

# Ch 3 and 4 Overview

Ryan Meyer

4/22/2022

## Contents

<b>Libraries</b>	<b>1</b>
<b>Chapter 3: plotting</b>	<b>2</b>
Starting with some useful geometries	2
A useful note	9
Adding Aesthetics to these	11
Getting a bit more advanced, mutating things within ggplot	17
Stat Summary, one of the best features of ggplot	19
Stat summary will do summary stats for you and plot them it great . . . . .	19
Facet grid and facet wrap - useful for data exploration	22
Last thing, making graphs pretty	24

## Libraries

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.6      v dplyr  1.0.8
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

## Chapter 3: plotting

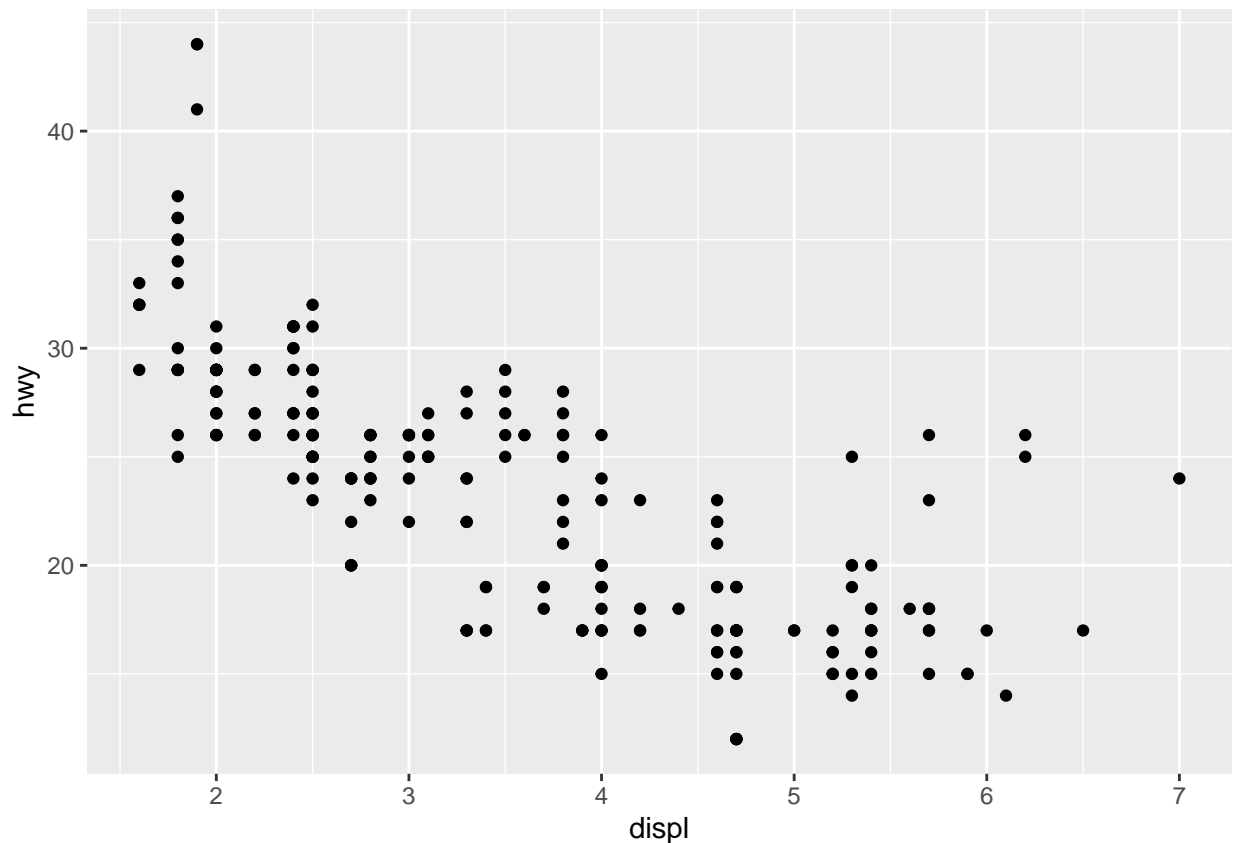
The syntax of plotting in ggplot2: `ggplot(data = data, mapping = aes(x = x, y = y, any other aes here)) + geometries_here()`

This can be shortened, I often do this `data %>% ggplot(aes(x,y)) + geom_here()`

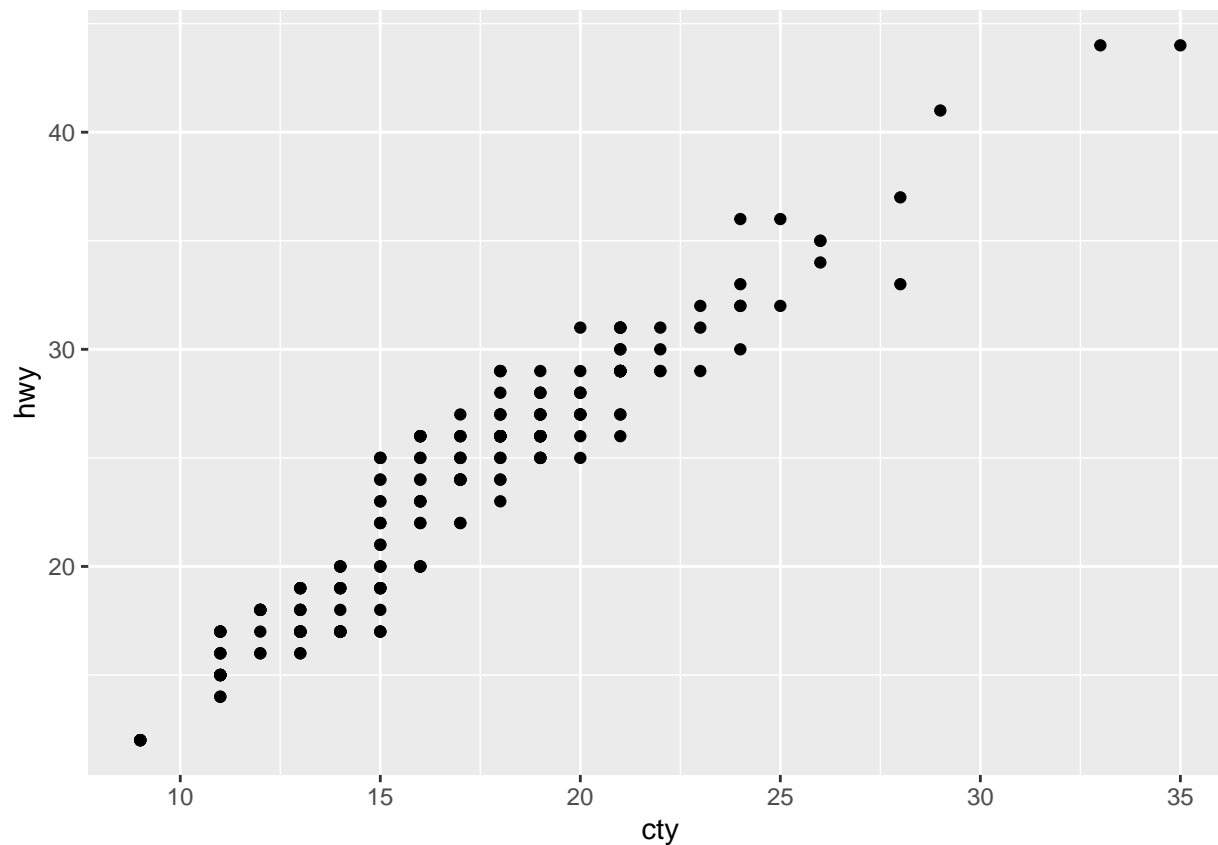
### Starting with some useful geometries

R has built in data, I'm going to be using the motor trend car data because cars are cool

```
# many different ways to write the code, lots of ways to do this!
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



```
# vs.
mpg %>%
  ggplot(aes(cty, hwy)) +
  geom_point()
```

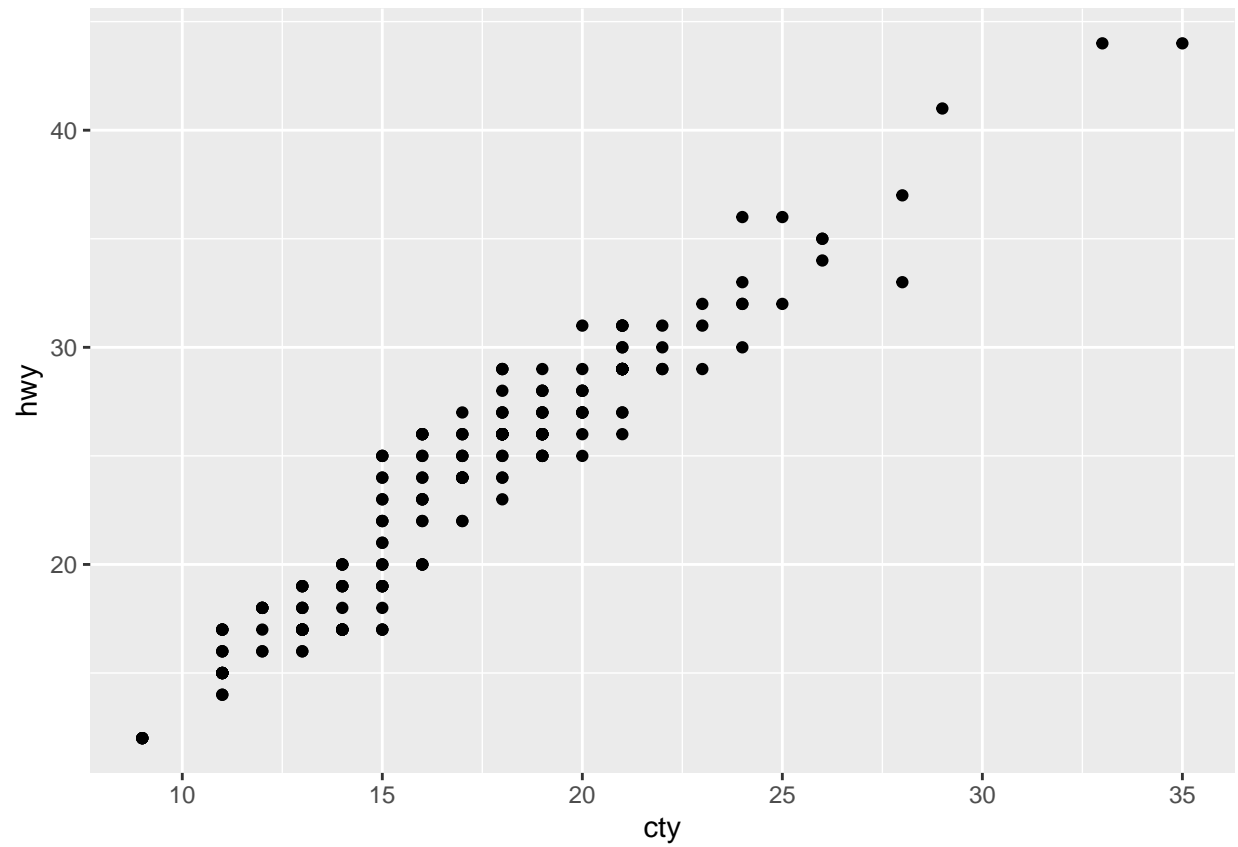


```
# some basic and useful geometries
```

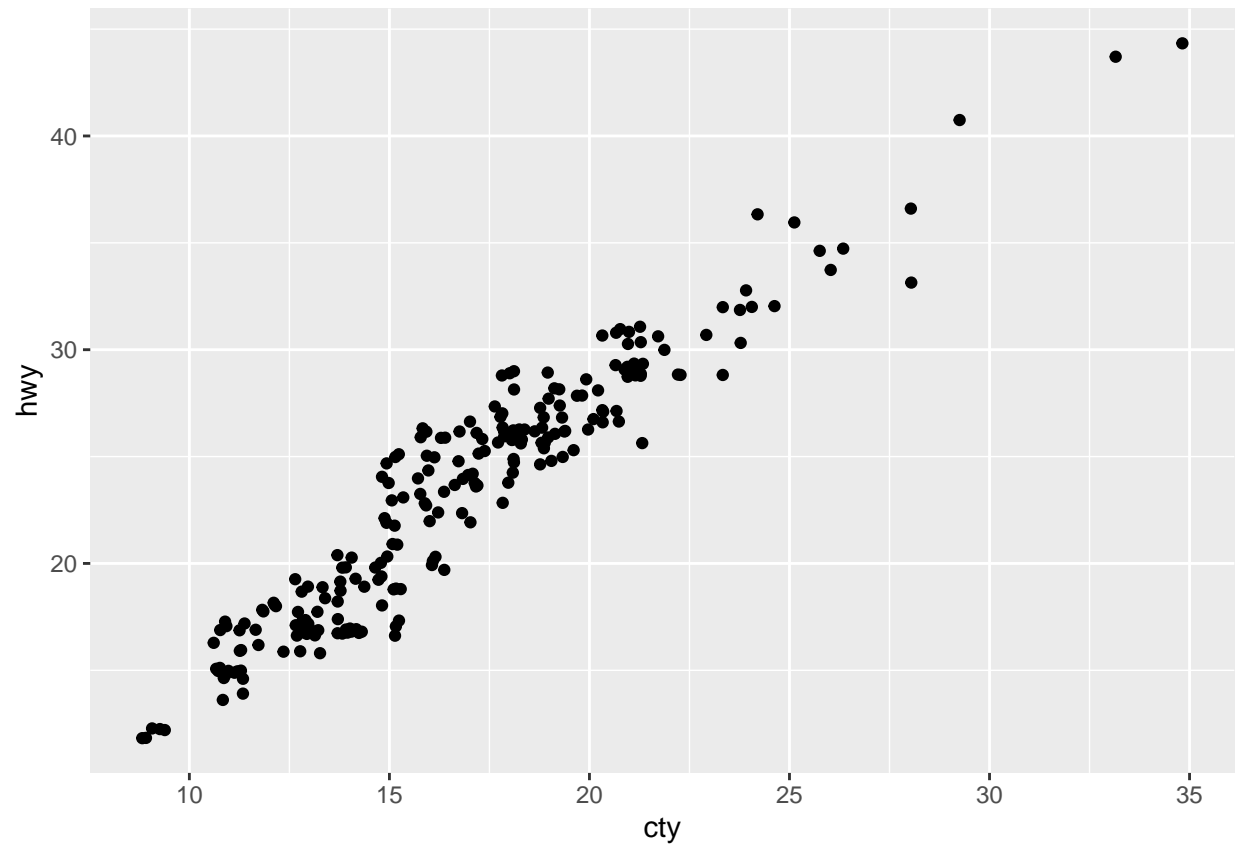
```
mpg
```

```
## # 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
```

```
mpg %>%
  ggplot(aes(cty, hwy)) +
  geom_point()
```

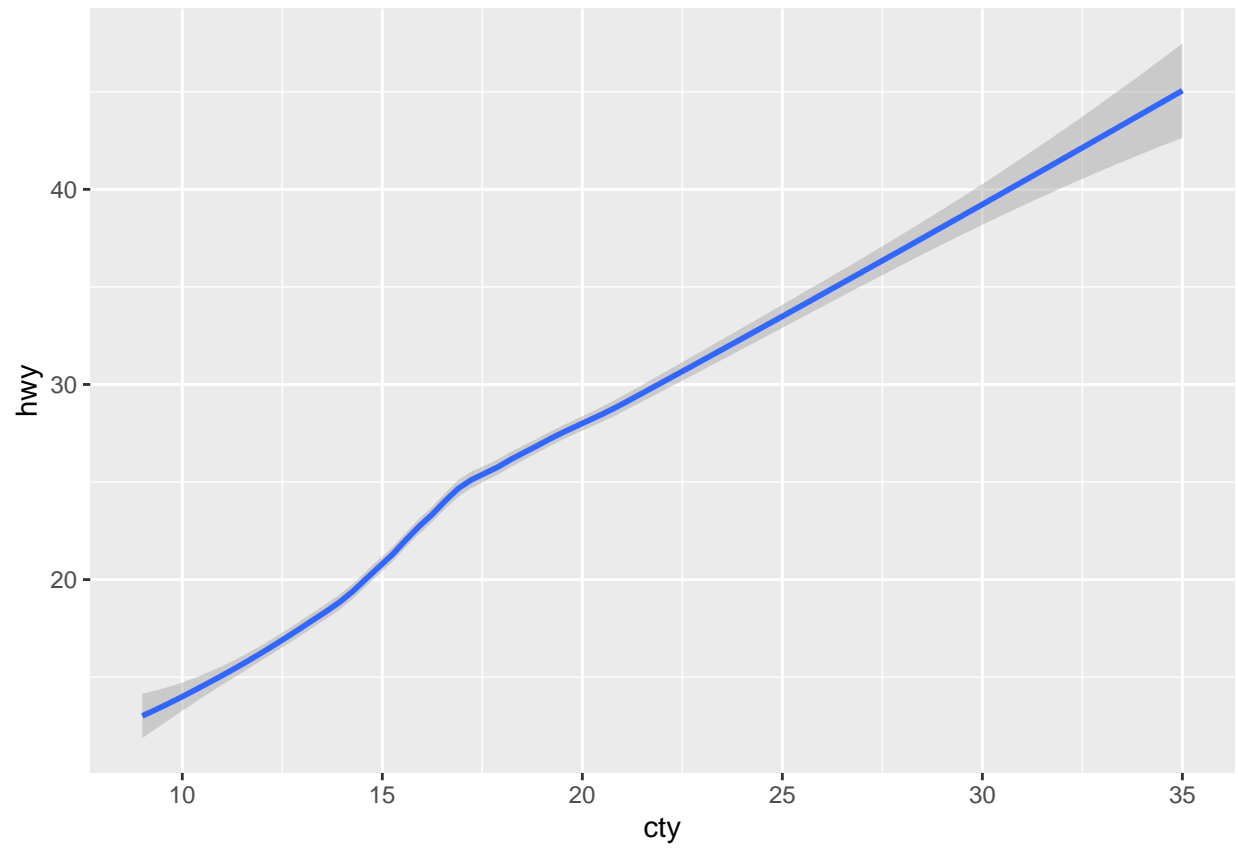


```
mpg %>%  
  ggplot(aes(cty, hwy)) +  
  geom_jitter()
```

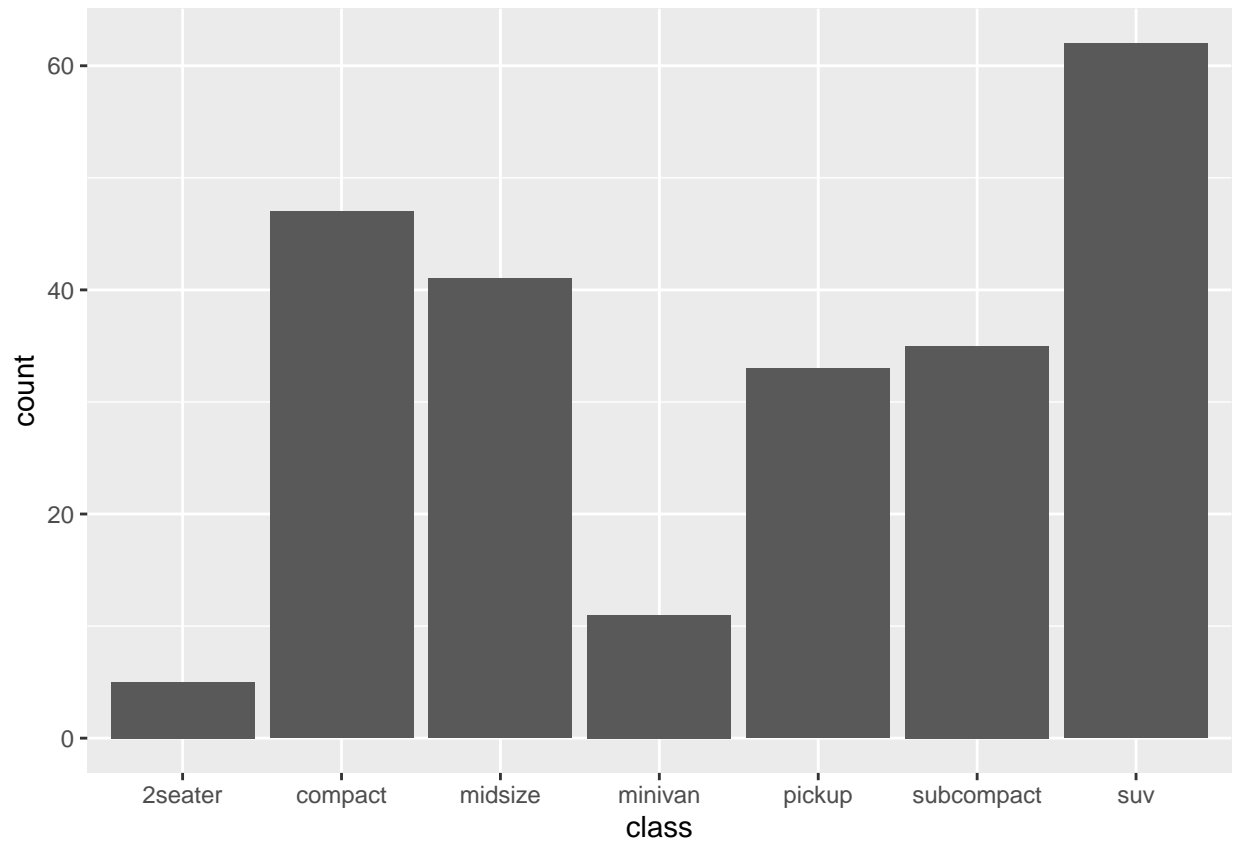


```
mpg %>%  
  ggplot(aes(cty, hwy)) +  
  geom_smooth()
```

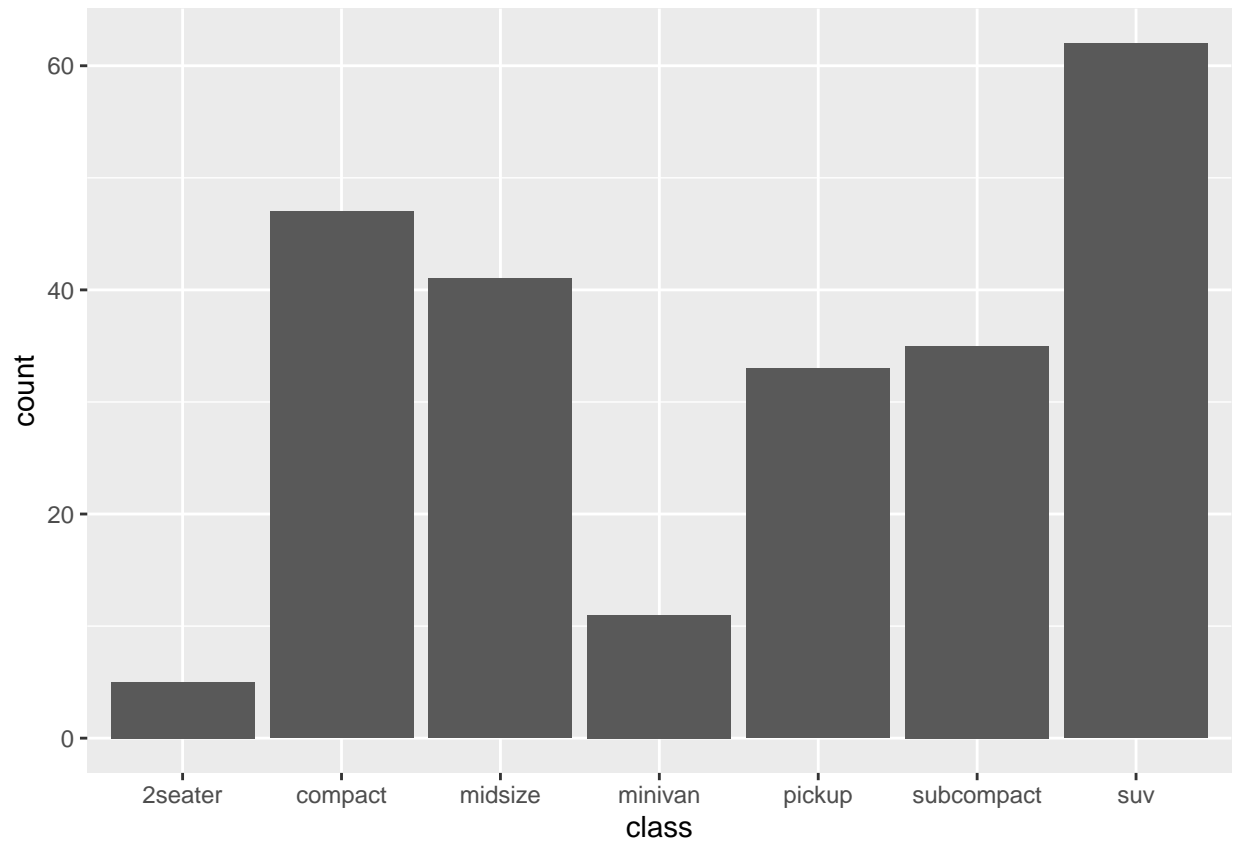
```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



```
mpg %>%  
  ggplot(aes(class)) +  
  geom_bar()
```



```
mpg %>%  
  ggplot(aes(class)) +  
  stat_count()
```

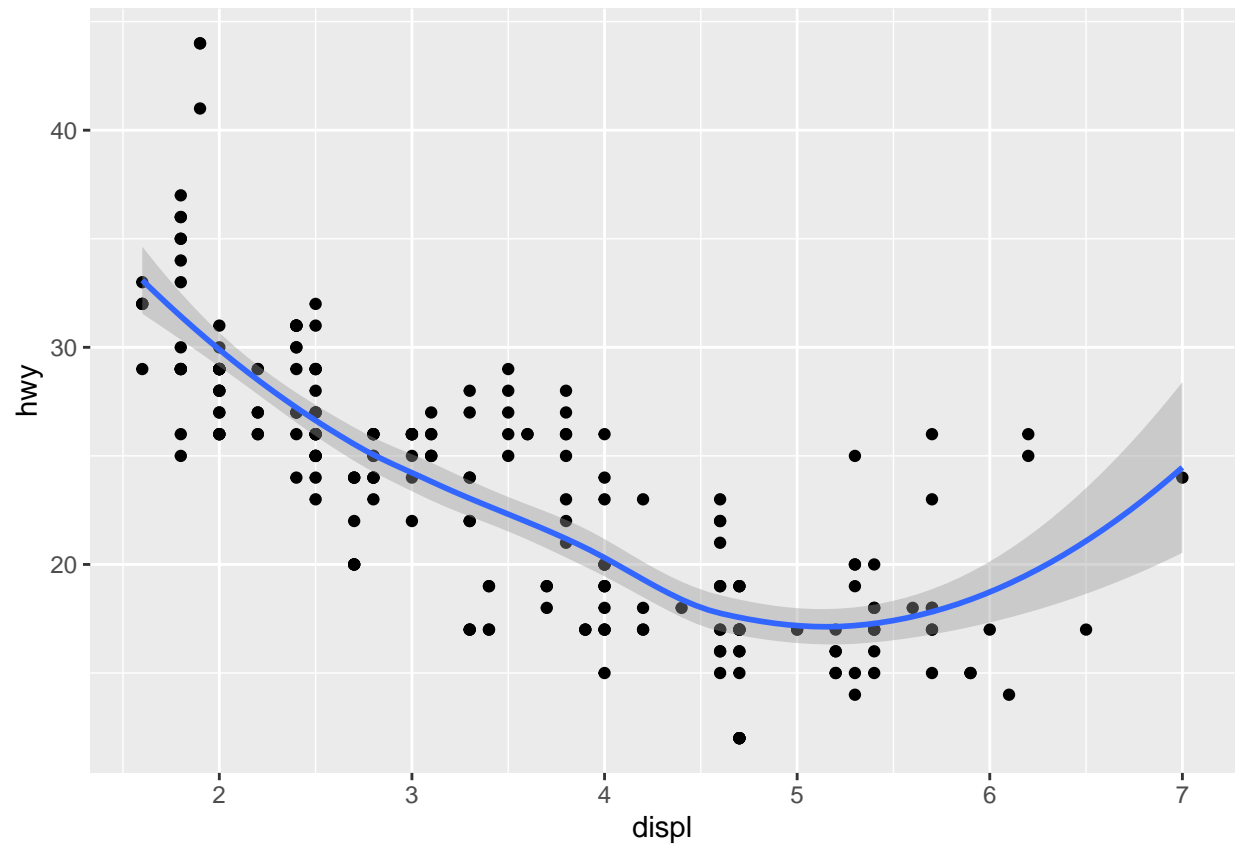


*# can have multiple geometries on the same plot*

```
mpg %>%  
  ggplot(aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(se = TRUE)
```

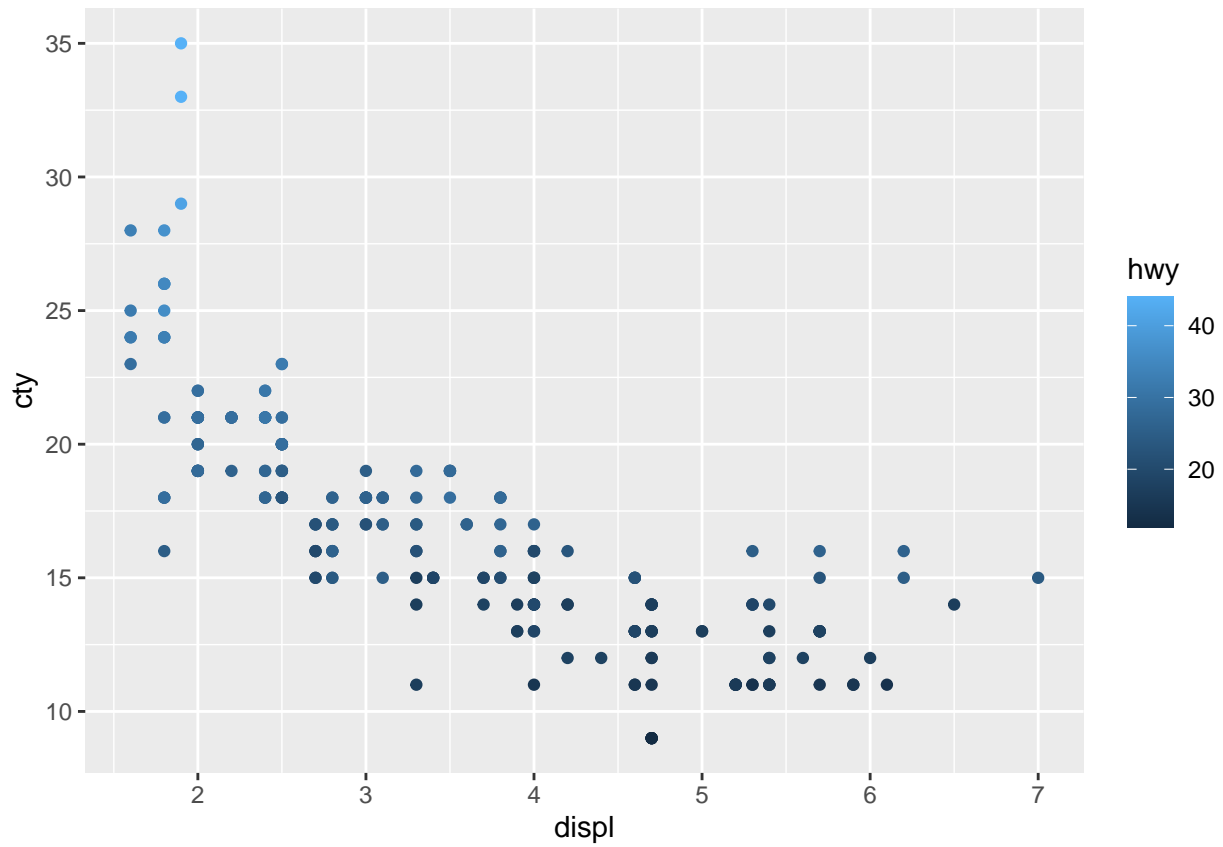
```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



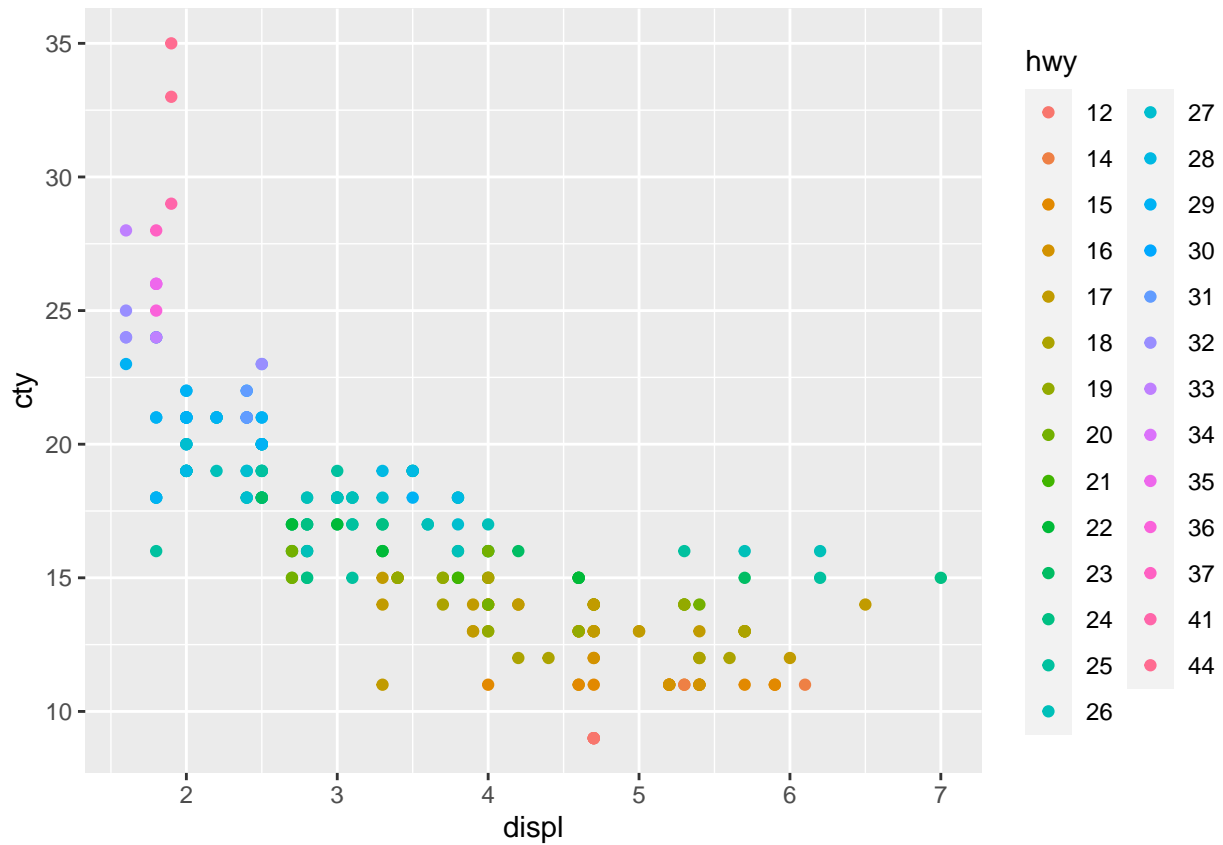


## A useful note

```
# sometimes when you want a continuous variable on your legend it can be useful to set it as a factor  
  
# is this the best example, nah, but it popped into my head pretty quick  
mpg %>%  
  ggplot(aes(displ, cty, color = hwy)) +  
  geom_point()
```



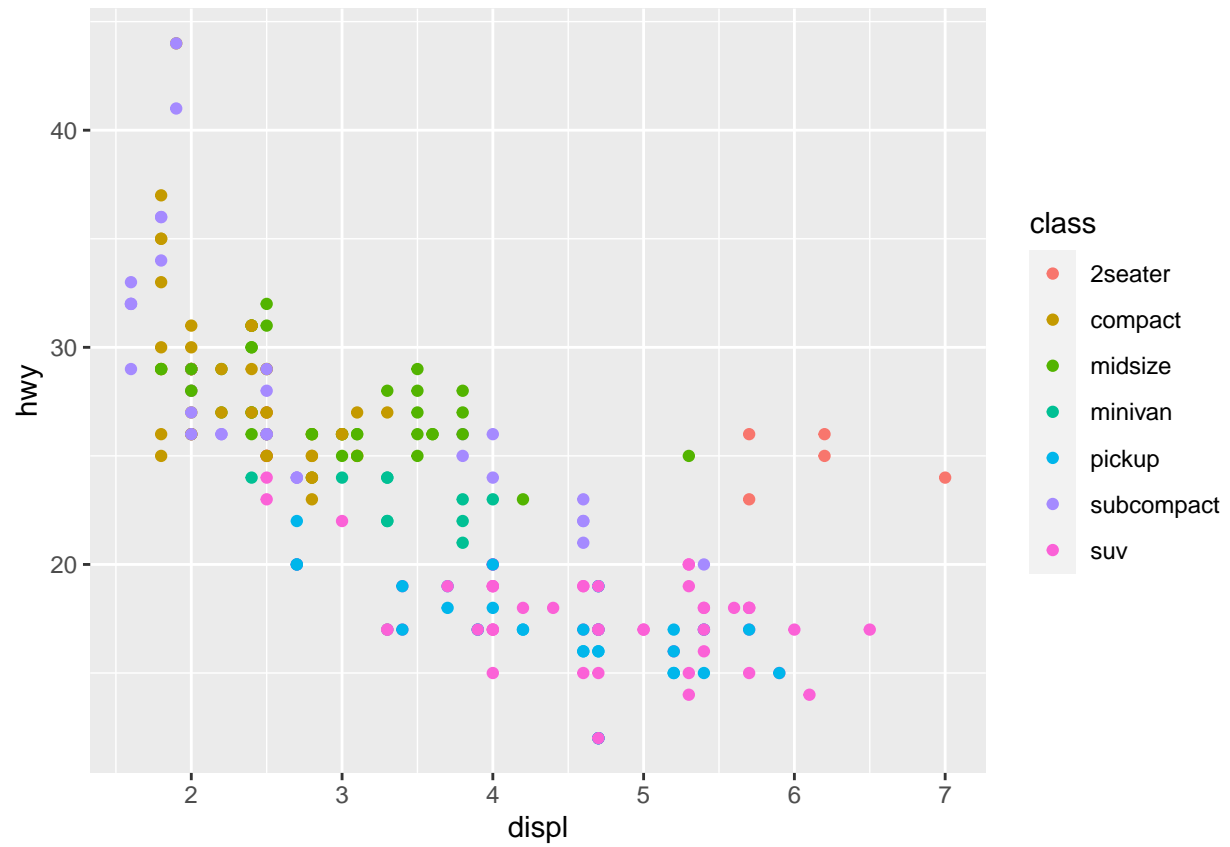
```
mpg %>%  
  mutate(hwy = as.factor(hwy)) %>%  
  ggplot(aes(displ, cty, color = hwy)) +  
  geom_point()
```



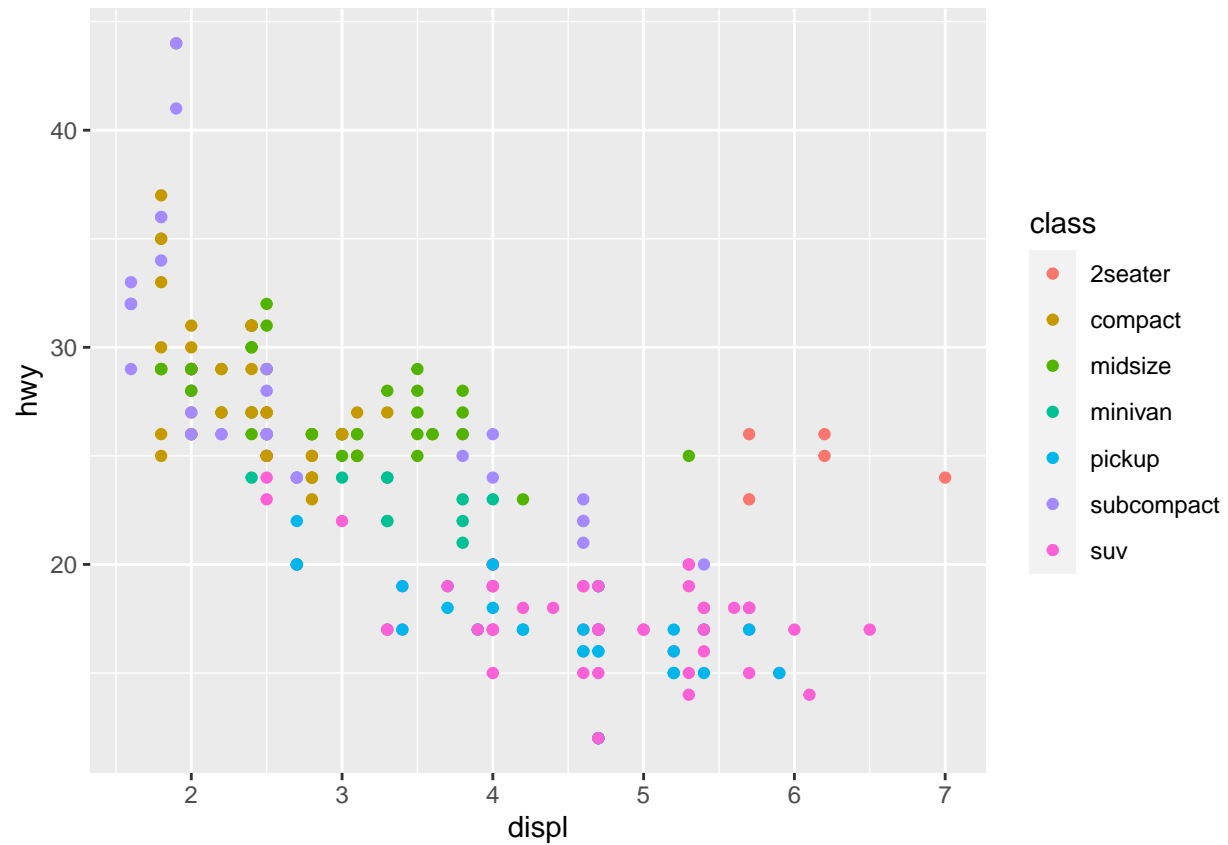
## Adding Aesthetics to these

```
# some examples of aesthetics, color, shape, size, alpha
# again, many different ways to do this, the method used in the textbook is useful for assigning an aes

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```



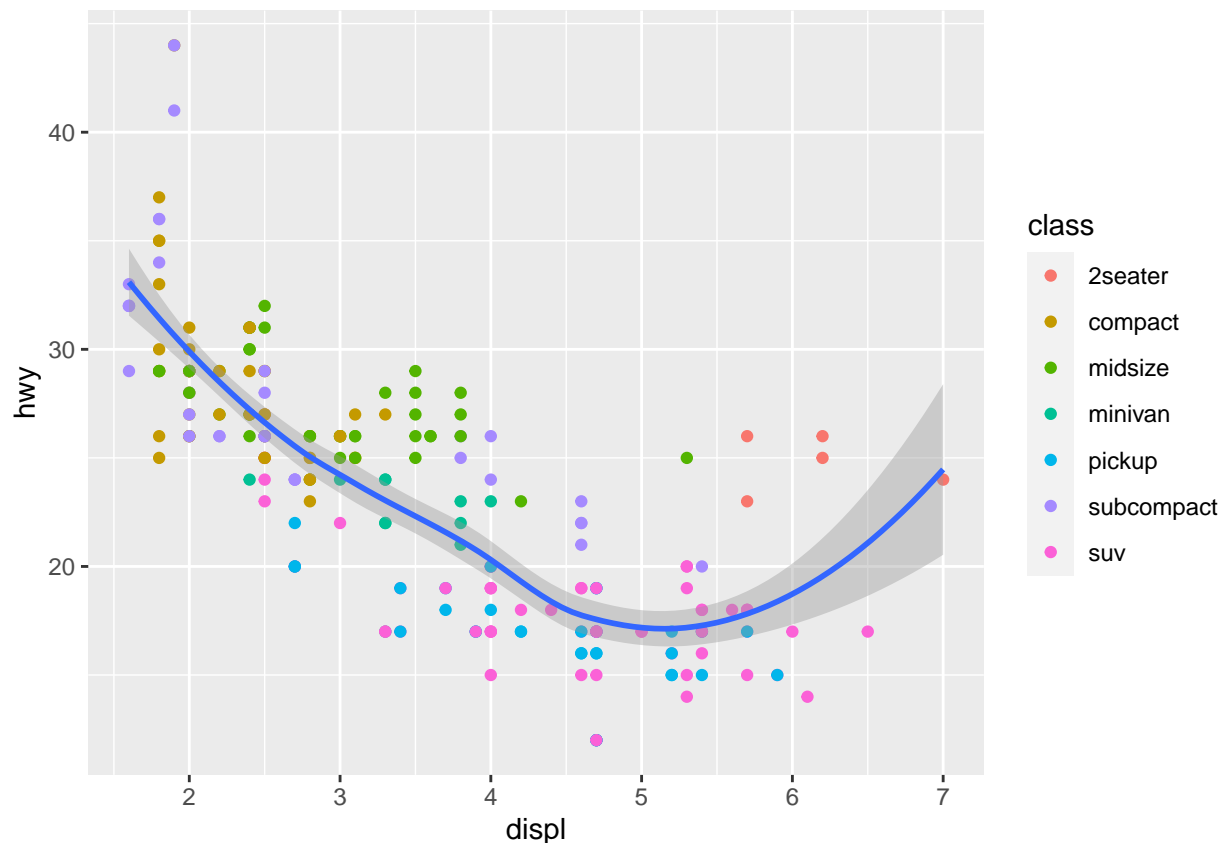
```
mpg %>%  
  ggplot(aes(displ, hwy, color = class)) +  
  geom_point()
```



*# sometimes you will need to put your aesthetics in the geom statement though, particularly if you want*

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point(mapping = aes(color = class)) +  
  geom_smooth()
```

## 'geom\_smooth()' using method = 'loess' and formula 'y ~ x'



*# if you put color in the ggplot statement it does this*

```
mpg %>%
  ggplot(aes(displ, hwy, color = class)) +
  geom_point() +
  geom_smooth()
```

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

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 5.6935
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.5065
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.65044
```

```

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : span too small. fewer
## data values than degrees of freedom.

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : pseudoinverse used at
## 5.6935

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : neighborhood radius
## 0.5065

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : reciprocal condition
## number 0

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : There are other near
## singularities as well. 0.65044

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 4.008

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.708

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.6135e-17

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.25

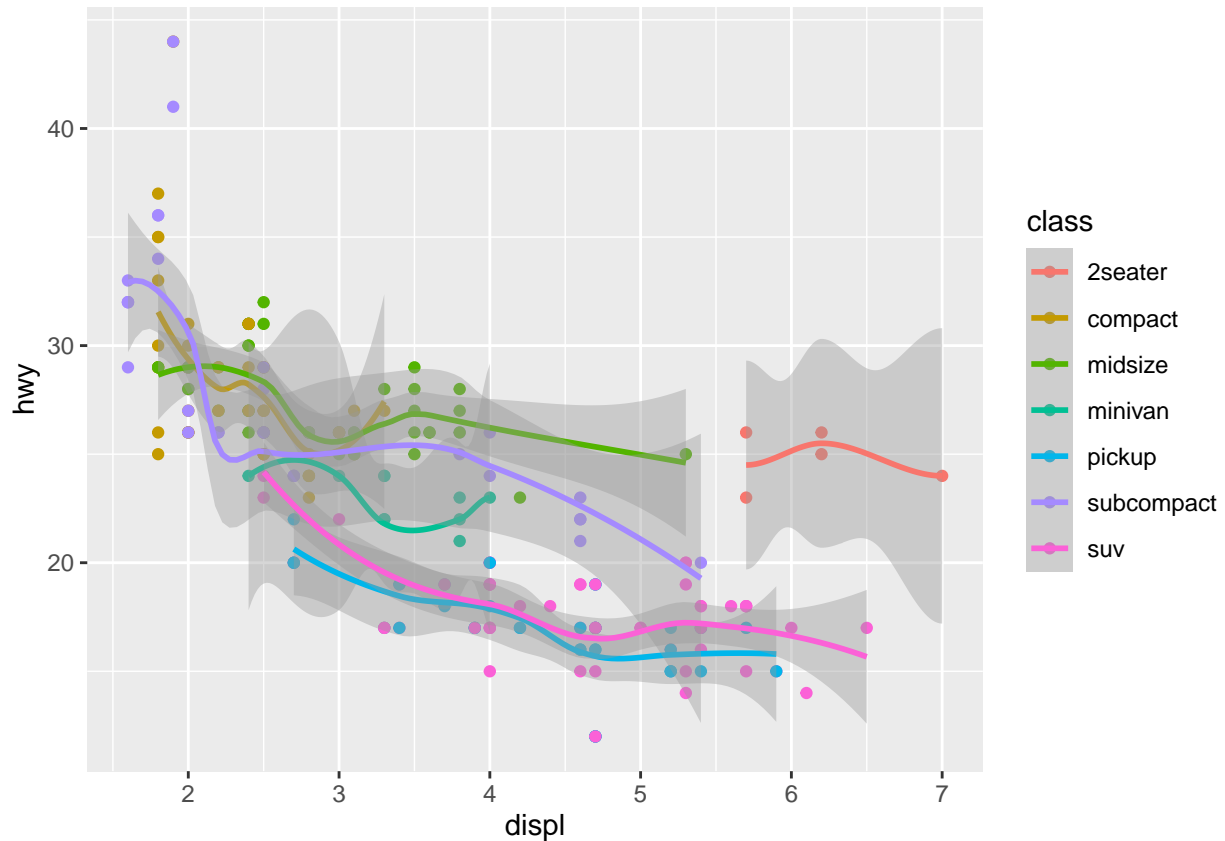
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : pseudoinverse used at
## 4.008

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : neighborhood radius
## 0.708

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : reciprocal condition
## number 1.6135e-17

```

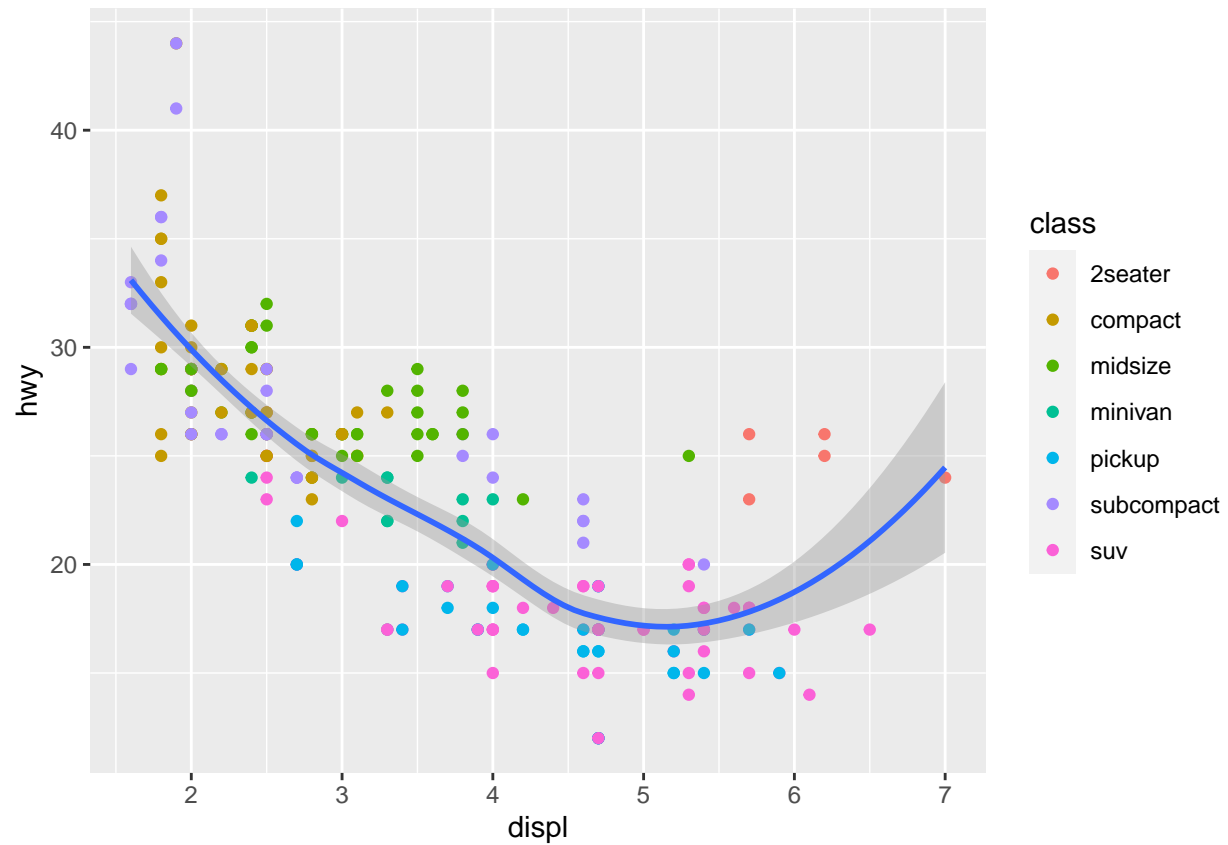
```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : There are other near
## singularities as well. 0.25
```



```
# the takeaway, there are many ways to do this, one isn't better than the other, but they are useful for
mpg %>%
  ggplot(aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth()
```

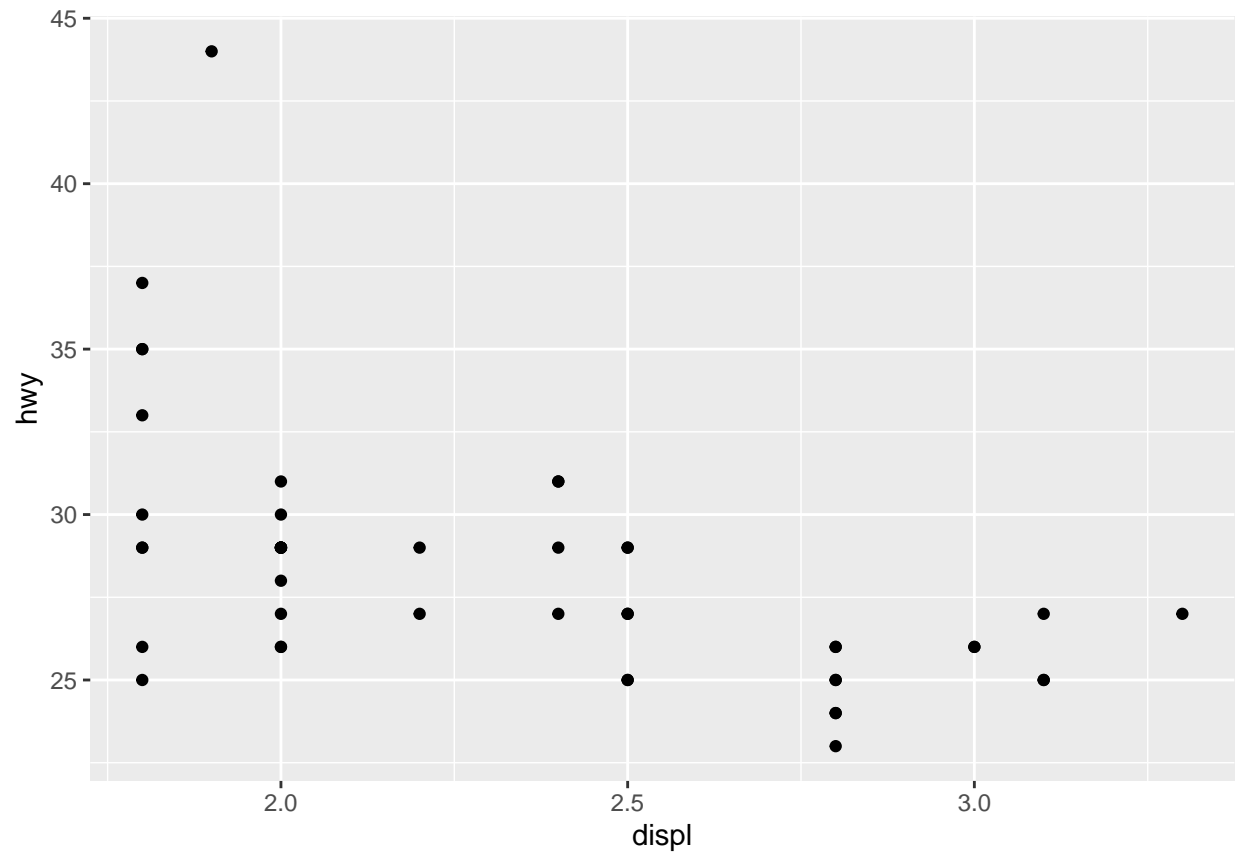
```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```





## Getting a bit more advanced, mutating things within ggplot

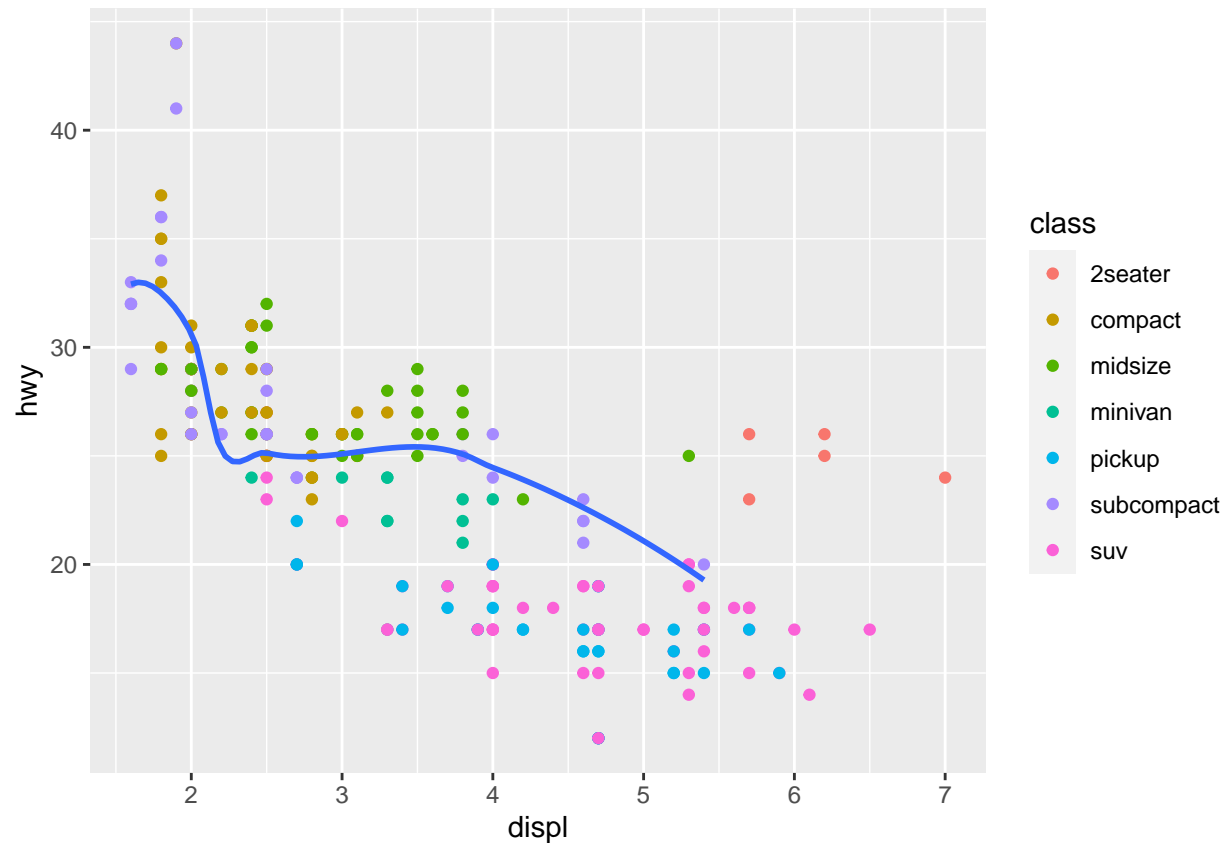
```
# you can mutate things within ggplot, you may sometimes want to make a new data frame to make a graph,  
mpg %>%  
  filter(class == "compact") %>%  
  ggplot(aes(displ, hwy)) +  
  geom_point()
```



*# but, it also matters where you put these mutations, you can put them within geometries, this is a cool*

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

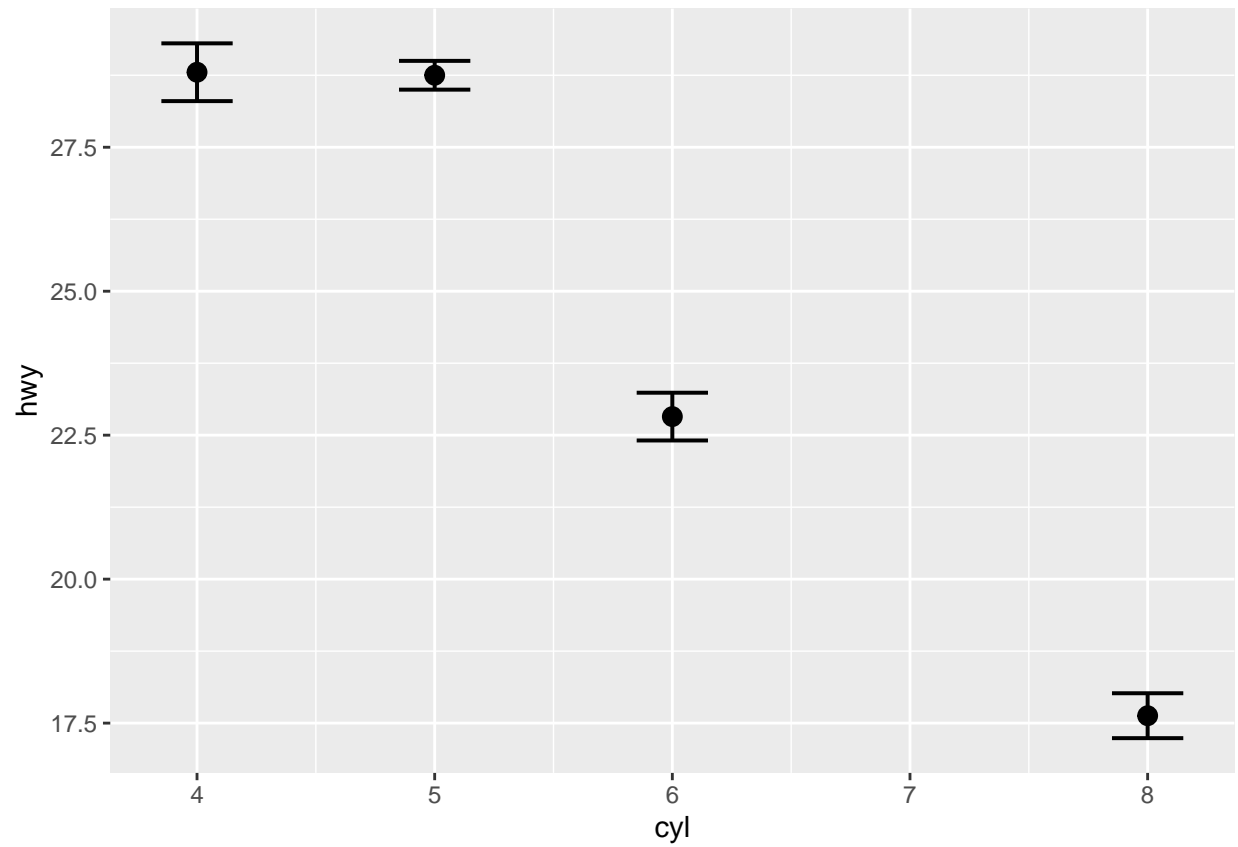
```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



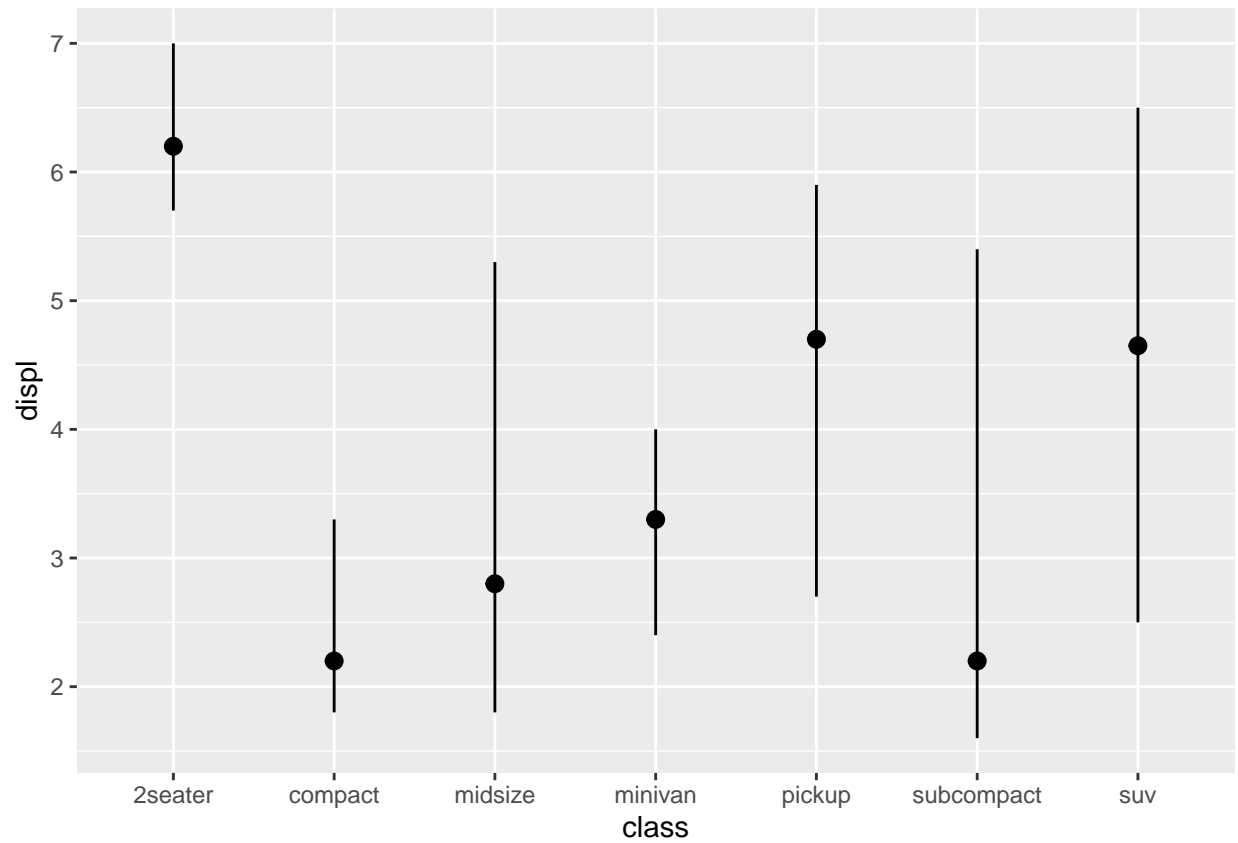
## Stat Summary, one of the best features of ggplot

Stat summary will do summary stats for you and plot them it great

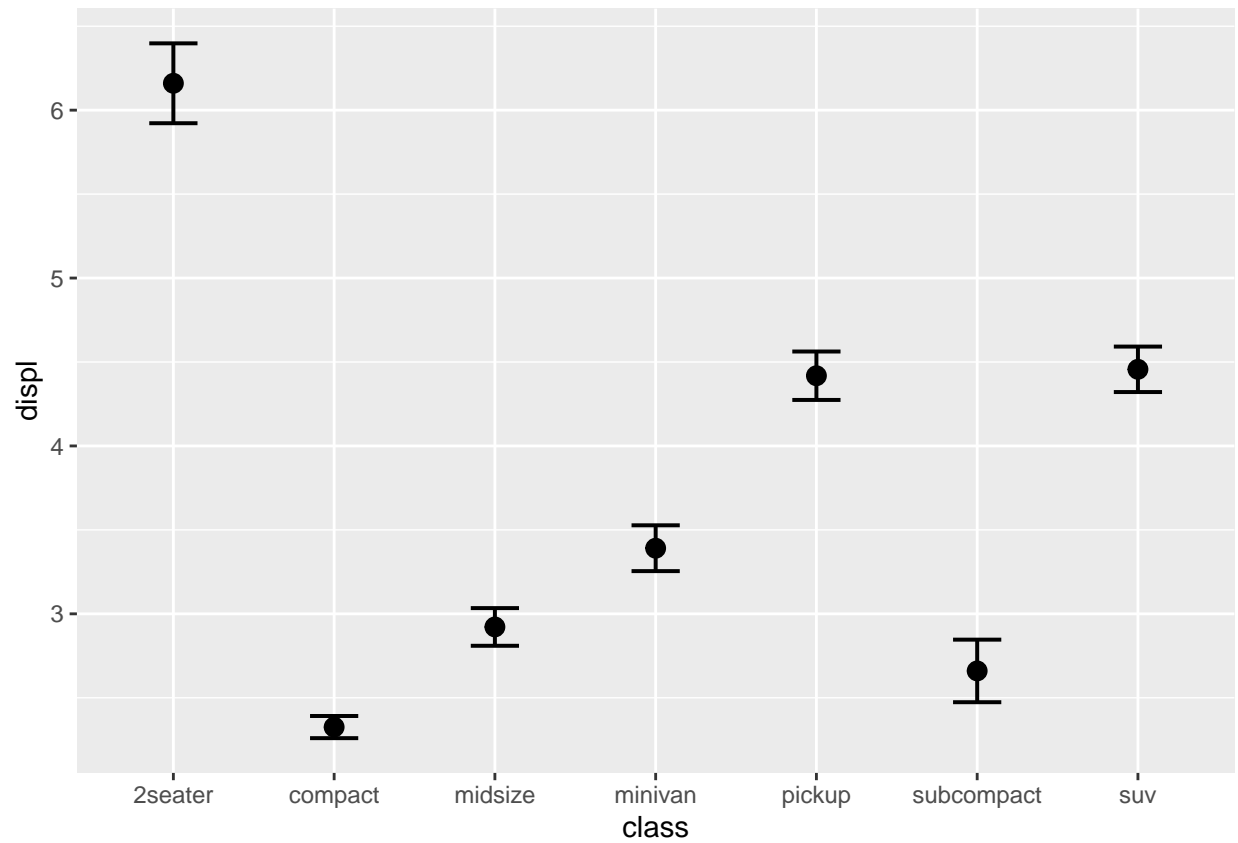
```
# mean and standard error plot
mpg %>%
  ggplot(aes(cyl, hwy)) +
  stat_summary(fun = mean, na.rm = TRUE, geom = "point", size = 3) +
  stat_summary(fun.data = mean_se, na.rm = TRUE, geom = "errorbar", width = .3, size = 0.7)
```



```
# min max and median examples
mpg %>%
  ggplot(aes(class, displ)) +
  stat_summary(
    fun.min = min,
    fun.max = max,
    fun = median
  )
```

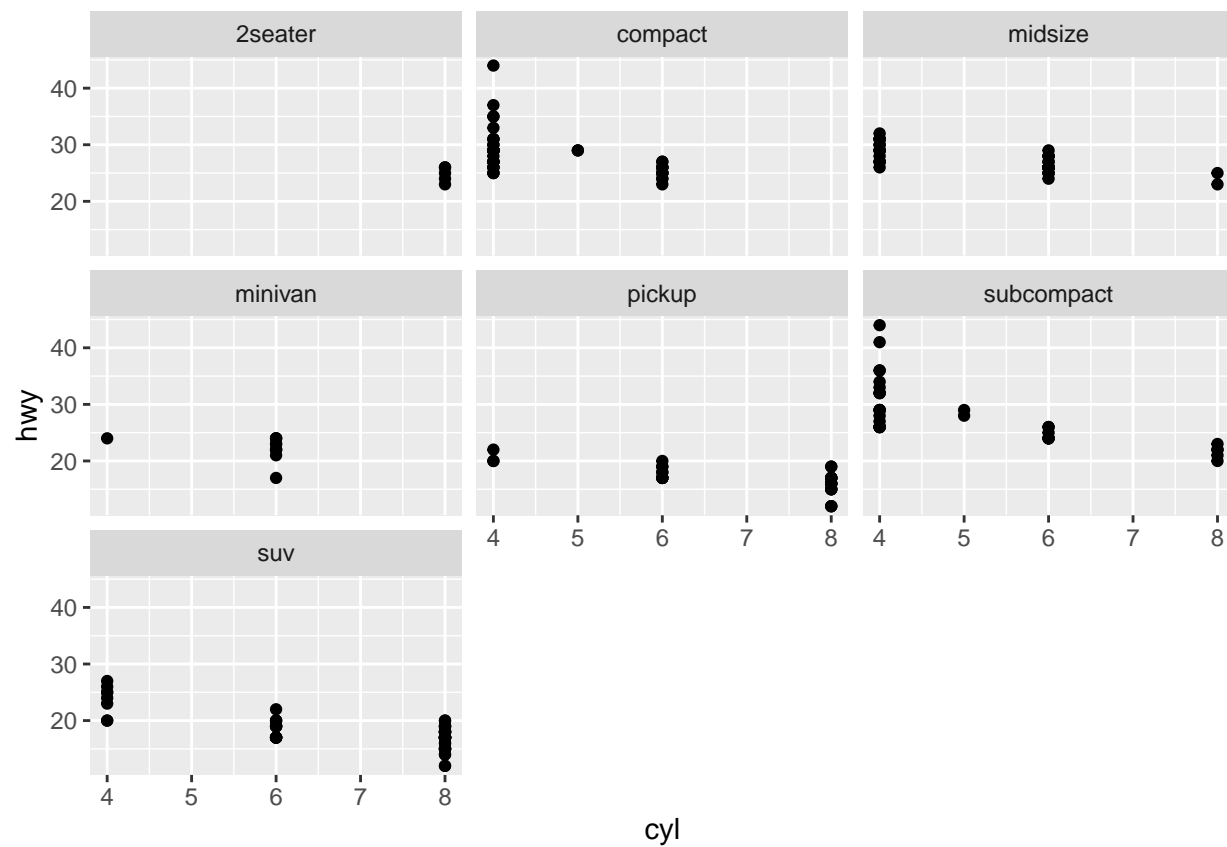


```
mpg %>%  
  ggplot(aes(class, displ)) +  
  stat_summary(fun = mean, na.rm = TRUE, geom = "point", size = 3) +  
  stat_summary(fun.data = mean_se, na.rm = TRUE, geom = "errorbar", width = 0.3, size = 0.7)
```

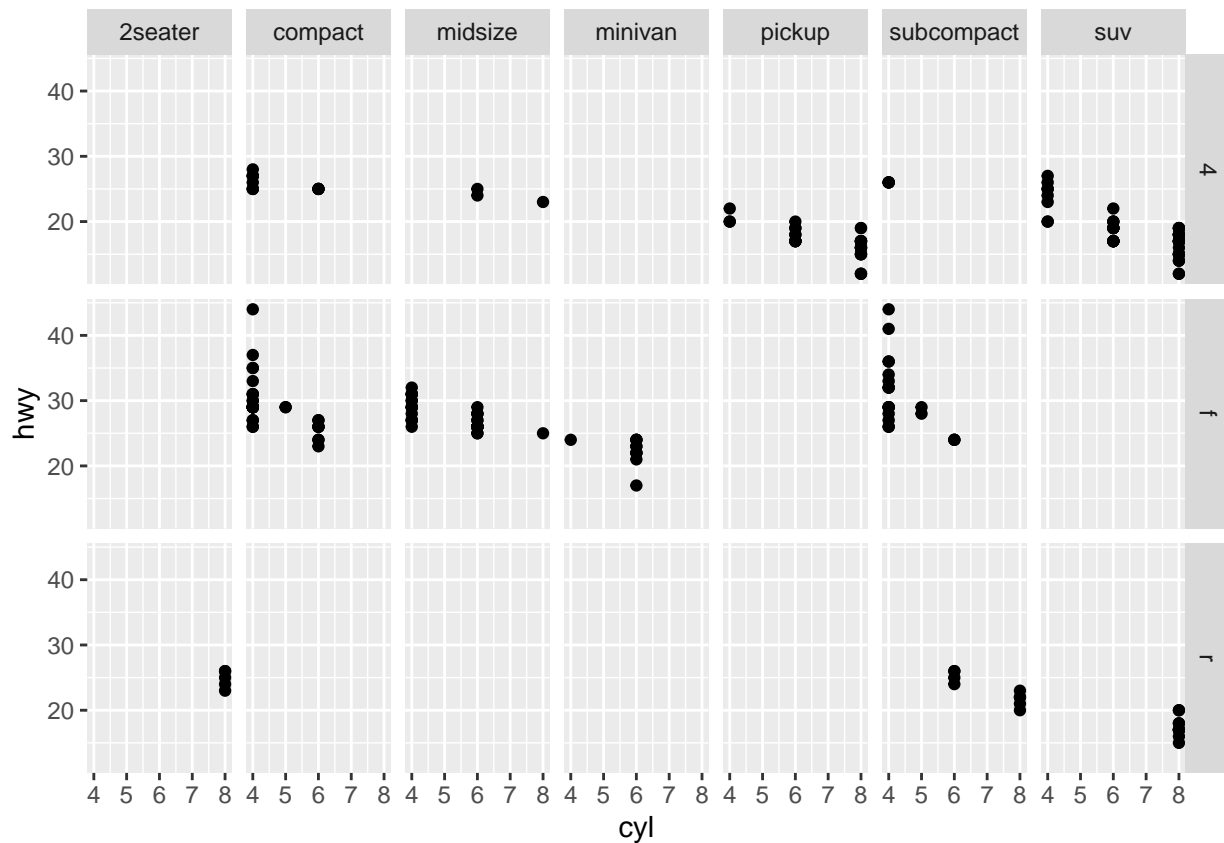


Facet grid and facet wrap - useful for data exploration

```
# facet wrap  
mpg %>%  
  ggplot(aes(cyl, hwy)) +  
  geom_point() +  
  facet_wrap(~class)
```



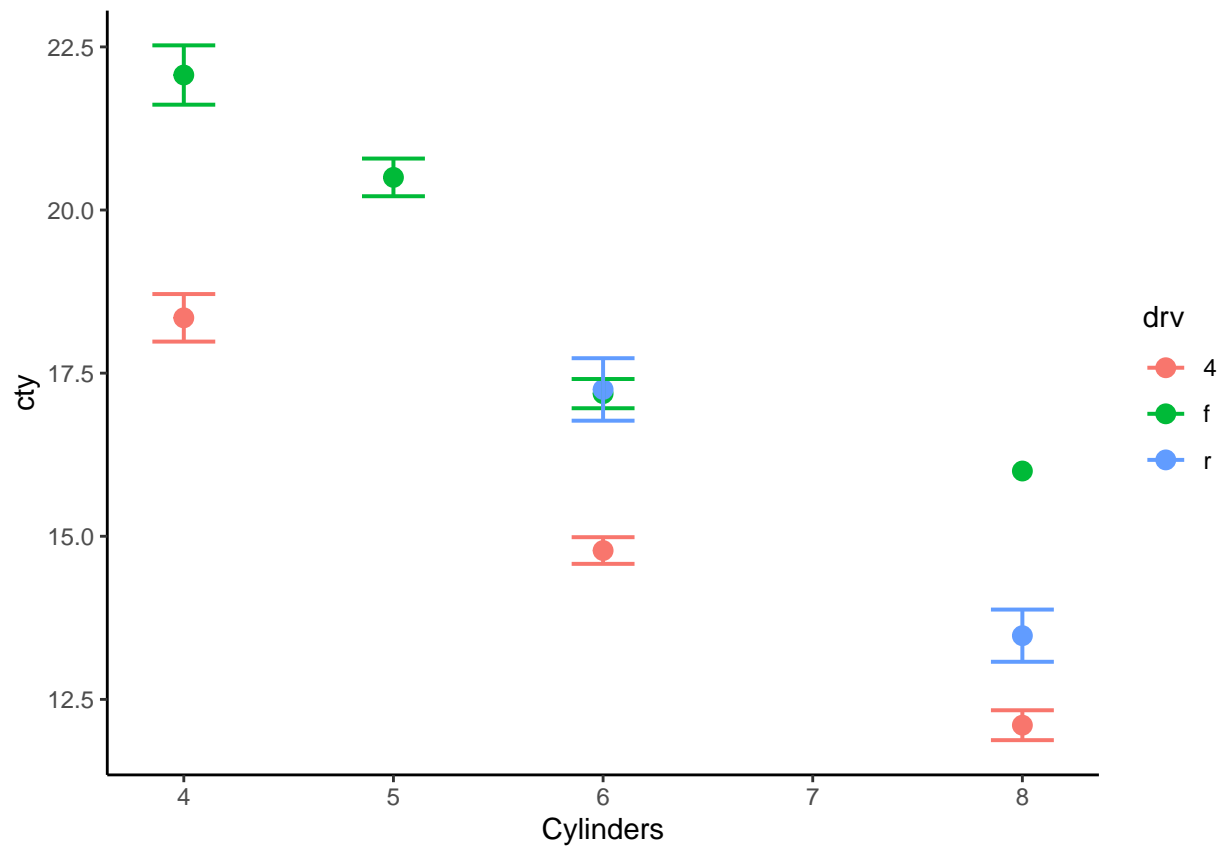
```
# facet grid
mpg %>%
  ggplot(aes(cyl, hwy)) +
  geom_point() +
  facet_grid(drv ~ class)
```



## Last thing, making graphs pretty

```
# install.packages(ggThemeAssist)
# use that
# can also add a whole host of theme_themes() to make graphs look nice easy
# labs is useful for adding axis labels
# theme classic is my favorite
mpg %>%
  ggplot(aes(cyl, cty, color = drv)) +
  stat_summary(fun = mean, na.rm = TRUE, geom = "point", size = 3) +
  stat_summary(fun.data = mean_se, na.rm = TRUE, geom = "errorbar", width = .3, size = 0.7) +
  labs(x = "Cylinders", "City MPG") +
  theme_classic()
```





```
# obligatory bar graph - so i made an interesting bar graph...ah jeez
mpg %>%
  ggplot(aes(class)) +
  stat_count(aes(fill = drv)) +
  theme_classic()
```

