

# COMPSCIX 415.2 Homework 2

*Santosh Kanutala*

*6/18/2018*

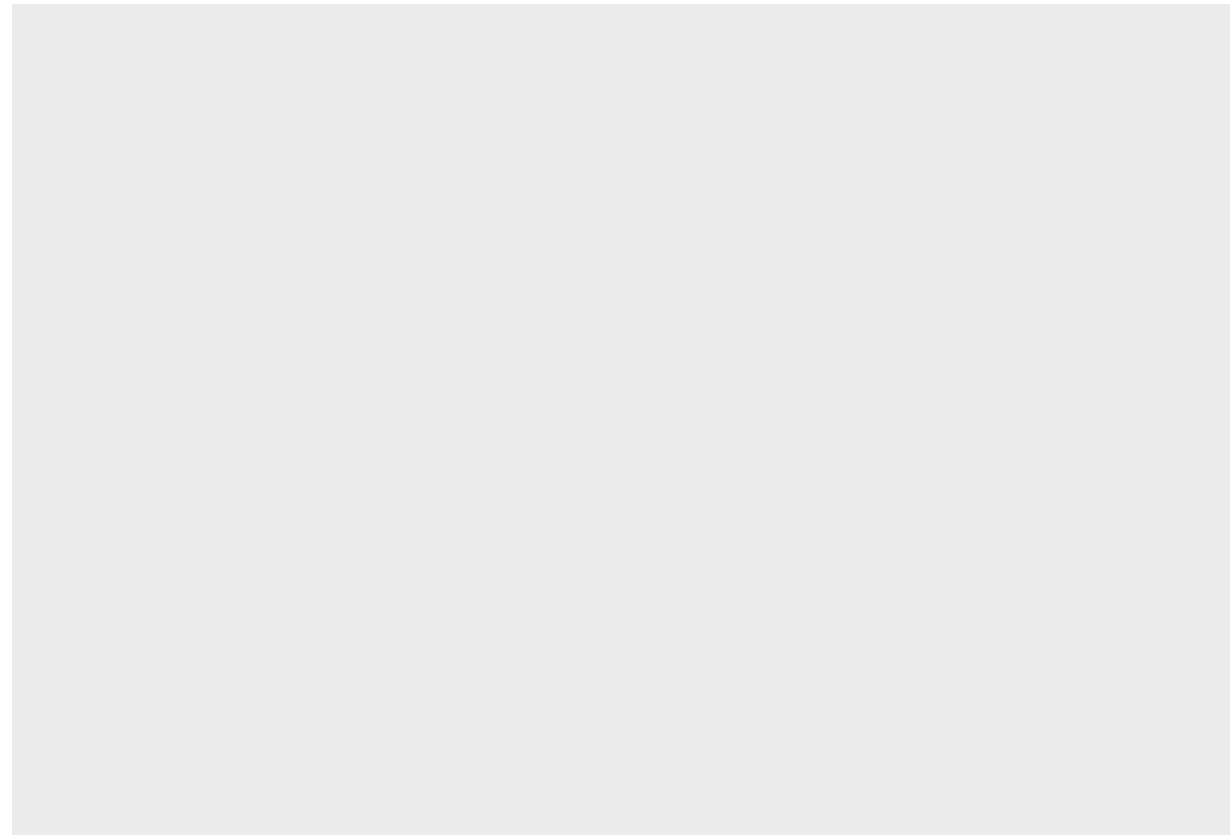
**\*\*My Github repository for my assignments can be found below URL: (<https://github.com/santumagic/compscix-415-2assignments.git>)\*\***

```
library(tidyverse)
library(mdsr)
```

## Section 3.2.4: all exercises

### Question 1:

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



### Answer:

From the above function, I see a blank graph, which is a coordinate system.

## Question 2:

```
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...
## $ 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..."
```

## Answer:

There are 234 rows (observations) and 11 columns (variables).

## Question 3:

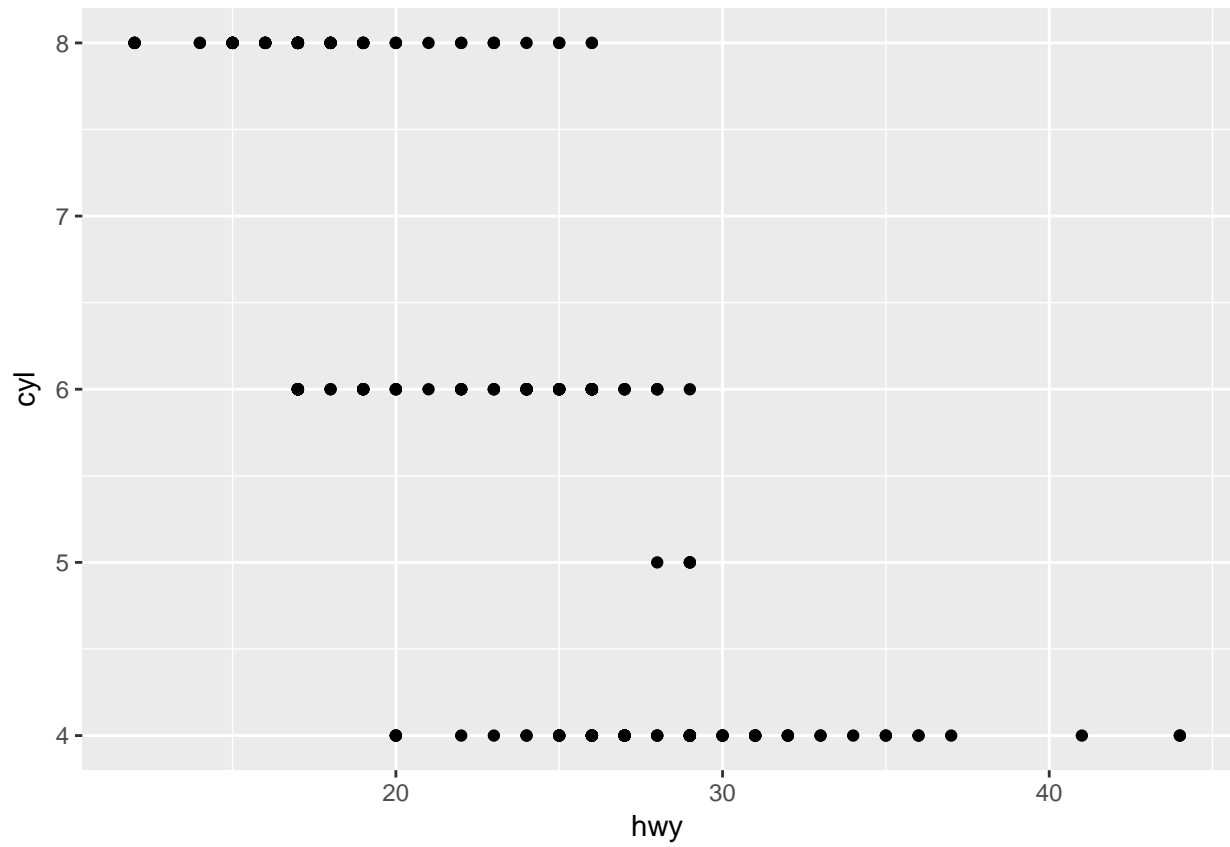
```
?mpg
```

## Answer:

drv variable describes the drive type of the vehicle like f = front wheel drive, r = rear wheel drive etc.

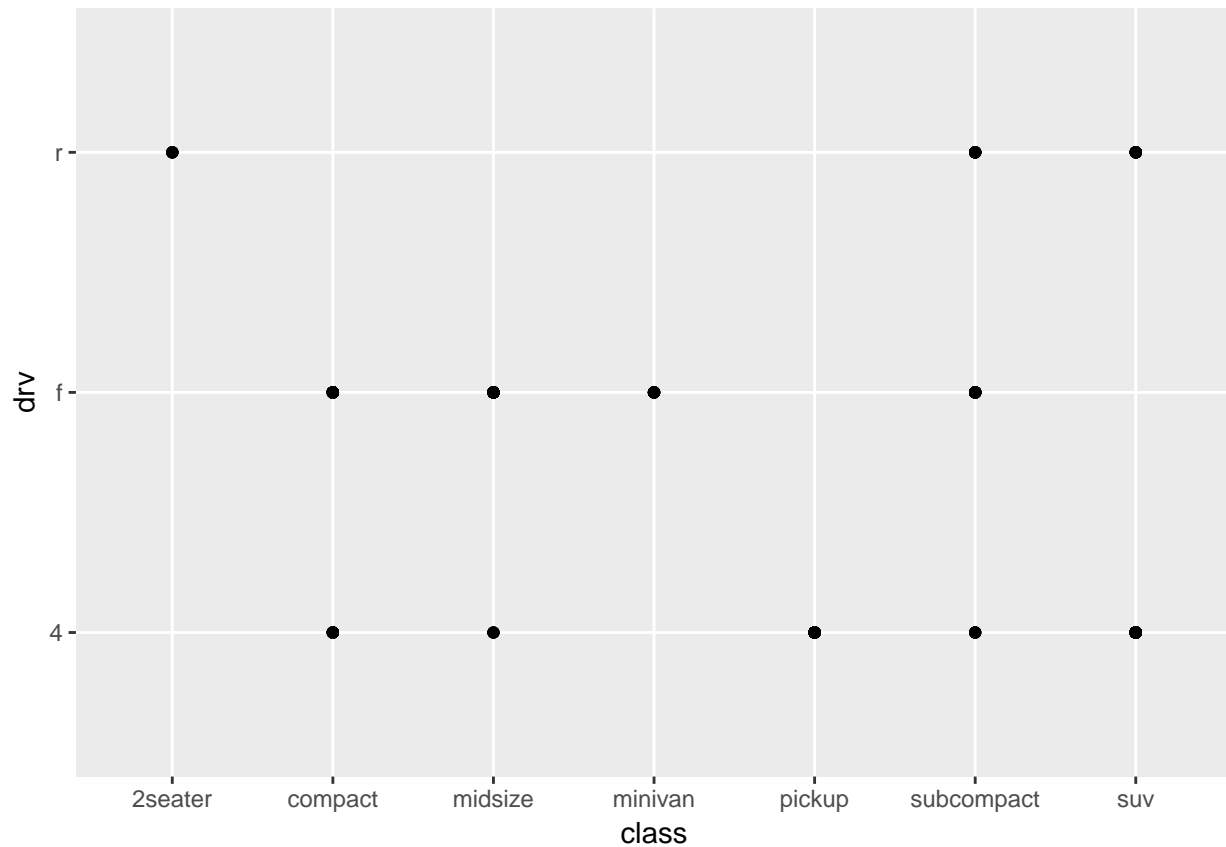
## Question 4:

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



Question 5:

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



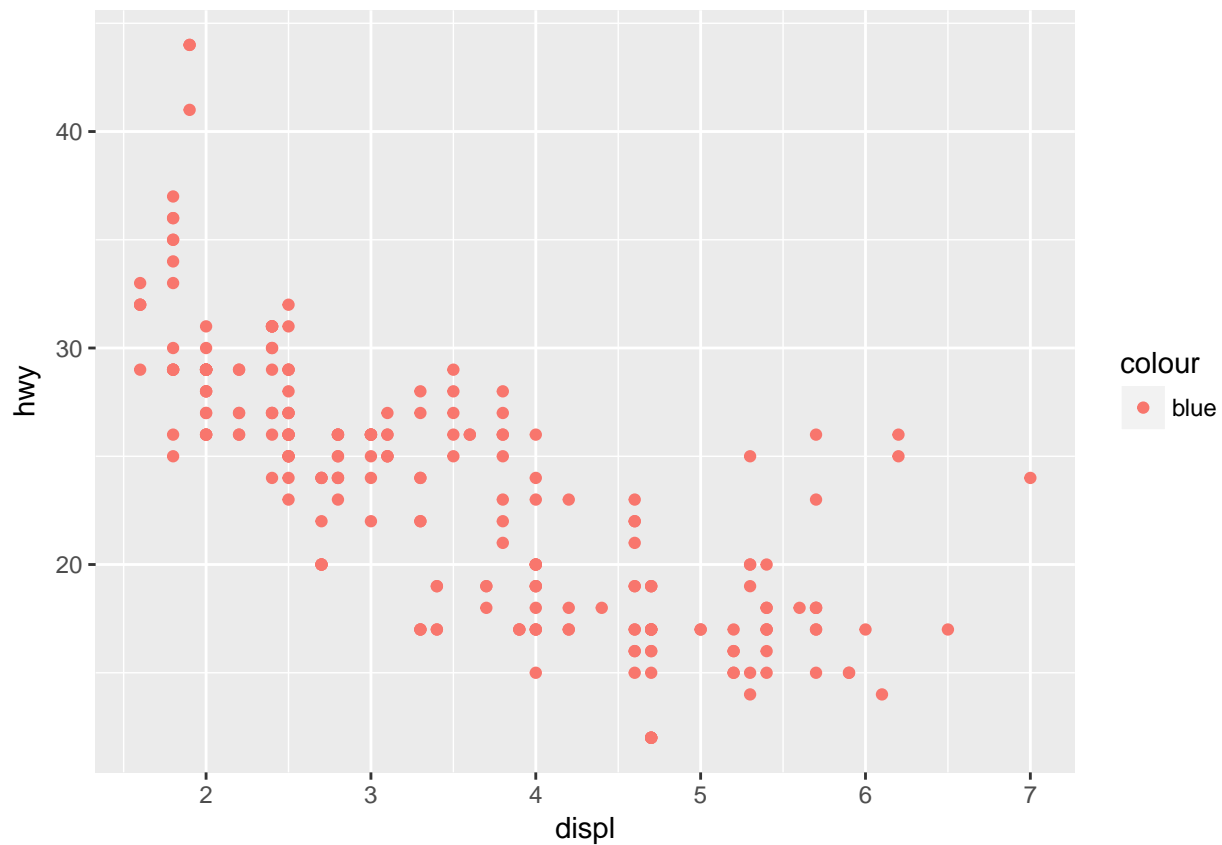
### Answer:

The graph is not useful because there is no clear relation between drv and class of a vehicle. For example, there are no front wheel and rear wheel drive types are 2 seater vehicles.

### Section 3.3.1: all excercises

#### Question 1:

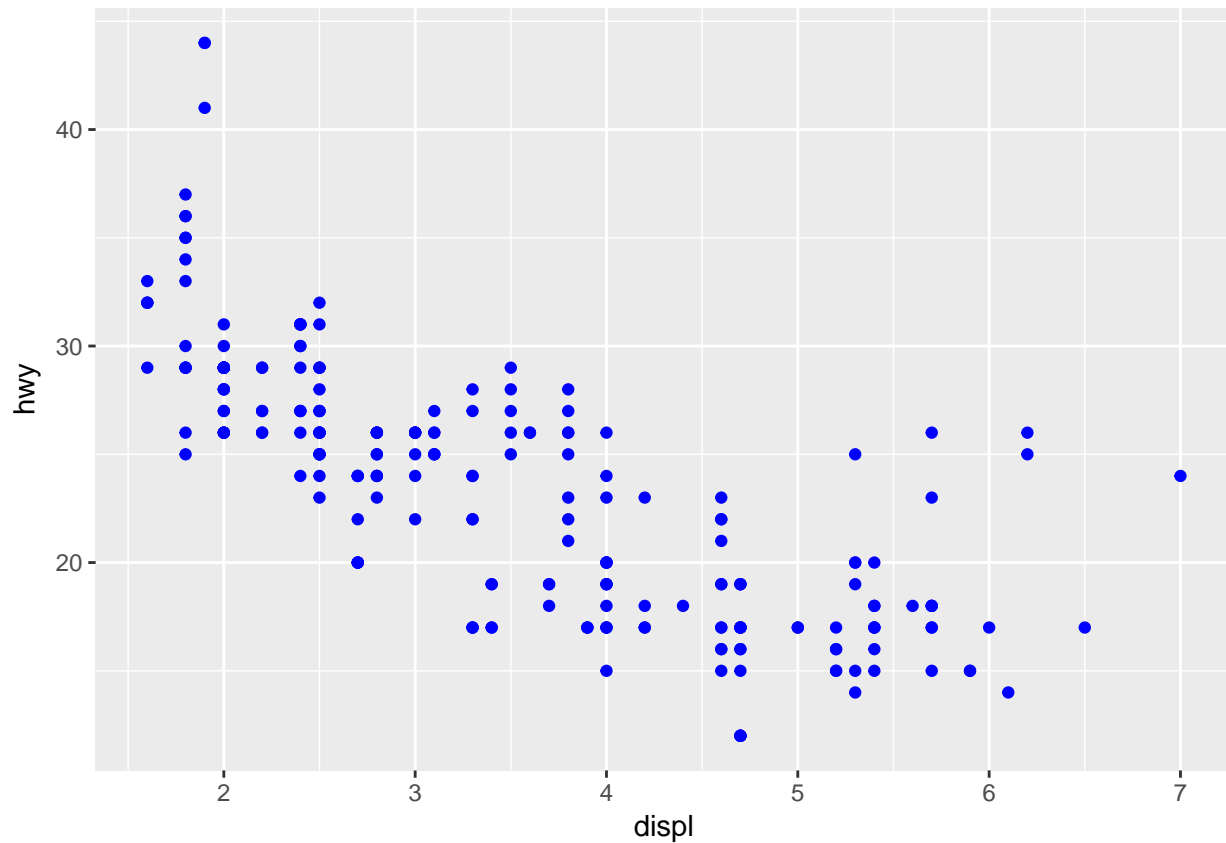
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color = "blue"))
```



### Answer:

In the above code we are trying to map the three aesthetics x, y, color to three variables of the dataset. But for the third aesthetic is assigned as a text “blue” same as displ and hwy. But we can manually override the color by placing the level outside of aes as shown below.

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



## Question 2:

?mpg

## Answer:

By looking at the above sample data, I can divide the variables as below.

### Categorical

manufacturer  
model  
cyl  
trans  
drv  
fl  
class

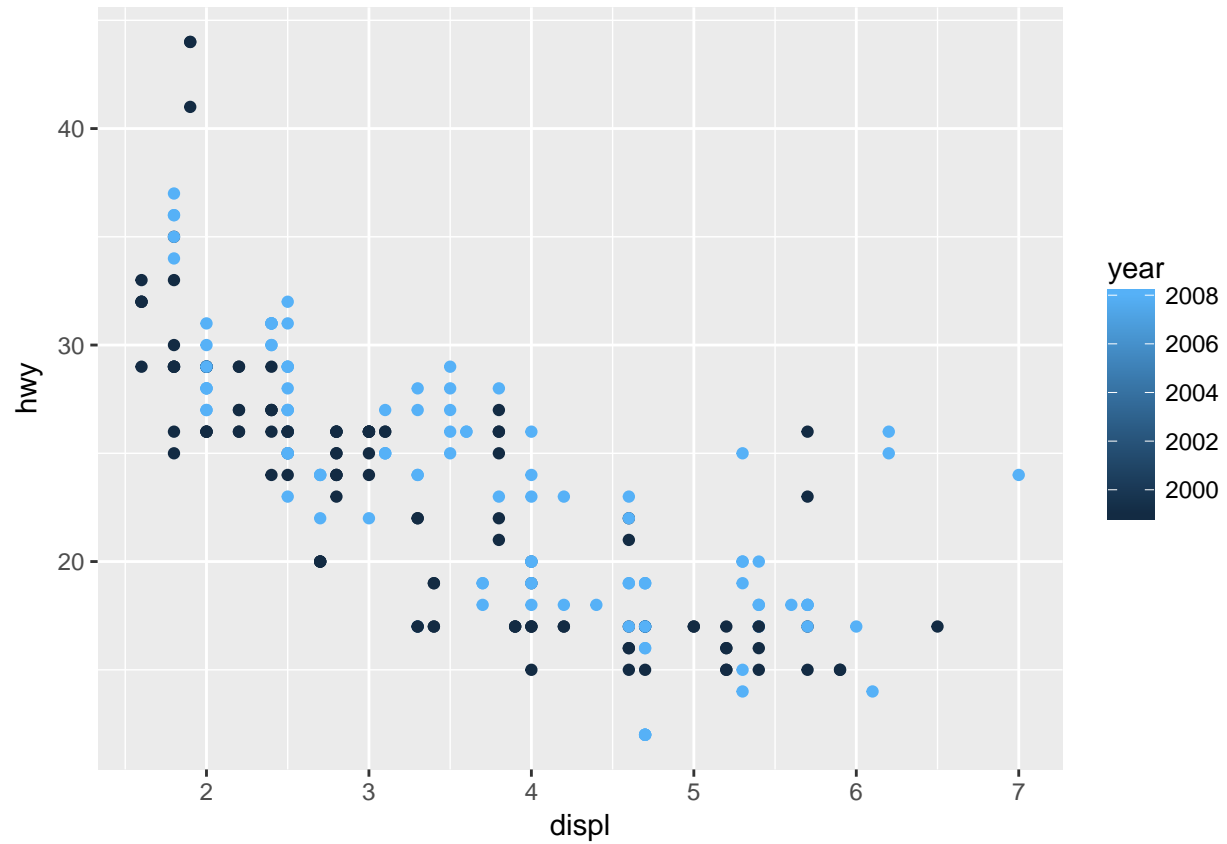
### Continuous

displ  
year

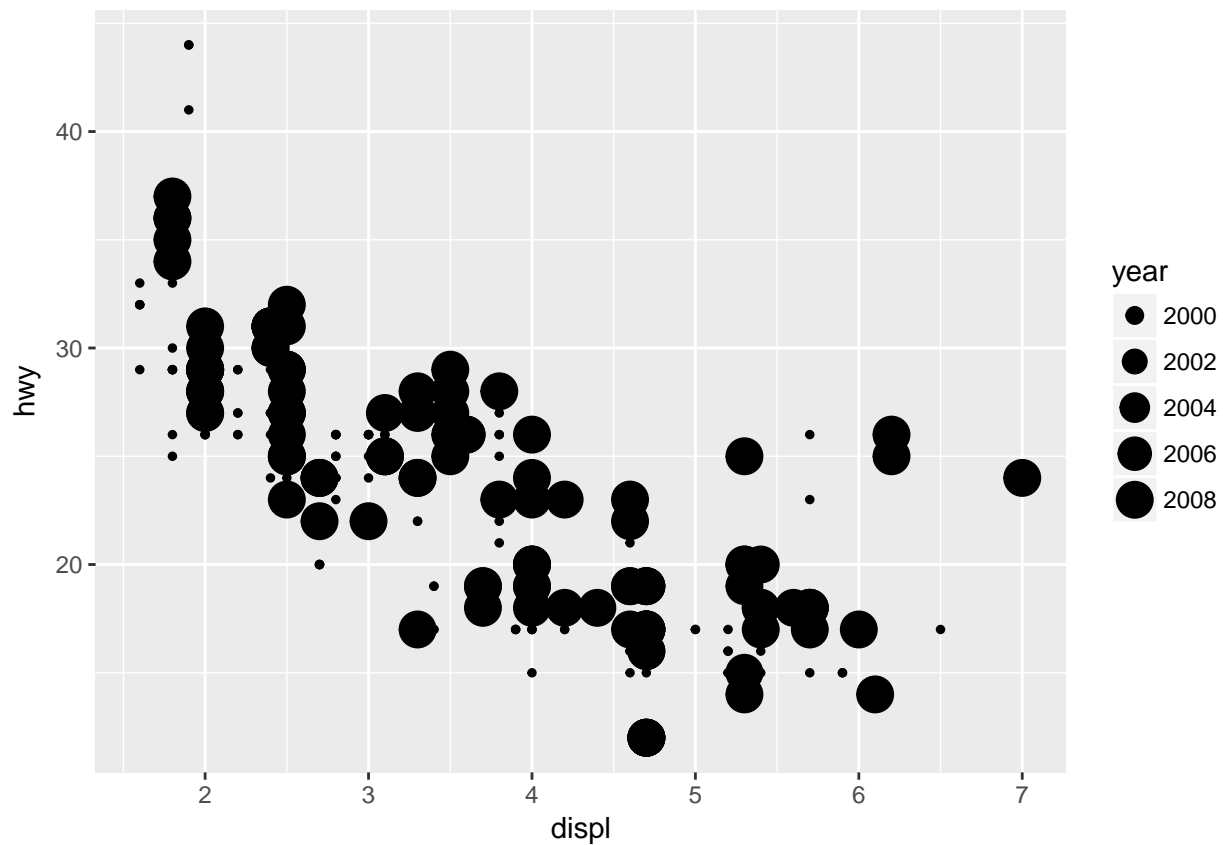
cty  
hwy

### Question 3:

```
## Mapping to Continuous Variables  
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = year))
```

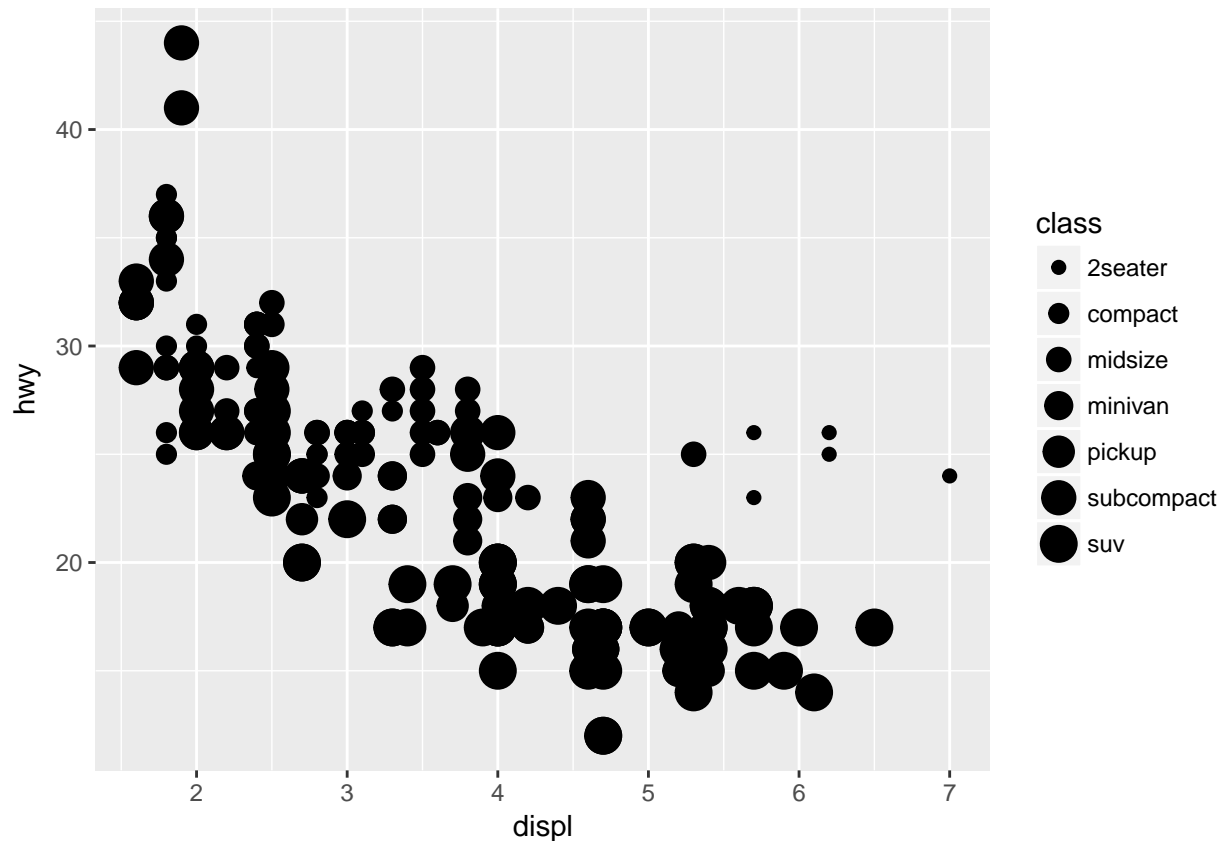


```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, size = year))
```



```
## ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, shape = year))  
  
## Mapping to discrete Variables  
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, size = class))
```





**Answer:**

### **\*\* Mapping to Continuous Variables\*\***

I have mapped the aesthetics color, size and shape to a continuous variable “year” and following are my observations.

When I mapped year to color, years are plotted with gradient colors on the map.

When I mapped year to size, years are plotted with different sizes and in this plot exact size of the point will explain the model years.

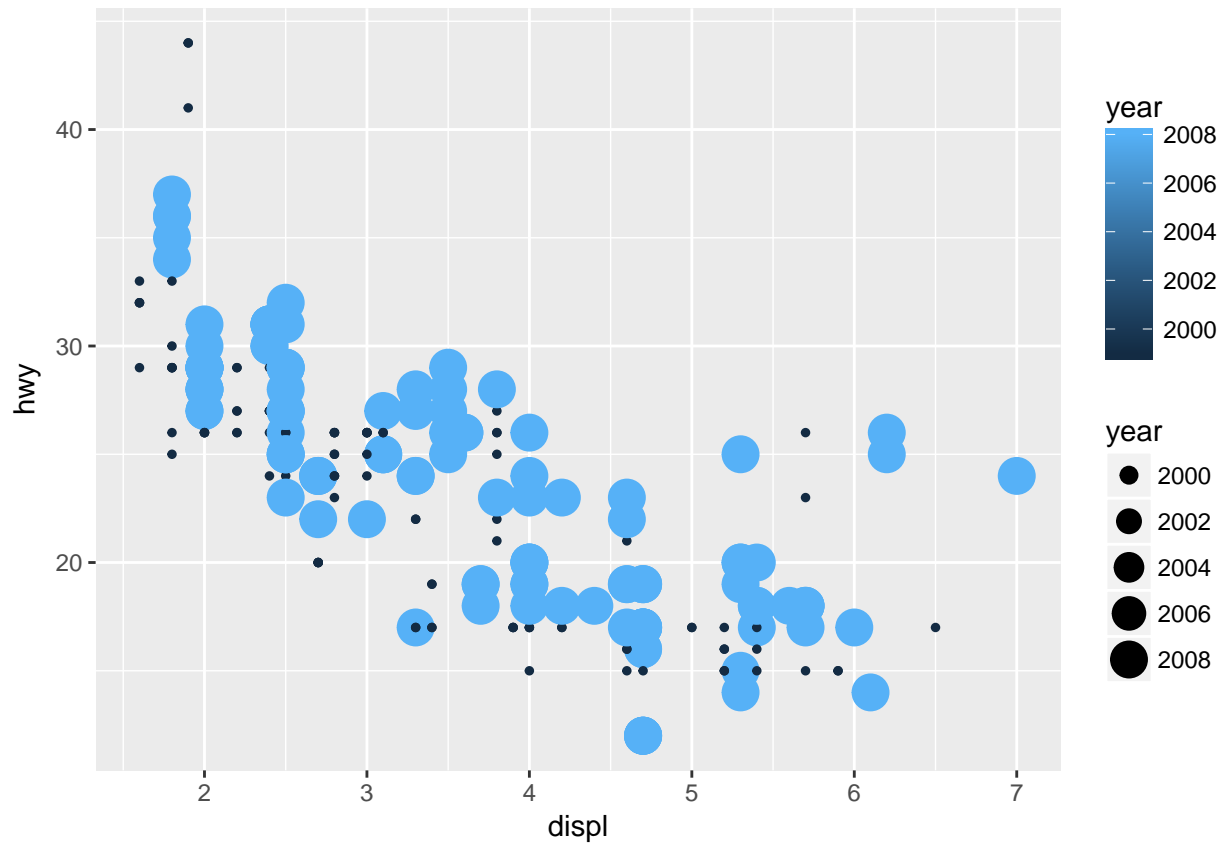
We can't map a continuous variable like year to shape aesthetic because shapes can't be ordered.

### **\*\* Mapping to categorical Variables\*\***

When we map an aesthetic to a categorical variable, category names are plotted instead of the scale.

**Question 4:**

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



**Answer:**

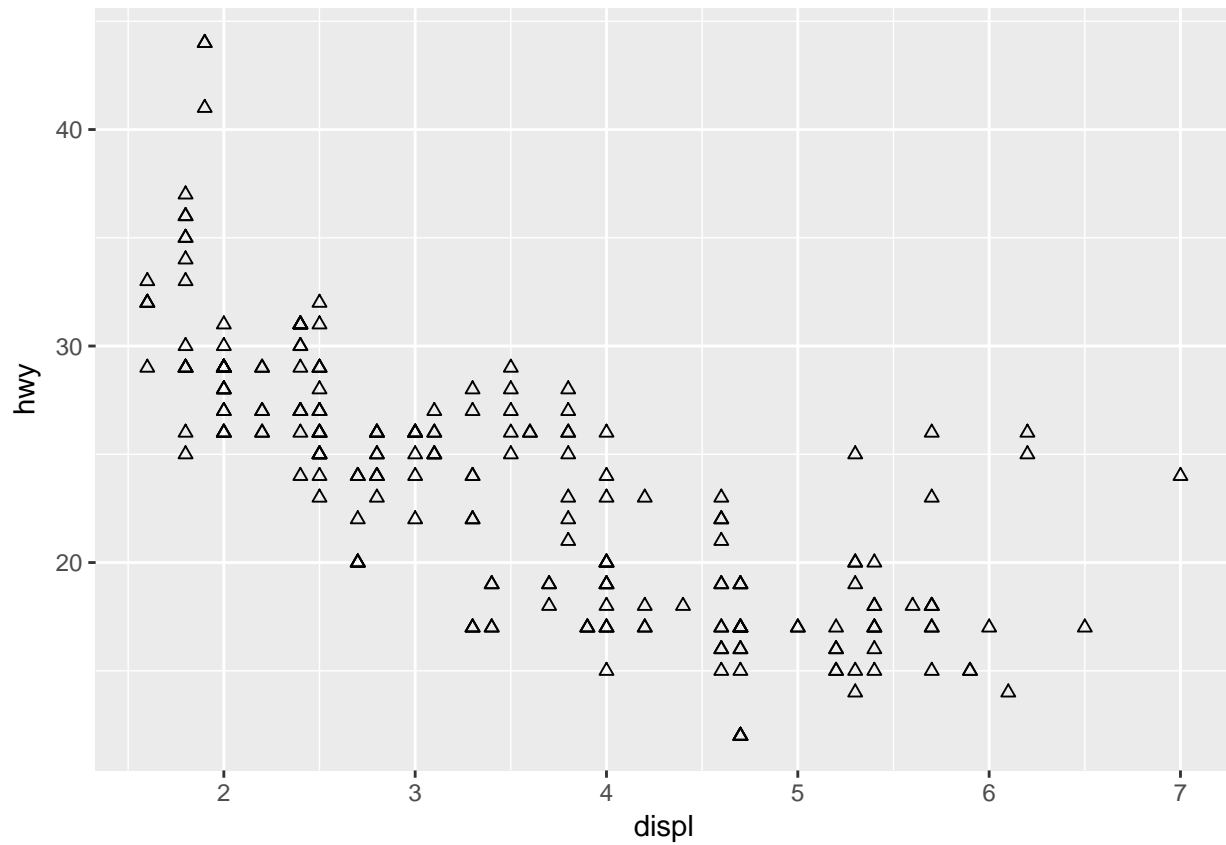
When we use a variable to map it to multiple aesthetics, simply two layers will be plotted on the same plot as shown above.

**Question 5:**

```
?geom_point

## Example by using stroke aesthetic

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, stroke = cty), shape = 24)
```

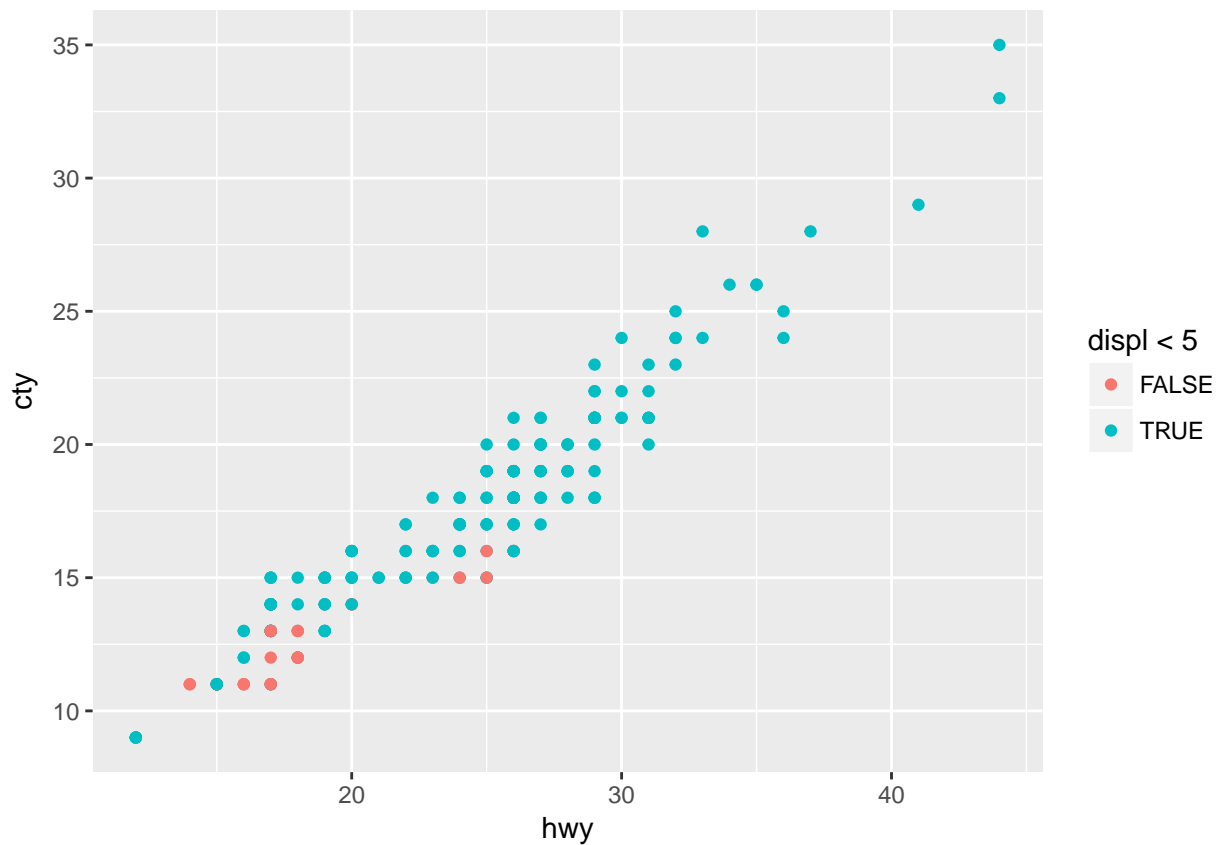


**Answer:**

stroke aesthetic controls the width of the border for any shapes with borders.

**Question 6:**

```
ggplot(data = mpg) + geom_point(mapping = aes(x = hwy, y = cty, colour = displ < 5))
```



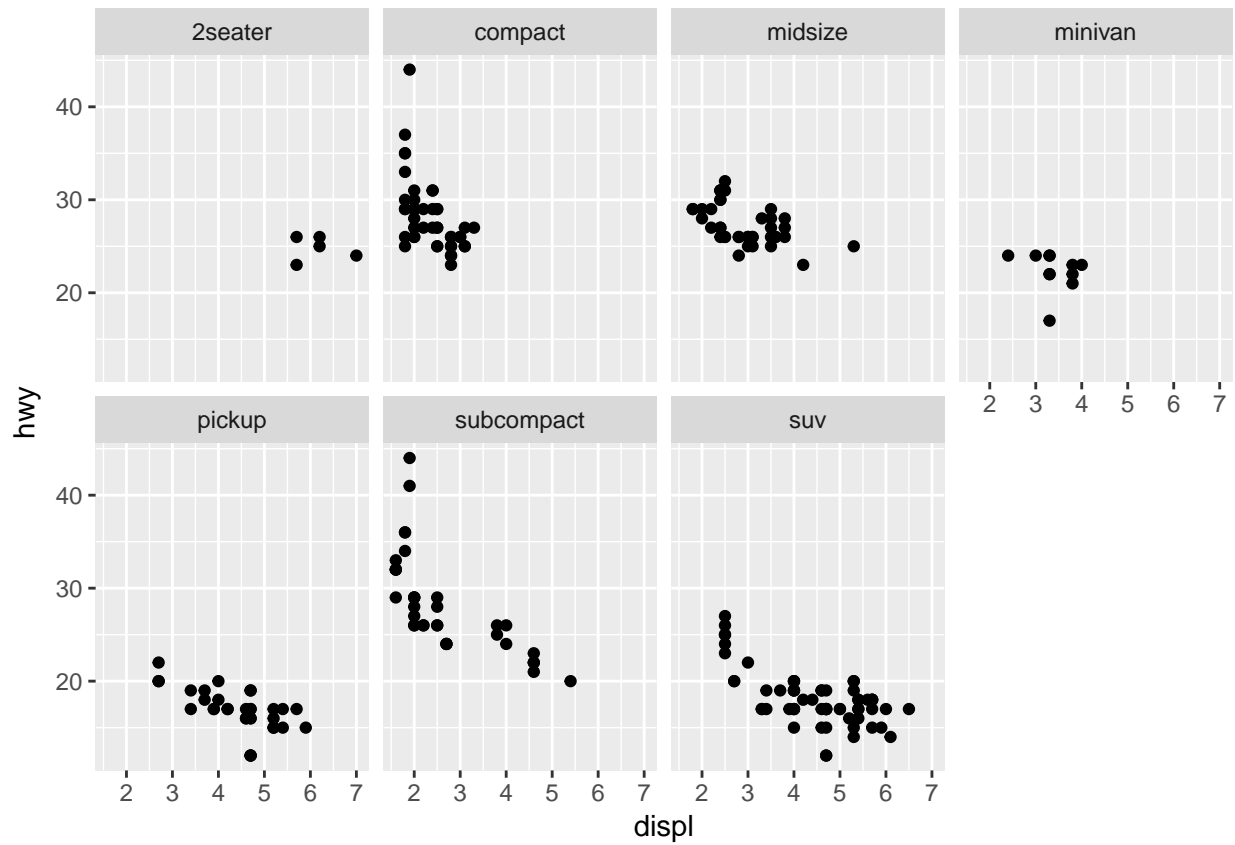
**Answer:**

When we map an aesthetic like colour to `displ < 5`, ggplot will takes it as an expression and maps that to True or False to the colour argument.

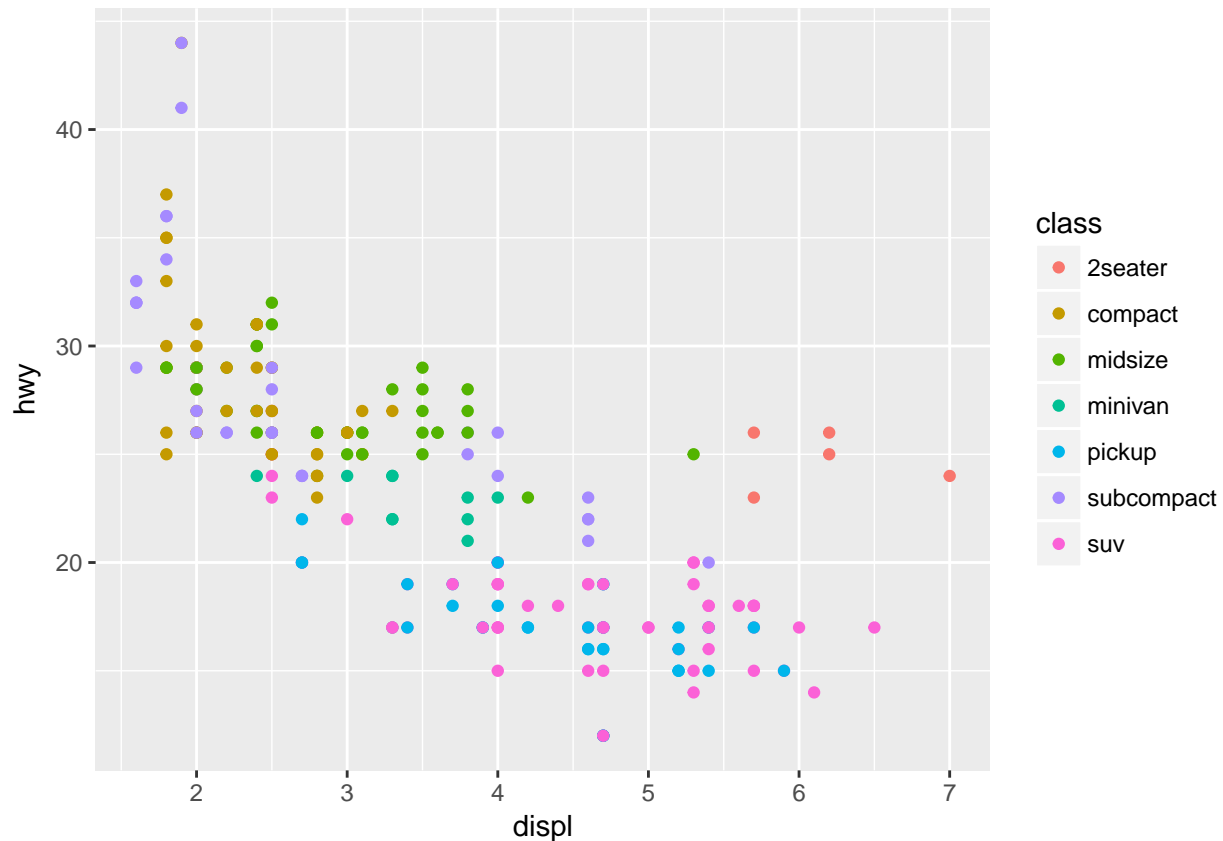
### Section 3.5.1: #4 and #5 only

**Question 4:**

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



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



**Answer:**

Basically with faceting we can observe the elements of each class separately where as with color aesthetics, it possible to observe the classes in a cluster like fashion. But it is quite difficult to read a graph with too many colors. On the otherhand with the larger datasets may be using coloring to create a overall clustering is better than reading individual clusters which are created by facets.

**Question 5:**

```
?facet_wrap
```

**Answer:**

nrows controls the number of rows and ncols controls the number of columns while faceting. scales is another option to control the axis of individual facets. We can use strip.position for controlling the label of each facet and we can use dir to control the orientation of individual facets.

### Section 3.6.1: #1-5. Extra Credit: Do #6

#### Question 1:

Answer:

line chart: `geom_line`

boxplot: `geom_boxplot`

histogram: `geom_histogram`

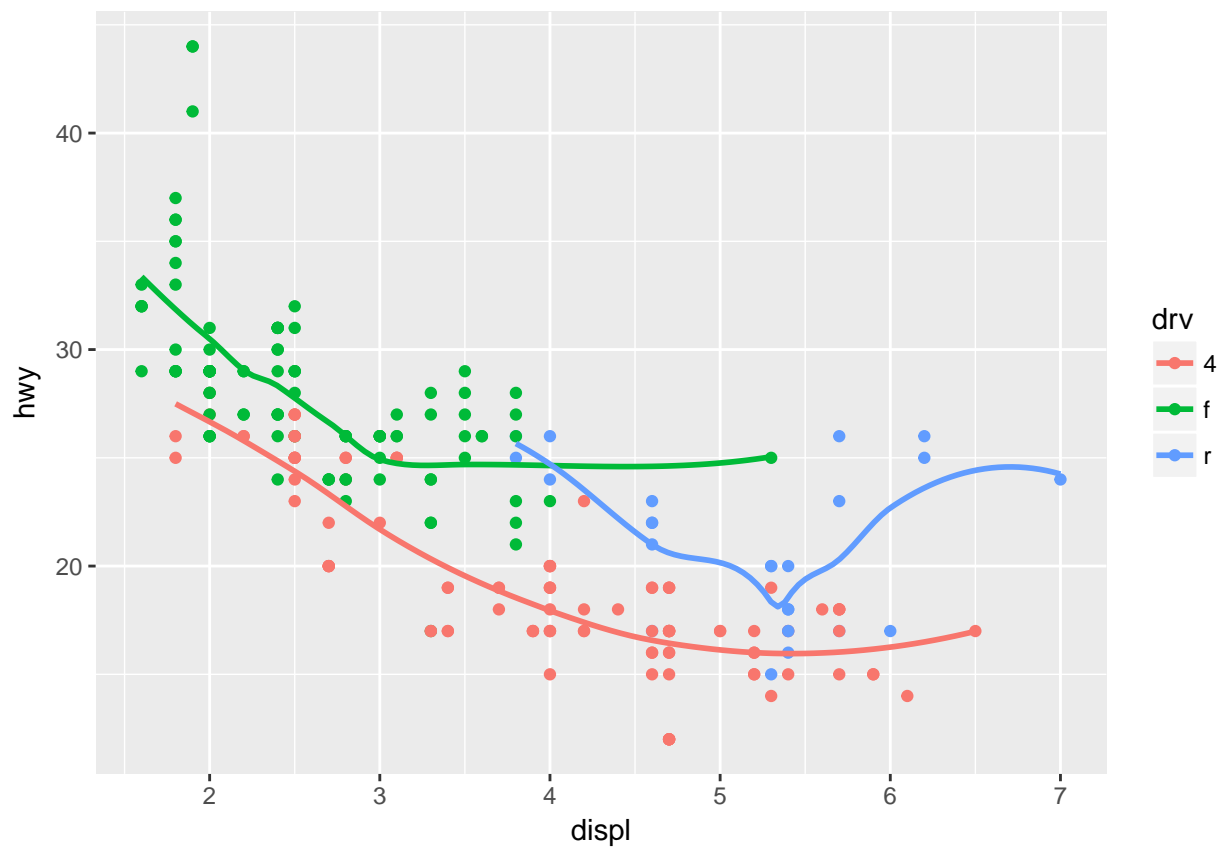
area chart: `geom_area`

#### Question 2:

Answer:

The given code will produce a scatter plot with `displ` on axis x and `hwy` on axis y. All the marks colored based on `drv` variable. On top of this there will be a smooth line graph layer is drawn without any standard errors.

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



### Question 3:

#### Answer:

`show.legend = FALSE` will remove / hide the legend from the graph. When we remove this the legend will appear and the graph will be shrunk on the available space.

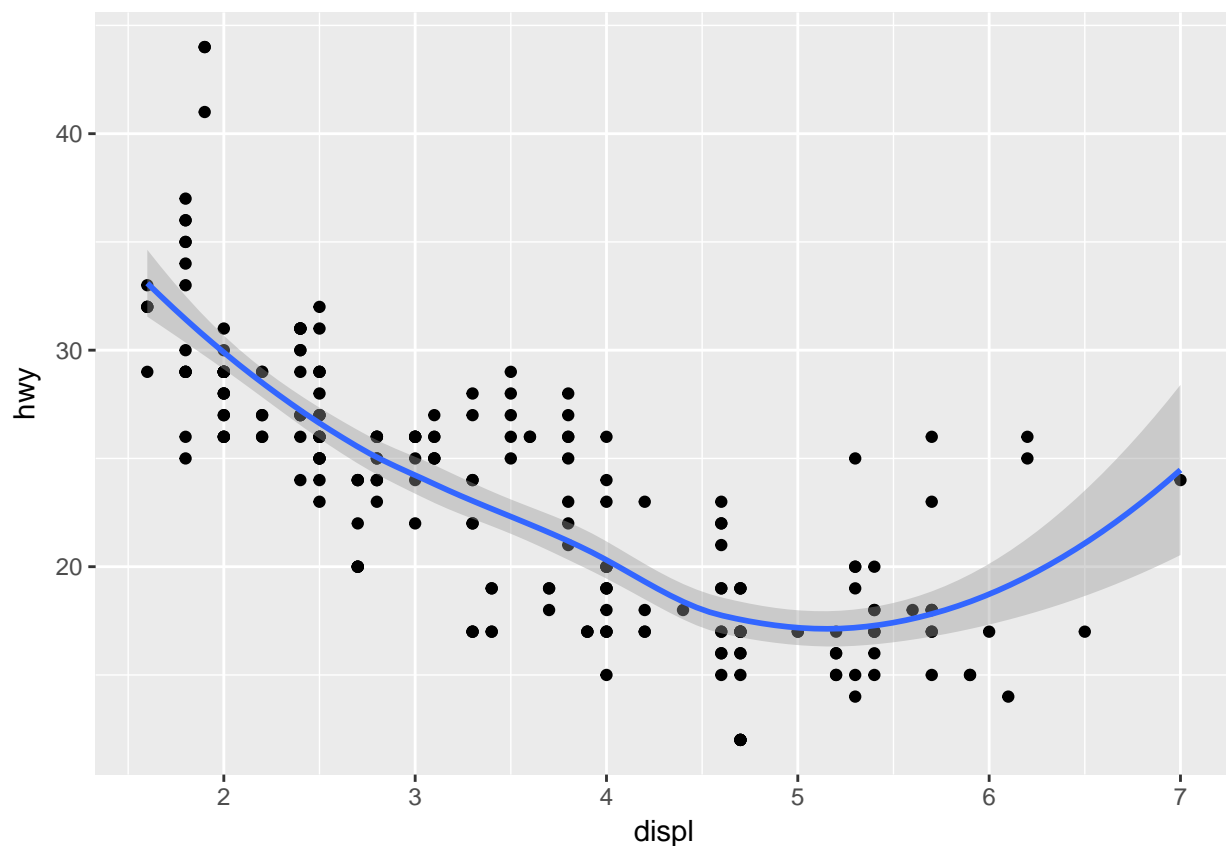
### Question 4:

#### Answer:

`se` specifies whether to add the standard error or not to the graph.

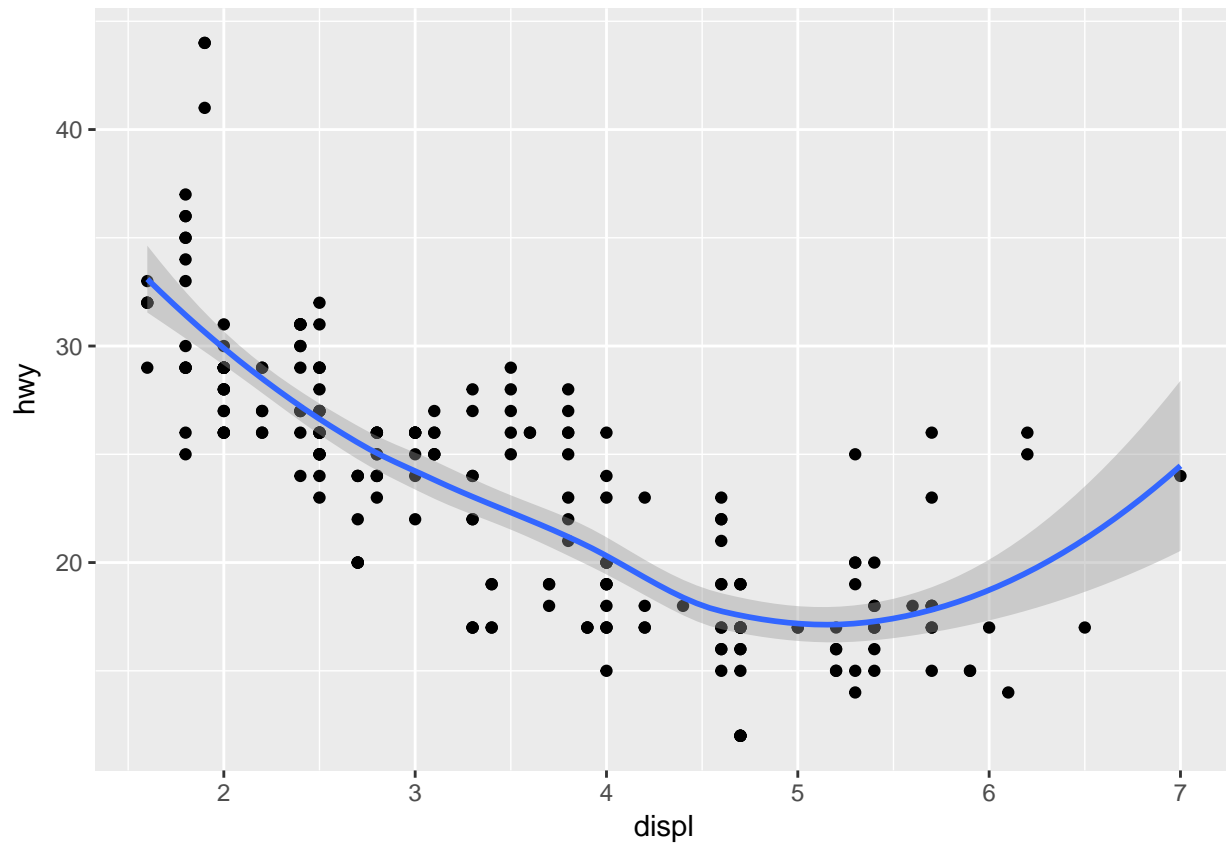
### Question 5:

```
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))
```





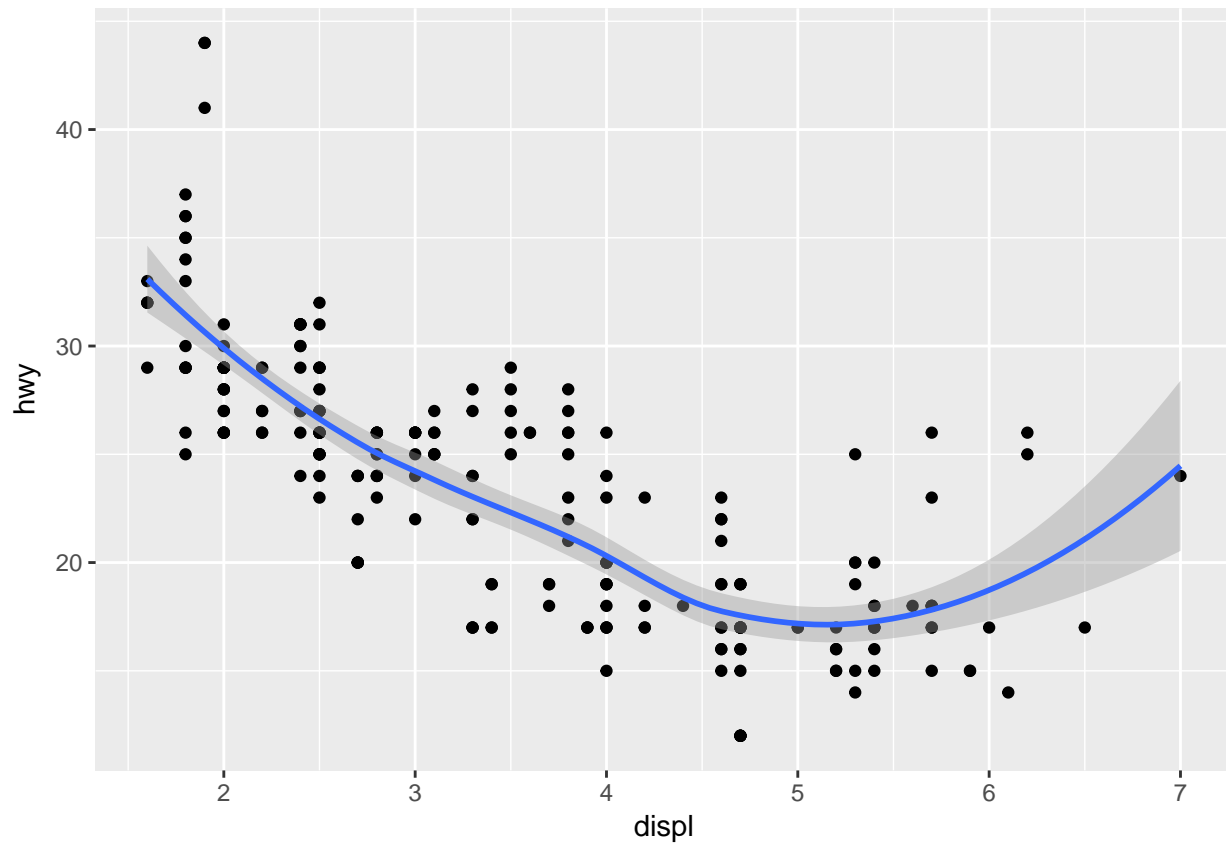
**Answer:**

No, above two graphs looks similar. As they use same data and mapping arguments, there is no need to specify individually. They can inherit the properties from global or parent ggplot object.

### Question 6:

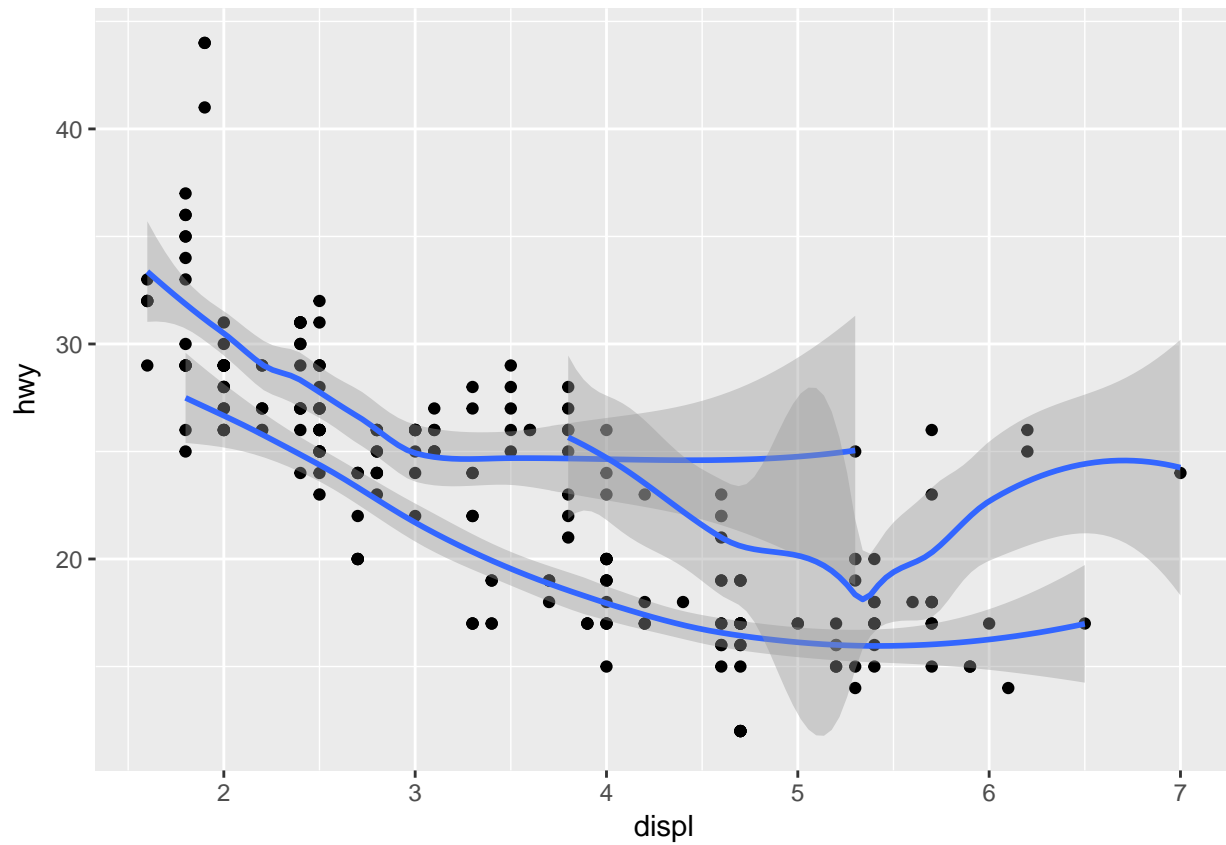
#### Graph 1

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



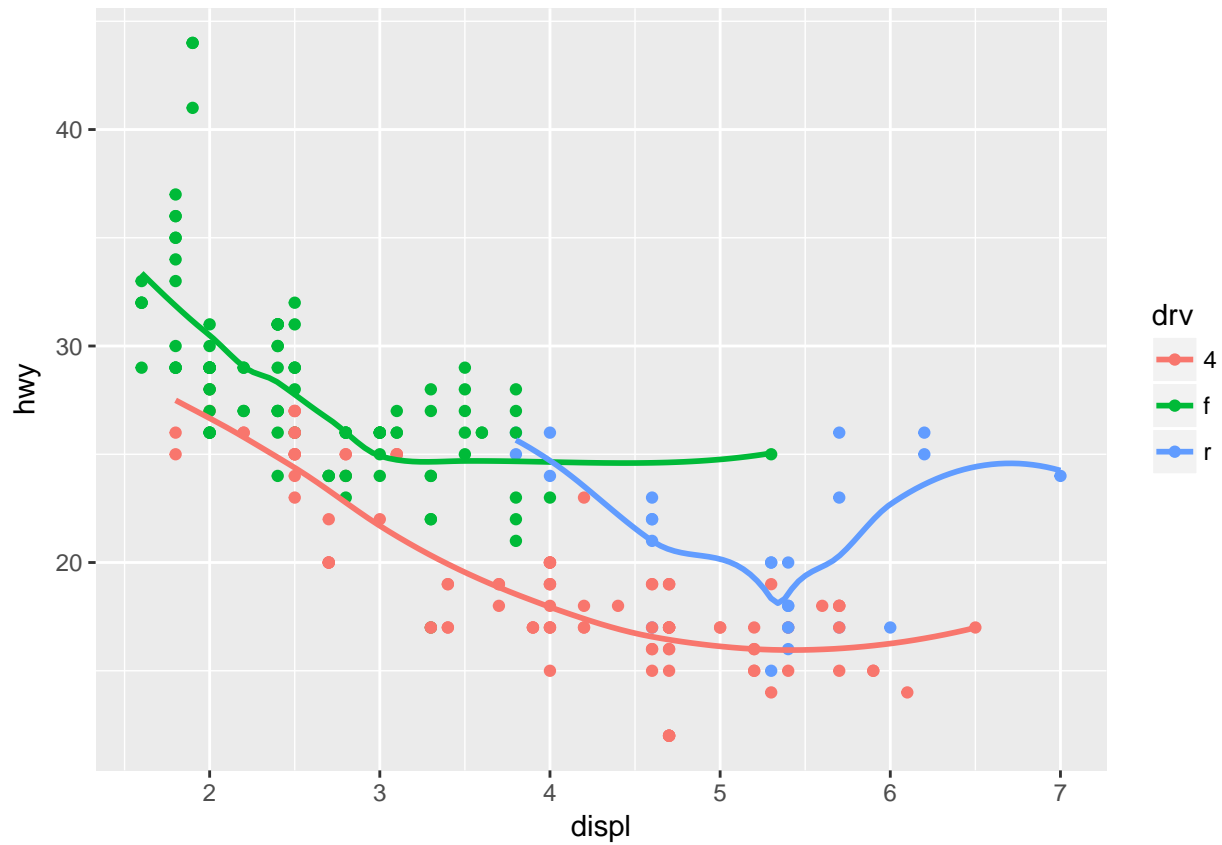
Graph 2

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



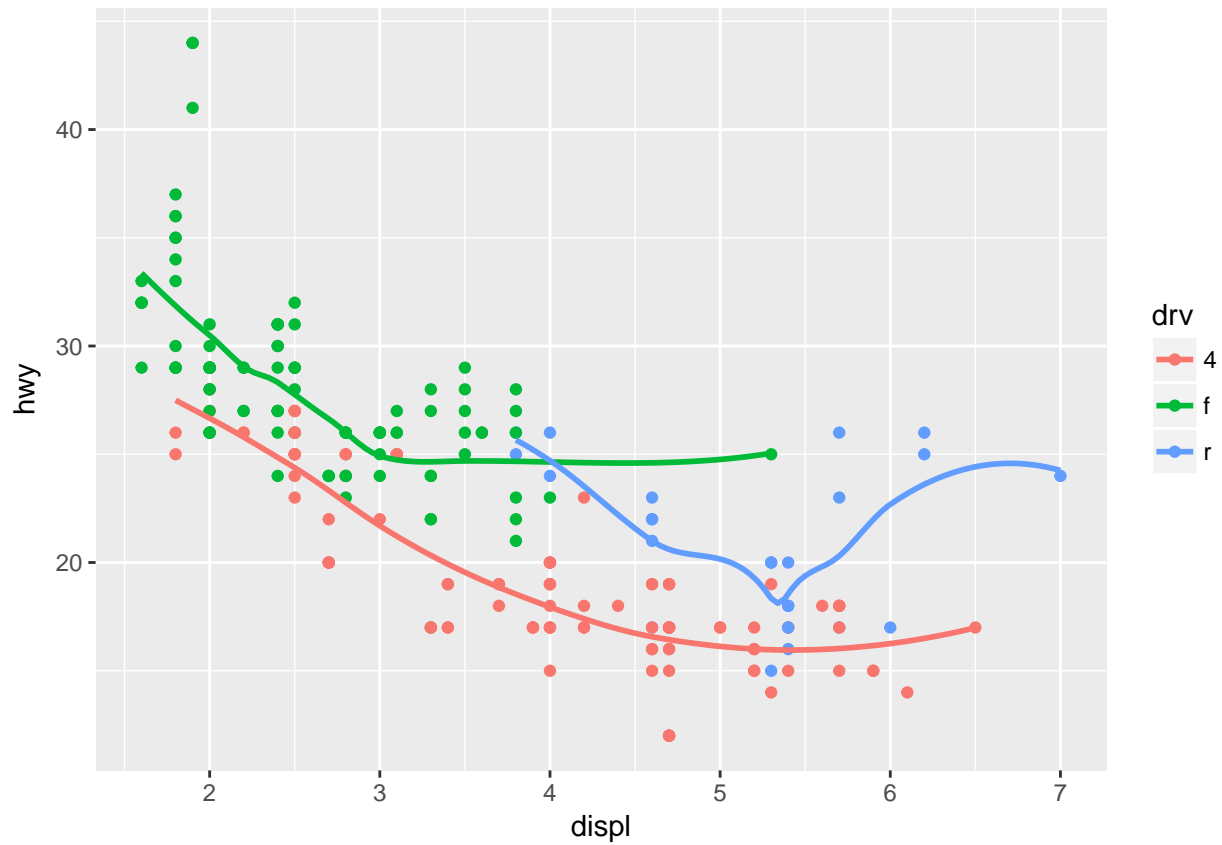
Graph 3

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



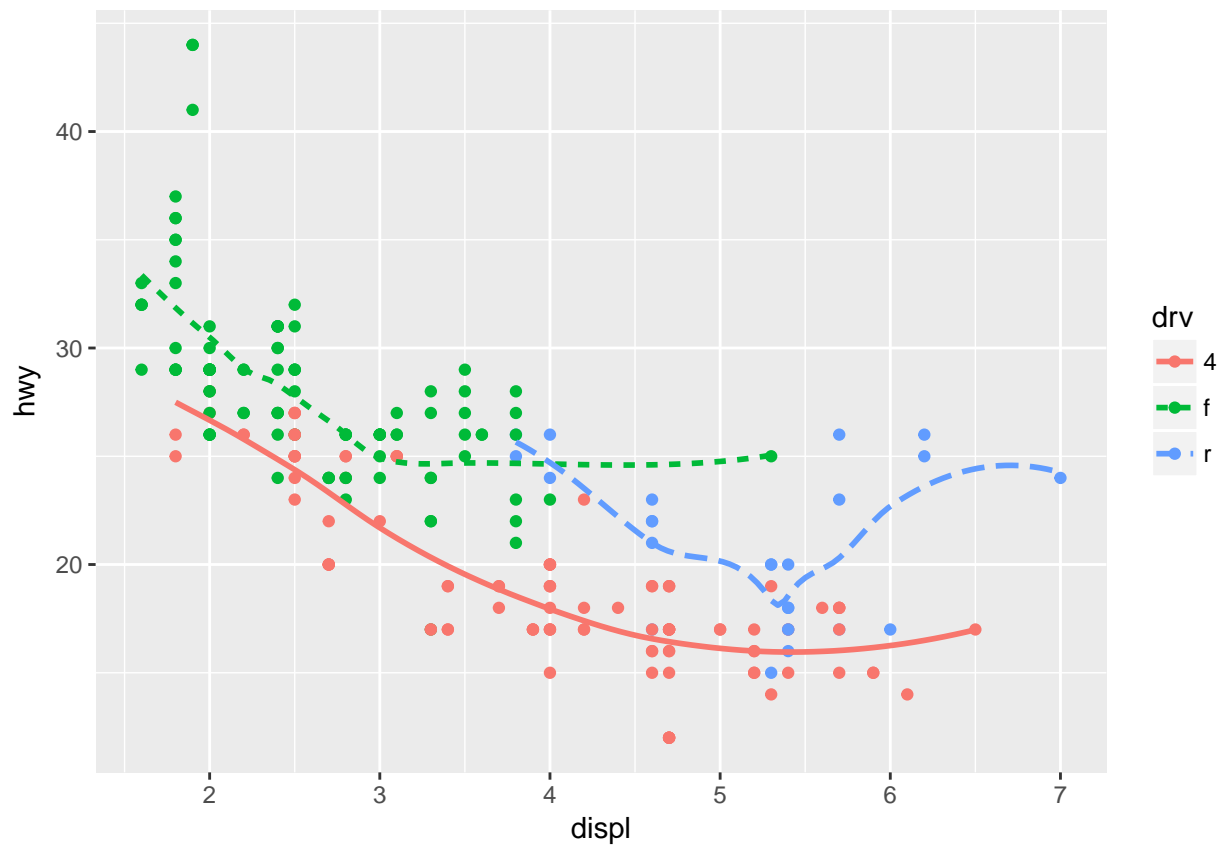
Graph 4

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



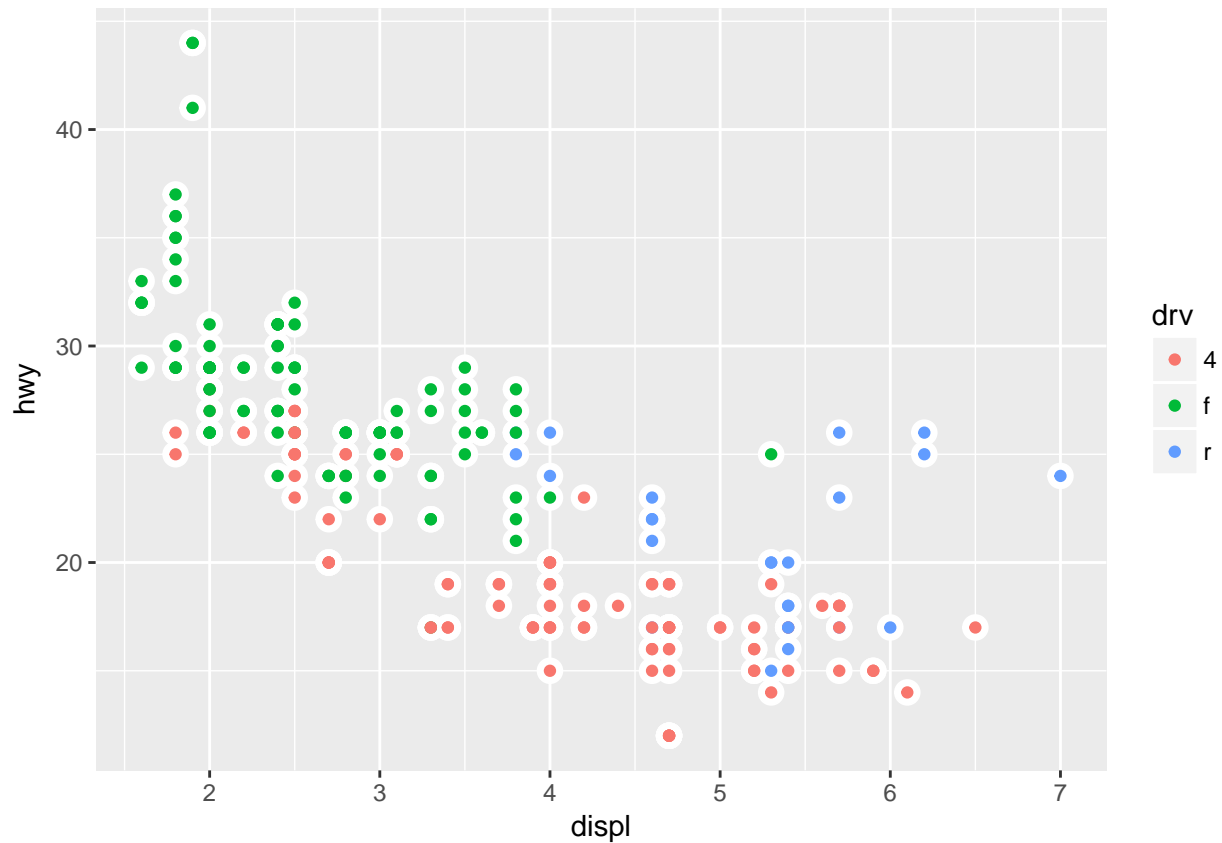
Graph 5

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



Graph 6

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) +
  geom_point(color = "White", size = 4) +
  geom_point(mapping = aes(color = drv))
```



### Section 3.7.1: #2 only

#### Question 2:

#### Answer:

geom\_bar & geom\_col are the two types of bar charts. geom\_bar draws the height of the bars relative to number of elements in the group, where are geom\_col plots data as is with out making any changes, in otherwords it uses the value of the data element to plot the data.