homework_2_vardhan_vishnu

Q 3.2.4

Answer

```
ggplot(data=mpg)
```



```
glimpse(mpg)
```

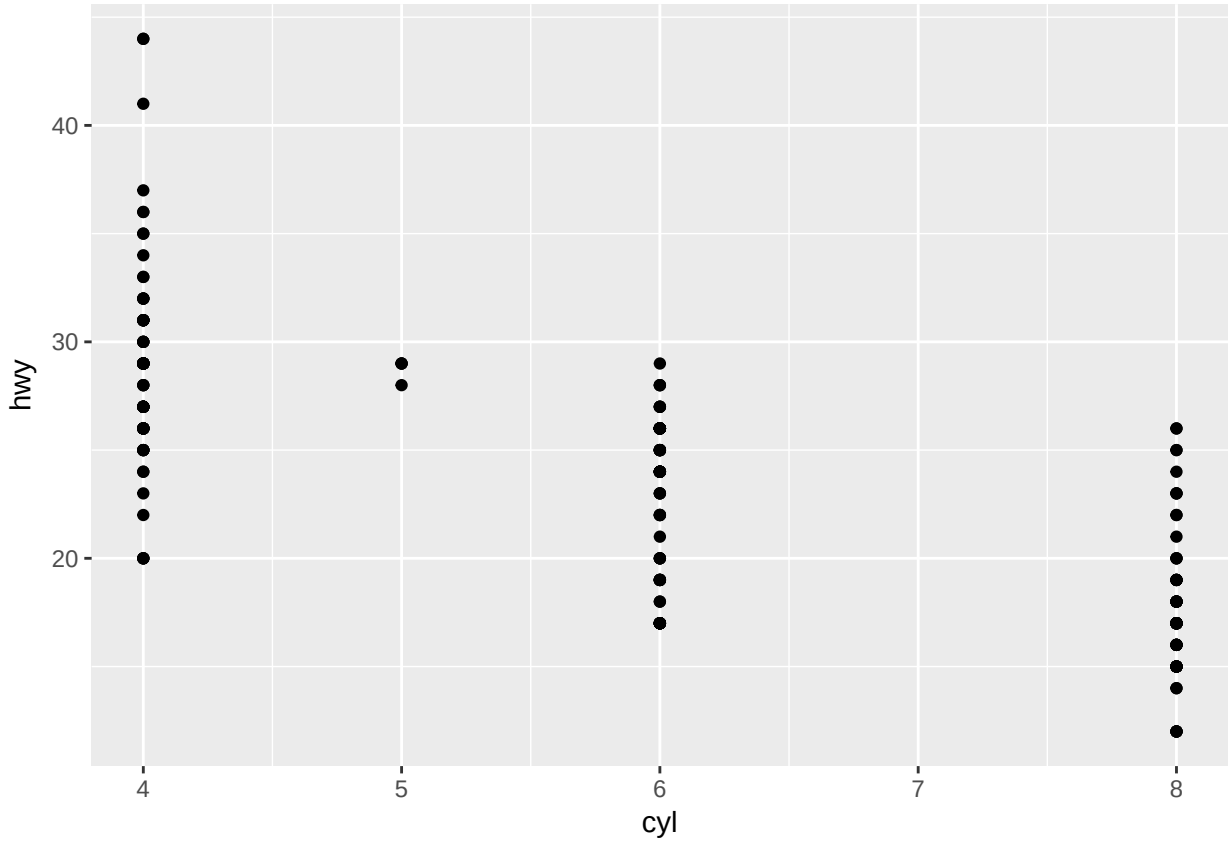
```
## Observations: 234
## Variables: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "...
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 qua...
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0,...
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1...
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6...
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)...
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4",...
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 1...
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 2...
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p",...
## $ class        <chr> "compact", "compact", "compact", "compact", "comp...
```

Dataset 'mpg' has 11 columns, and 234 rows

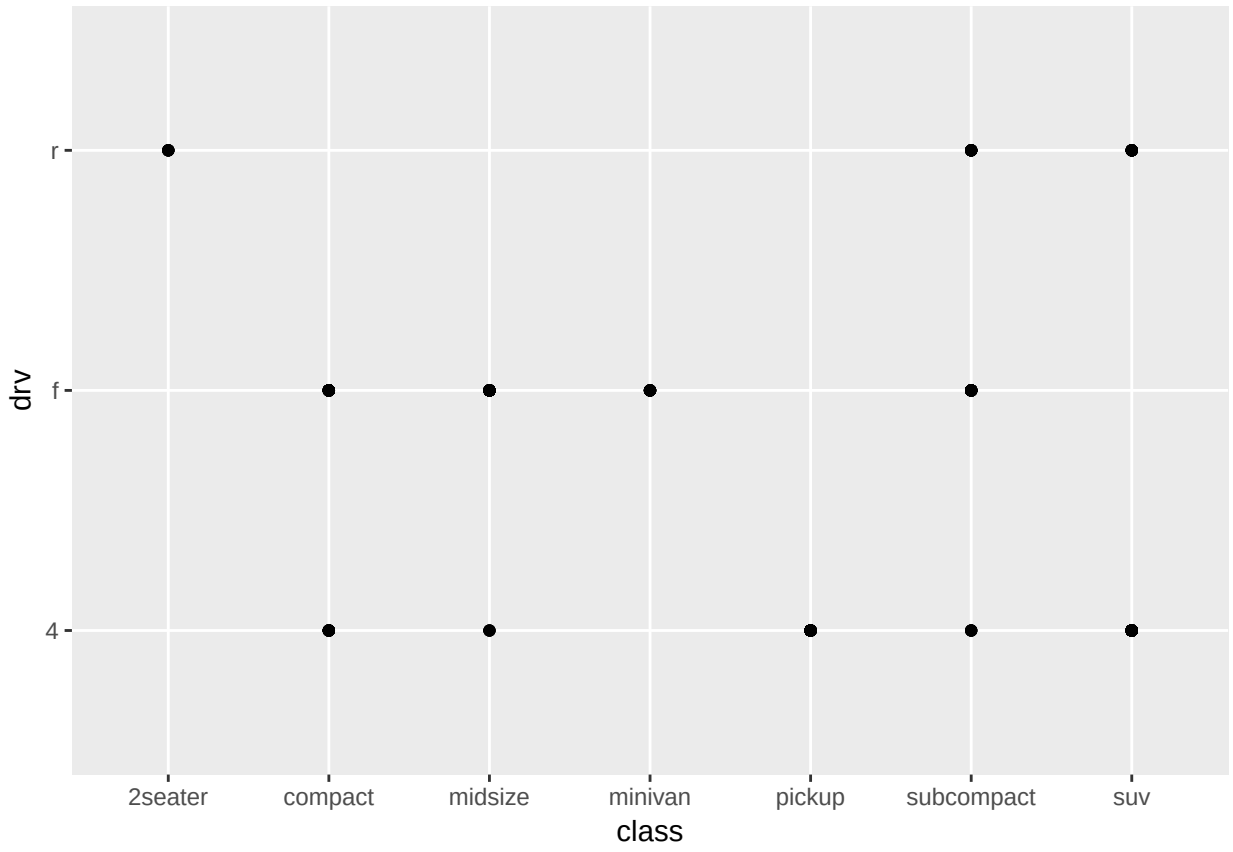
```
?mpg
```

'drv' represents the kind of drive the car has. One of 3 options (4 wheel drive, front wheel drive, rear wheel drive)

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




```
ggplot(data = mpg, mapping =aes(x=class, y=drv)) + geom_point()
```



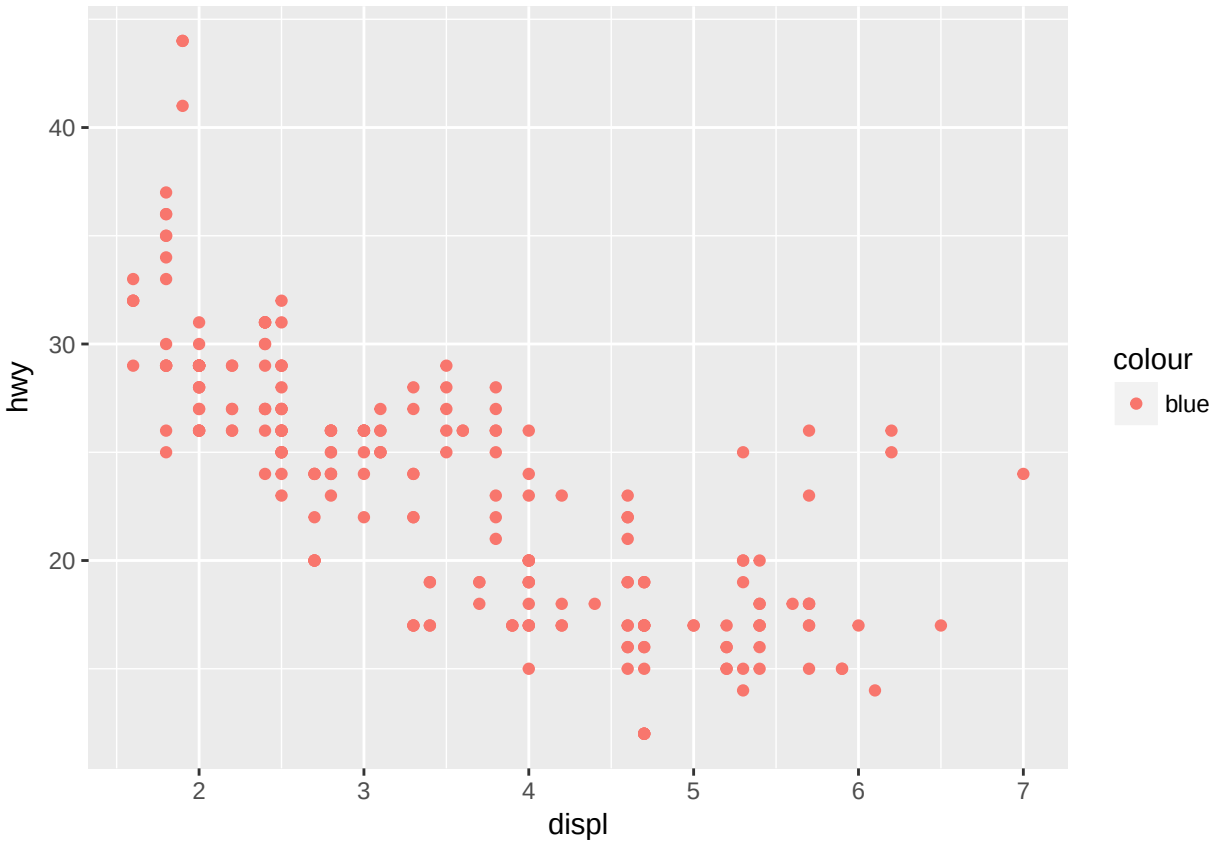
This scatter plot is not useful, because we are plotting two categorical variables versus each other. It (a scatter plot) finds it hard to reflect the number of instances of a particular combination, which is what we are looking for.

Q 3.3.1

What's gone wrong with this code? Why are the points not blue?

Answer

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



The colour is not blue, because ggplot is searching for a column with the value "blue", and not the colour 'blue'

Q.3.3.2

Which variables in mpg are categorical? Which variables are continuous? (Hint: type ?mpg to read the documentation for the dataset). How can you see this information when you run mpg?

Answer

```
mpg
```

```
## # A tibble: 234 x 11
##    manufacturer      model displ  year   cyl      trans   drv   cty   hwy
##           <chr>      <chr> <dbl> <int> <int>      <chr> <chr> <int> <int>
##  1         audi         a4   1.8  1999     4   auto(l5)     f    18    29
##  2         audi         a4   1.8  1999     4 manual(m5)     f    21    29
##  3         audi         a4   2.0  2008     4 manual(m6)     f    20    31
##  4         audi         a4   2.0  2008     4   auto(av)     f    21    30
##  5         audi         a4   2.8  1999     6   auto(l5)     f    16    26
##  6         audi         a4   2.8  1999     6 manual(m5)     f    18    26
##  7         audi         a4   3.1  2008     6   auto(av)     f    18    27
##  8         audi a4 quattro   1.8  1999     4 manual(m5)     4    18    26
##  9         audi a4 quattro   1.8  1999     4   auto(l5)     4    16    25
## 10         audi a4 quattro   2.0  2008     4 manual(m6)     4    20    28
## # ... with 224 more rows, and 2 more variables: fl <chr>, class <chr>
```

```
?mpg
```

Categorical variables are: Model, Manufacturer, displ, year, cyl, trans, drv, fl, class Non Categorical variables are: city miles per gallon, highway miles per gallon

```
unique(mpg$displ)
```

```
##  [1] 1.8 2.0 2.8 3.1 4.2 5.3 5.7 6.0 6.2 7.0 6.5 2.4 3.5 3.6 3.0 3.3 3.8
## [18] 4.0 3.7 3.9 4.7 5.2 5.9 4.6 5.4 5.0 1.6 2.5 2.7 6.1 4.4 5.6 2.2 3.4
## [35] 1.9
```

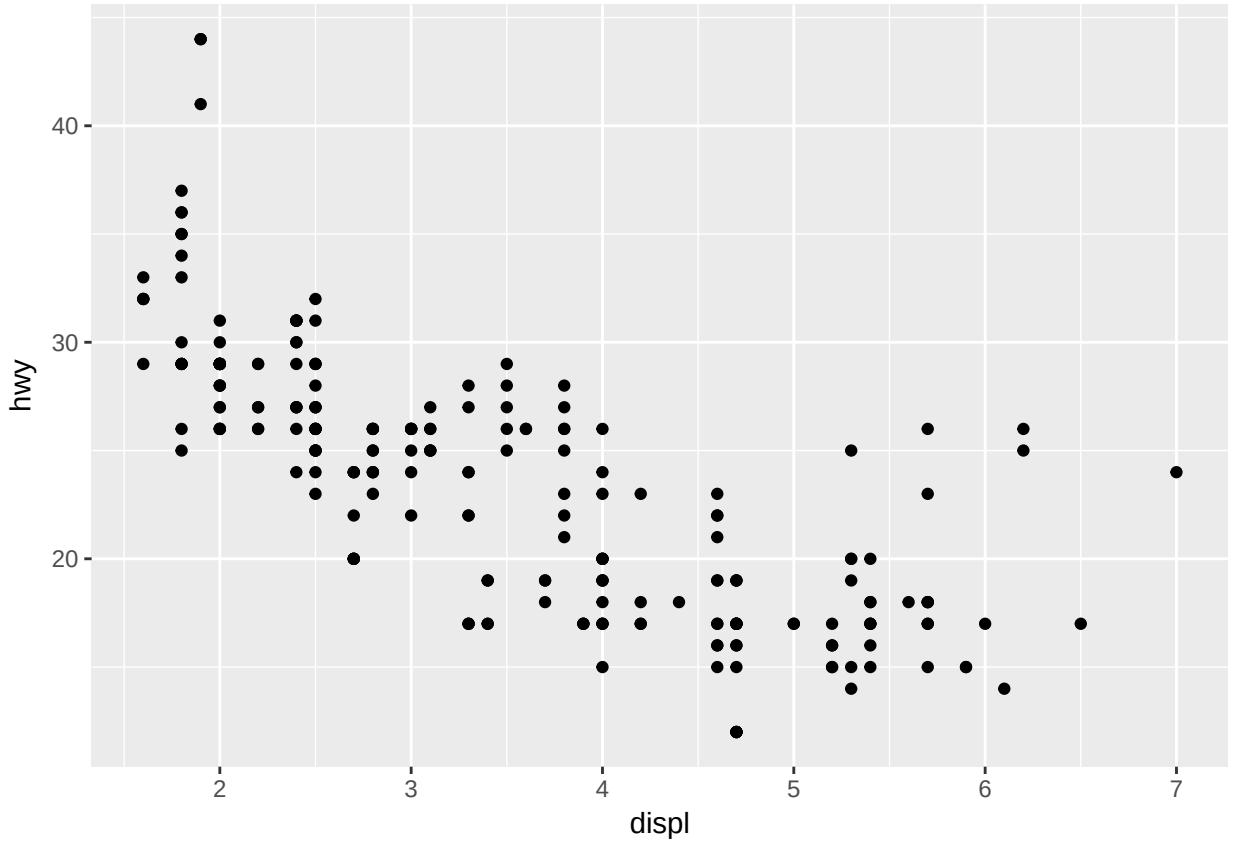
Q 3.3.3

Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?

Answer

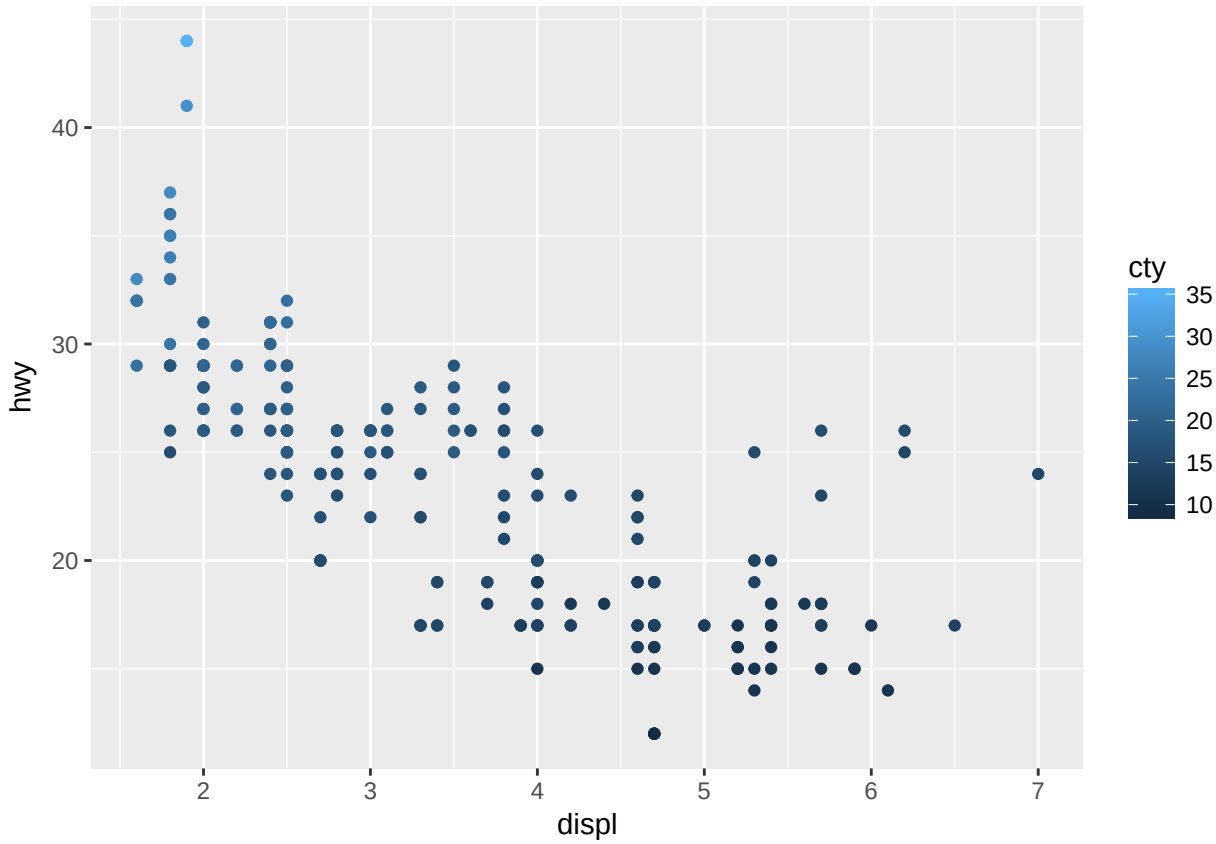
Lets plot it without any asthetics

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



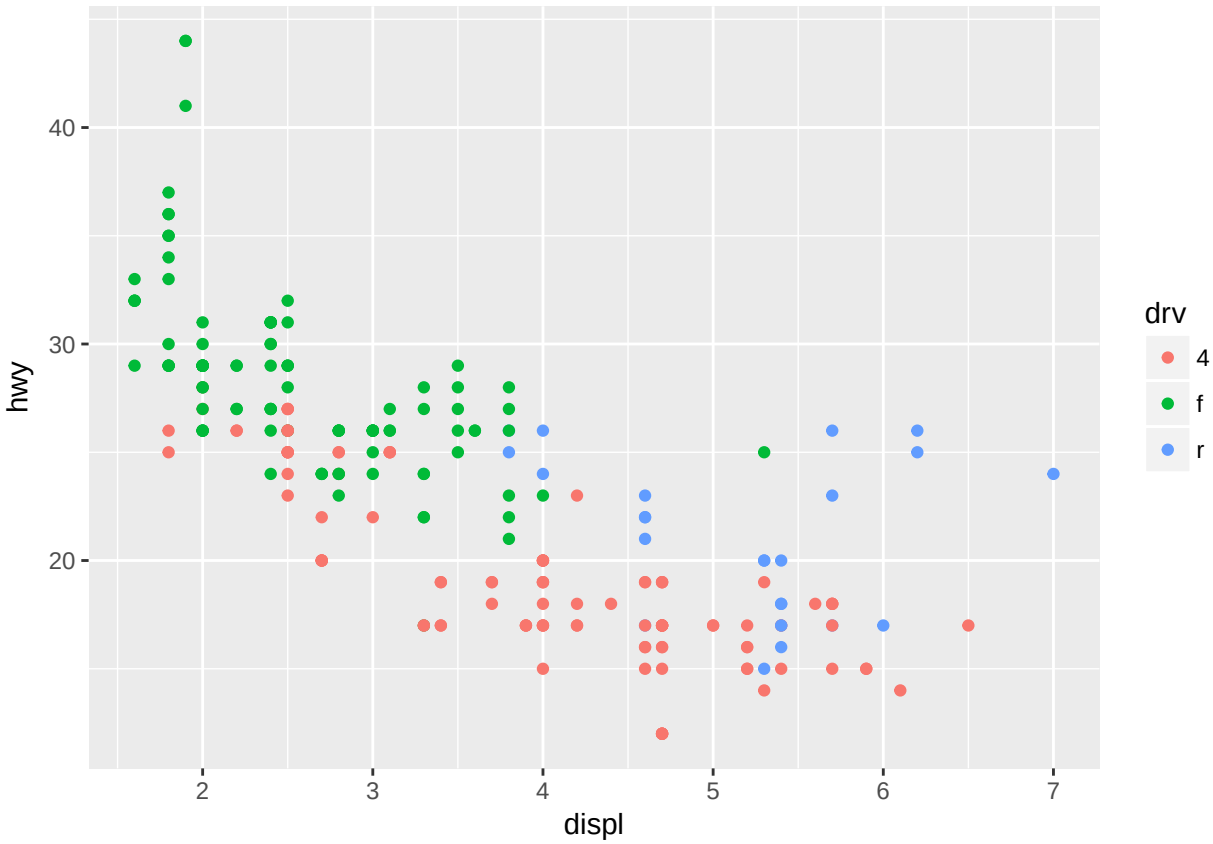
Now lets add a continuous variable to colour

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



now, lets add a categorical variable to color

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



The difference between a categorical and continous variable in the selection of the colour, is which colour band it selects. With continuous variables it is using different shades of one colour while with categorical R is using different colours

Q 3.3.4

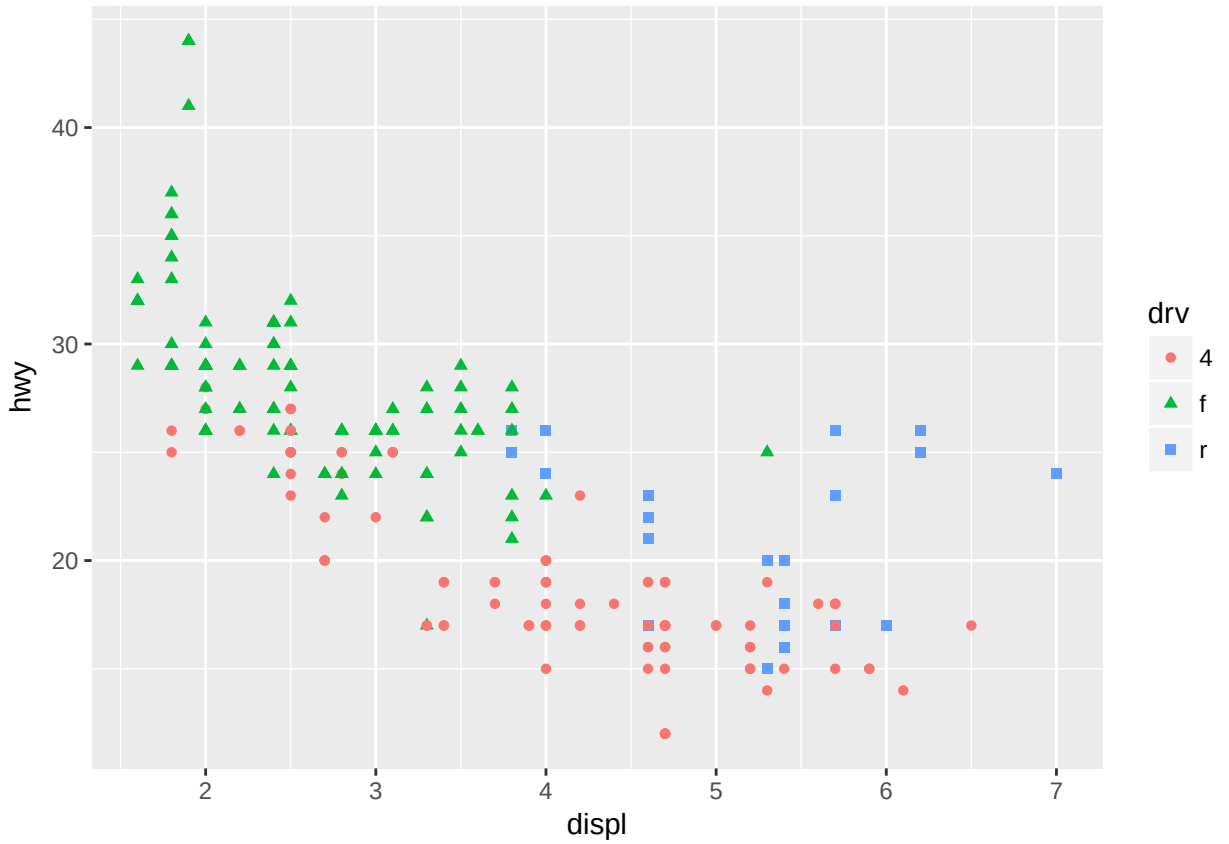
What happens if you map the same variable to multiple aesthetics?

Answer

You are essentially wasting one additional visual cue that can be used to discriminate data.

Below 'drv' is used for both shape and colour, versus using shape for showing an additional dimension of data

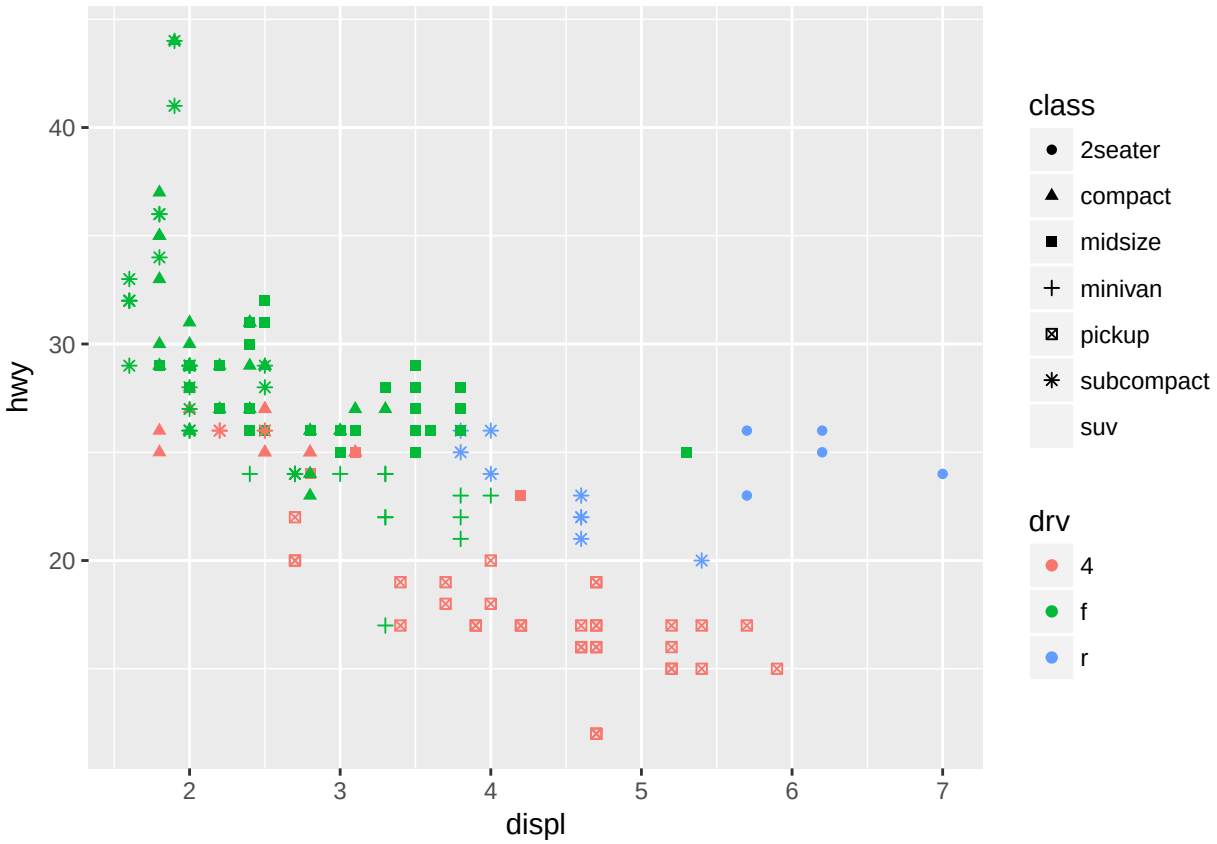
```
ggplot(data = mpg, mapping = aes(x = displ, y =hwy, color = drv, shape = drv)) + geom_point()
```




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

```
## 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).
```



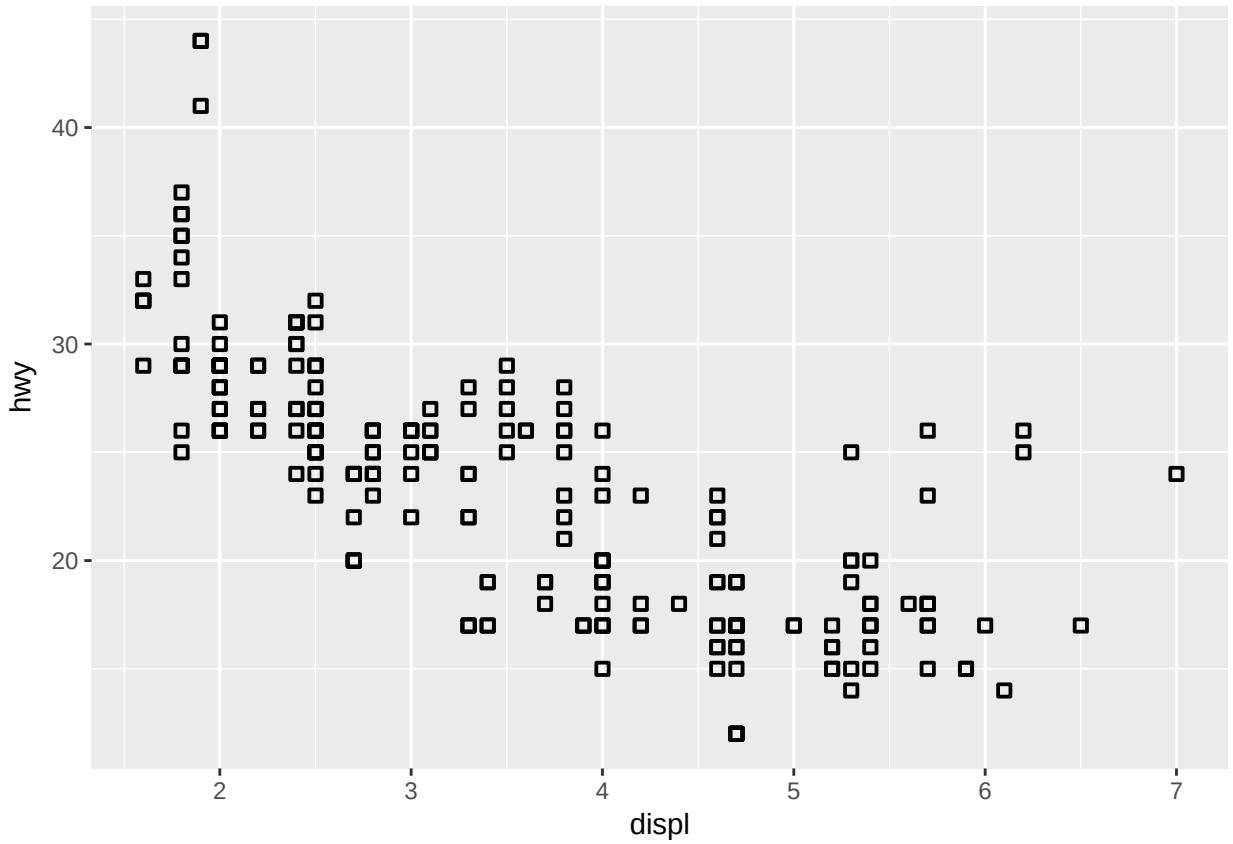
Q 3.3.5

What does the stroke aesthetic do? What shapes does it work with? (Hint: use ?geom_point)

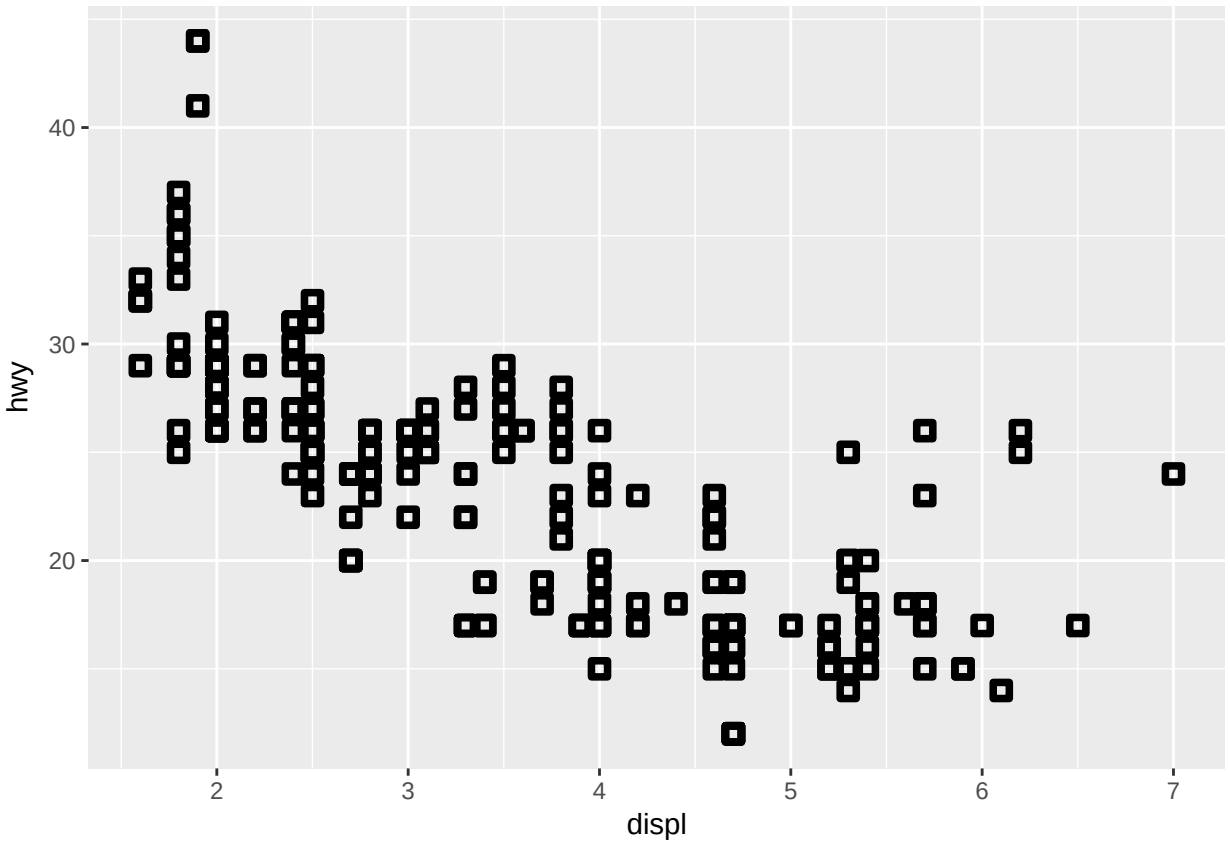
Answer

'stroke' specifies the thickness of the border of a particular shape, measured in 'mm'. As you can see below, changing stroke from 1 to 2 doubles the thickness of the border

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, stroke = 1)) + geom_point(shape = 0)
```




```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, stroke = 2)) + geom_point(shape = 0)
```



Q 3.3.6

What happens if you map an aesthetic to something other than a variable name, like aes(colour = displ < 5)?

Answer

Using a condition, plots the results as a true / false statement. See the step by step exploration below.

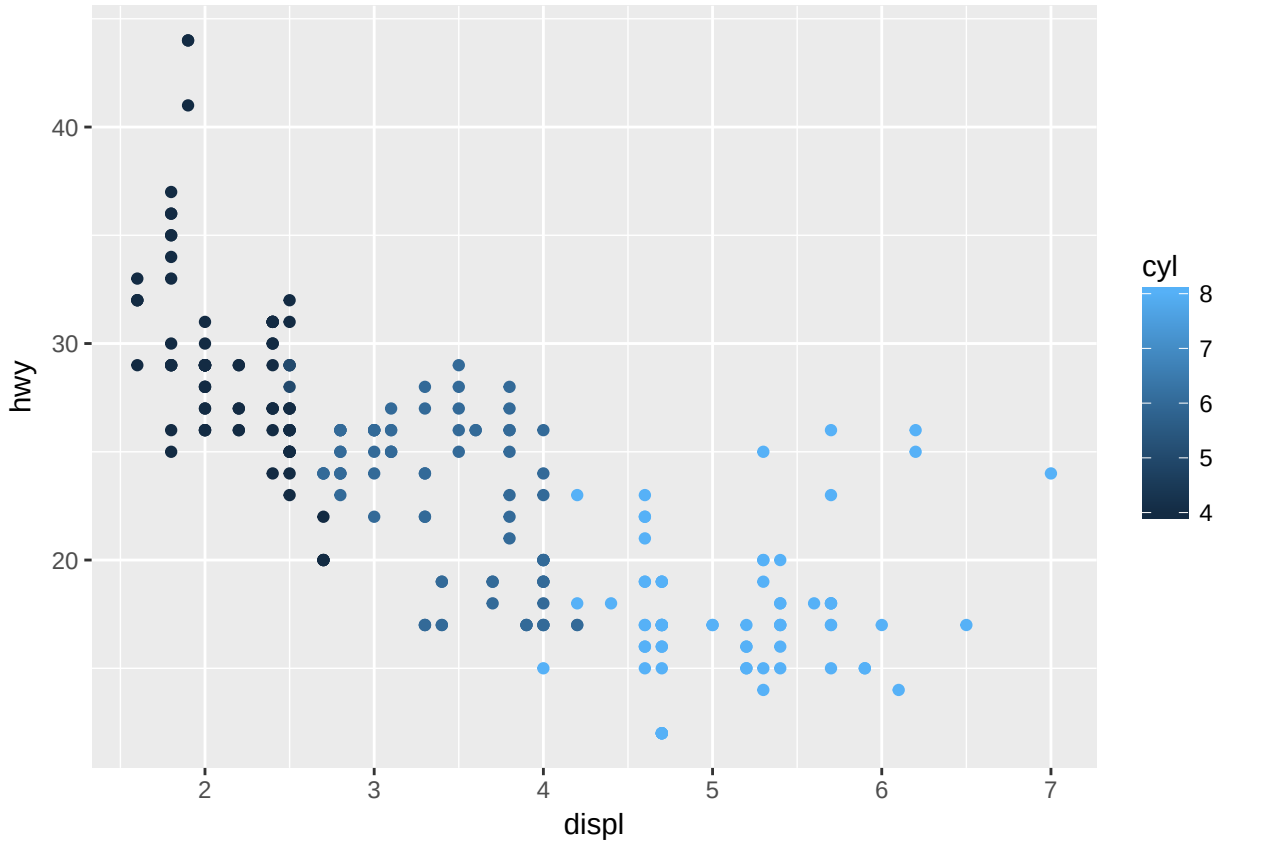
Lets check the variables we have to play with, as a refresher

```
glimpse(mpg)
```

```
## Observations: 234
## Variables: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "...
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 qua...
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0,...
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1...
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6...
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)...
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4",...
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 1...
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 2...
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p",...
## $ class        <chr> "compact", "compact", "compact", "compact", "comp...
```

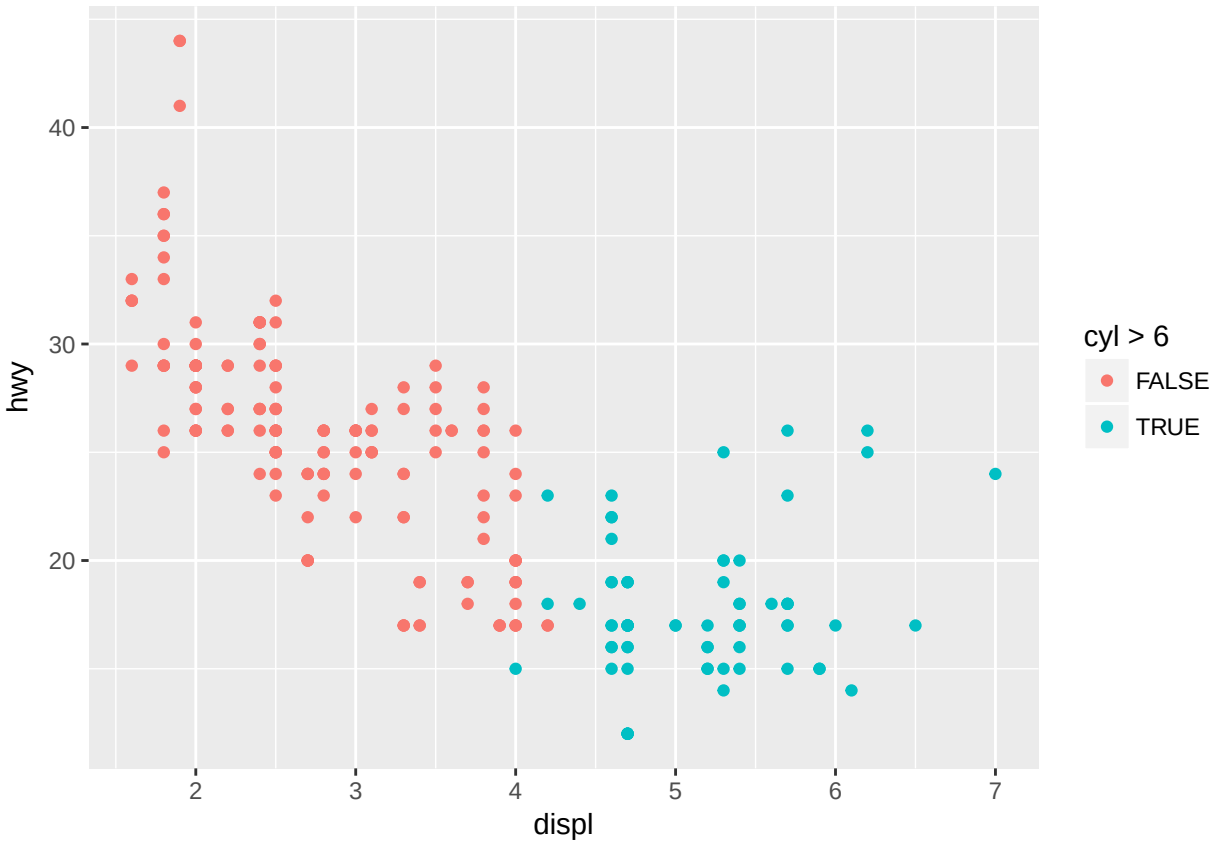
Lets plot the highway gallons per mile vs the displacement, and colour the output by number of cylinders

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



Now, lets plot this as a condition, as asked in the question. As you can see below, it converts this to a 'true/false' and plots the results.

```
ggplot(data=mpg, mapping = aes (x=displ, y = hwy, color = cyl > 6)) + geom_point()
```



Q 3.5.4

Take the first faceted plot in this section:

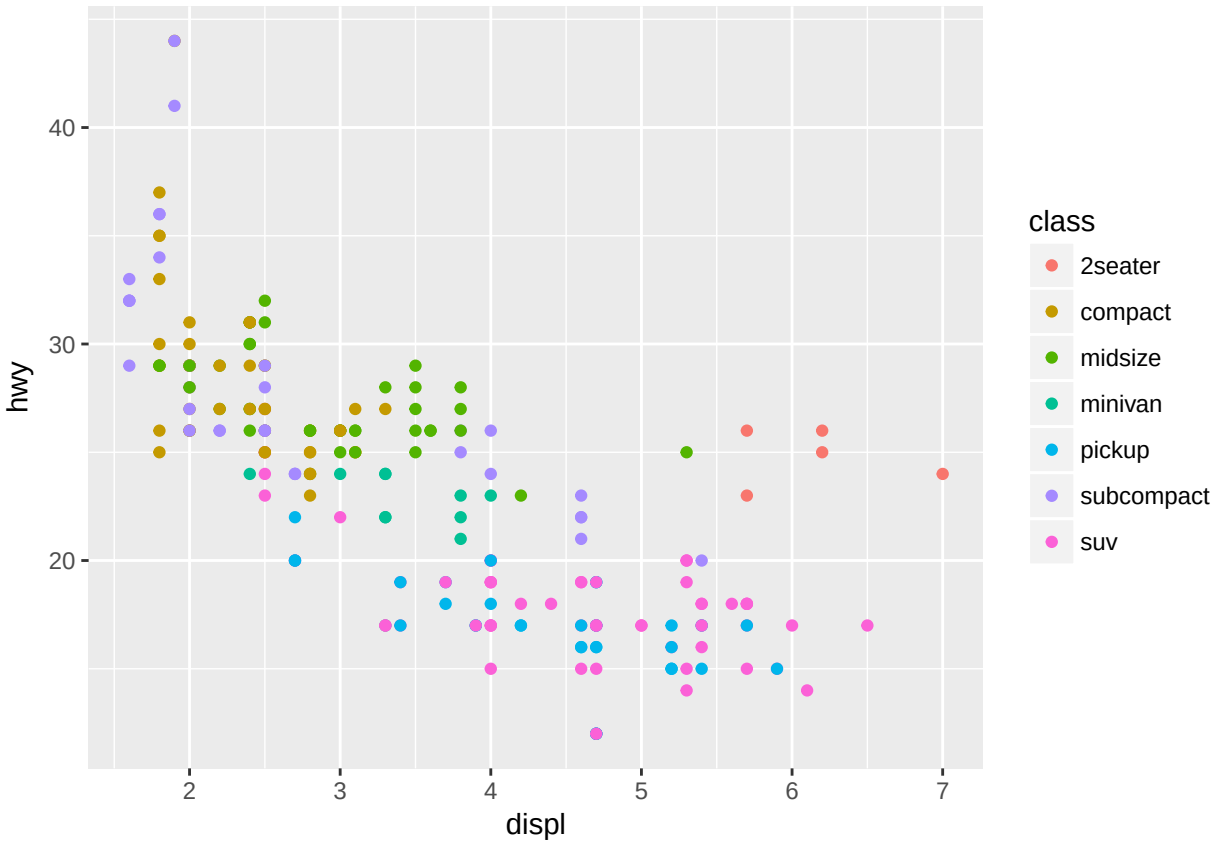
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)) + facet_wrap(~ class, nrow = 2)

What are the advantages to using faceting instead of the colour aesthetic? What are the disadvantages? How might the balance change if you had a larger dataset?

Answer:

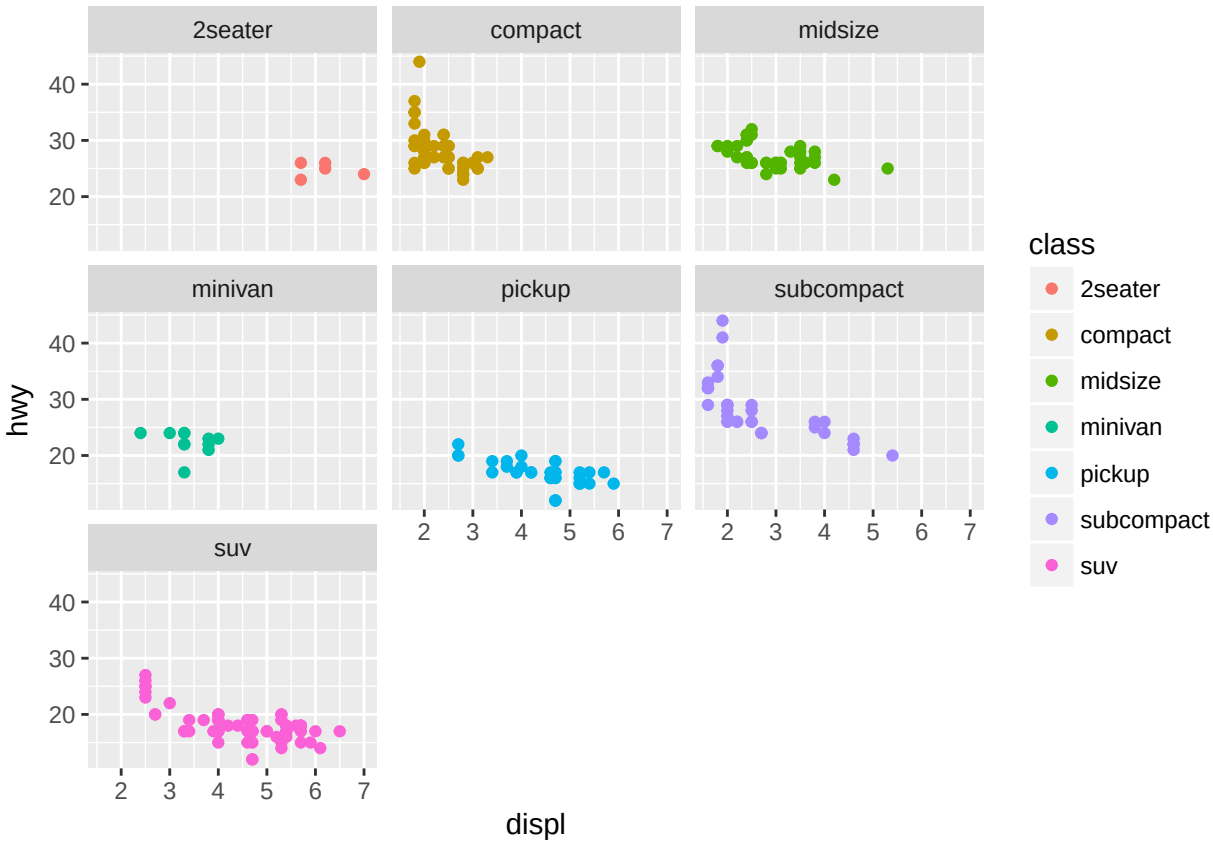
Lets plot this first to see how it looks, without the facet. You can see some trends, but not as clear.

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



Lets add some facets. Clearly, there is more clear conclusions that can be drawn pretty quickly, that are lost if only colours were used.

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



The disadvantage is that exact data points are harder to discern, but trends are much easier. Also, if the number of categorical variables was large, it would be hard to use facets to make out the differences between the different values.

Q 3.6.5

Read ?facet_wrap. What does nrow do? What does ncol do? What other options control the layout of the individual panels? Why doesn't facet_grid() have nrow and ncol argument?

Answer:

nrow and ncol, specifies the number of rows and columns to plot the categories of a single categorical variable. facet_grid is used when we have two variables, and the rows and columns are fixed in the case of facet_grid by the categories of the two variables specified. In the case of a single variable, instead of layting out each graph sequentially, they can be laid out as a table for more efficient representation.

While laying out a set of graphs, you could either go row first, or column first, as specified by the 'dir' argument. By default all the graphs have the same scales (so it is easy to compare), but you can change the scale using the 'scales' argument, and once you change the scale, you may need to adjust the output using 'shrink'. Changing the scale may make it harder to compare accross different variables.

'strip.position' allows you to control where the categorical label is displayed. Previously, 'switch' was used, but the switch command has since been deprecated.

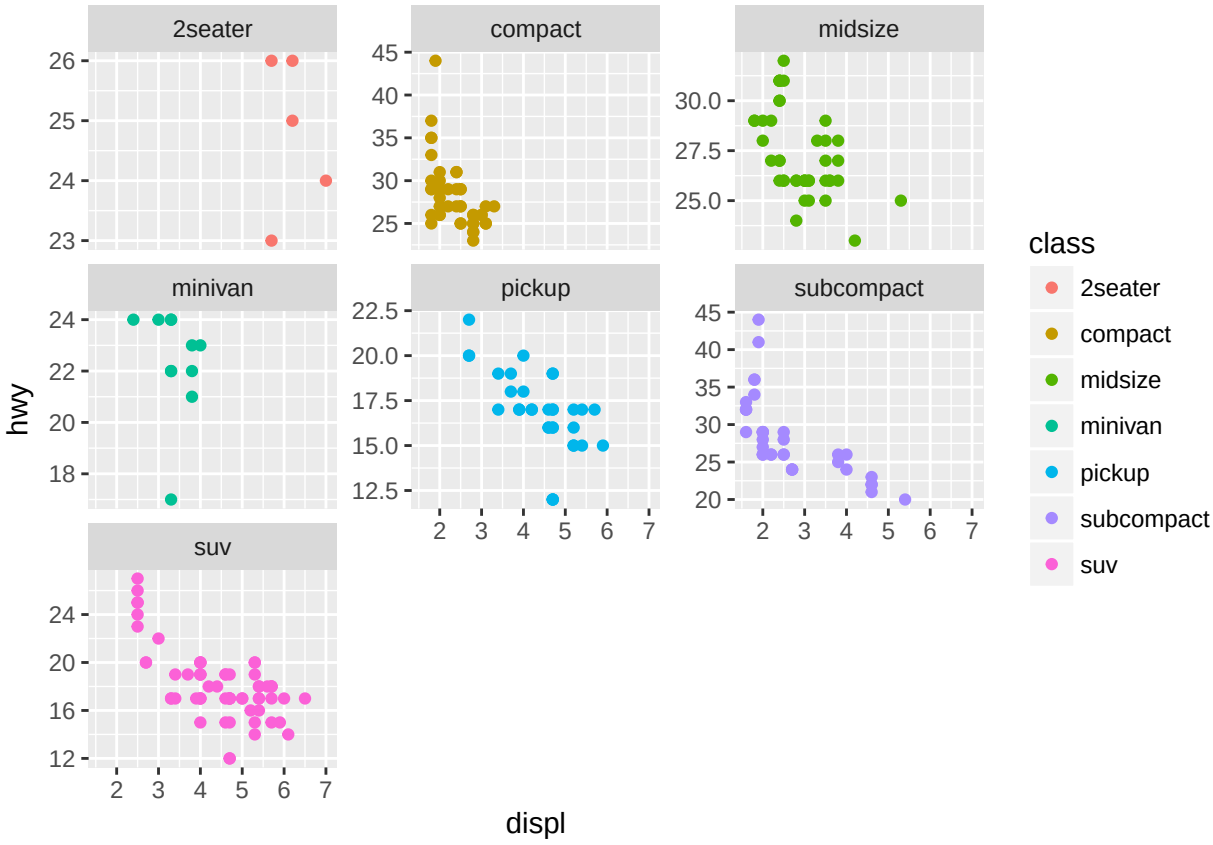
The detailed workings are outlined below.

```
?facet_wrap
```

```
?facet_grid
```

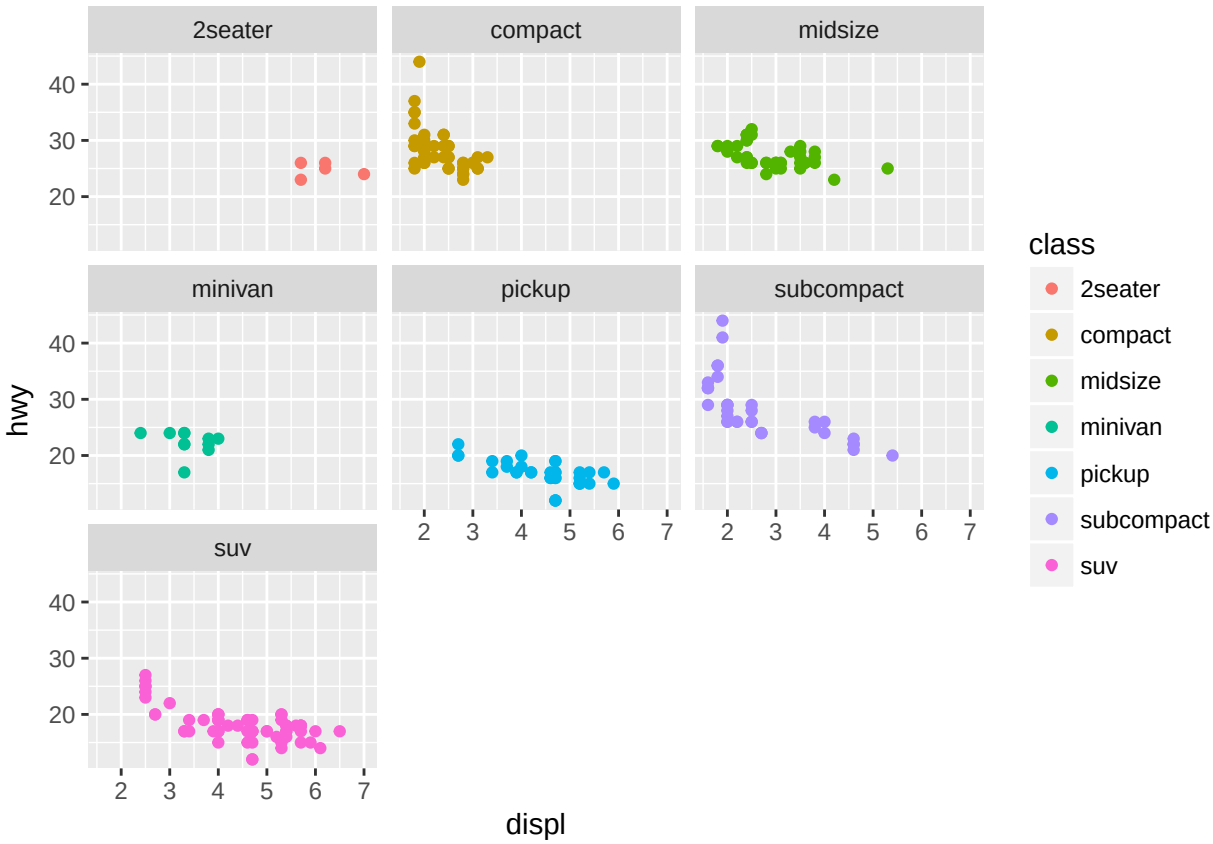
Using a 'scale'

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



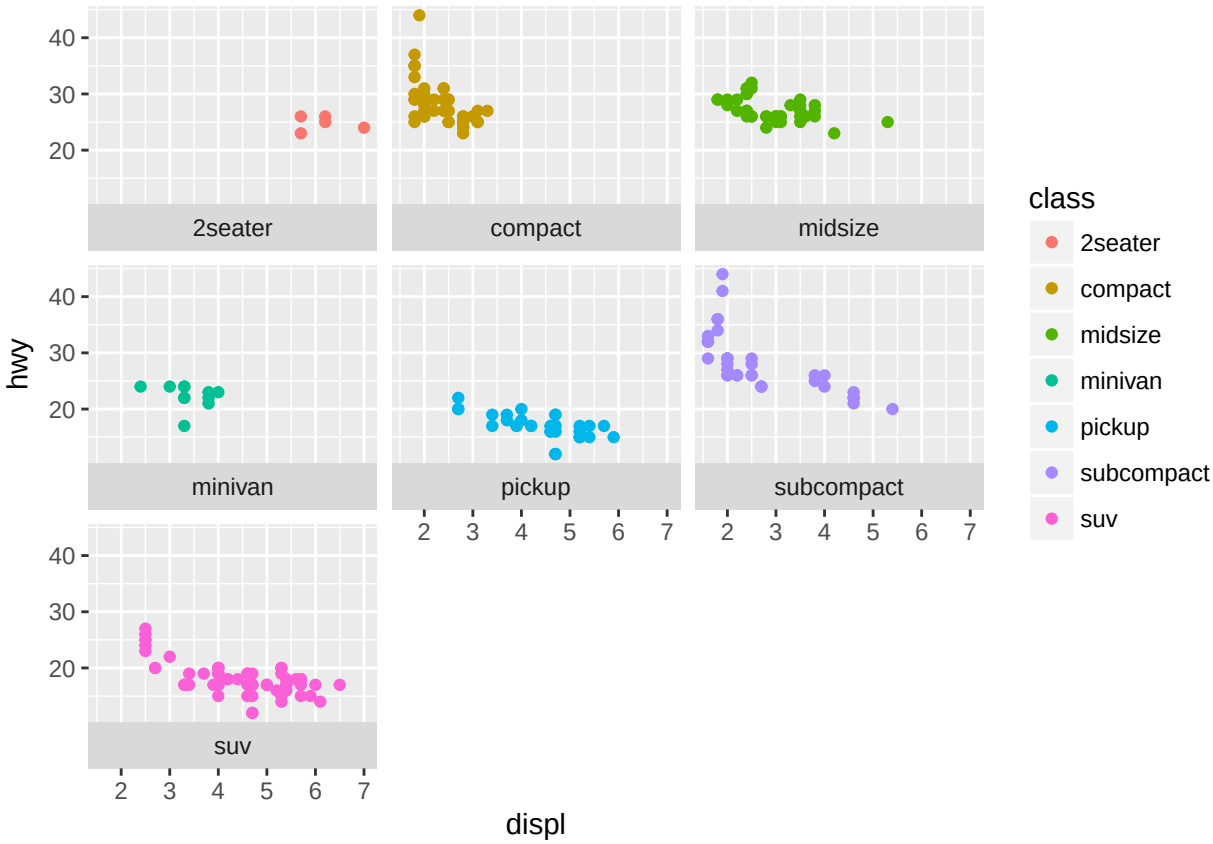
using 'as.table'

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



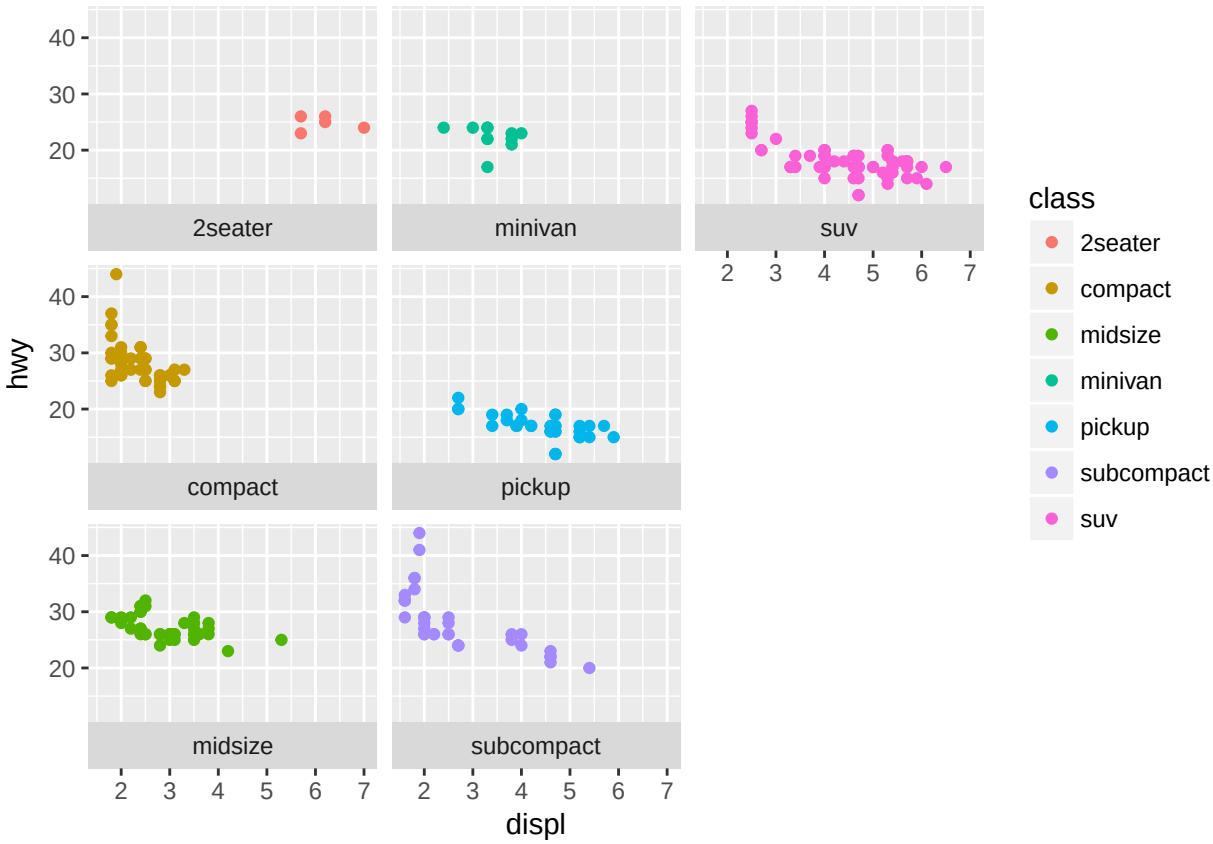
using 'strip.position'

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



using 'strip.position' and 'dir'

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



Q 3.6.1

What geom would you use to draw a line chart? A boxplot? A histogram? An area chart?

Answer:

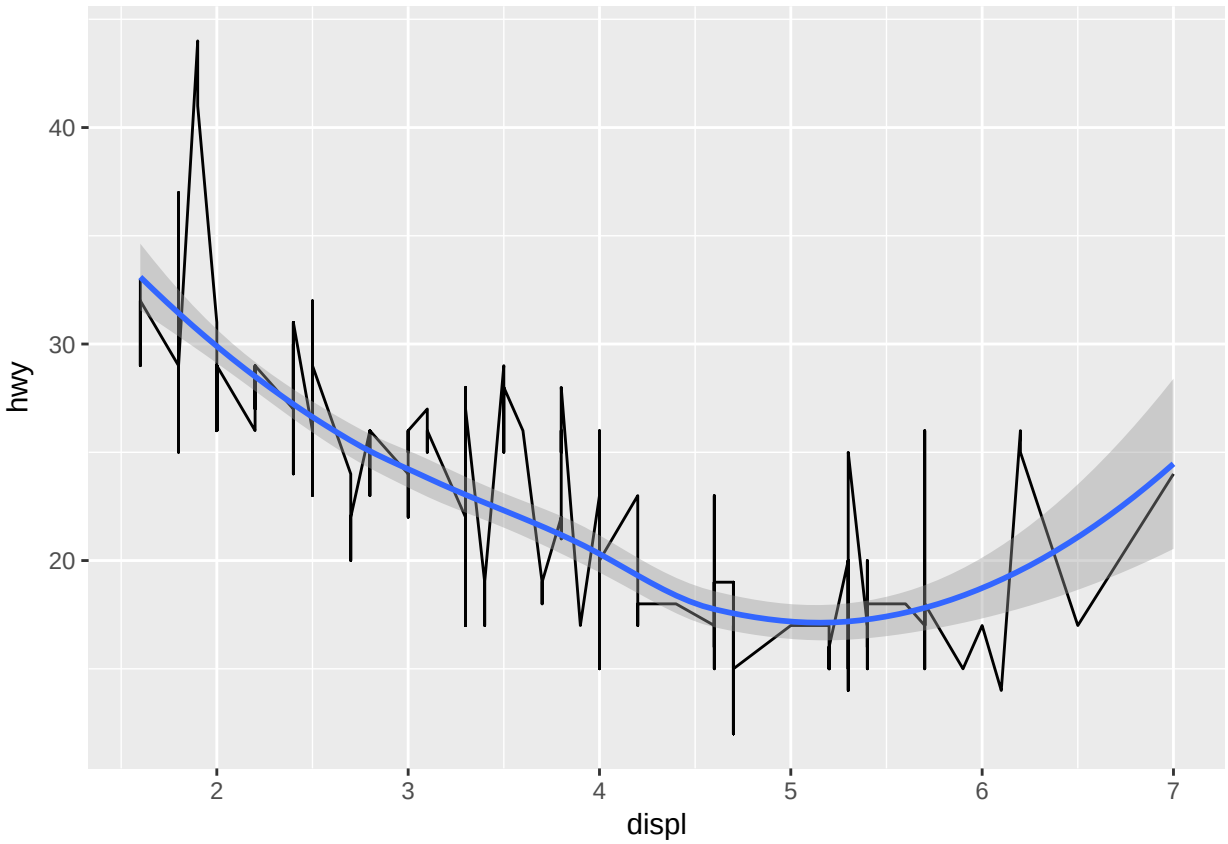
Line charts can use geom_line, or geom_smoth. Boxplots use geom_boxplot Historgrams use geom_histogram Area chargs use geom_ribbon or geom_area.

Because i don't have a good sample data set to use for geom_area, i am not plotting it, but showing plots for the other graphs.

First, we plot displacement vs highway miles, and draw line graphs for them.

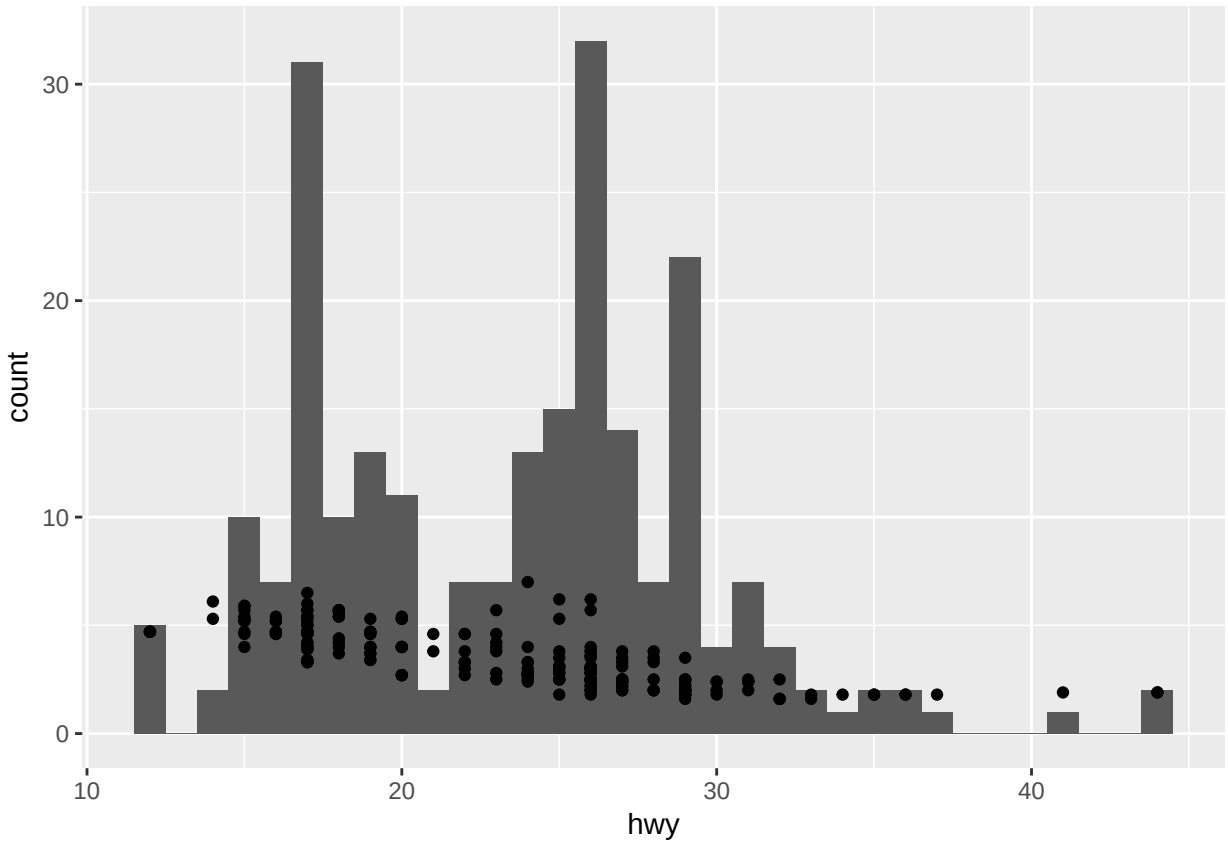
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_line() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```



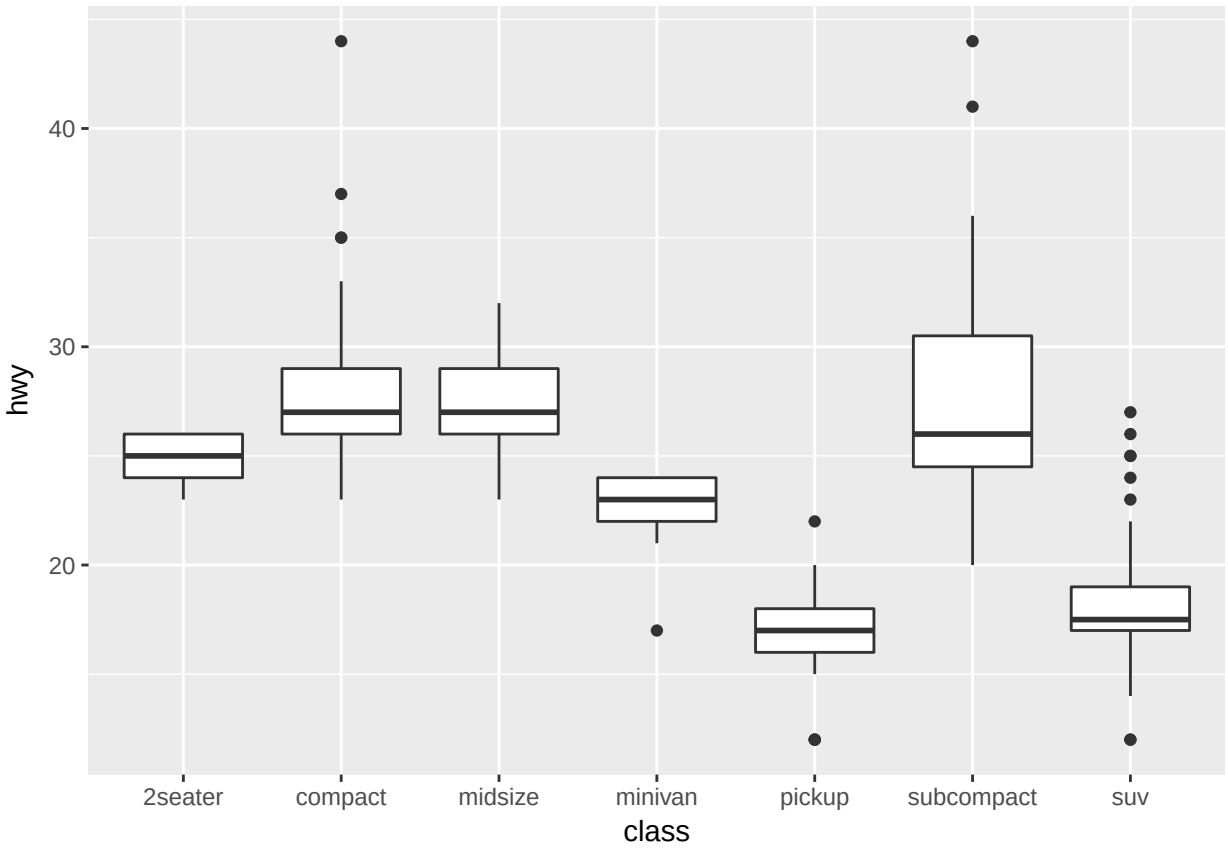
Then, we count the number of cars with particular highway miles, and plot a histogram.

```
ggplot(data = mpg, mapping = aes(x = hwy)) + geom_histogram(binwidth = 1) + geom_point(mapping = aes(y=
```



And finally, we plot a boxplot of the above histogram

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



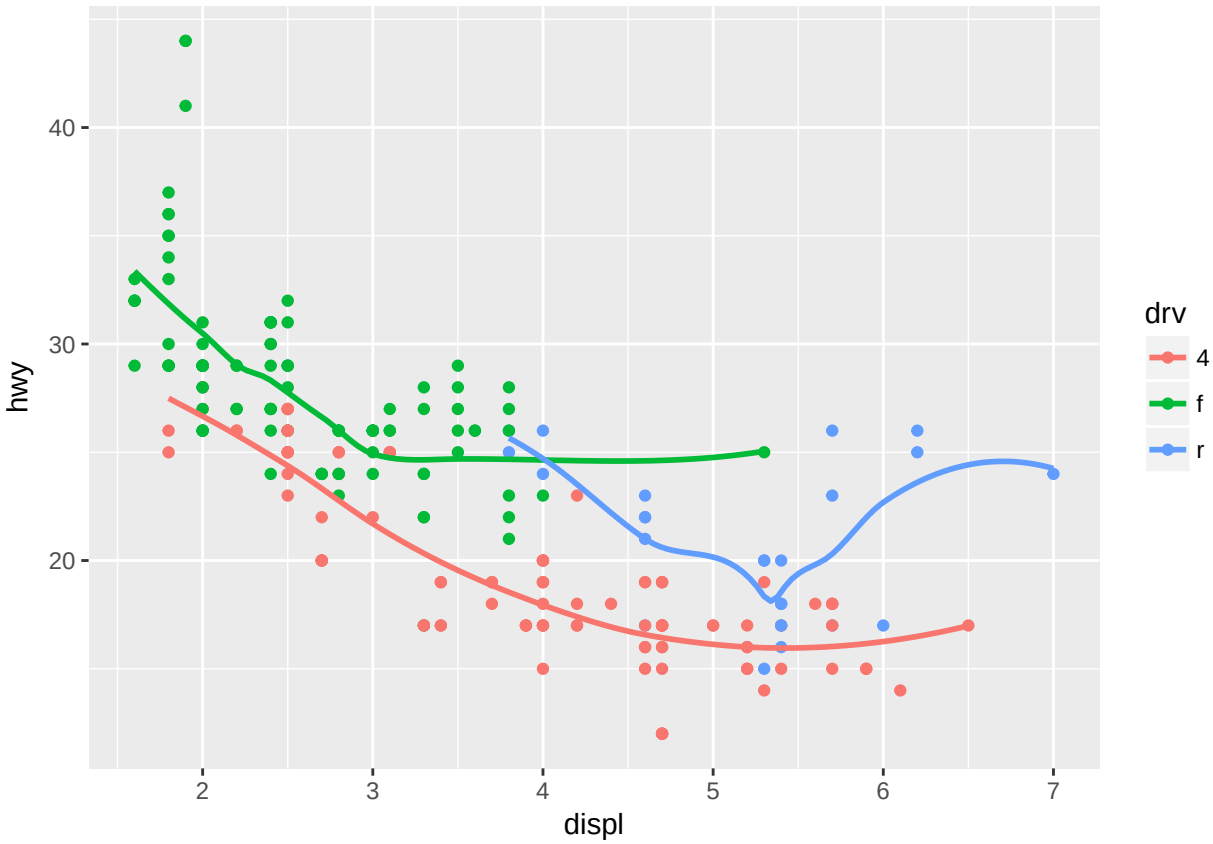
Q 3.6.2

Run this code in your head and predict what the output will look like. Then, run the code in R and check your predictions.

ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) + geom_point() + geom_smooth(se = FALSE)

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

```
## `geom_smooth()` using method = 'loess'
```



Q 3.6.3

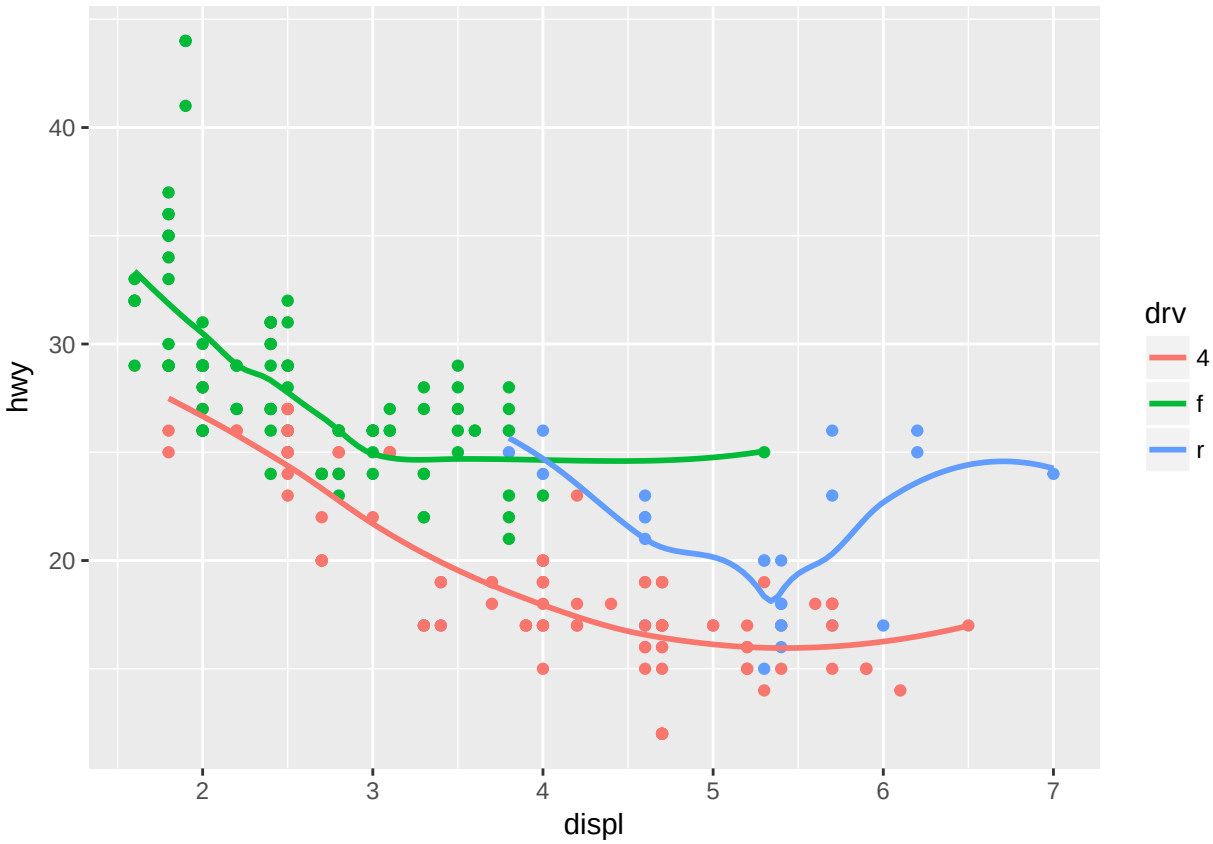
What does show.legend = FALSE do? What happens if you remove it? Why do you think I used it earlier in the chapter?

Answer:

It does not plot the legend for the particular layer. Lets plot exercise 3.6.2 without legends for the points and see what happens.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) +
  geom_point(show.legend = FALSE) +
  geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```



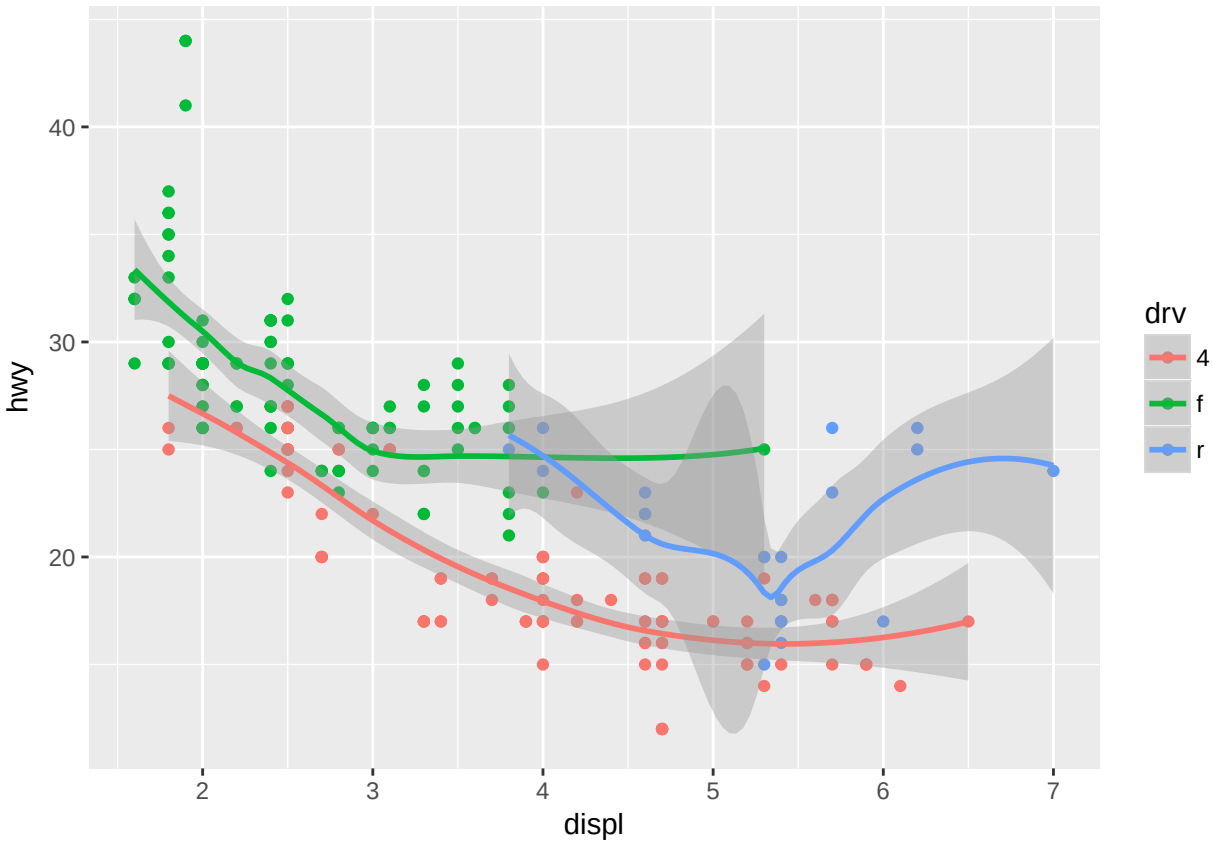
Q 3.6.4

What does the se argument to geom_smooth() do?

Answer: 'se' argument controls printing of the confidence interval. As so

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

```
## `geom_smooth()` using method = 'loess'
```



Q 3.6.5 Will these two graphs look different? Why/why not?

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

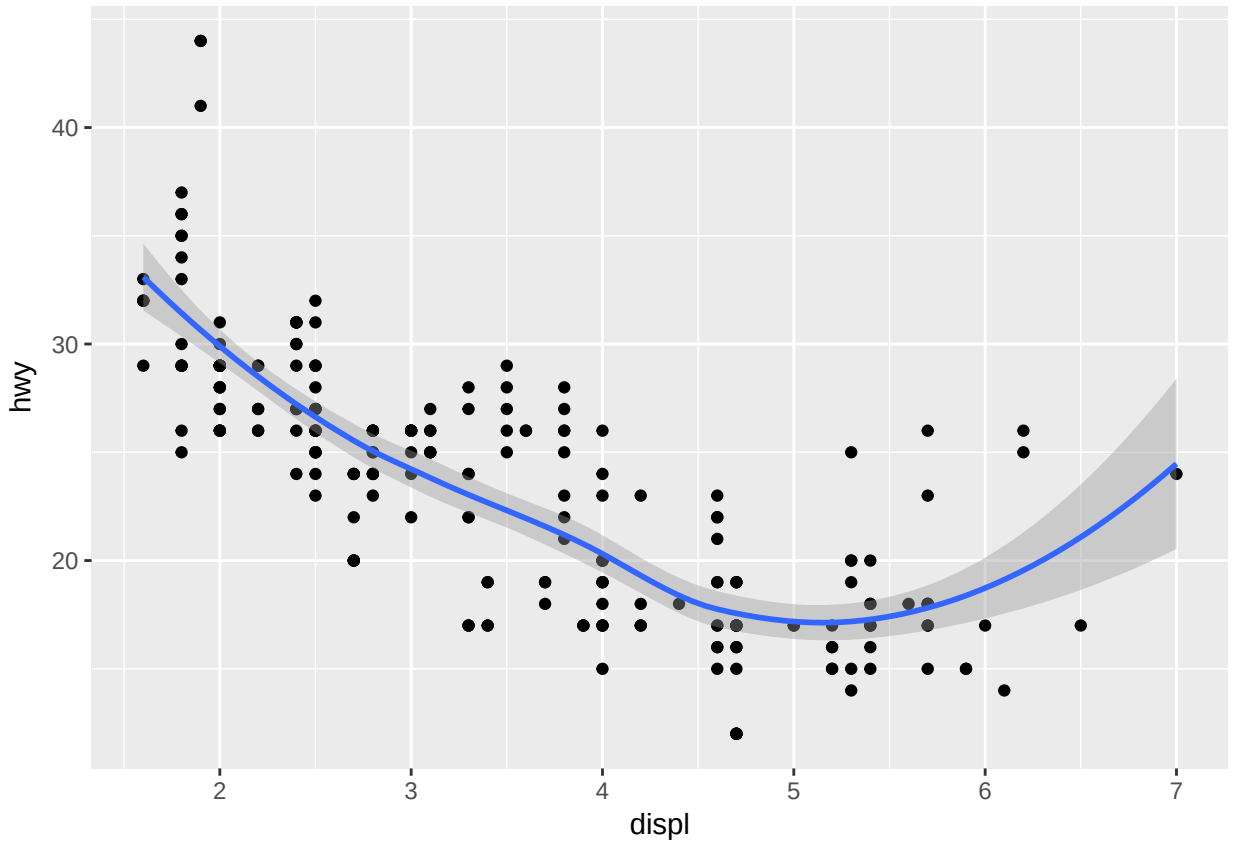
ggplot() + geom_point(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_smooth(data = mpg, mapping = aes(x = displ, y = hwy))

Answer:

Both graphs should look identical. Lets plot them and we see they are indeed identical.

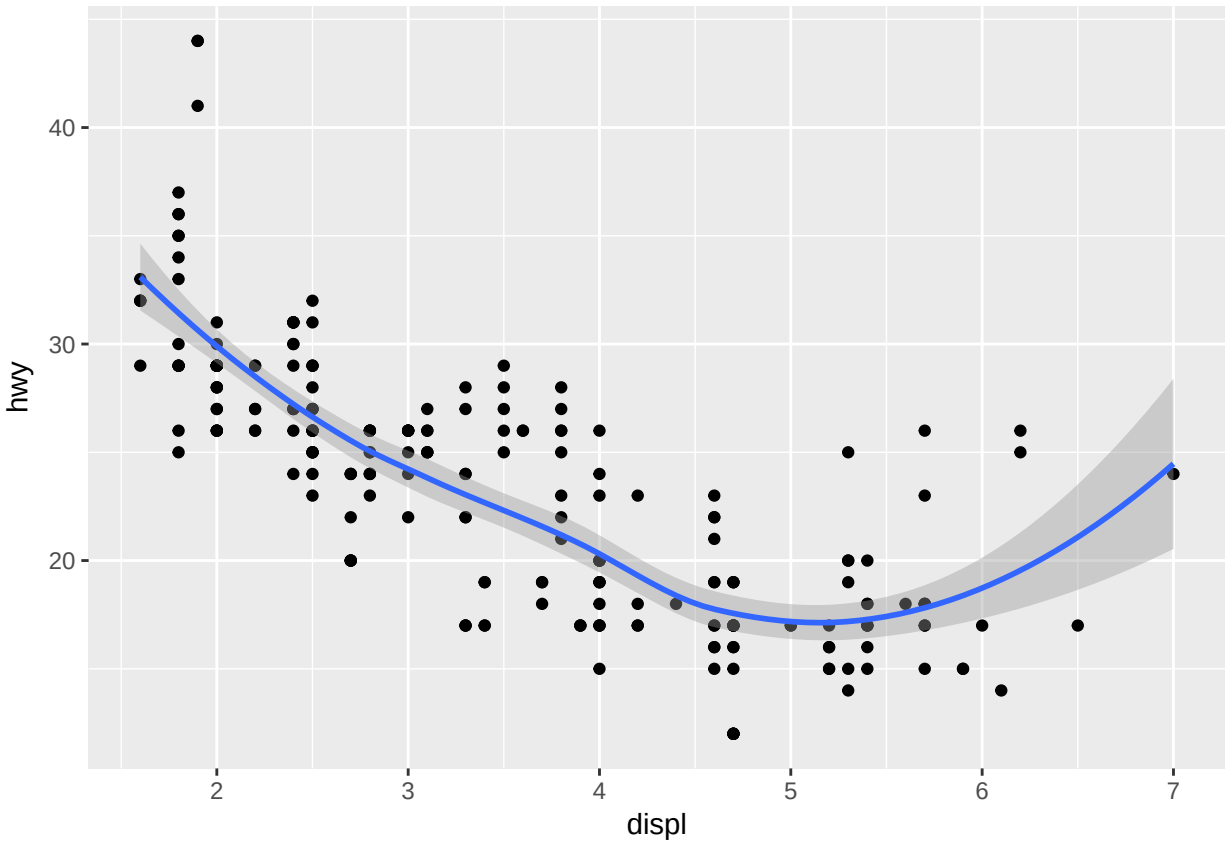
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```




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

```
## `geom_smooth()` using method = 'loess'
```



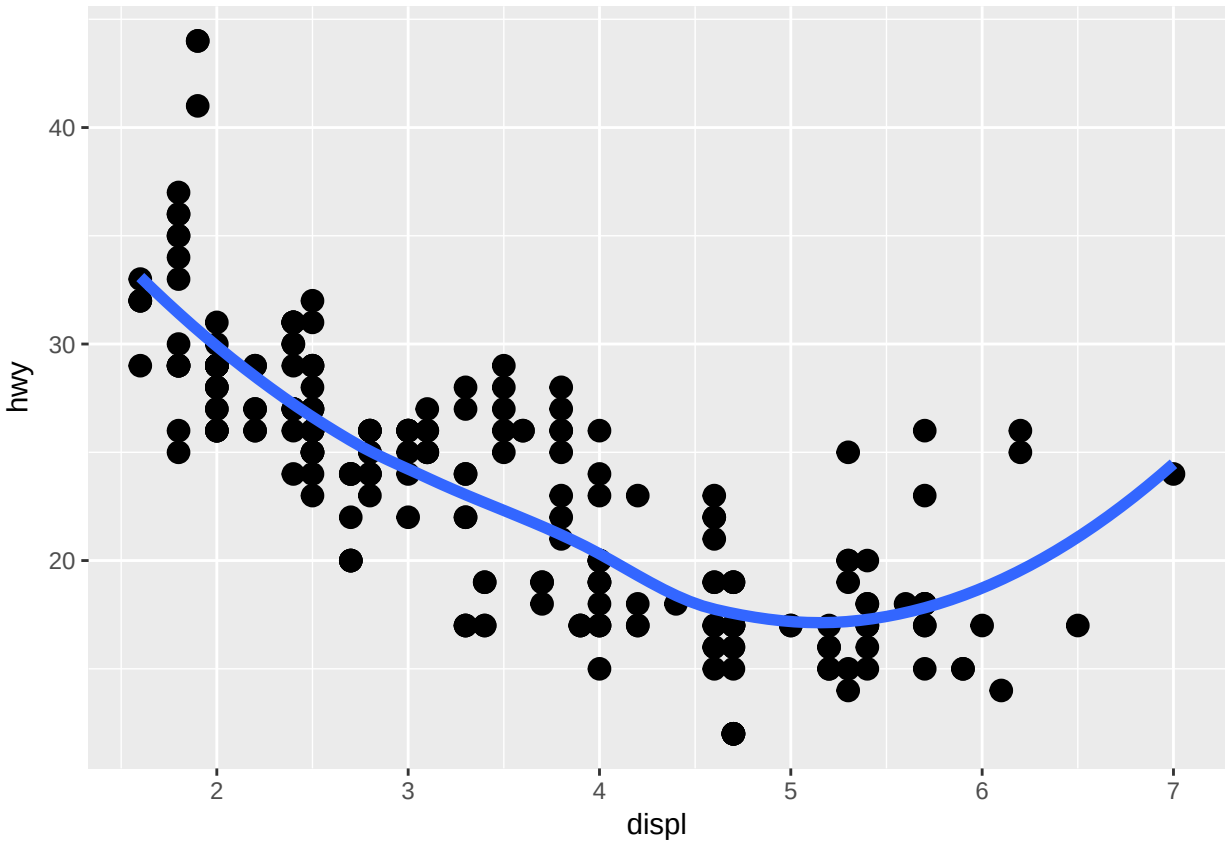
Q 3.6.6 Recreate the R code necessary to generate the following graphs.

Answer:

Graph 1:

```
ggplot(data = mpg, mapping = aes(x=displ, y=hwy)) + geom_point(stroke=2) + geom_smooth(se = FALSE, size=
```

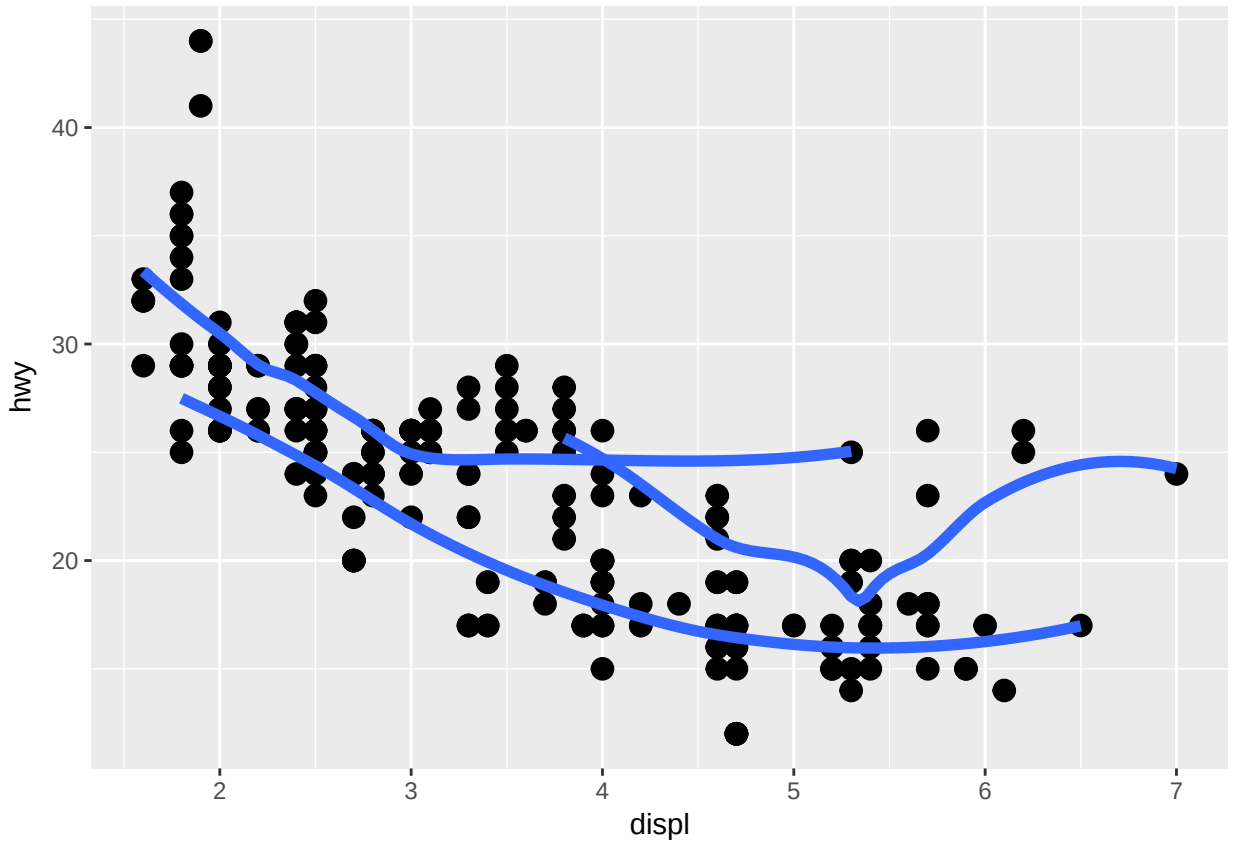
```
## `geom_smooth()` using method = 'loess'
```



Graph 2:

```
ggplot(data = mpg, mapping = aes(x=displ, y=hwy)) + geom_point(stroke=2) + geom_smooth(mapping = aes(gr
```

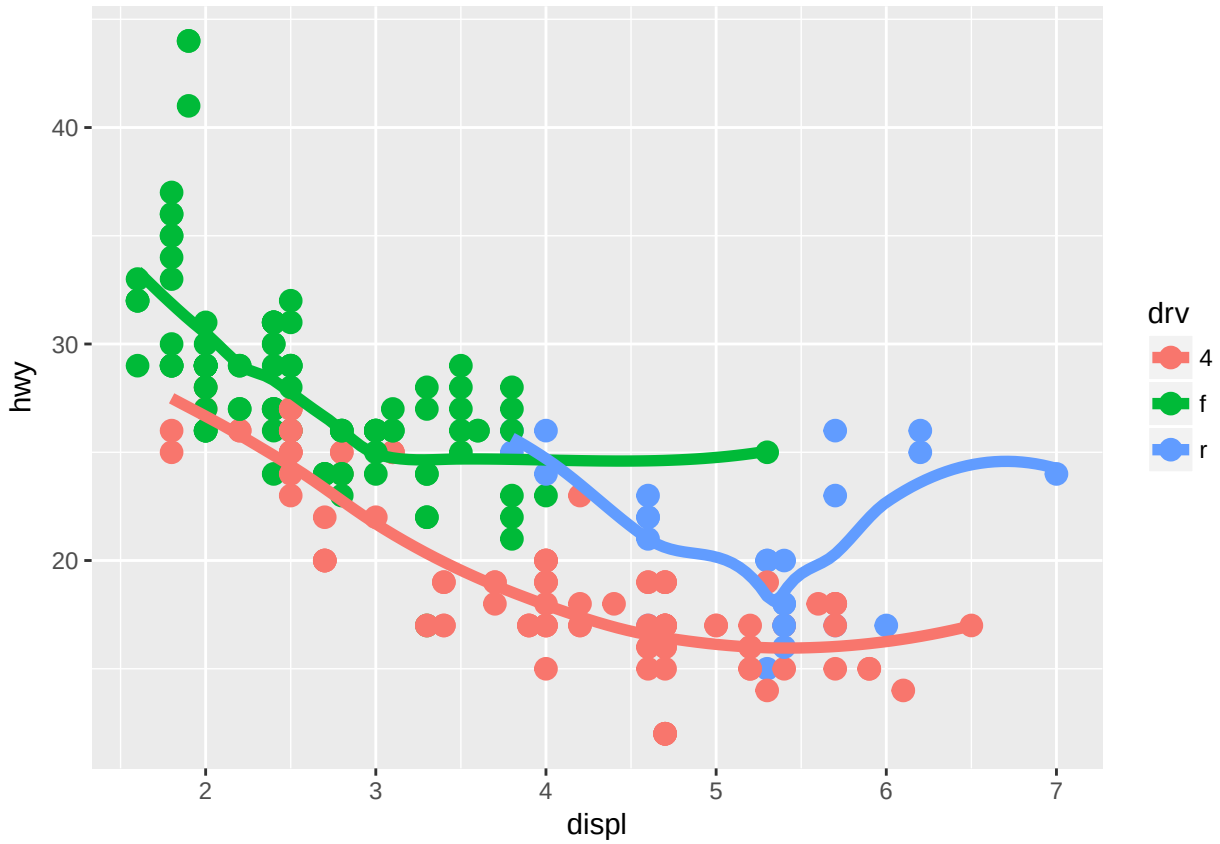
```
## `geom_smooth()` using method = 'loess'
```



Graph 3:

```
ggplot(data = mpg, mapping = aes(x=displ, y=hwy,color = drv)) + geom_point( stroke = 2) + geom_smooth(s
```

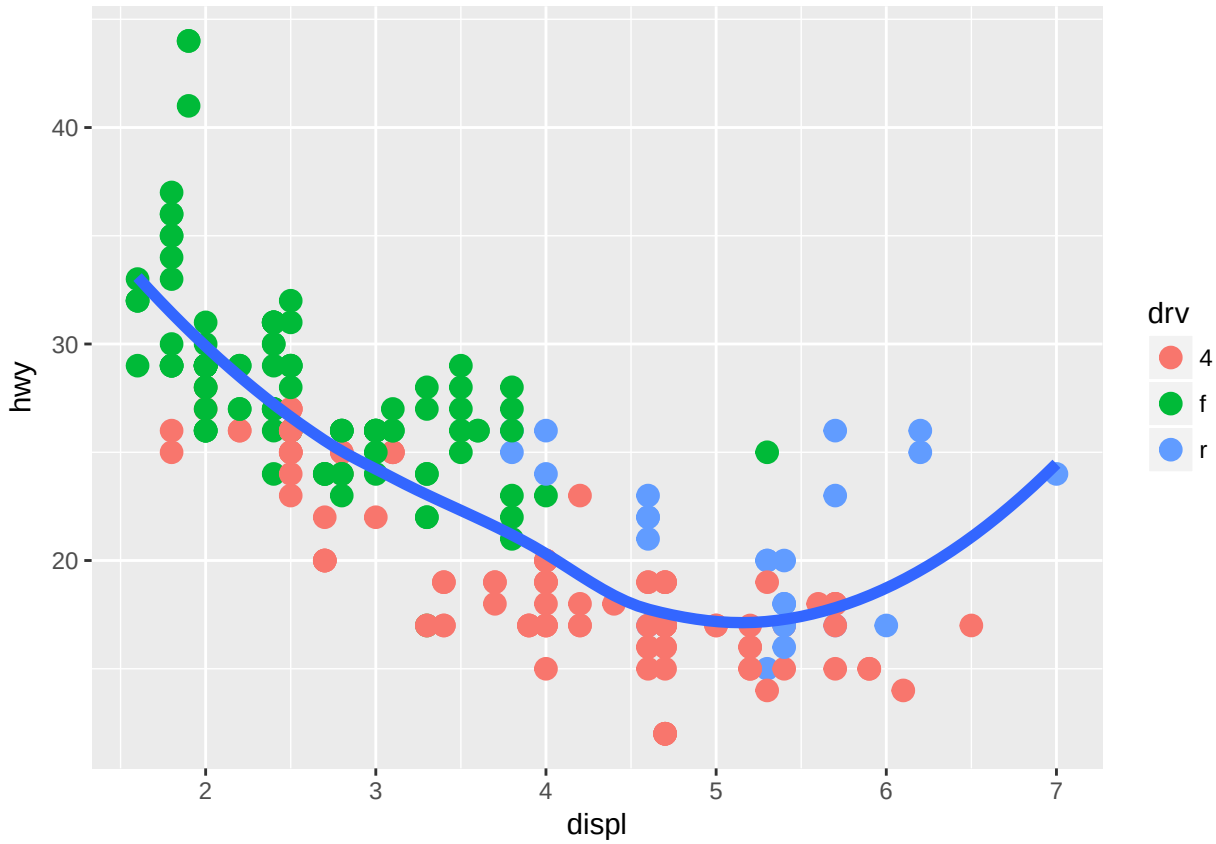
```
## `geom_smooth()` using method = 'loess'
```



Graph 4:

```
ggplot(data = mpg, mapping = aes(x=displ, y=hwy)) + geom_point( mapping = aes(color = drv), stroke = 2)
```

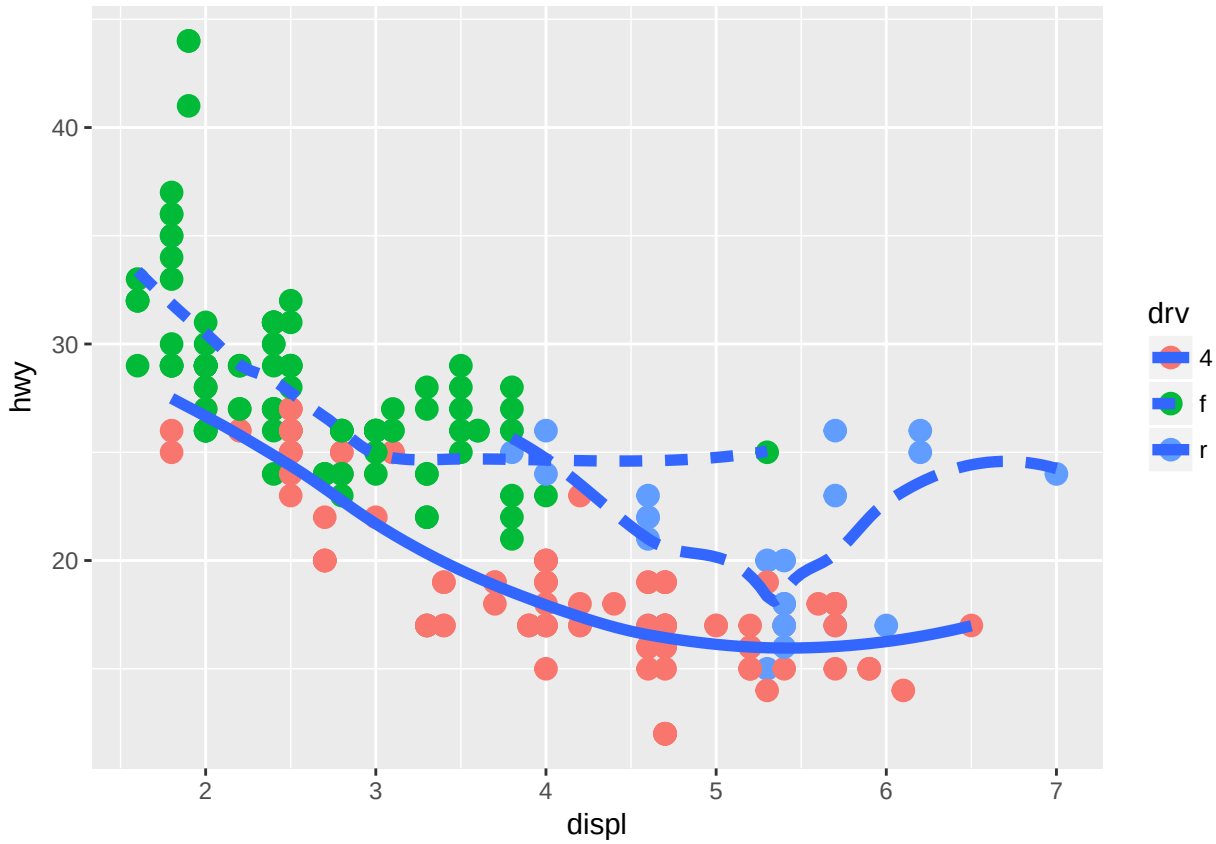
```
## `geom_smooth()` using method = 'loess'
```



Graph 5:

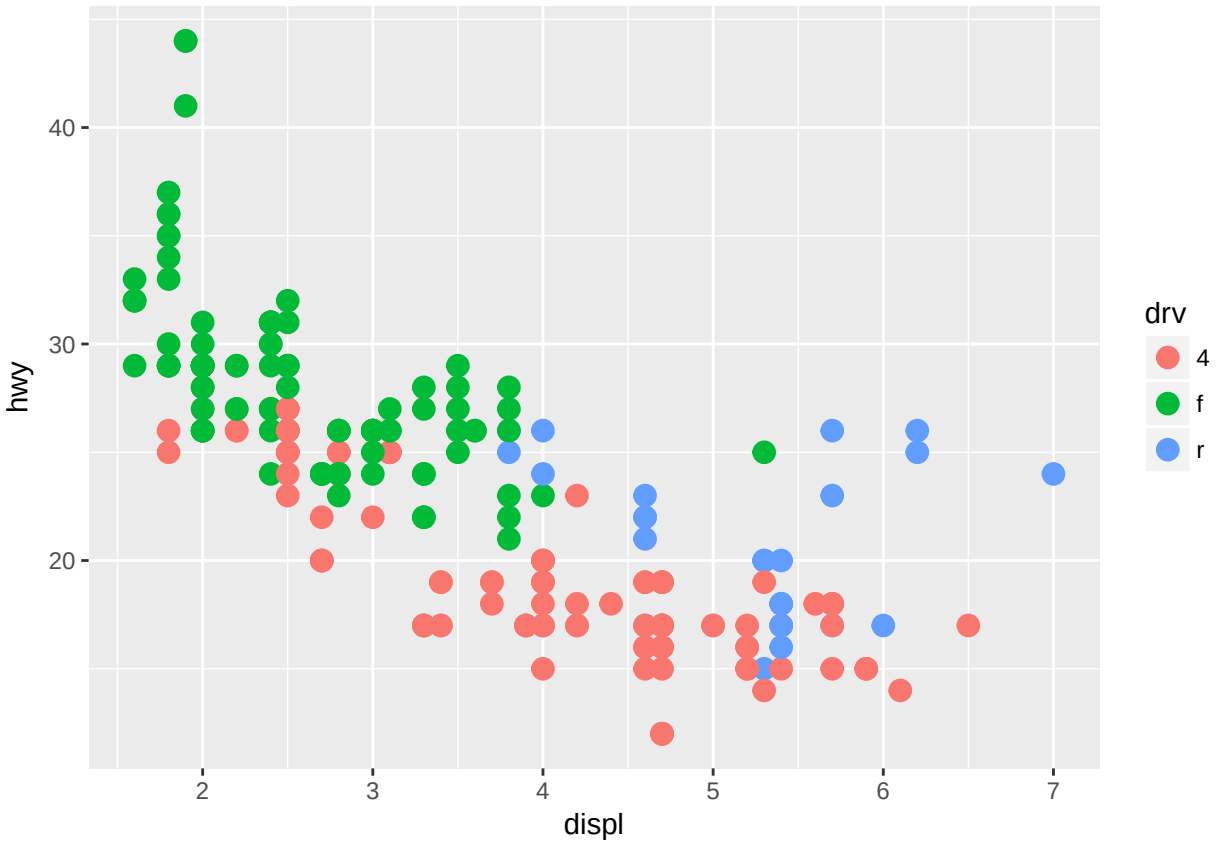
```
ggplot(data = mpg, mapping = aes(x=displ, y=hwy)) + geom_point( mapping = aes(color = drv), stroke = 2)
```

```
## `geom_smooth()` using method = 'loess'
```



Graph 6:

```
ggplot(data = mpg, mapping = aes(x=displ, y=hwy)) + geom_point( mapping = aes(color = drv), stroke = 2)
```



Q 3.7.1.1

What is the default geom associated with stat_summary()? How could you rewrite the previous plot to use that geom function instead of the stat function?

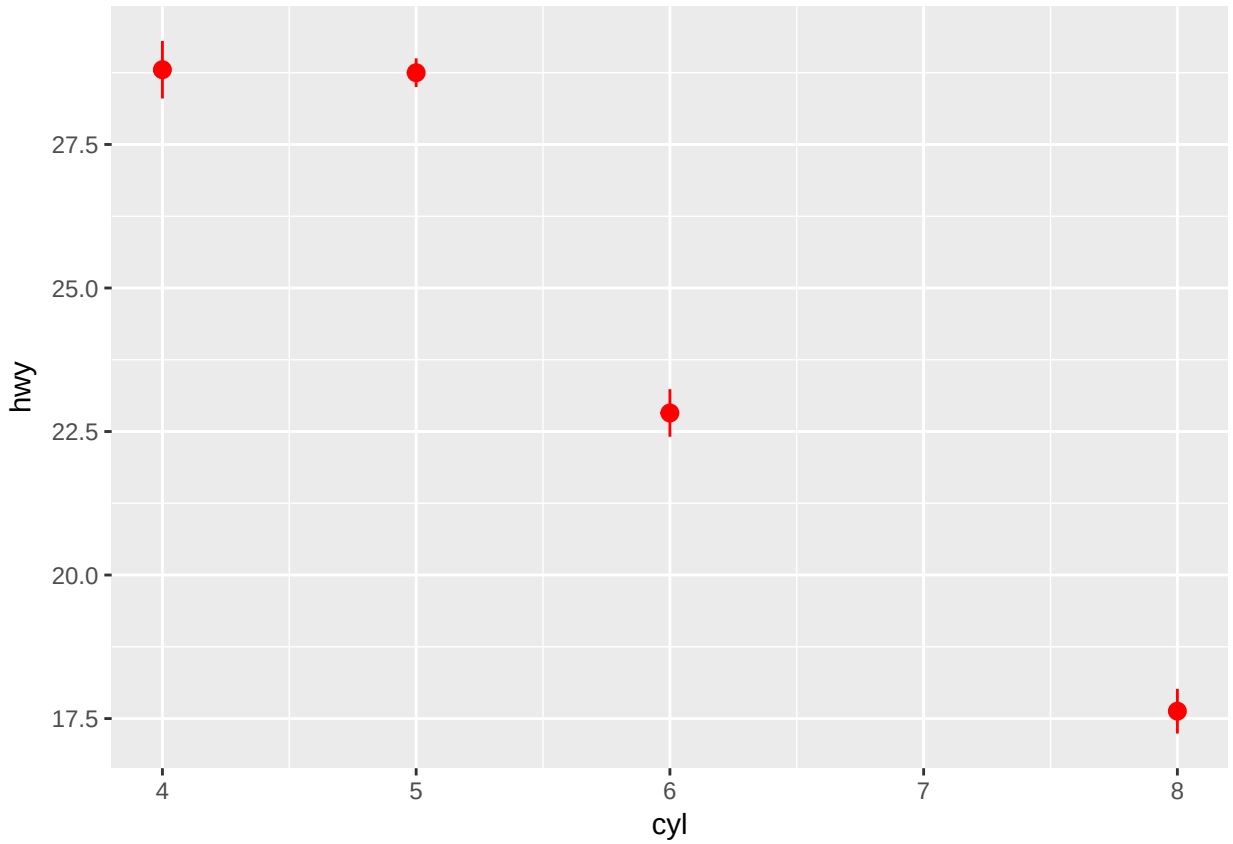
Answer:

TBD:

Documentation says it uses geom_pointrage, but it is unclear how this fits in.. trying to use geom_pointrange does not seem to work.

```
ggplot(data=mpg, aes(cyl, hwy)) +  stat_summary(color='red')
```

```
## No summary function supplied, defaulting to `mean_se()
```



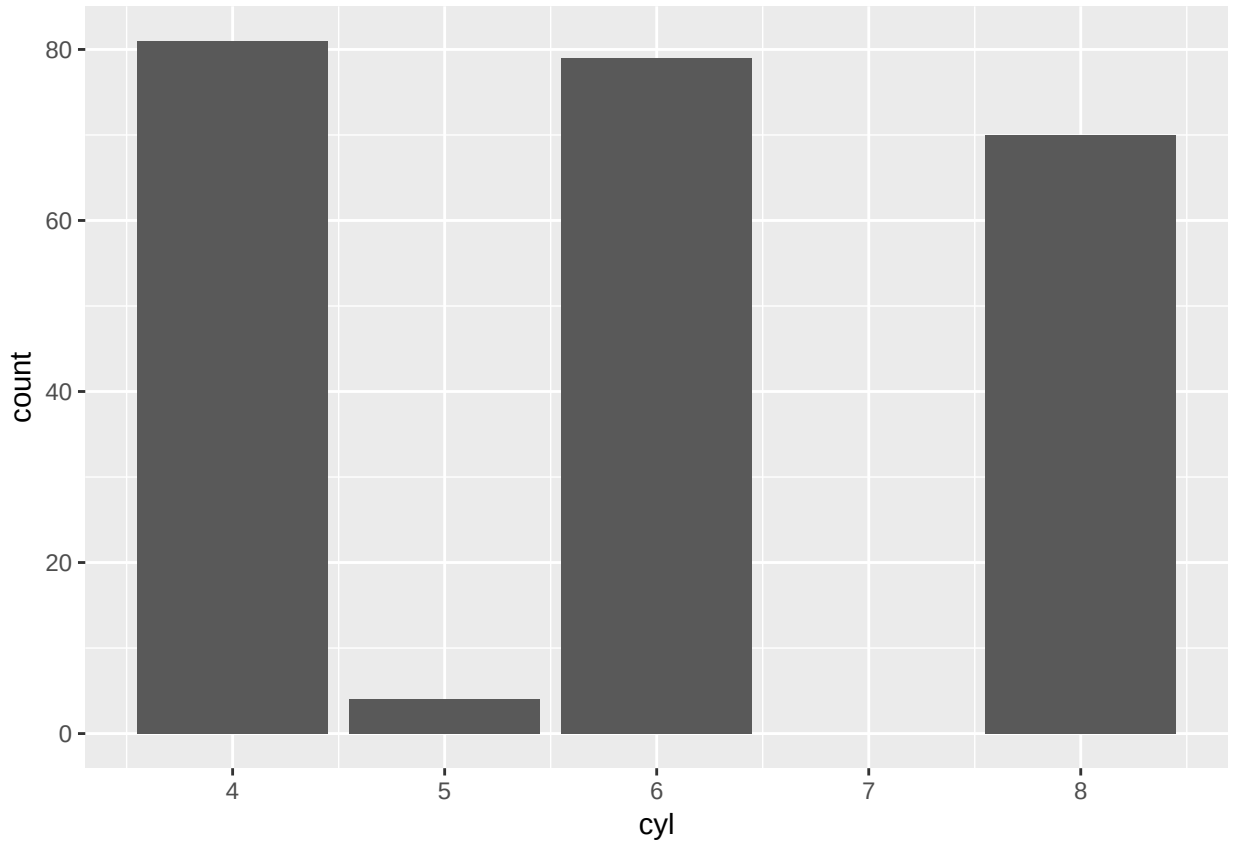
Q 3.7.1.2 What does geom_col() do? How is it different to geom_bar()?

Answer:

geom_bar plots the count of an x-axis variable. In this case, the count of 'cyl', but if you wanted to plot another variable from the dataset, then you need to use geom_col.

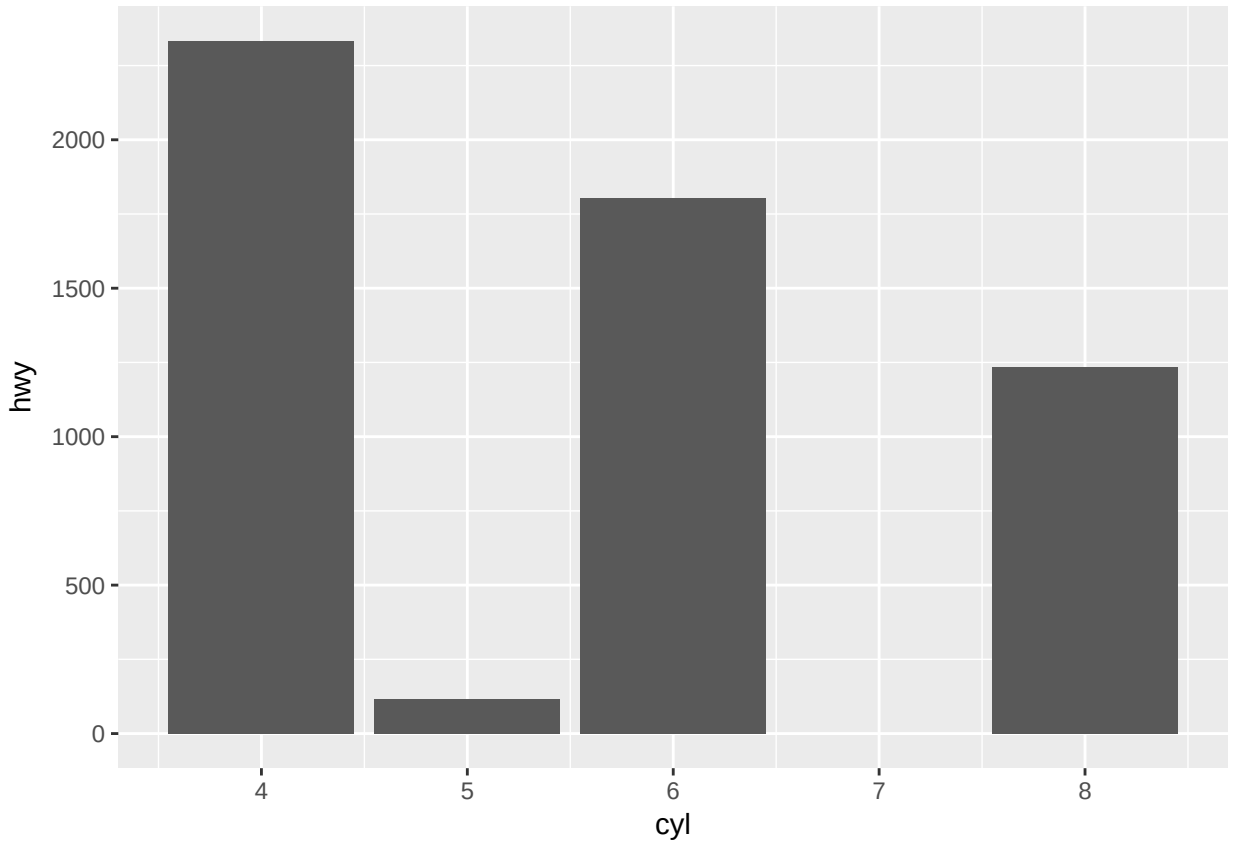
e.g Using geom_bar()

```
ggplot(data = mpg, mapping = aes(cyl)) + geom_bar()
```



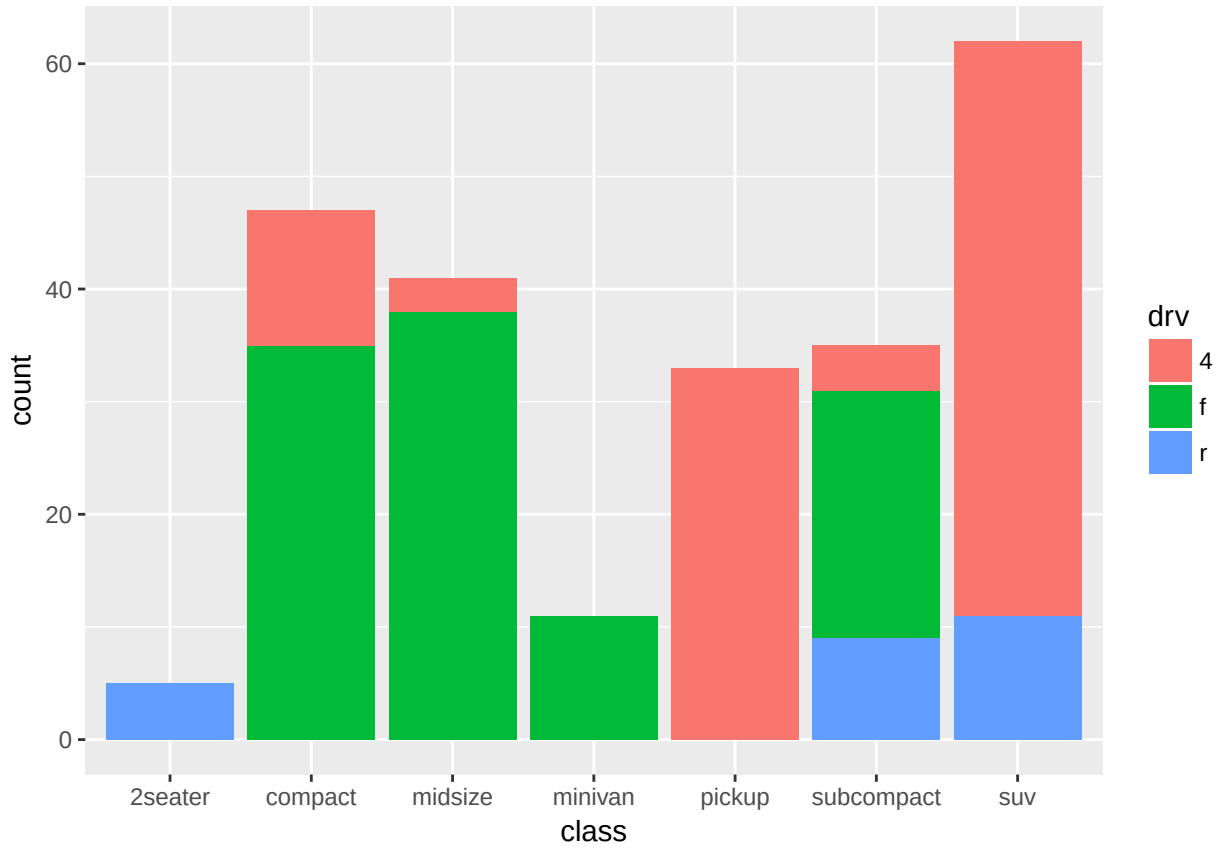
same example, using geom_col, but plotting 'cylinders' vs 'total number of hwy miles'. It seems like geom_col is adding all the values of 'hwy', but this was never stated anywhere.

```
ggplot(data = mpg, mapping = aes(cyl, hwy)) + geom_col()
```



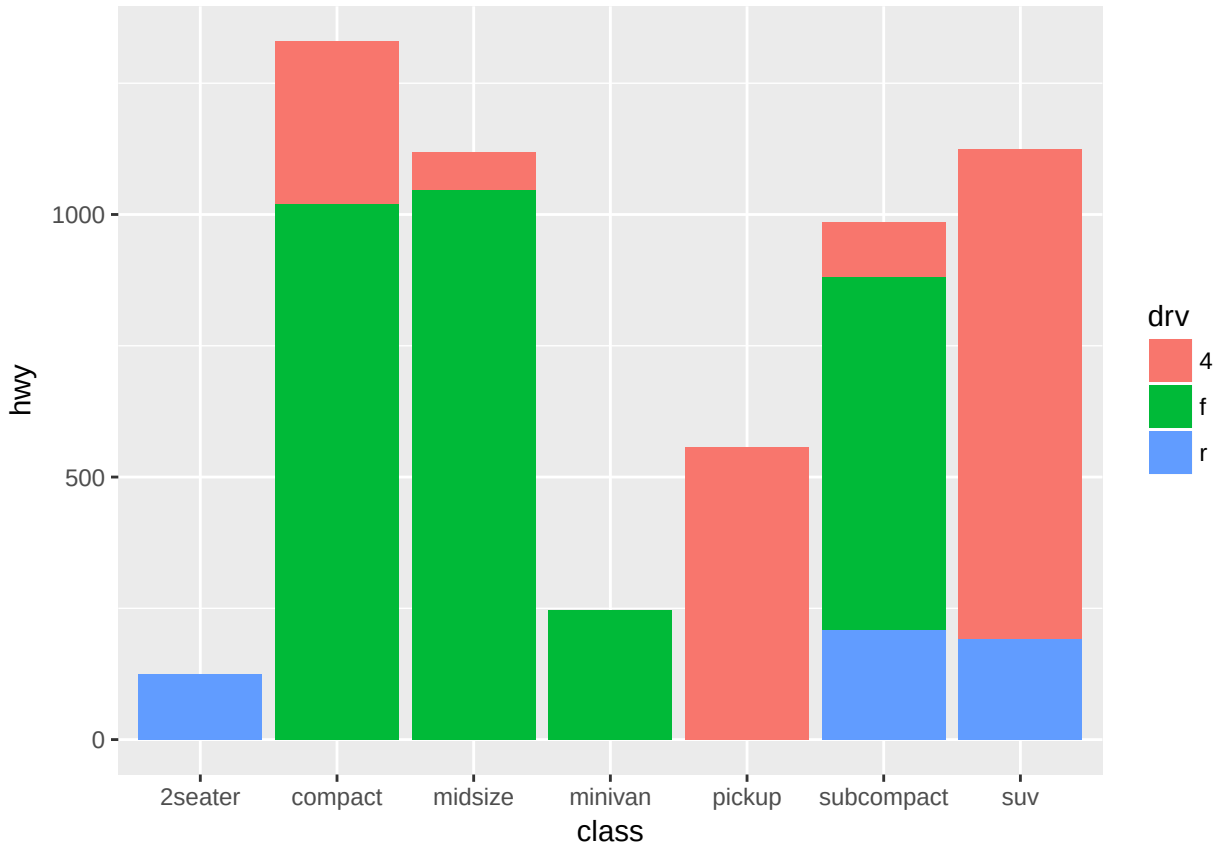
Using fill for fun..

```
ggplot(data=mpg, mapping = aes(class)) + geom_bar(aes(fill = drv))
```



and combining with geom_col.. interesting results

```
ggplot(data=mpg, mapping = aes(class,hwy)) + geom_col( mapping  =aes (fill = drv))
```



Q 3.7.1.3 Most geoms and stats come in pairs that are almost always used in concert. Read through the documentation and make a list of all the pairs. What do they have in common?

Answer: Geom Stat geom_freqpoly stat_bin geom_histogram stat_bin geom_bar stat_count geom_col stat_count geom_bin2d stat_bin_2d geom_boxplot stat_boxplot geom_contour stat_contour geom_count stat_sum geom_density stat_density geom_density_2d stat_density_2d

Stopping at this point, since there seem to be many more such geoms

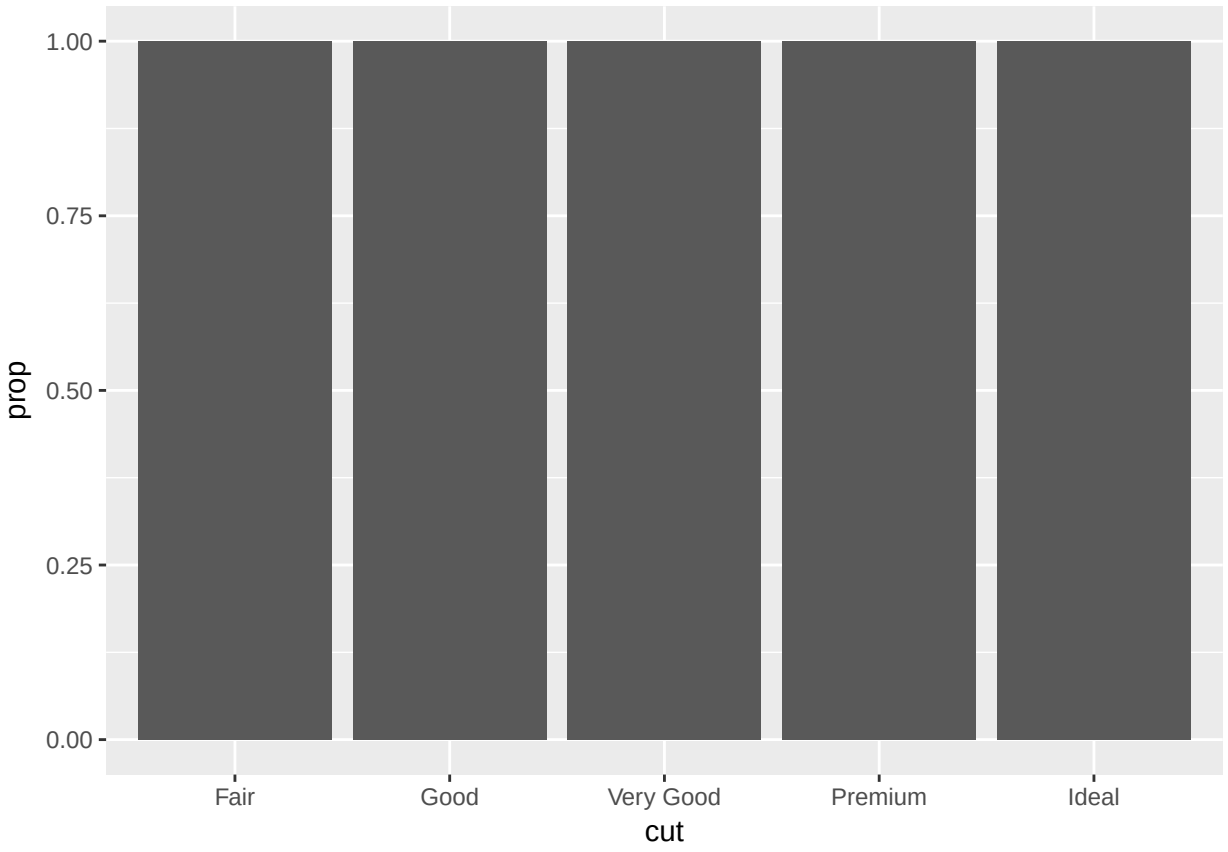
Q 3.7.1.4 What variables does stat_smooth() compute? What parameters control its behaviour?

Answer: stat_smooth calculates the mean at a particular point, and the standard error around that point. stat_smooth computes 4 variables, the predicted value (y), the upper and lower confidence interval around the mean, and the standard error.

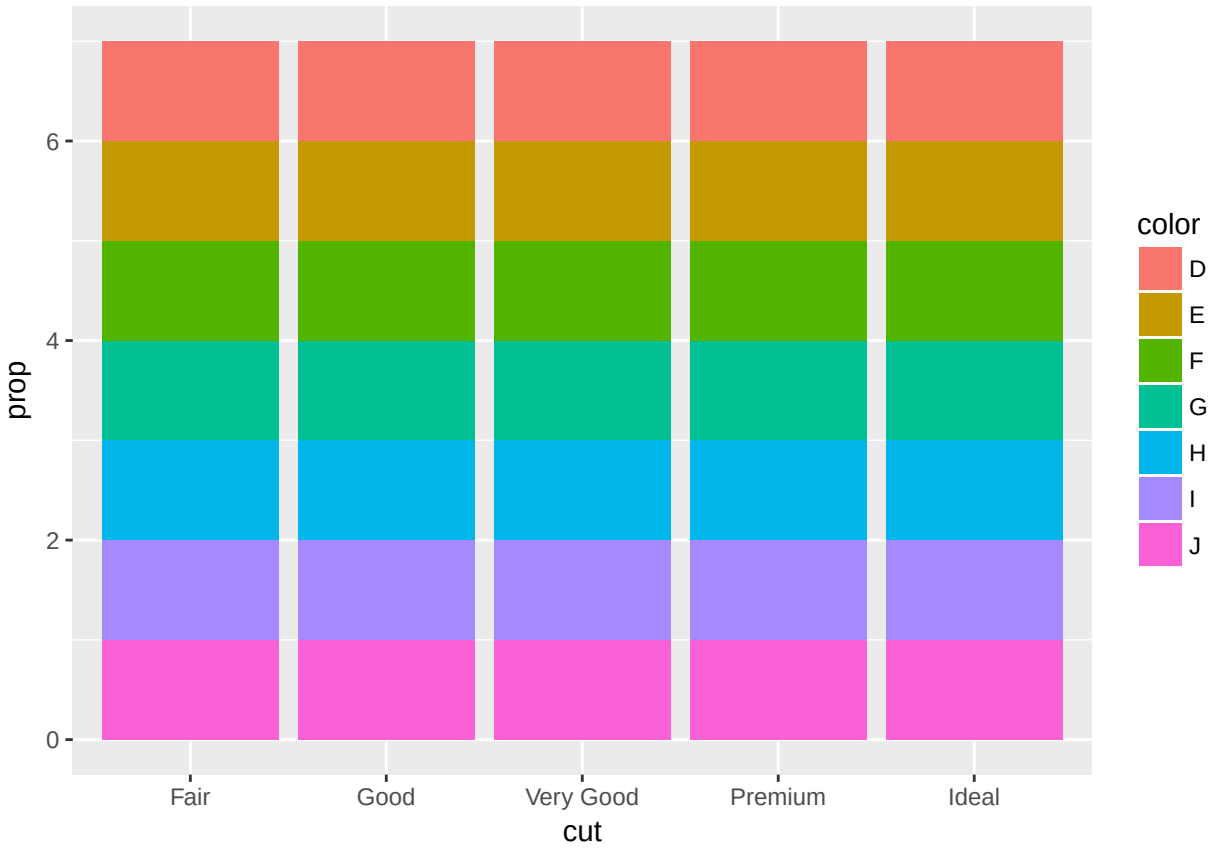
The parameters that control stat_smooth are all visible by ?stat_smooth. Particular interest are the parameters - method, se, n, span, fullrange, level and method.args

Q 3.7.1.5 In our proportion bar chart, we need to set group = 1. Why? In other words what is the problem with these two graphs?

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = ..prop..))
```




```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = color, y = ..prop..))
```

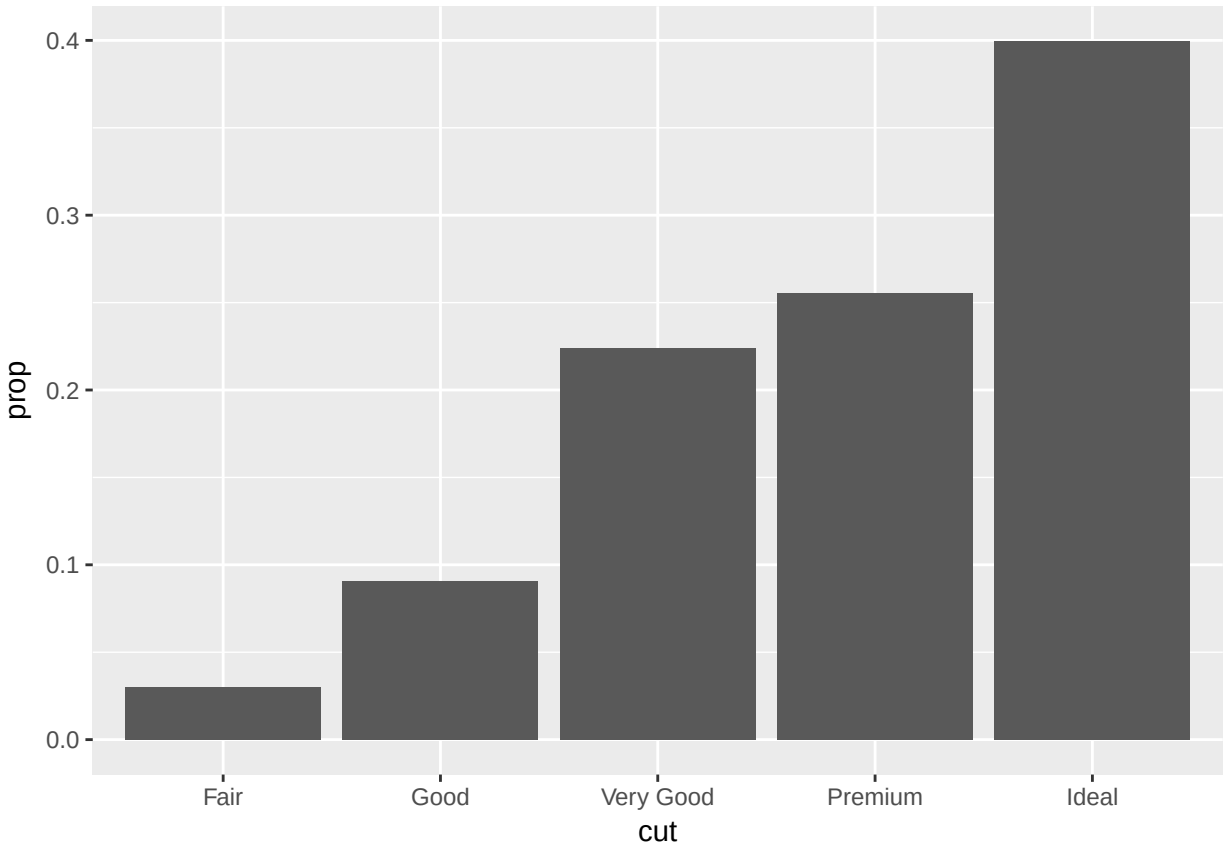


Answer:

The problem is that there is no variable to show propotion on, and so propotion of 100% is 100%, as shown in the graph in the question.

If the desierd result was to show propotion of the diamonds by the 'cut' - e.g x% diamonds are 'fair', while y% are 'good' the plot would need to be - which needs the group=1 variable. In fact, any number for the group seems acceptable (e.g group=2, group=3) all producing the same result, and it is not clear why the error is not flagged.

```
ggplot(data = diamonds) + geom_bar(mapping = aes(cut, y = ..prop.., group=1))
```



This treats all the values as one group, and calculates the proportion for them. Without the group variable, the propotions are being calculated for each individual variable.

Last Question

Look at the data graphics at the following link: What is a Data Scientist. Please briefly critique the designer's choices. What works? What doesn't work? What would you have done differently?

Answer

Use of PI chart, though not ideal, seems to look good (though does not communicate the data it needs) Bar chart for "Best source of New Data Science Talent" is nice "Biggest obstacle to data science adoption" - Numbers are good "Data scientists are significantly more likely to advanced degrees" - Bar charts are bad "Data Scientists have more varied backgrounds" - Trying to use a chart for eye candy. Hard to understand "Characteristics of Data Scientsits" - Nice simple difference between Big Data Science, and Normal Data Science "Data scients are likely to be involved accross the data lifecycle" - multiple bar charts is confusing.