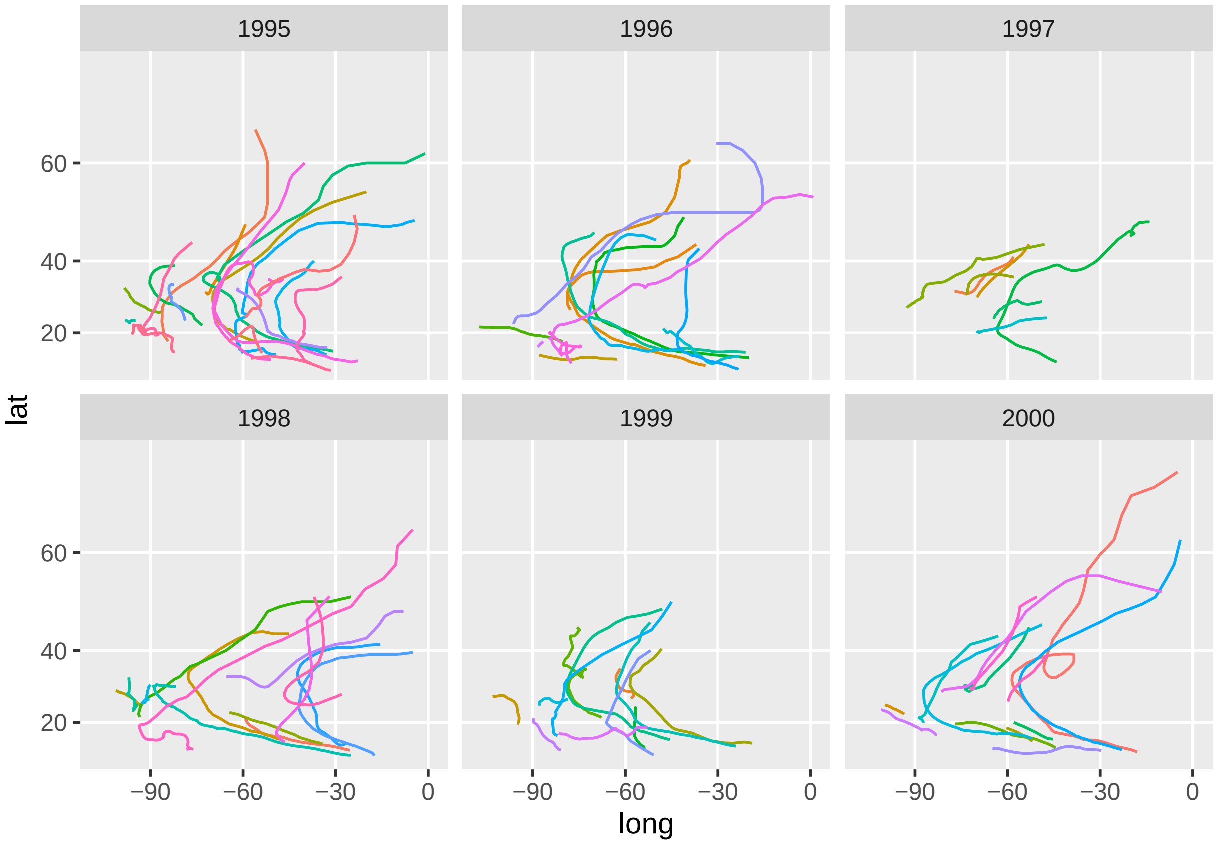
Problem 3: Storm paths by year

```
# install.packages("nasaweather")
data(storms, package = "nasaweather")
```

a.

answer: write your answer here

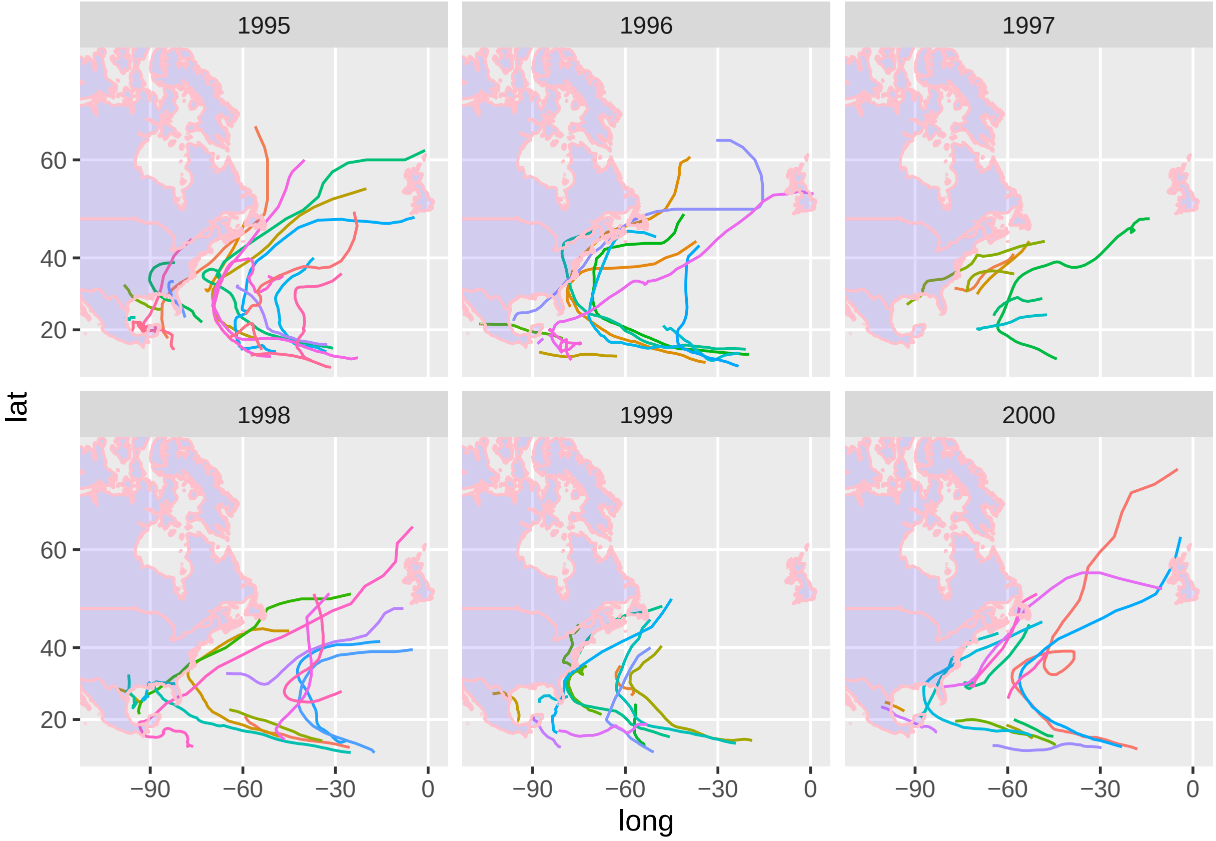
```
storm.path <- ggplot(storms, aes(x = long, y = lat)) +
  geom_path(aes(color=name)) +
  facet_wrap(~year) +
  scale_color_discrete(guide="none")
storm.path + coord_map()
```



b.

answer: write your answer here

```
country_data <- map_data("world", region = c("usa", "mexico", "canada", "uk"))
storm.path + geom_polygon(data = country_data,
                          aes(x = long, y = lat, group = group),
                          alpha=0.2, col="pink", fill = "#7050ff") +
  coord_map(xlim = range(storms$long),
```



Problem 4: explain command (no R needed)

```
mydata <- data.frame(classType = c('C', 'C', 'C', 'S', 'S'),
                     m = c(10, 3, 7, 2, 7),
                     w = c(4, 1, 3, 7, 10))
mydata
##   classType  m  w
## 1         C 10  4
## 2         C  3  1
## 3         C  7  3
## 4         S  2  7
## 5         S  7 10
```

a.

answer: Let's go by parts. mydata %>% filter(classType == "C") takes the dataset mydata and sends it to filter. Filter gets the dataset mydata and returns only the rows where the classType is C. In this case, it will return the first three rows. This updated dataset will be sent to select, which will choose the columns m and w from the dataset (the end result will not have classType).

In other words, what it is doing, is for all CS classes it prints the number of Mac users and the number of Windows users.

So, the final output will be like

m	w
10	4
3	1
7	3

b.

answer: mydata %>% mutate(ratioW = w/sum(w)) Takes the dataset mydata and sends it to the mutate function. Mutate will create an extra column in the result (the name of the column is ratioW). The value of this extra column for each row will be w/sum(w). Where, w is the value of w in the current row, and sum(w) is the sum of all w in the dataset.

This is the ratio of windows users who are taking a specific class.

The results will be like

classType	m	w	ratioW
C	10	4	0.16
C	3	1	0.04
C	7	3	0.12
S	2	7	0.28
S	7	10	0.40

c.

answer: Here mydata %>% group_by(classType) takes the mydata dataset and sends it to function group_by, which is similar to the SQL command GROUP BY. The function group_by will group the rows based on column, classType. So, CS classes will be grouped together and statistics classes will be grouped together. This modified dataset will then be sent to mutate, that will create a new column (that will be named ratioW). The value of ratioW will be w/sum(w). Here w is the value of the column w in the current row. However, sum(w) is the sum of all w within a group (so, the sum of all w where classType == 'C', and the sum of all w where classType == 'S'). w/sum(w) gets the value of w in the current row and divides it by the sum of w of the group of the current row.

This is effectively the ratio of windows computers in each class (grouped by the subject of the class, CS or Statistics).

The result is going to be the following

classType	m	w	ratioW
C	10	4	0.5
C	3	1	0.125
C	7	3	0.375
S	2	7	0.412
S	7	10	0.588

d.

answer: Here, mydata %>% group_by(classType) takes the dataset mydata and sends it to group_by(classType). group_by will group the data on this dataset depending on the values on column classType. So, two groups will be created, one for rows with value C and one for rows with value S. Then, %>% summarize(Y = sum(w+m)) will send this dataset to function summarize. Function summarize will print the values of the sum of w and m for groups C and S. The sum of w and m is all the values of w in a group plus all the values of m in the same group. The sum will be in a column named Y.

This represents the total number of mac and windows machines in CS classes, and the number of mac and windows machines at Stats classes.

The result will be the following

classType	Y
C	28
S	26

e.

answer: Here, mydata %>% group_by(classType) sends the dataset mydata to the group_by function, that will group the data based on the values on column classType. Two groups will be created, one for when classType is C (for the CS classes) and one for when classType is S (for Stats classes).

So %>% mutate(X = w+m, Y = sum(w+m)) will send the results of the previous operation to mutate, which will create two new columns on the dataset. Column X which is the sum of the values of w and m in the current row.Y is the sum of all the ws and ms of the group of the current row.

THis represents the total number of windows and mac computers in the current class, and the total number of windows and mac computers across all CS, and across all Stats Classes.

The results are the following

classType	m	w	X	Y
C	10	4	14	28
C	3	1	4	28
C	7	3	10	28
S	2	7	9	26
S	7	10	17	26