Module 2 Exercises

Brady Lamson

2/7/2022

3: Data Visualization (Graphics)

3.2: A Taxonomy for Data Graphics

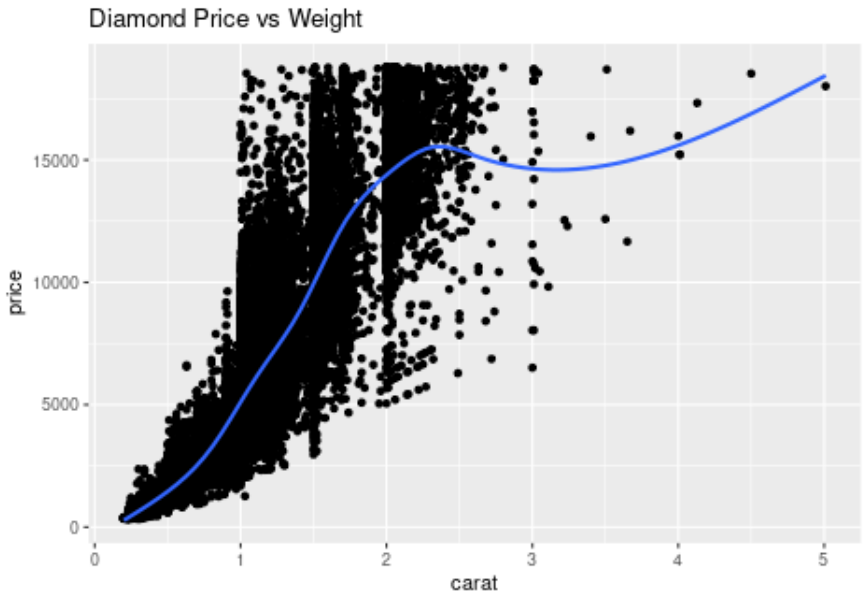
Exercise 1:

For each graph indicate:

- The **visual cues** that are used
- The **coordinate system** that's used
- The **scales** that are used
- How **context** is provided

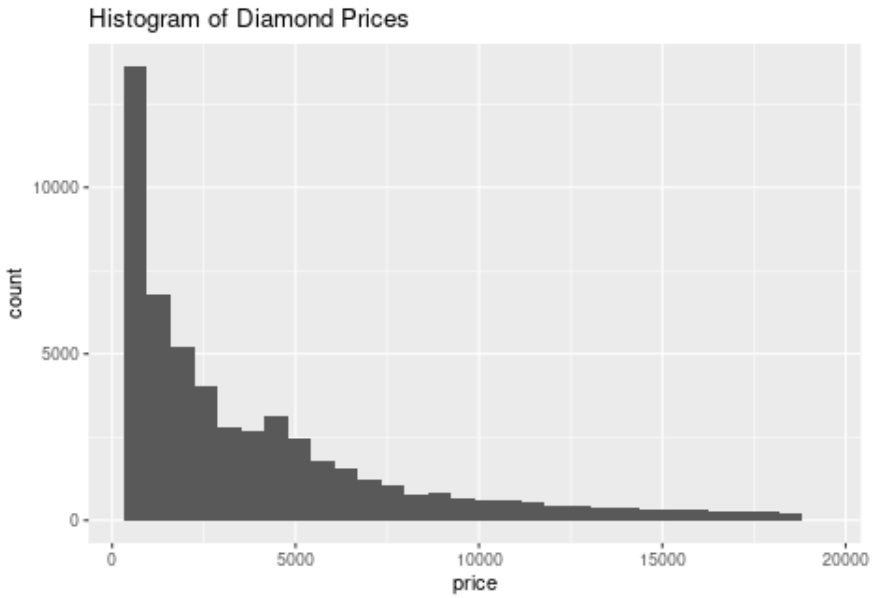
a)
- **Visual Cues:** *Position* along the x and y axis and *color*.
- **Coordinate System:** *Cartesian*
- **Scale:** *Numerical* for the axes scale *Categorical* for the colors
- **Context:** Legend, x and y-axis labels, title

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



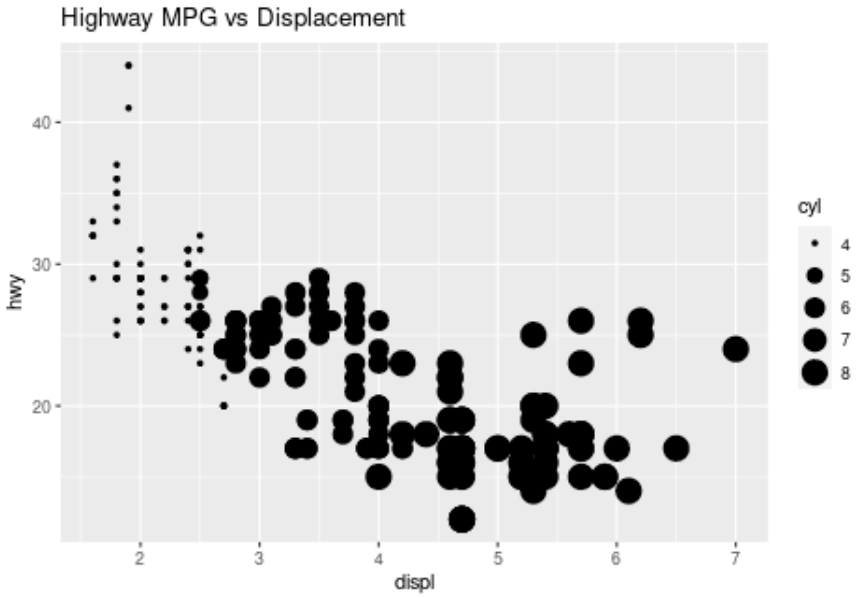
b)
- **Visual Cues:** *Position* along the x and y axis. *Angle / direction* of the line
- **Coordinate System:** *Cartesian*
- **Scale:** *Numerical*
- **Context:** Legend, x and y-axis labels, title

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

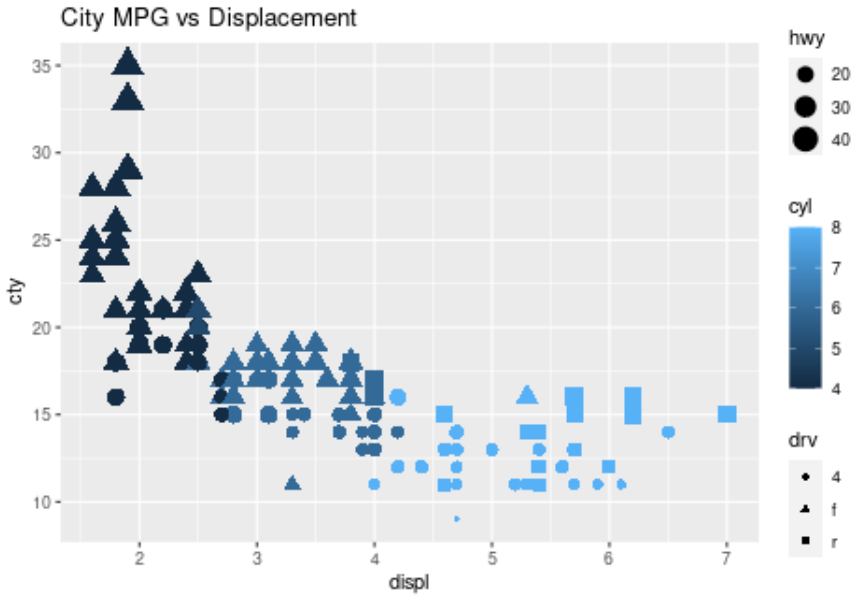


c)
- **Visual Cues:** *Position* along the x and y axis, *length* of the histogram bars. *Area* of the histogram bars.
- **Coordinate System:** *Cartesian*
- **Scale:** *Numerical*
- **Context:** X and y-axis lables, title.

Exercise 2:



a)
- **Visual Cues:** *Position* along the x and y axis, *area* of the circles.
- **Coordinate System:** *Cartesian*
- **Scale:** *Numerical*
- **Context:** X and y-axis lables, title, legend.



b)
- **Visual Cues:** *Position* along the x and y axis, *shade*, *shape*, *area*.
- **Coordinate System:** *Cartesian*
- **Scale:** *Numerical*
- **Context:** Title, x and y-axis, three different legends

4: A Grammar for Graphics with "ggplot2"

4.1: Introduction

Exercise 3:

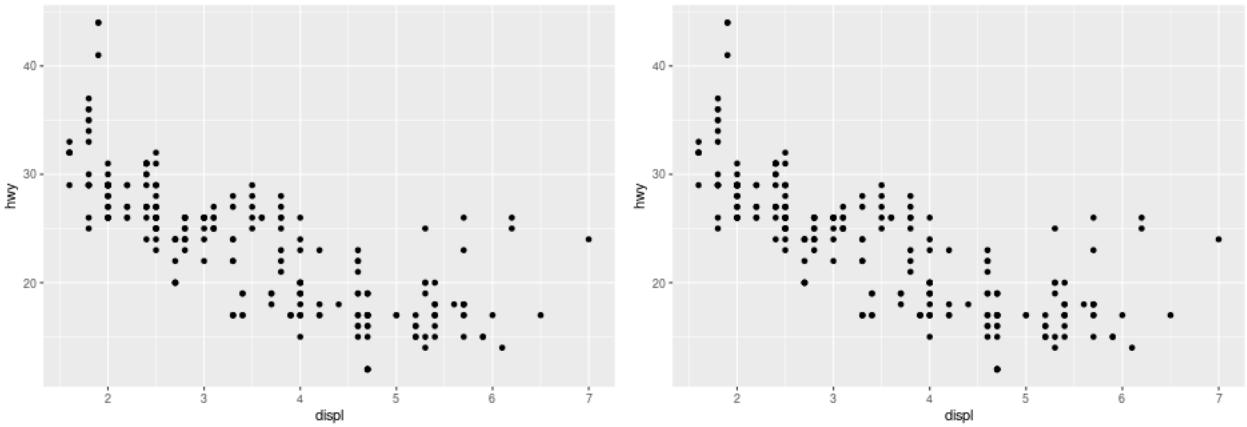
This outputs a blank box, the box that will have more and more information added onto it once more things are specified.

Exercise 4:

Guess whether the following commands both make the same scatterplot, then check your answer:

I would guess **yes**. These would be different *if* there was another geom_* used with a different dataset. These, I think, should be functionally equivelant.

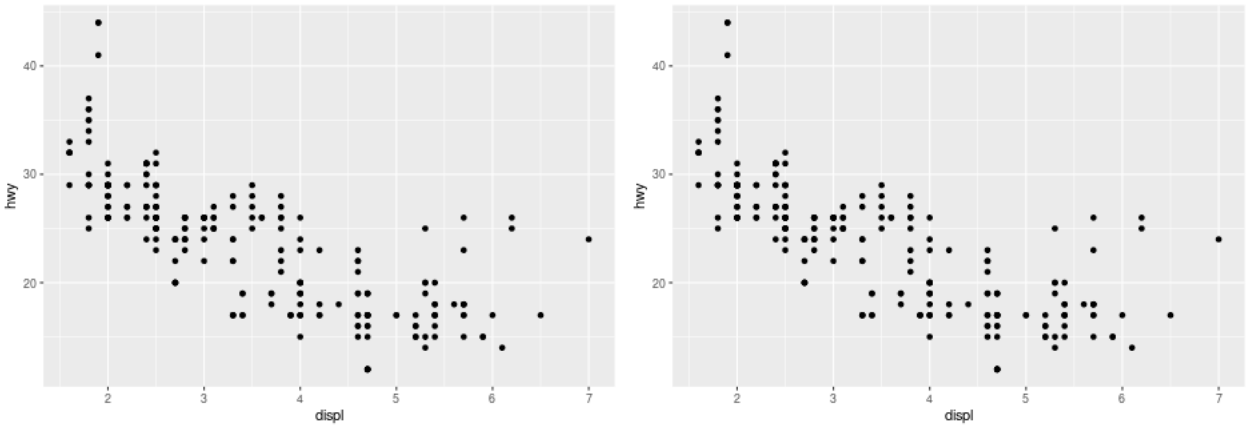
```
## Specify data in ggplot():
ggplot(data = mpg) +
geom_point(mapping = aes(x = displ, y = hwy))

## Specify data in geom_*() function:
ggplot() +
geom_point(data = mpg, mapping = aes(x = displ, y = hwy))
```



Exercise 5:

I would make the same guess that **yes**, these are equivalent expressions. Not enough is really going on to impact the graph.

```
## Specify aesthetics in geom_*() function:
ggplot(data = mpg) +
geom_point(mapping = aes(x = displ, y = hwy))
## Specify aesthetics in ggplot():
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
geom_point()
```



Exercise 6

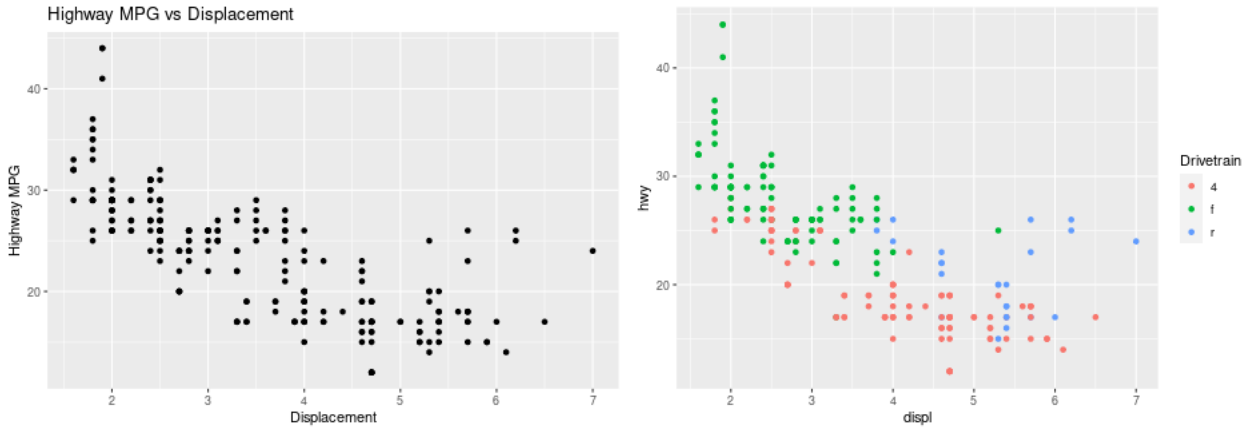
a) Guess what the **ggtitle(), xlab()**, and **ylab()** commands do to the scatterplot below and to the left. Then check your answers.

- ggtitle will put a title at the top with the text "Highway MPG vs Displacement"
- xlab and ylab will put text labels on the x-axis and y-axis respectively.

b) Guess what the **labs()** command does to the scatterplot below and to the right. Then check your answer.

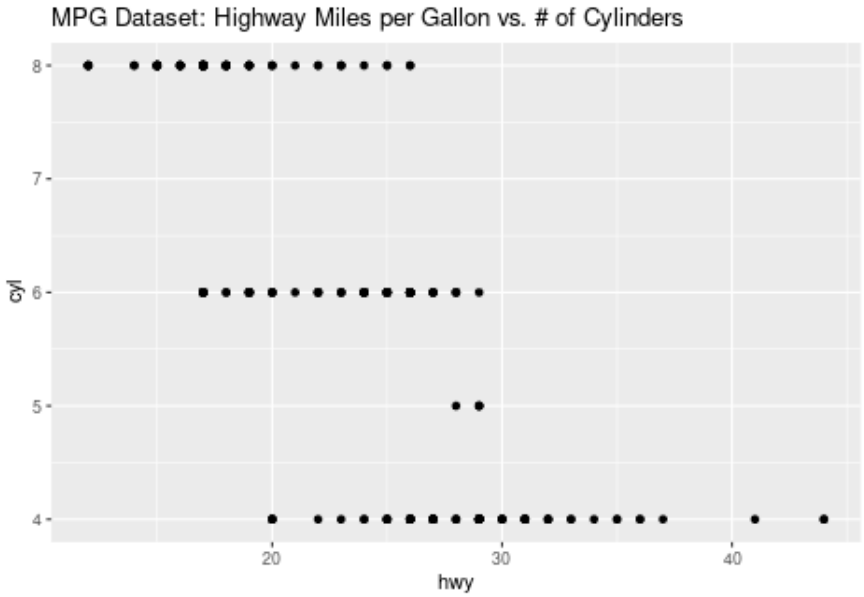
- I'm not sure actually. col = drv means we have some categorical color usage here and my gut instinct says the only way a label would make sense here is with a legend. Let's go with that then! This will create a legend with the text label above it saying "Drivetrain".

```
# A
ggplot(data = mpg) +
geom_point(mapping = aes(x = displ, y = hwy)) +
ggtitle(label = "Highway MPG vs Displacement") +
xlab(label = "Displacement") +
ylab(label = "Highway MPG")

# B
ggplot(data = mpg) +
geom_point(mapping = aes(x = displ, y = hwy, color = drv)) +
labs(color = "Drivetrain")
```



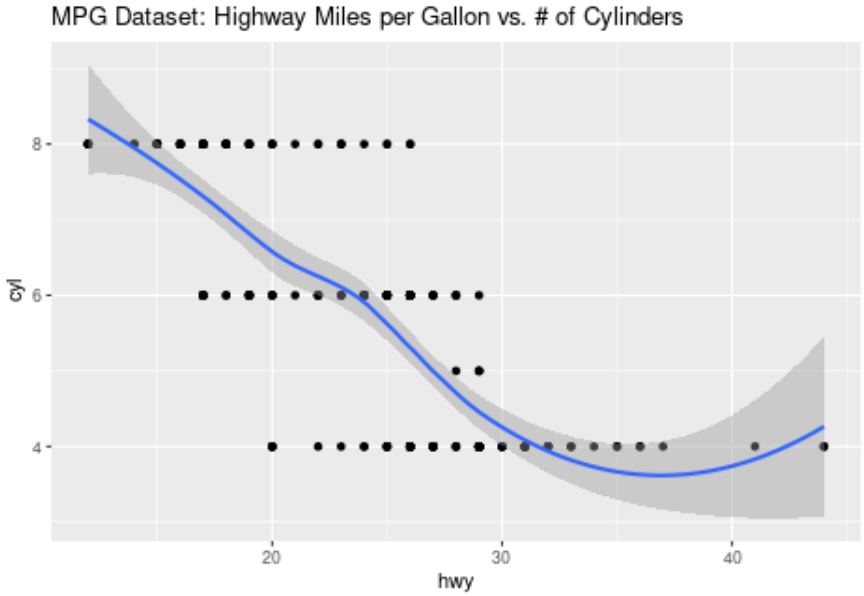
Exercise 7:

a) Make a scatterplot of hwy (on the y-axis) versus cyl (x-axis). Report your R commands.

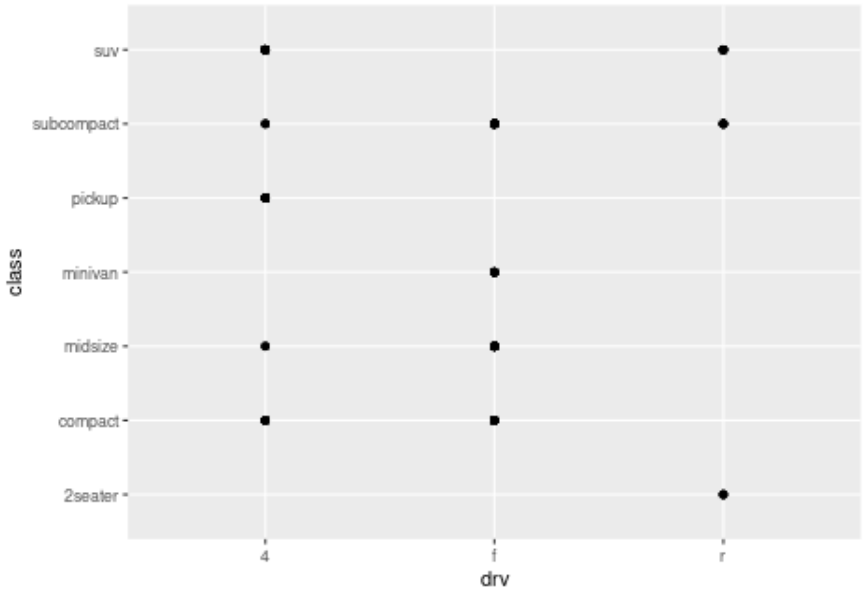


b) Reproduce the scatterplot of Part a, but now add a second layer to the plot using **geom_smooth()**. Report your R command(s).

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



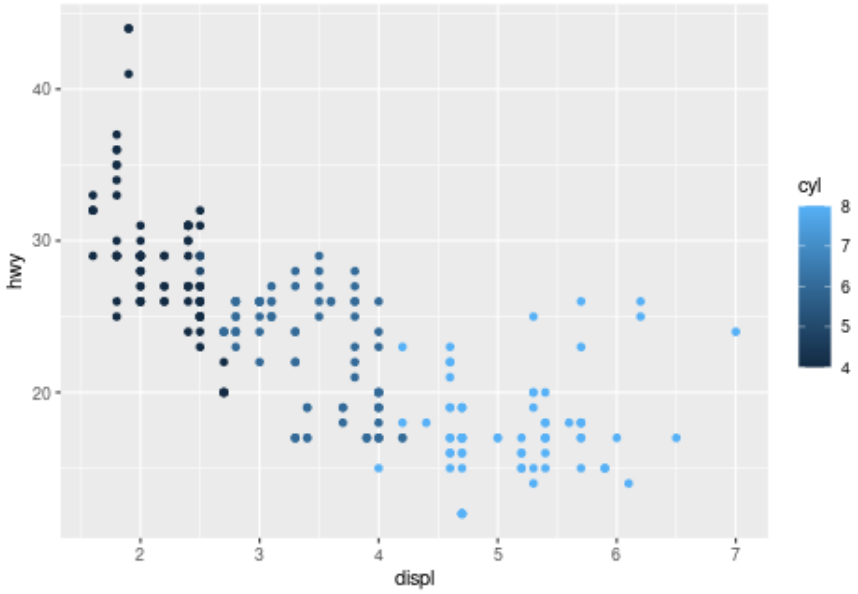
c) Make a scatterplot of class (y-axis) versus drv (x-axis)? What happens? Why is the plot not useful?



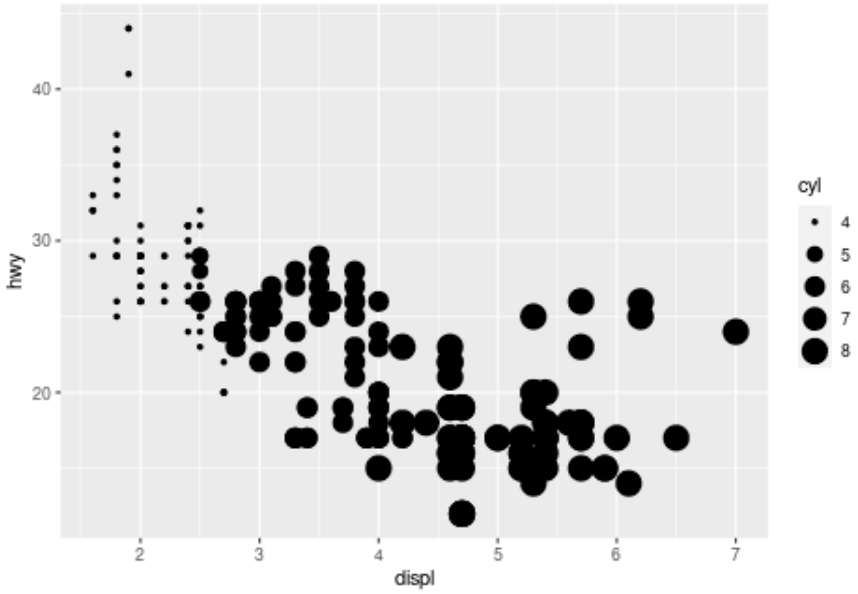
This plot is trying to use Cartesian coordinates for two different categorical values. The plot just doesn't make any sense as none of the visual cues carry any relevant information.

4.2: More on Aesthetic Mappings

Exercise 8



a) Reproduce the plot, but with cyl mapped to the size aesthetic (instead of color). How does the plot differ from the one above?



It makes the scatterplot a lot busier but instead of the number of the cylinder column determing the points color, it now determines its size. Bigger cylinder is represented by a bigger point.

b) What happens when you try to map cyl to the shape aesthetic?

- It throws an error "a continuous variable can not be mapped to shape. This is particularly weird because of the following:

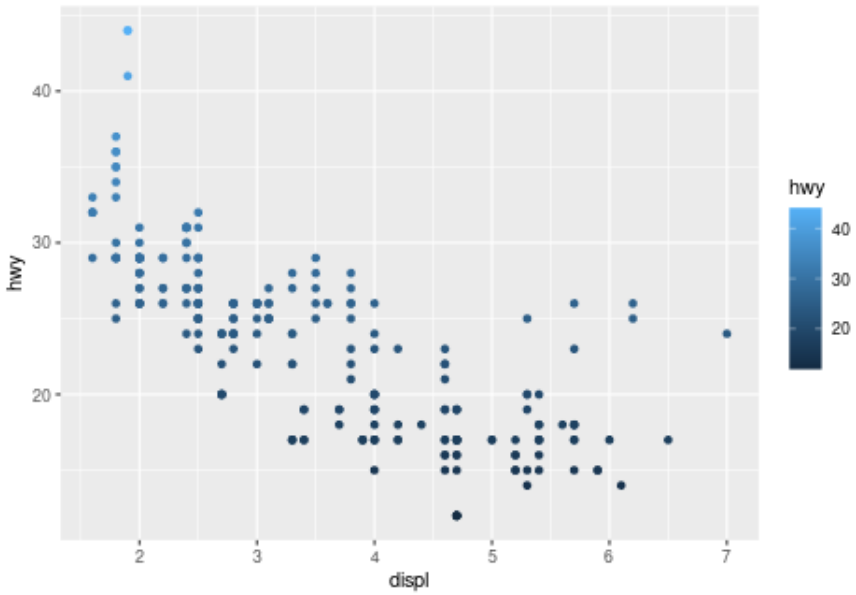
```
## [1] "integer"
```

c) What happens when you map the "logical" values in displ < 5 to an aesthetic property?

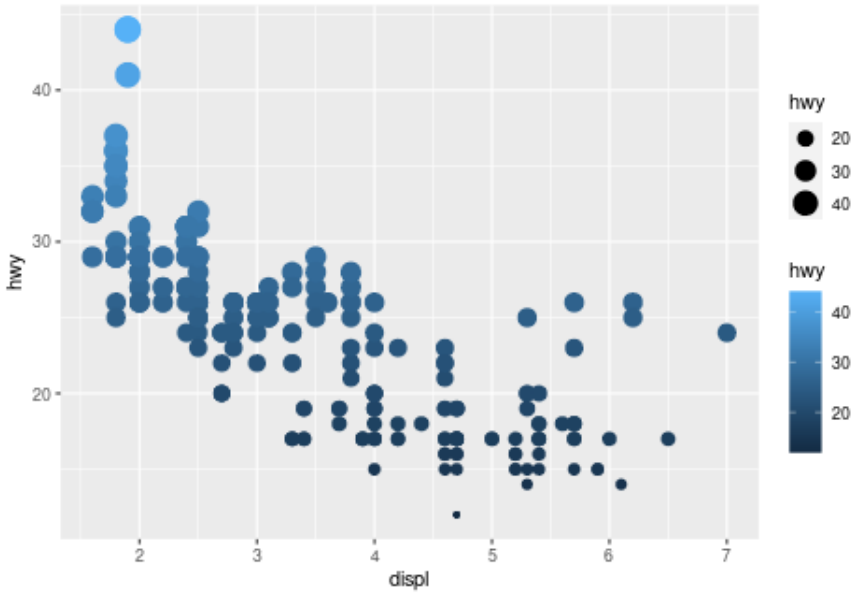
We get a scatterplot with two colors, blue and red. Any of the points left of 5 displ are blue, all to the right of 5 are red.

d) What happens when you map the same variable to multiple aesthetics?

- Map hwy to *both* y and color.
- Map hwy to y, color, *and* size.



Hwy now has both a positional and shaded component. Values high on the y axis are a lighter shade of blue.

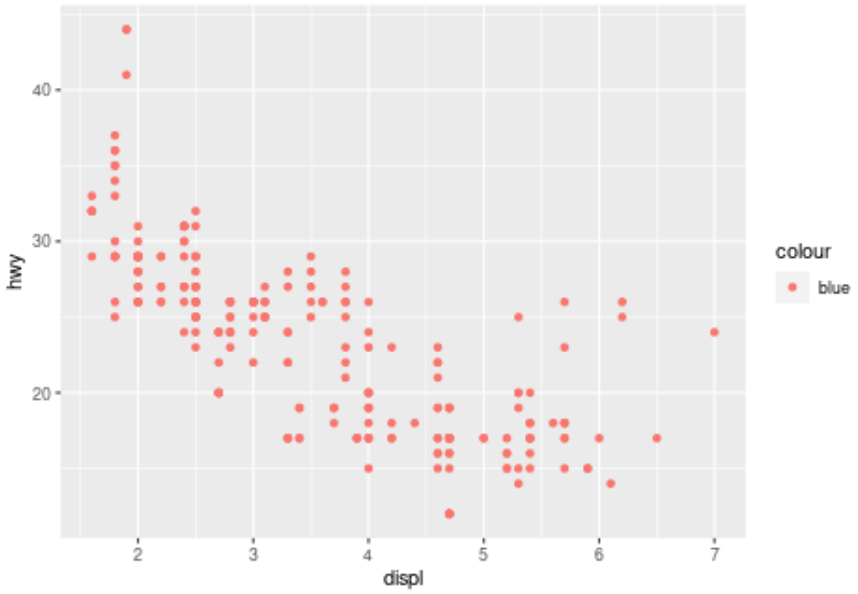


Similar to the previous plot, but now values higher on the y-axis also get progressively larger.

Exercise 9

You can set the color aesthetic of the points *manually* in the geom_*() function, *outside* aes(). What happens when you try and do this inside of aes()?

- It just creates a legend that says the color represents blue, whatever that means.



4.3: More on Layers

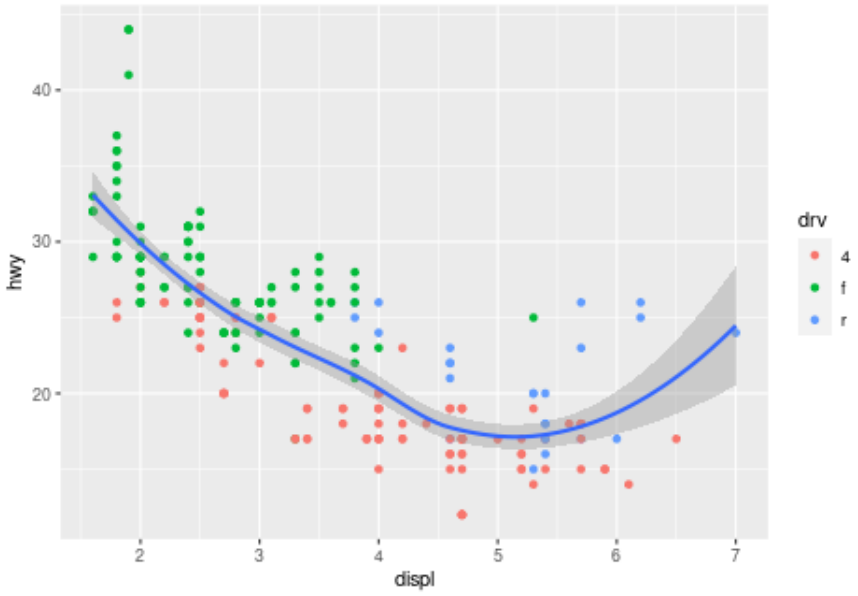
Exercise 10:

Try and predict what the following graph will look like and predict your answer.

a)

- What we will see from the below code is displacement on the x-axis, highway MPG on the y-axis and a color shaded component based on the type of drive train. Also there will be a line of best fit applied to the plot as well.

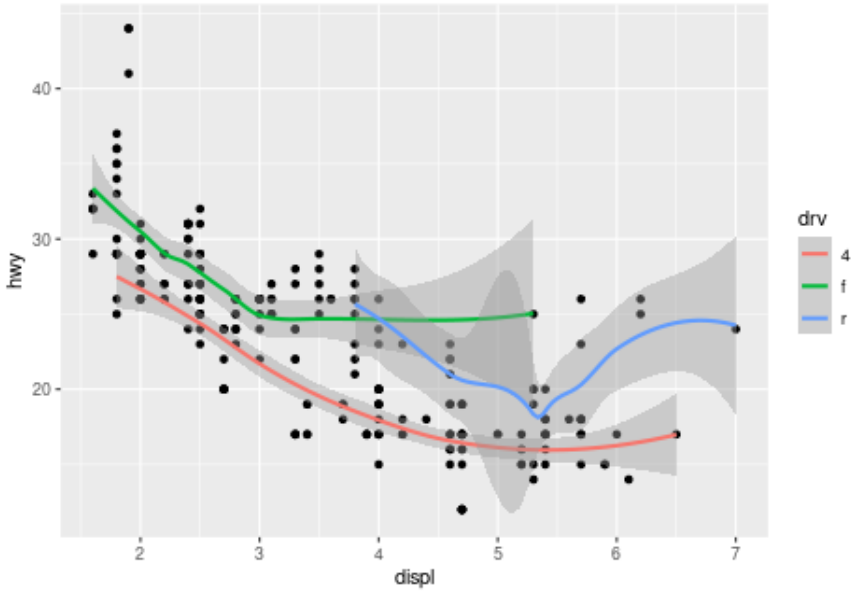
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



b)

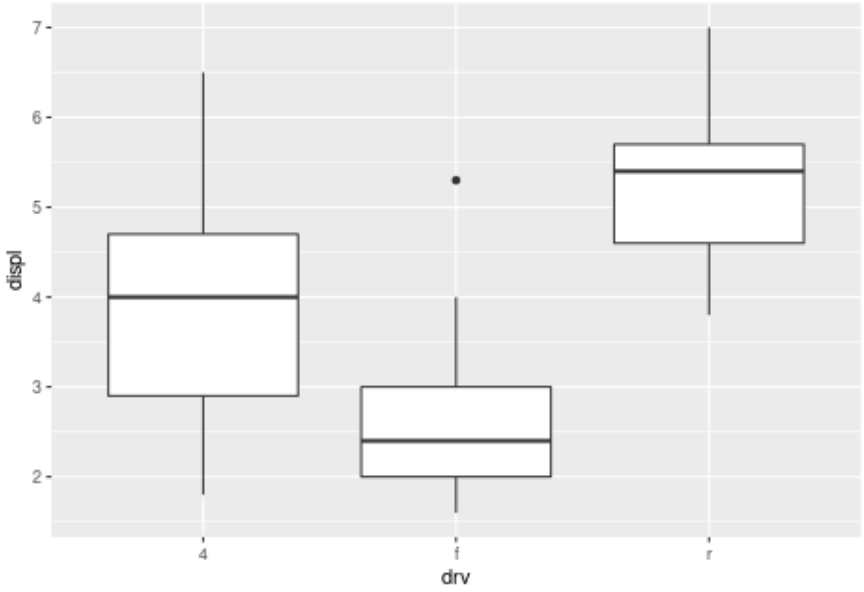
- Similar plot as last time, but we will get a different smooth line per drive.

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



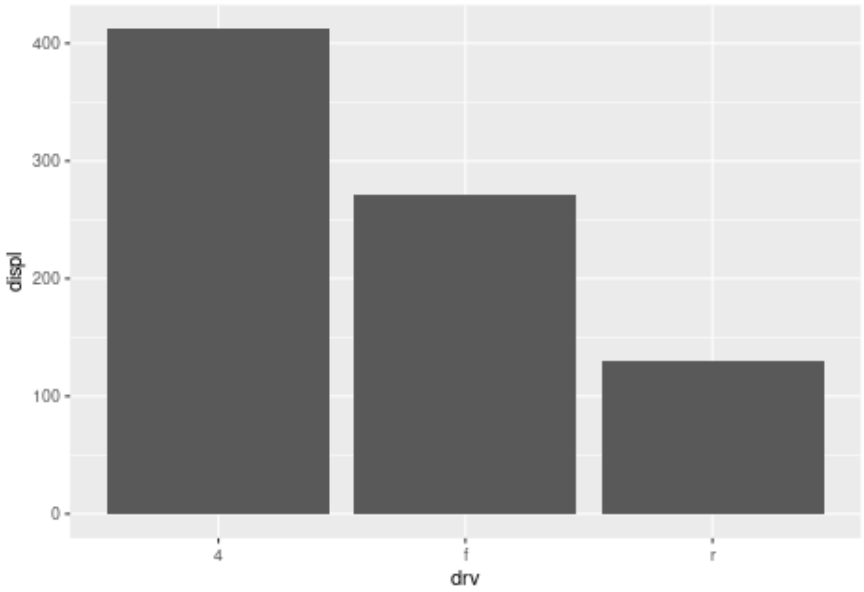
Exercise 11:

a) Modify the command, ggplot(data = mpg), so that the following layer is added to the plot.



We get a boxplot showing summary statistics of placement based on different values of drv.

b) What happens if you use geom_col() in place of geom_boxplot()?

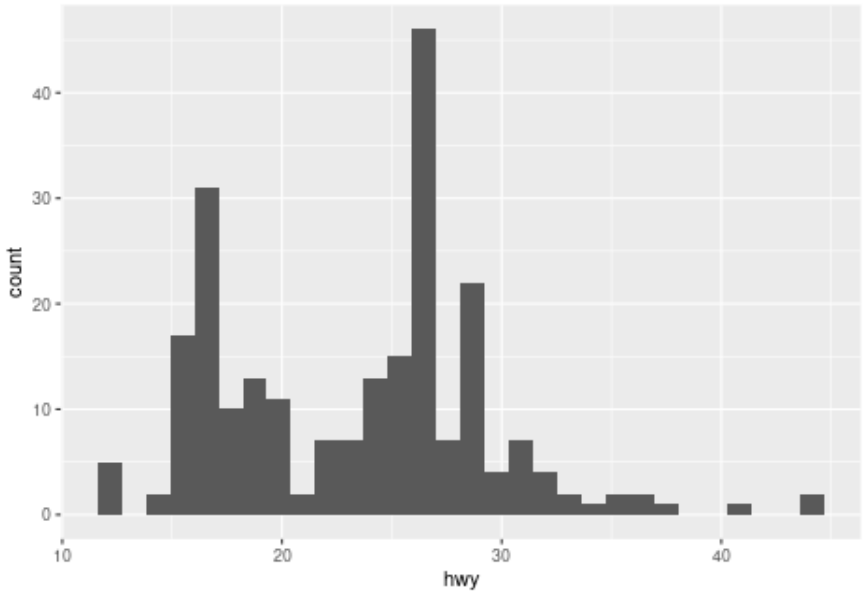


This shows a box plot of the sum of displacements based on the type of drv.

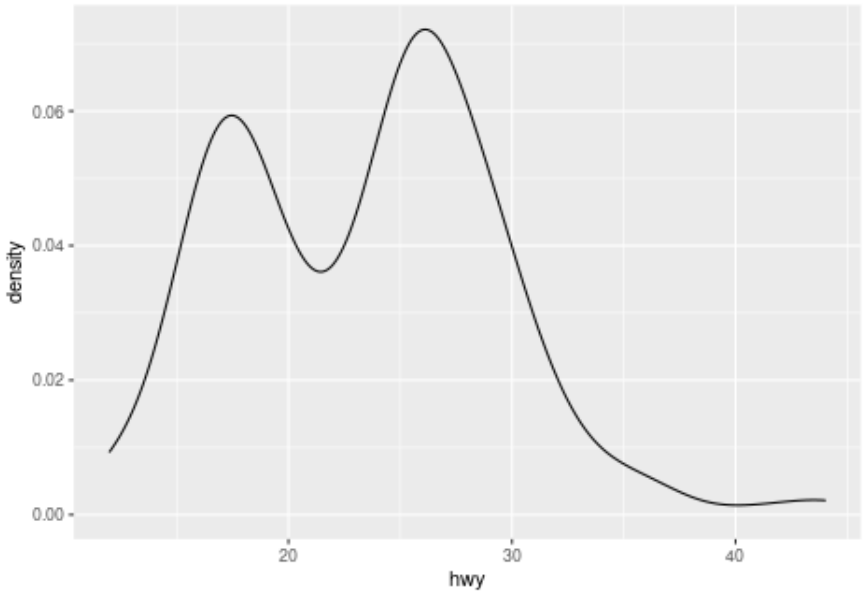
Exercise 12:

a) Use ggplot() and geom_histogram() to make a histogram of hwy.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

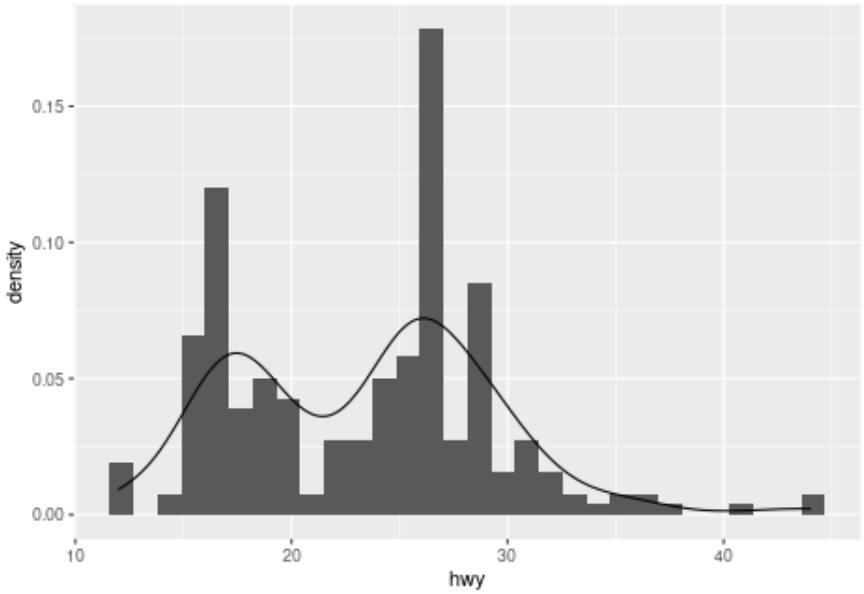


b) Replace geom_histogram() in your command with geom_density.



c) Stack both geom_histogram and geom_density on the same plot.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



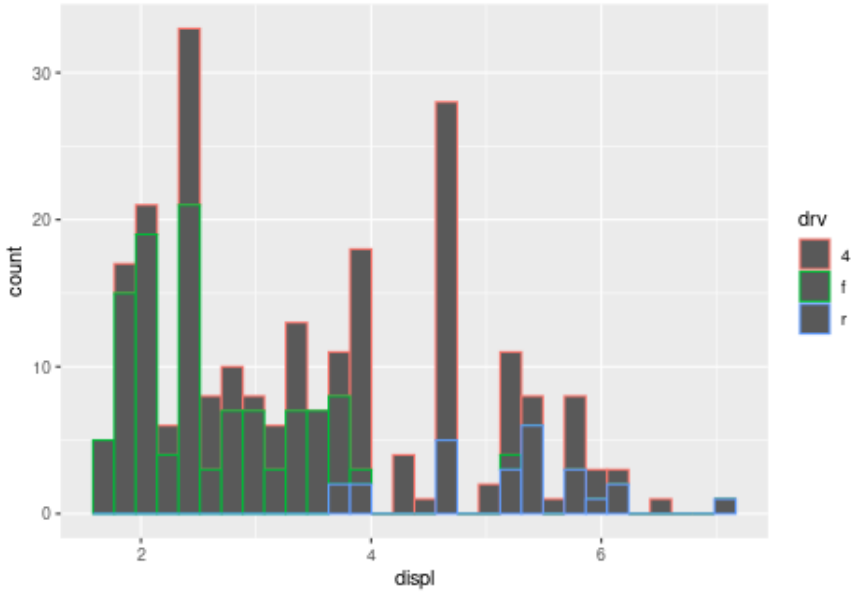
We have both the histogram and the density plot on the same graphic.

4.4: More on Faceting

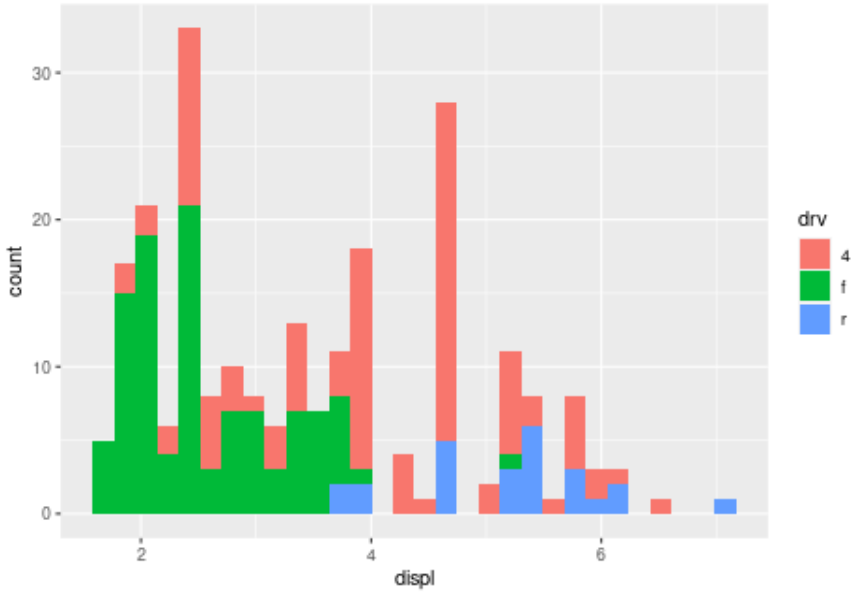
Exercise 13:

a) Which graph do you prefer?

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



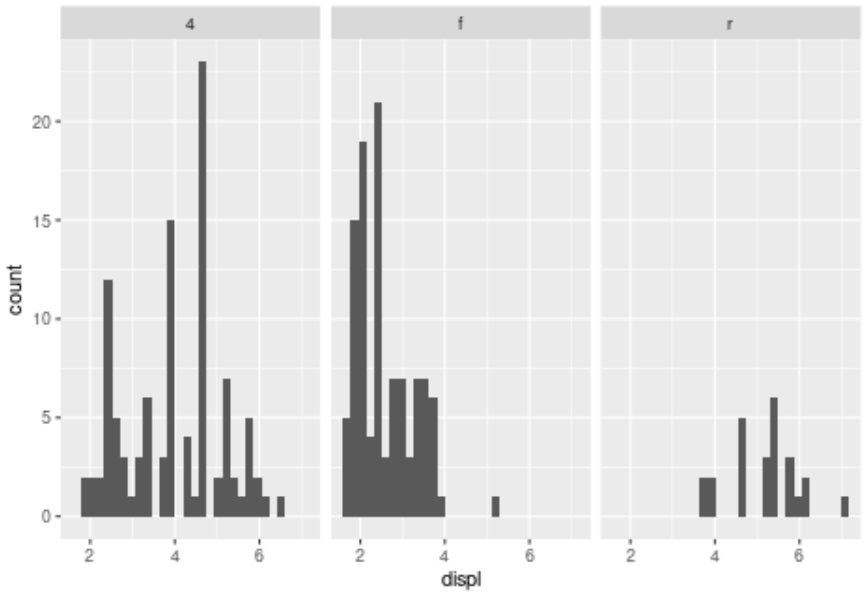

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



I think the second plot is far more legible. I much prefer it, though it could probably use some outlines as well.

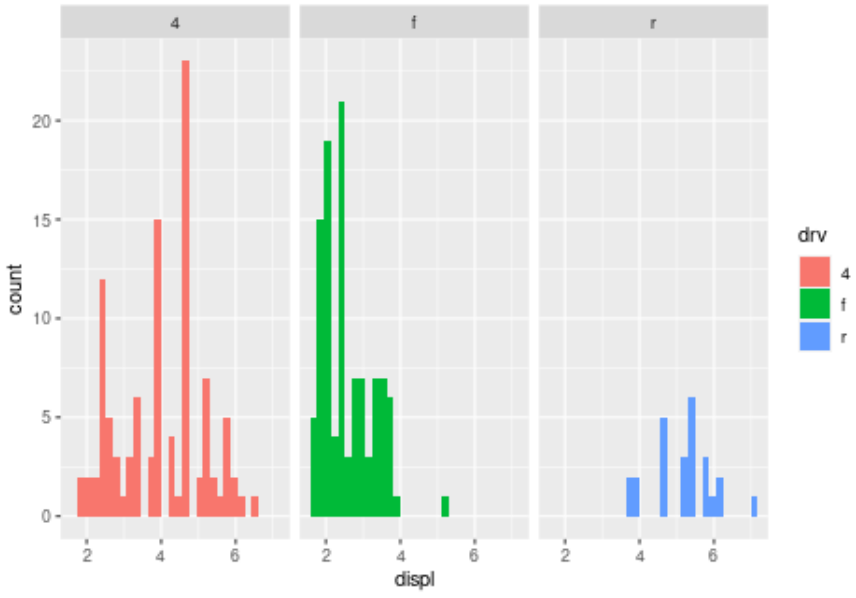
b) Alter the given code to utilize facet_wrap(), with facets = ~ drv.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



c) Add another aesthetic mapping, fill = drv to the previous plot.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



4.5: Statistical Transformations

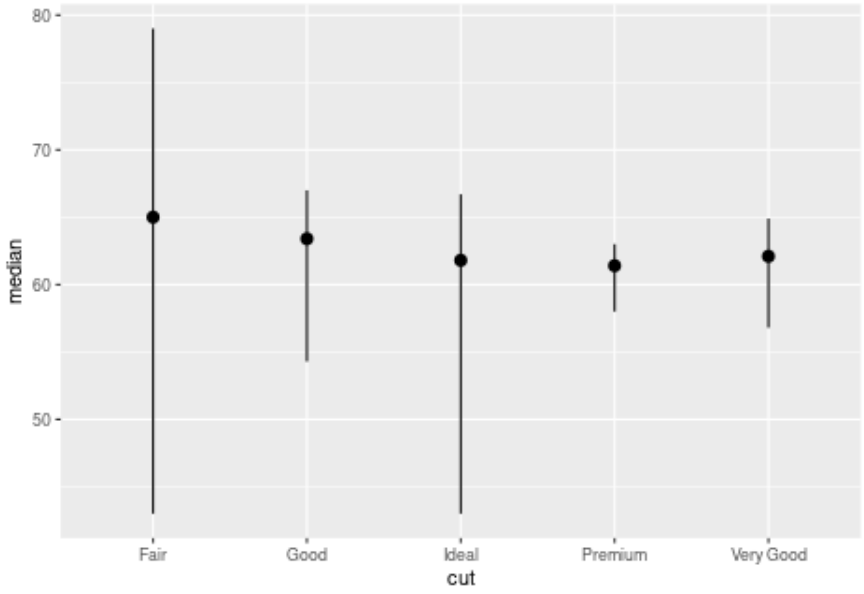
Exercise 14:

a) What is the default type of **geometric object**?

- **pointrange**

b) Verify that **geom_pointrange** can be used to duplicate the function of stat_summary.

```
grouped_by_cut <- data.frame(
    cut = c("Fair", "Good", "Very Good",
            "Premium", "Ideal"),
    lower = c(43.0, 54.3, 56.8, 58.0, 43.0),
    upper = c(79.0, 67.0, 64.9, 63.0, 66.7),
    median = c(65.0, 63.4, 62.1, 61.4, 61.8))
ggplot(data = grouped_by_cut) +
    geom_pointrange(mapping = aes(x = cut,
                                  y = median,
                                  ymin = lower,
                                  ymax = upper))
```



Exercise 15 / 16

Look under computed variables in the help page for stat_smooth(). What statistical values does it compute?

- predicted value, lower and upper pointwise confidence intervals.

What does geom_col()? How does it differ from geom_bar?

- geom_bar makes the height of the bars reference the *count* of that value, in other words it references the *frequency* of a variable in a data set. Geom_col on the other hand uses values in the data itself as the height.

4.6: Position Adjustments

Exercise 17

What is geom_bar()'s default **position adjustment**?

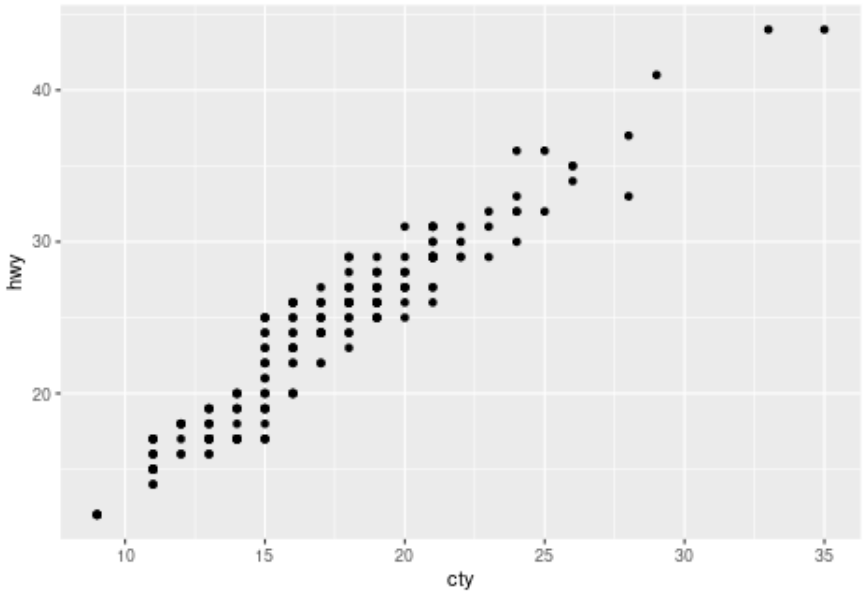
- **stack**

What is the default position adjustment for geom_point()?

- **identity**

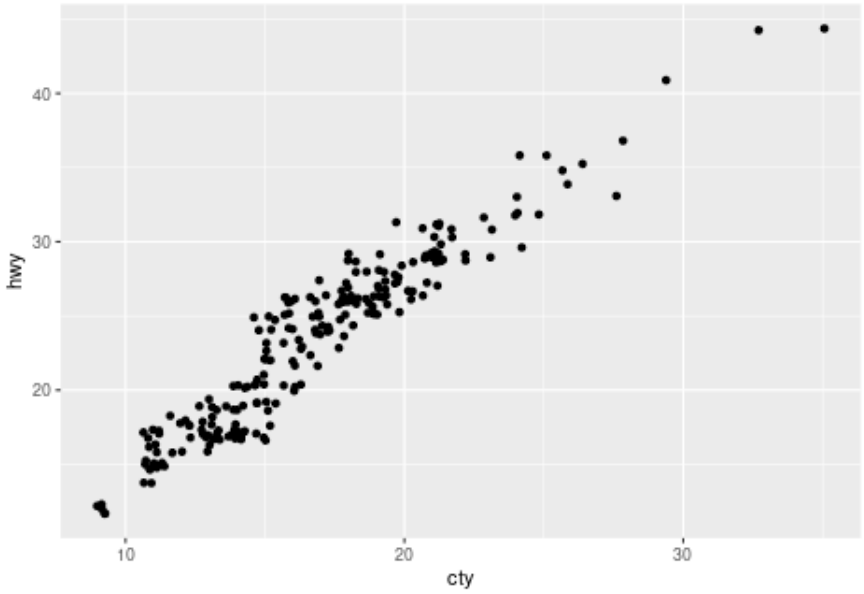
Exercise 18:

a) **Overplotting** is when points in a scatterplot overlap? What's the problem with the following plot? **Hint:** The mpg data set contains 234 observations.



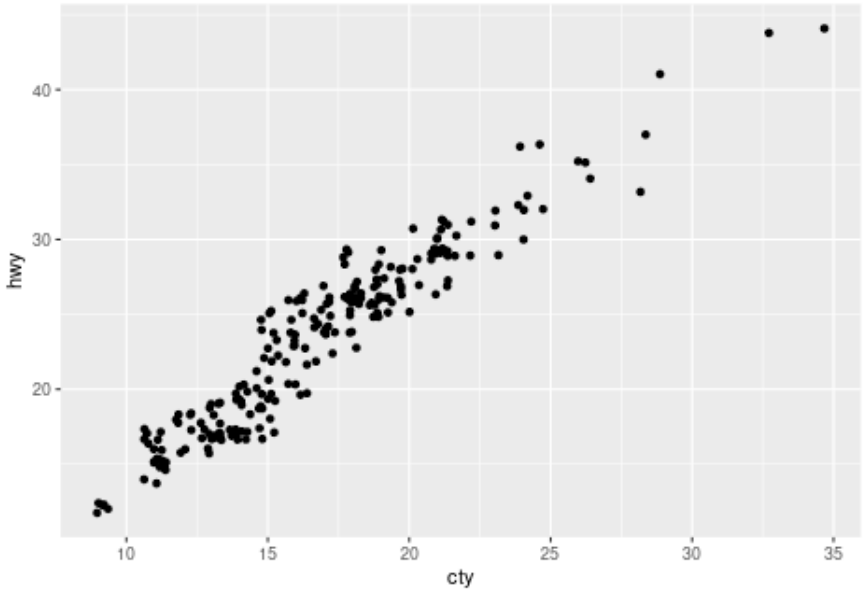
There are definitely not 234 points on this map. It's misrepresenting the density of the points because they're overlapping. It's not an honest representation of the data.

b) Re-run the command above using position = "jitter". Describe the difference.

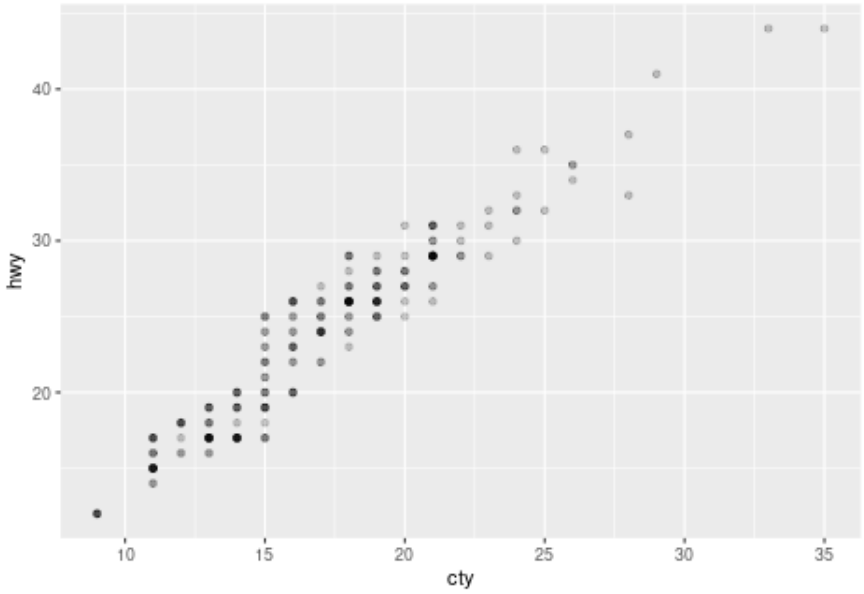


The dots are actually allowed to overlap here! It looks far more natural and doesn't so neatly align with the grid. Tinkering with it a different formation of the points is generated every time. That's due to the random noise that's generated to encourage less overlapping.

c) Jittering is a very common practice with scatterplots, so common in fact there is a geom for it! Use it here.



d) Another solution to overplotting is setting the opacity value lower.



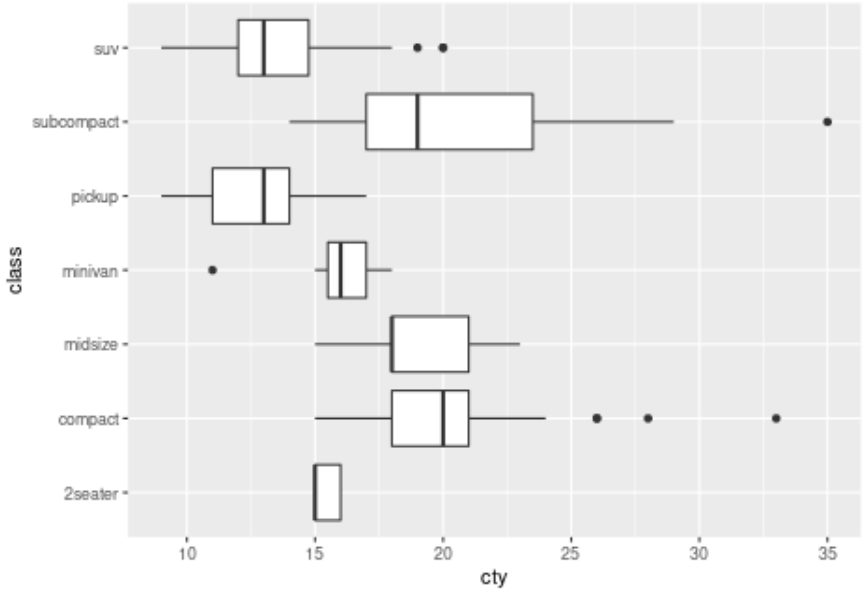
This is interesting, I suppose this is similar to having a shaded visual component to the plot. The darker the points the more points there are, because enough .2 opacity points still creates a dark circle. So this allows you to show density without adding in random noise.

4.7: Coordinate Systems

Exercise 19:

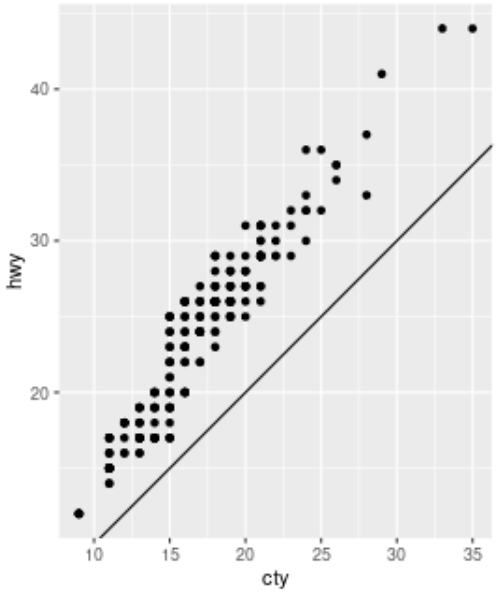
Using the code provided flip the coordinates of the boxplot to make it horizontal.

```
ggplot(data = mpg) +
    geom_boxplot(mapping = aes(x = class, y = cty)) +
    coord_flip()
```



Exercise 20:

What does the given plot tell you about city and highway mpg? Why is coord_fixed() important? What does geom_abline() do?



What this shows is that highway miles per gallon is pretty much always higher than city miles per gallon. That's what geom_abline() is for. It's a line with a slope of 1, so points being above or below that line give information about which (x or y) is larger for a specific point. And, with all of points together, you get a story about a deeper relationship! coord_fixed() is very useful for this as well, as it forces the same scale for the x and y values.

4.10: Mosaic Plots

Exercise 21:

a) Make a mosaic plot of the cut and color variables in the **diamonds** data set.

```
# ggplot(data = diamonds) +
#     ggmosaic::geom_mosaic(
#         mapping = aes(x = ggmosaic::product(cut, color)
#         )
#     )
```

This code throws an error. I had to finish this up at home, I remember in class that you mentioned there is a typo in this problem. I'm not sure what that typo is so I unfortunately cannot complete this exercise! I can at least do part c though.

c) Generate a table of color and cut.

```
##
##     Fair Good Very Good Premium Ideal
##   D  163  662      1513    1603  2834
##   E  224  933      2400    2337  3903
##   F  312  909      2164    2331  3826
##   G  314  871      2299    2924  4884
##   H  303  702      1824    2360  3115
##   I  175  522      1204    1428  2093
##   J  119  307       678     808   896
```