notes

Intro.

Using the mpg data set, we can get a start to fucking up daddy fung. Make sure u get all the releveant libraries coz ur gonna need to be using them.

ytb

to load up the data, just write the name of the data set in the console, it'll give a fat tibble table. if u dunno anything, just put a ? mark inf front and you get a definition.

```
mpg
```

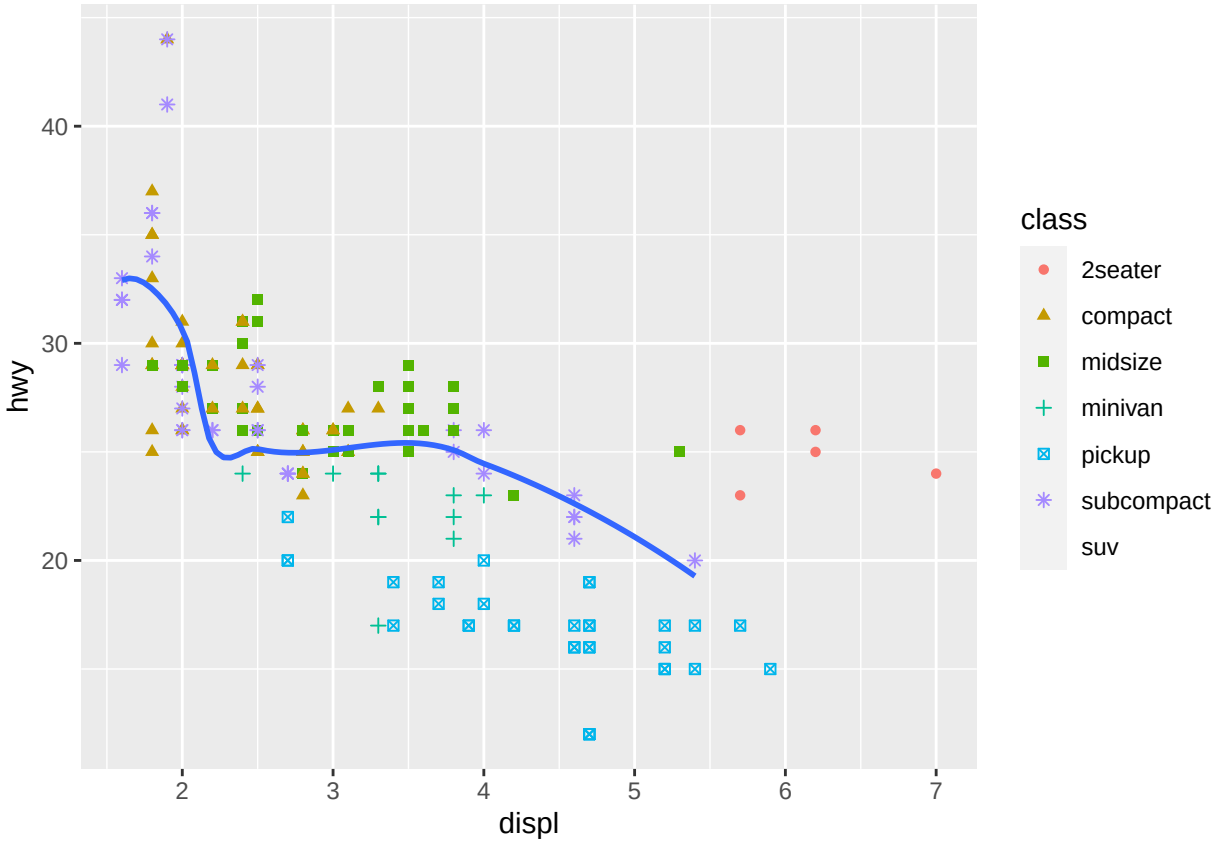
```
## # A tibble: 234 x 11
##    manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
##    <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
##  1 audi         a4           1.8  1999     4 auto~ f        18    29 p     comp~
##  2 audi         a4           1.8  1999     4 manu~ f        21    29 p     comp~
##  3 audi         a4           2    2008     4 manu~ f        20    31 p     comp~
##  4 audi         a4           2    2008     4 auto~ f        21    30 p     comp~
##  5 audi         a4           2.8  1999     6 auto~ f        16    26 p     comp~
##  6 audi         a4           2.8  1999     6 manu~ f        18    26 p     comp~
##  7 audi         a4           3.1  2008     6 auto~ f        18    27 p     comp~
##  8 audi         a4 quattro   1.8  1999     4 manu~ 4        18    26 p     comp~
##  9 audi         a4 quattro   1.8  1999     4 auto~ 4        16    25 p     comp~
## 10 audi         a4 quattro   2    2008     4 manu~ 4        20    28 p     comp~
## # ... with 224 more rows
```

```
library(tidyverse)
library(ggplot2)

carData <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class, shape = class)) +
  geom_smooth(data = filter(mpg, class == "subcompact"), se = FALSE)

carData
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 62 rows containing missing values (geom_point).
```

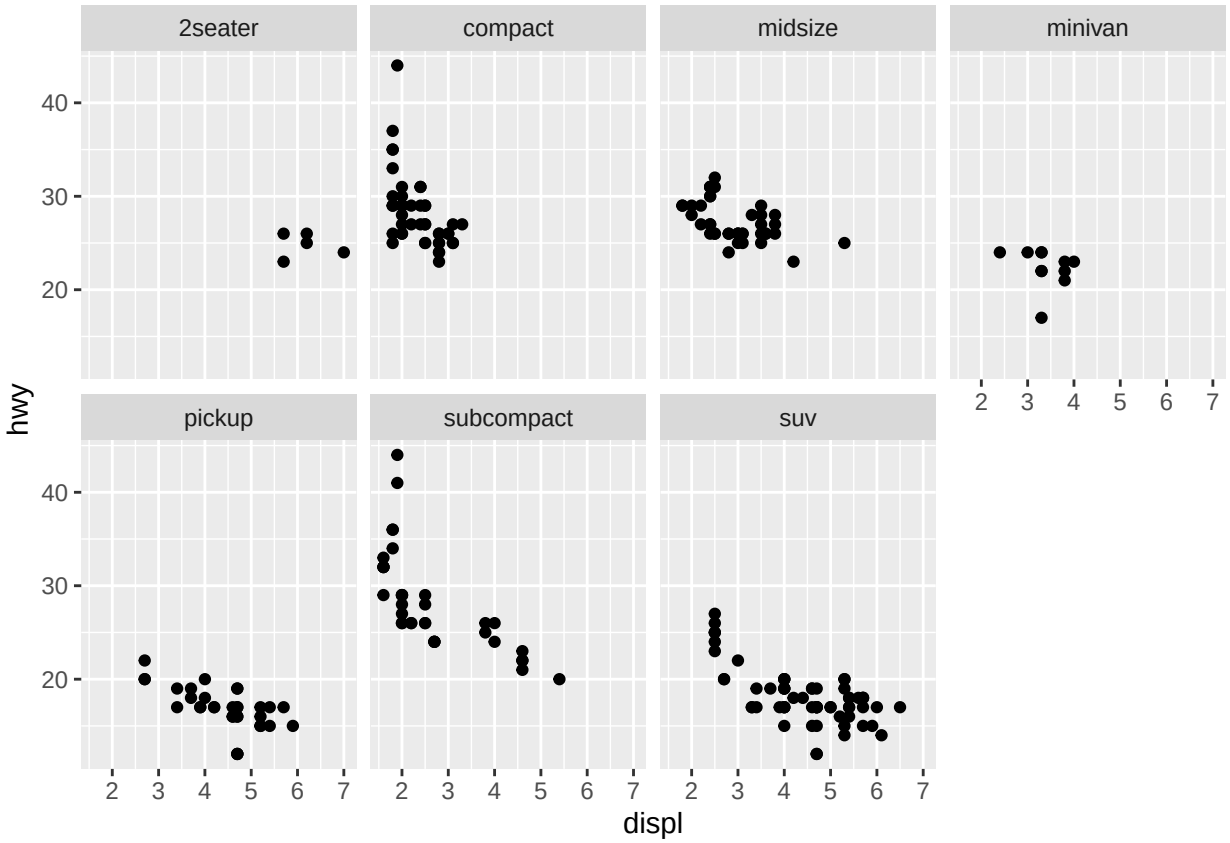


This carData graph uses data mpg, which also has a mapping argument, this makes the data "Mapped" to certain properties. In this case, the x and y axis is displ and hwy, this is easily exchanged for other shit. geom point is the dots on the plot, which can be any shape / color depending on how you set it. You can either go LGBTQ+ or set it to a random variable. Up to you. Geom Smooth is the line given, which we can use the filter function to find and make a line of best fit for the shit that we want. In this case, we have to declare the data used and what category that we want.

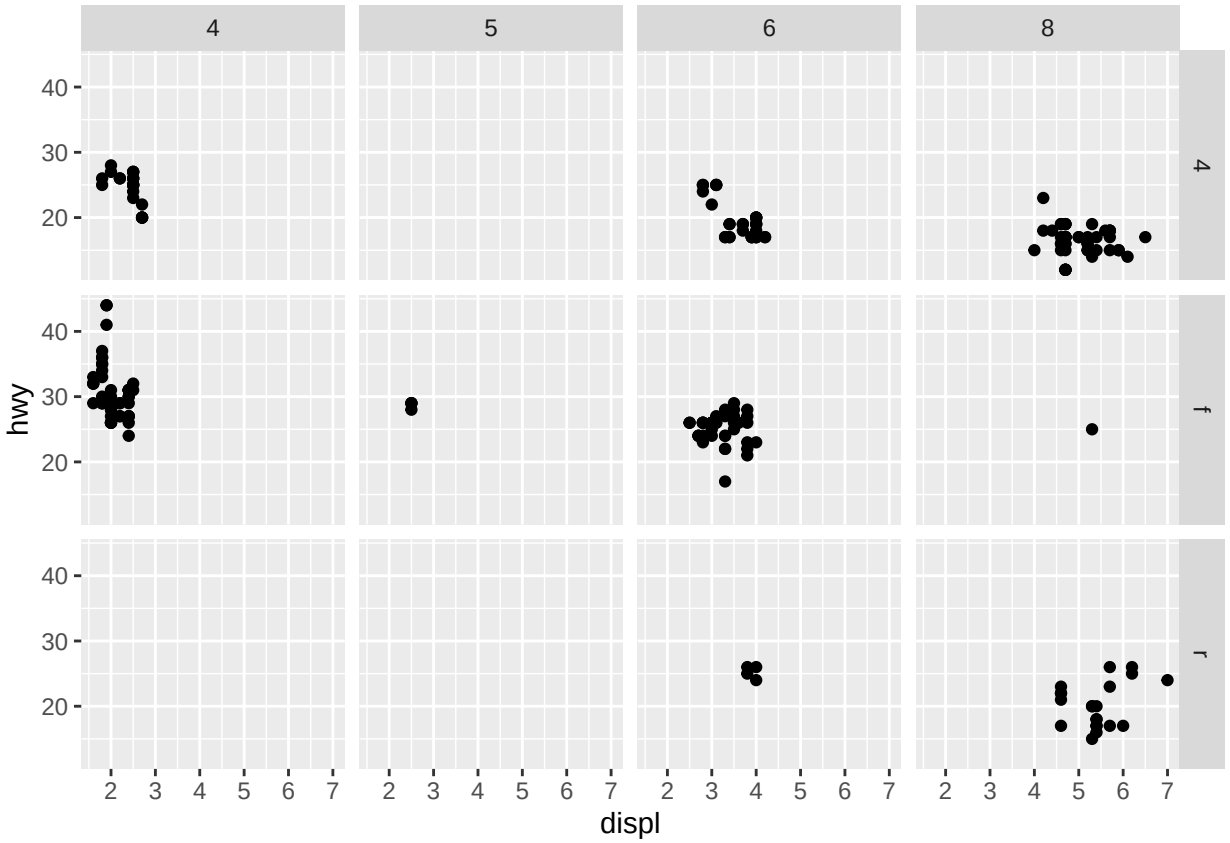
idk what the fuck se is lmao

Facuet Failure.

Facet is a way to seperate the data into different graphs so its not clunky as fuck. Facet_wrap is to set how many rows int he data. We can also put same x and y variables in the data to save space.




```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ cyl)
```



What does facet wrap do? fuck use this ? but, it wraps your data into different sets. Making the shit look pretty.

Stuff from the lecture.

Using the data given from Thomas, we are given an encoder and a data of the pulses of some people.

```
dat = read.table("pulse.dat", header=TRUE)
```

Something that reads the data, a header is a logical variable which shows which working directory its in.

```
attach(dat)
```

We are making the data set saved, and easy to find.

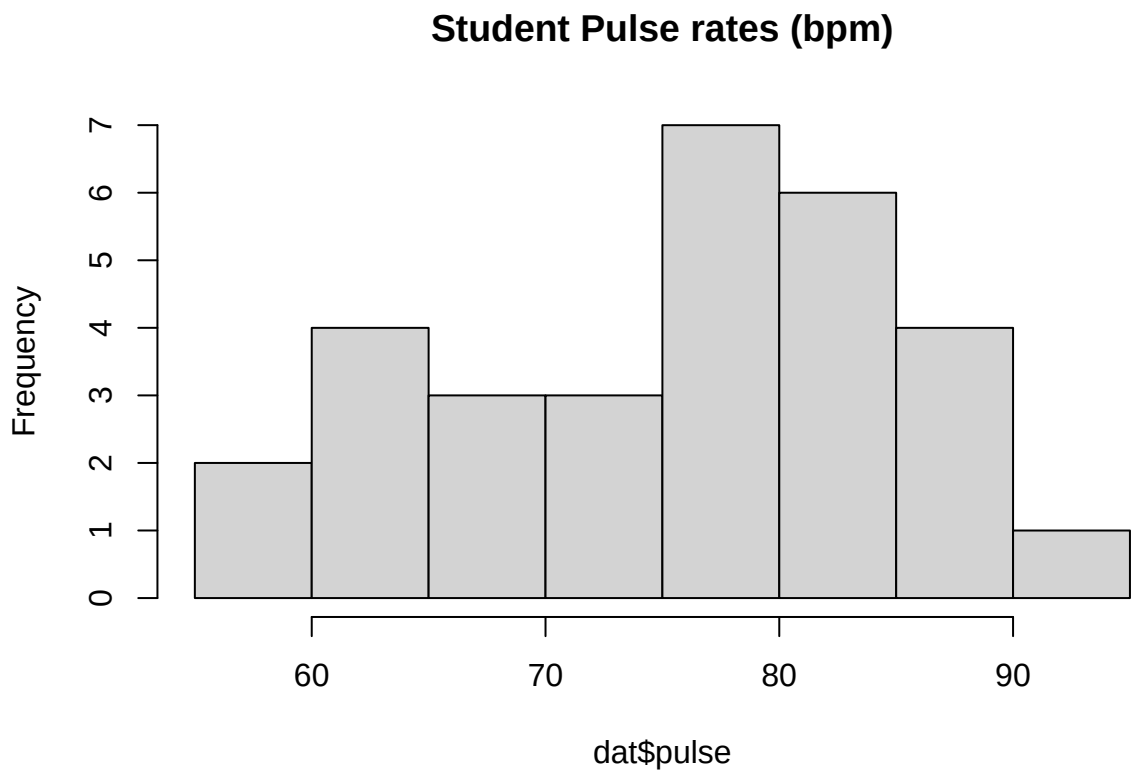
```
stem(dat$pulse)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   5 | 56
##   6 | 12
##   6 | 55689
##   7 | 133
```

```
##   7 | 667777
##   8 | 01112
##   8 | 556667
##   9 | 3
```

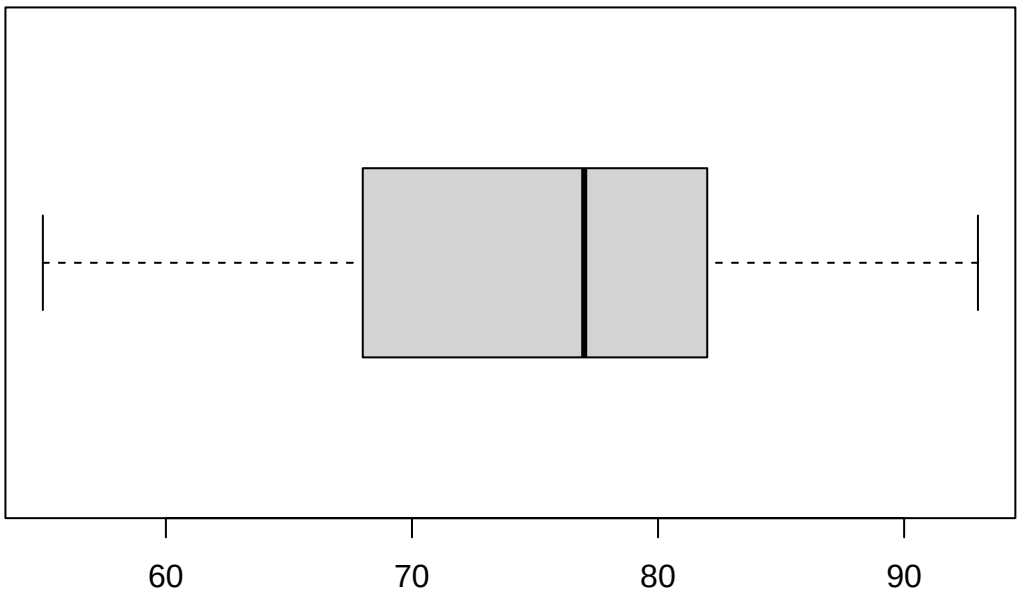
Creates a stem and leaf plot, from the sub category pulse, pulse has been "Head" From the data, which means we take a single column from the data set.

```
hist(dat$pulse, main = "Student Pulse rates (bpm)")
```



Creates a histogram of the data, which in this case is Pulse

```
boxplot(dat$pulse, horizontal = TRUE,
        main = "Student Pulse rates (bpm)")
```



Same shit, but a box plot.

```
summary(dat$pulse)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   55.00   68.25   77.00   75.23   81.75   93.00
```

Gives General Information of the data set, which we can see from the results the Min, 1st Quartile, Median, Mean, 3rd Quartile and Max is showing

```
sd(dat$pulse)
```

```
## [1] 9.729739
```

Standard deviation.

```
mean(dat$pulse)
```

```
## [1] 75.23333
```

Mean

This is Thomas' T test, I have altered it slightly to automate to reject or do not reject the null hypothesis.

```r
dat1 = read.table("encoder.txt", header = TRUE)
str(dat1) # Look at the data object
```

```
## 'data.frame':    10 obs. of  1 variable:
##  $ time: num  18.3 17.9 19.1 16.8 18.9 17.4 19.6 18.3 19.6 16.3
```

```r
dat1 # Look at all observations
```

```
##    time
## 1  18.3
## 2  17.9
## 3  19.1
## 4  16.8
## 5  18.9
## 6  17.4
## 7  19.6
## 8  18.3
## 9  19.6
## 10 16.3
```

```r
head(dat1) # display first six observations
```

```
##   time
## 1 18.3
## 2 17.9
## 3 19.1
## 4 16.8
## 5 18.9
## 6 17.4
```

```r
summary(dat1$time) # obtain descriptive statistics
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.30   17.52   18.30   18.22   19.05   19.60
```

```r
sd(dat1$time)^2 #Find the SD.
```

```
## [1] 1.281778
```

```r
pValue <- t.test(dat1$time, mu = 19.25)$p.value
# ifelse(bob < 0.05 <-yes = print("Reject"), no = print("Do not reject"))
if(pValue < 0.05) {
  print("Reject")
} else {
  print("Do Not Reject")
}
```

```
## [1] "Reject"
```

```r
# ifelse(pValue<0.05, print("Reject"), print("Don't reject"))
# t.test(dat1$time, mu = 19.25, alternative = "less")

# t.test(dat1$time, mu = 19.25, alternative = "greater")
```

Okay, we didn't know why the fuck mu is 19.25, mu is the mean of the POPULATION When we find the descriptive statistics of the data, we realize the mean is 18.22, lol what the fuck right? No, this is coz this is this the mean of the SAMPLE

When we are constructing our hypothesis, we will need to create a Null Hypothesis and a Alternative Hypothsis.

In this case, the sample mean is NOT 19.25, but 18.22 from our t test. Therefore, We reject the null hypothesis AND the sample mean was significantly less than the population

— If the ***p-value is less than or equal to 0.05 (critical value), the null hypothesis is rejected.*** Hence, the conclusion to be drawn is that there is *a difference* between means of heights of the population from two countries.

— If the ***p-value is greater than 0.05 (critical value), the null hypothesis is accepted.*** Hence, the conclusion to be drawn is that there is *no difference* between the means of heights of the population from two countries.

Figure 1: **Figure here shows when to reject or do not reject.**

Anyway, thats all the notes for this week. Thomas, fuck you.