Data Visualisation - R for Data Science

Margaret Kinyanjui

March 21, 2019

R Markdown

Nairobi Women in Machine Learning and Data Science cohort 2 R exercises.

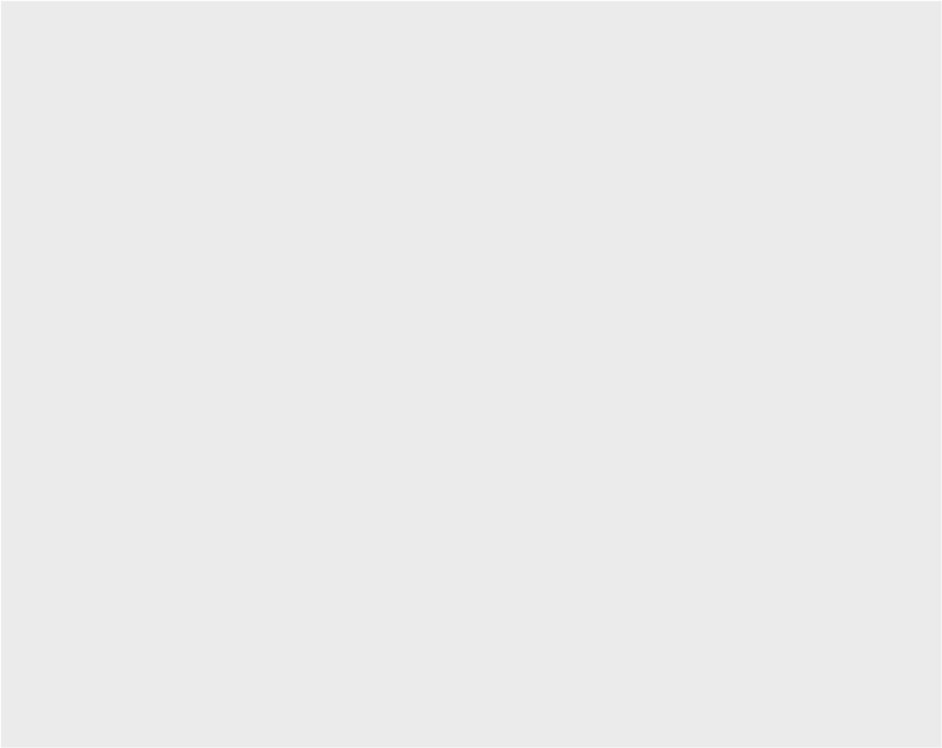
These are exercises of chapter 3 - Data Visualisation of the book titled “R for Data Science”. For more details on these exercises, see https://r4ds.had.co.nz/data-visualisation.html

loading the required package

```
library(tidyverse)

## -- Attaching packages ------------------------------------------------
tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.1     v stringr 1.4.0
## v readr   1.1.1     v forcats 0.3.0

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

3.2.4 Exercises

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



mpg - Fuel economy data from 1999 and 2008 for 38 popular models of car

Remark- I see a blank page plot

234 rows, 11 columns

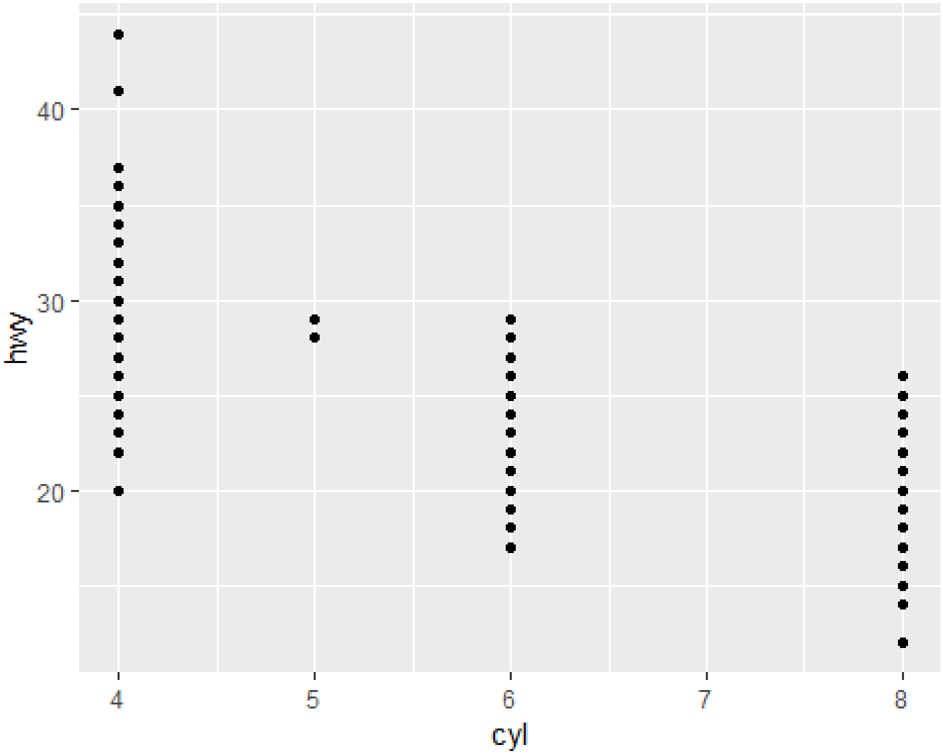
```
dim(mpg)

## [1] 234  11
```

meaning of drv: f = front-wheel drive, r = rear wheel drive, 4 = 4wd

```
?mpg

## starting httpd help server ... done
```

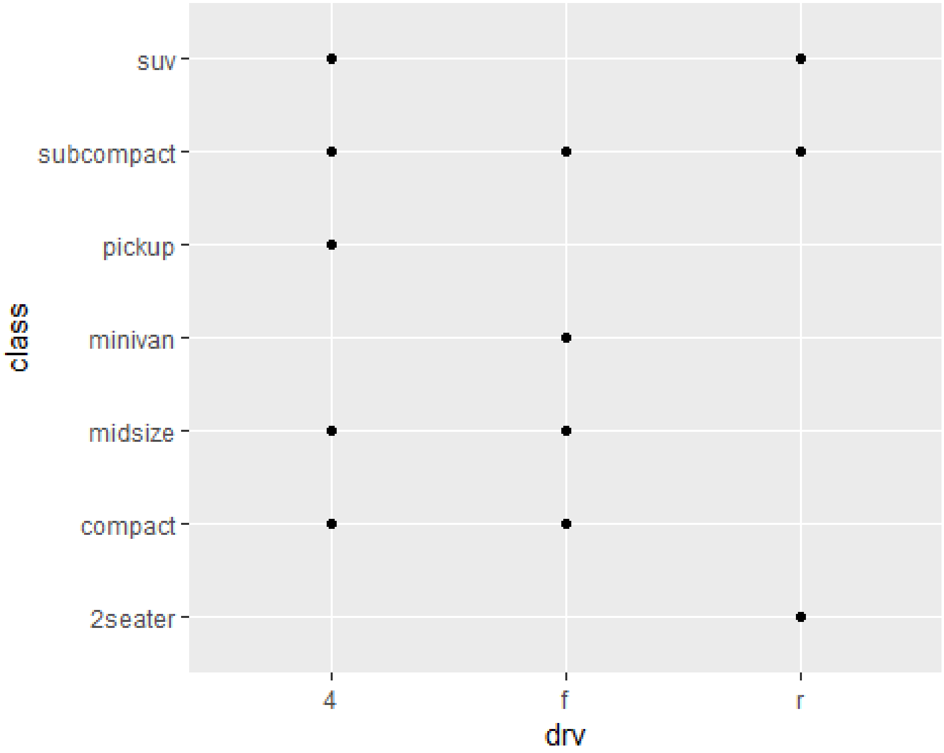
Scatter plot of hwy vs cyl

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



Scatterplot of class vs drv?

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



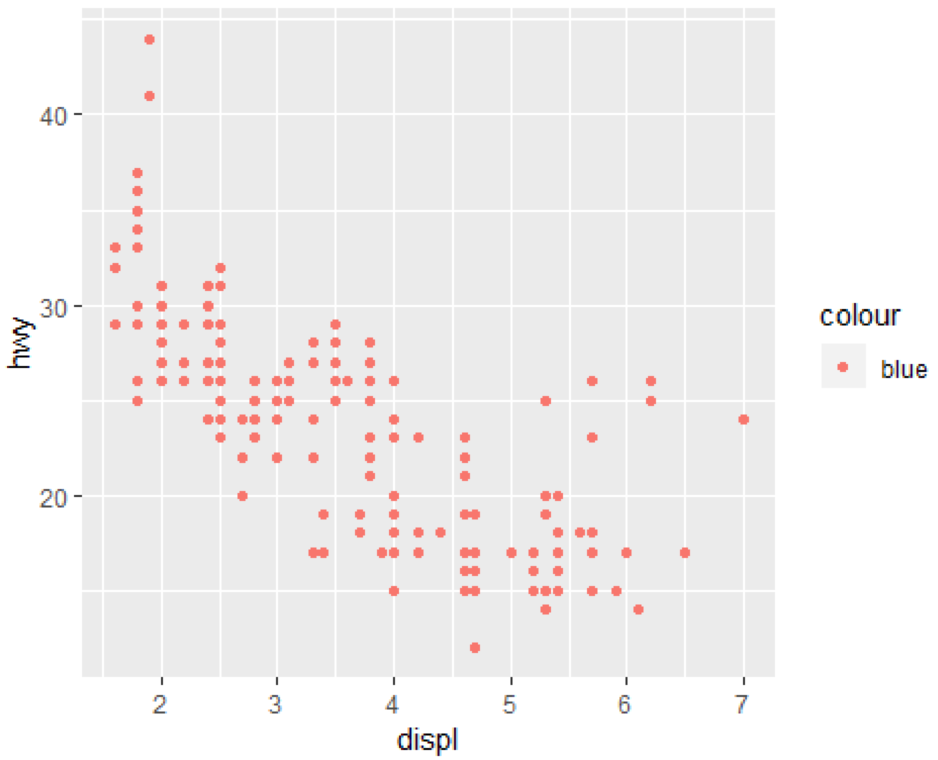
The plot is not useful because this is a comparison of two discrete(categorical variables)

3.3.1 Exercises

Why are the points not blue?

Since color is part of the aesthetic mapping, it assigns the points the default color(red)

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



Categorical/ Continuous variables?

Categorical: manufacturer, model, drv, fl, class, cyl

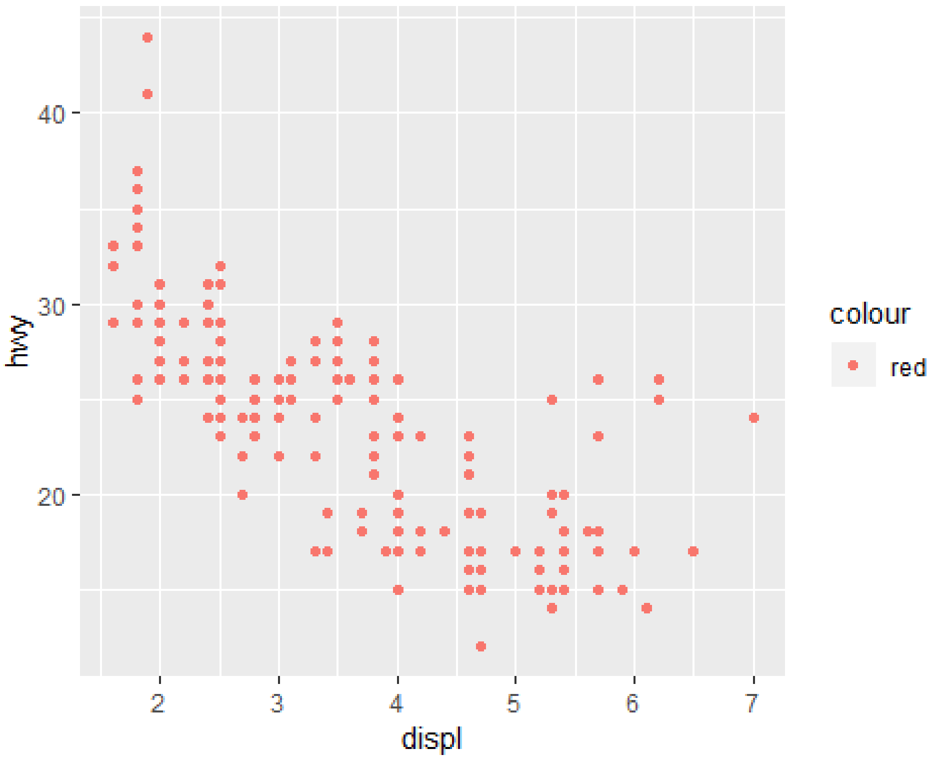
Continuous: cty, hwy, displ

```
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~
```

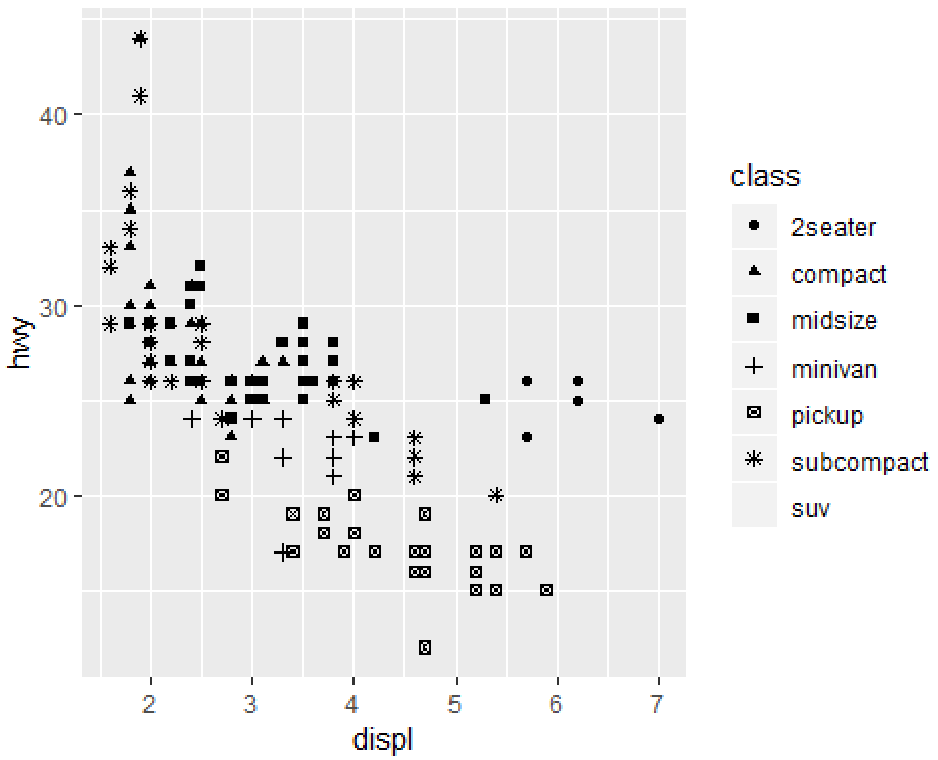
```
##  8 audi         a4 q~   1.8  1999     4 manu~ 4        18     26 p     
comp~
##  9 audi         a4 q~   1.8  1999     4 auto~ 4        16     25 p     
comp~
## 10 audi         a4 q~   2    2008     4 manu~ 4        20     28 p     
comp~
## # ... with 224 more rows
```

Mapping continuous and categorical variables

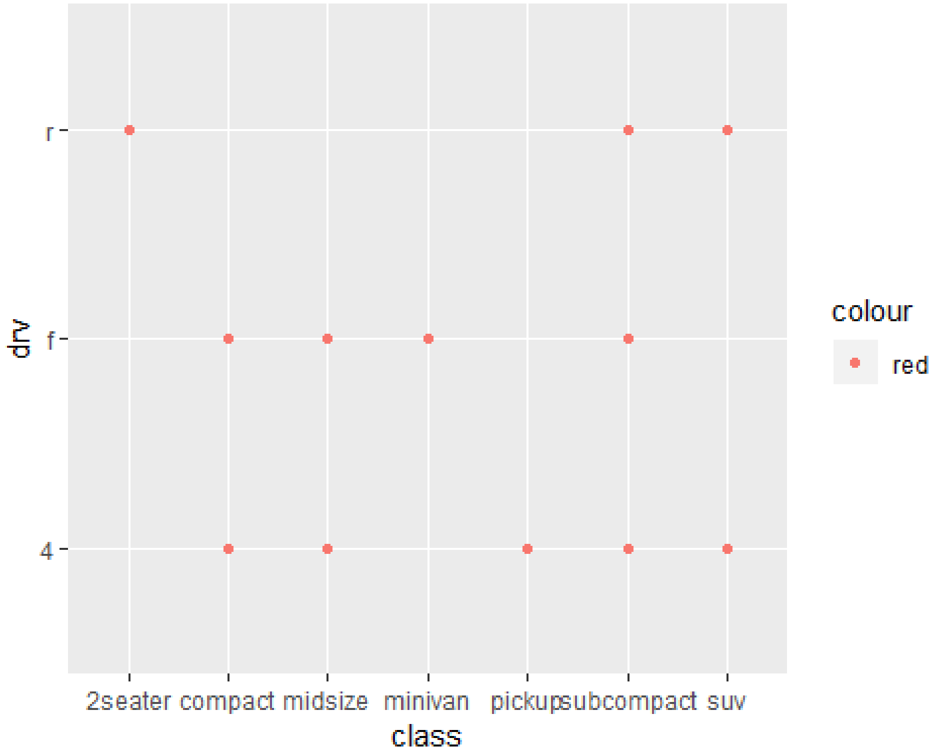
```
ggplot(data = mpg)+
  geom_point(mapping = aes(x = displ, y = hwy, color = "red"))
```



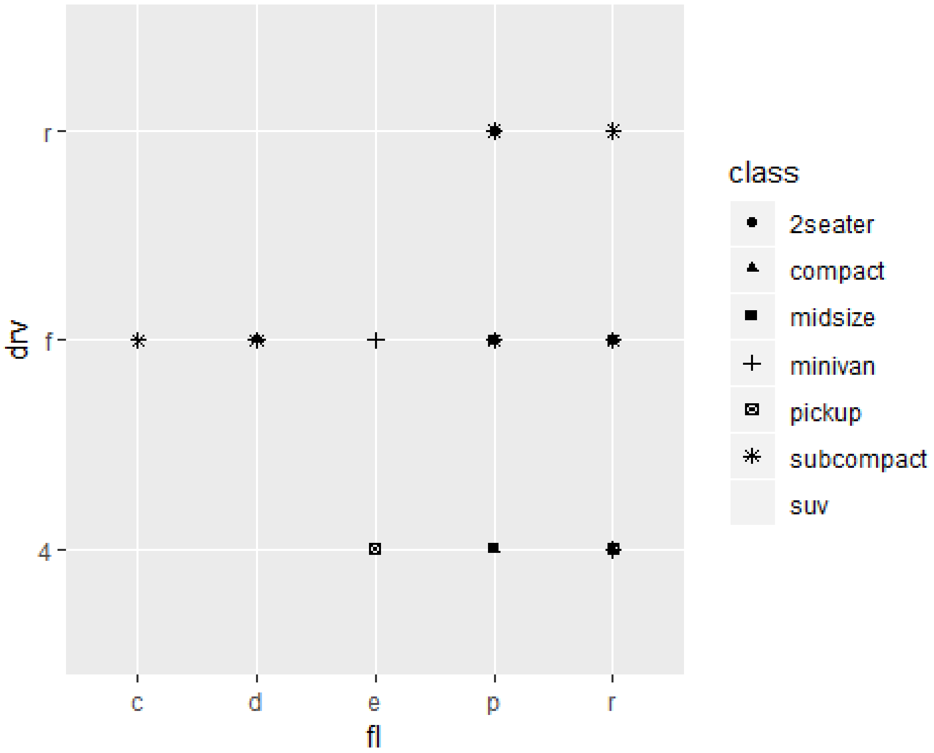
```
ggplot(data = mpg)+
  geom_point(mapping = aes(x = displ, y = hwy, shape = class))

## 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).
```



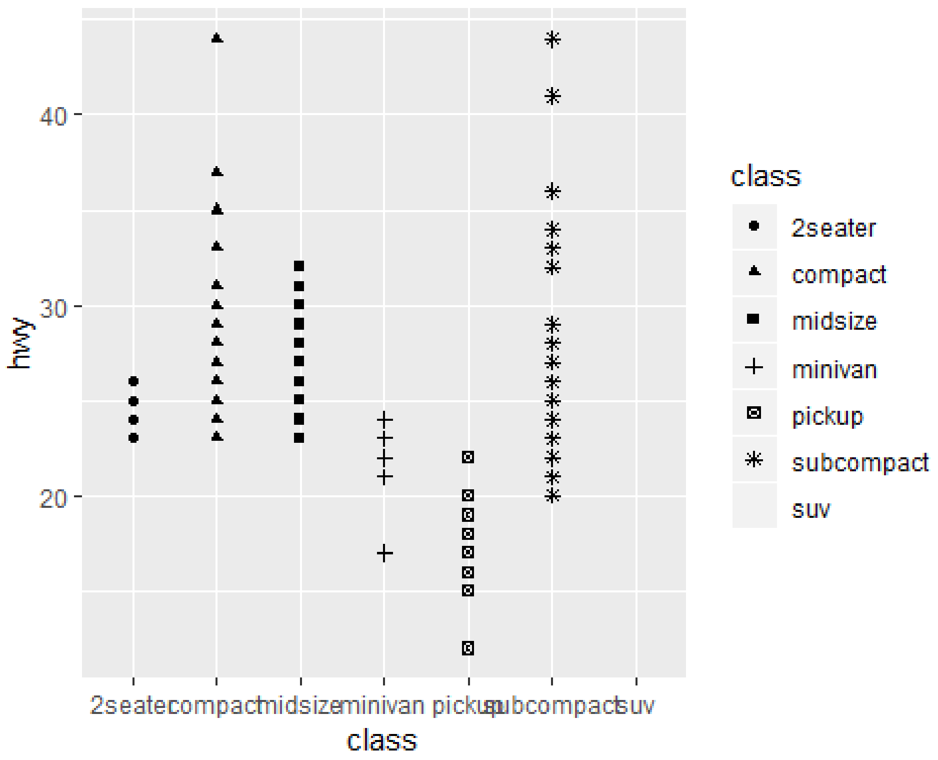
```
ggplot(data = mpg)+
  geom_point(mapping = aes(x = class, y = drv, color = "red"))
```



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

## 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).
```



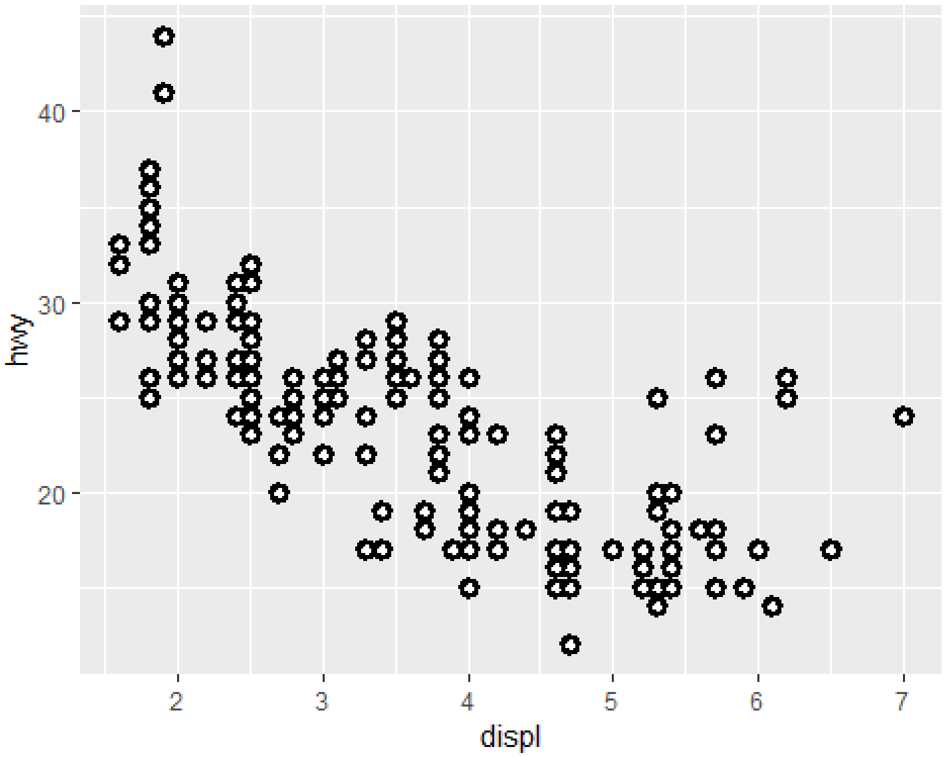
```
#mapping the same variable to multiple aesthetics

ggplot(data = mpg)+
  geom_point(aes(x = class, y = hwy, shape = class))

## 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).
```



What does the stroke aesthetic do?

For shapes that have a border (like 21), you can colour the inside and outside separately. Use the stroke aesthetic to modify the width of the border

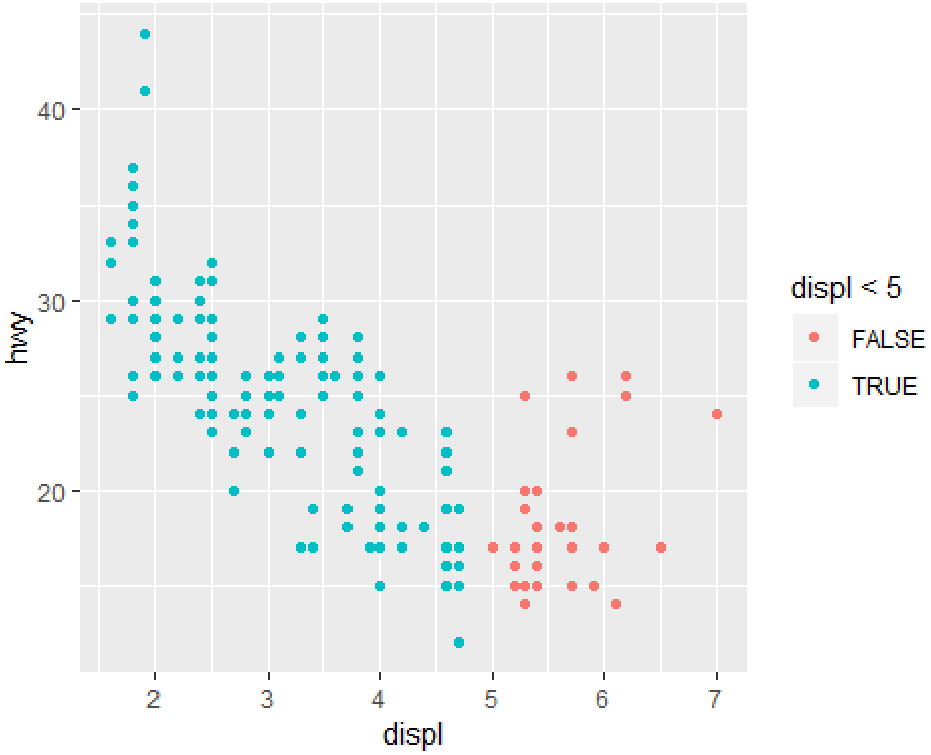
```
ggplot(mpg, aes(displ, hwy))+
  geom_point(shape = 21, color = "black", fill = "white", size = 2, stroke =
2)
```



Notice there is a split of points by color when displ<5 is mapped.

If true it is assigned a blue color, otherwise it is red.

```
ggplot(mpg, aes(displ, hwy))+
  geom_point(aes(color = displ<5))
```



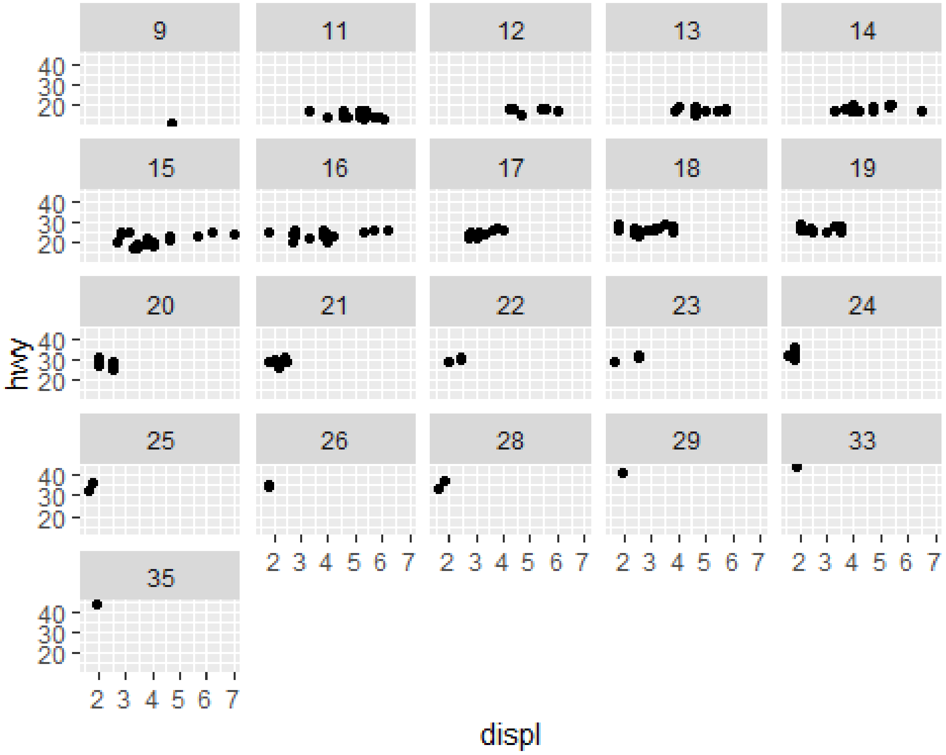
Get help about any R function by running ?function_name

3.5.1 Exercises

Try facet on a continuous variable?

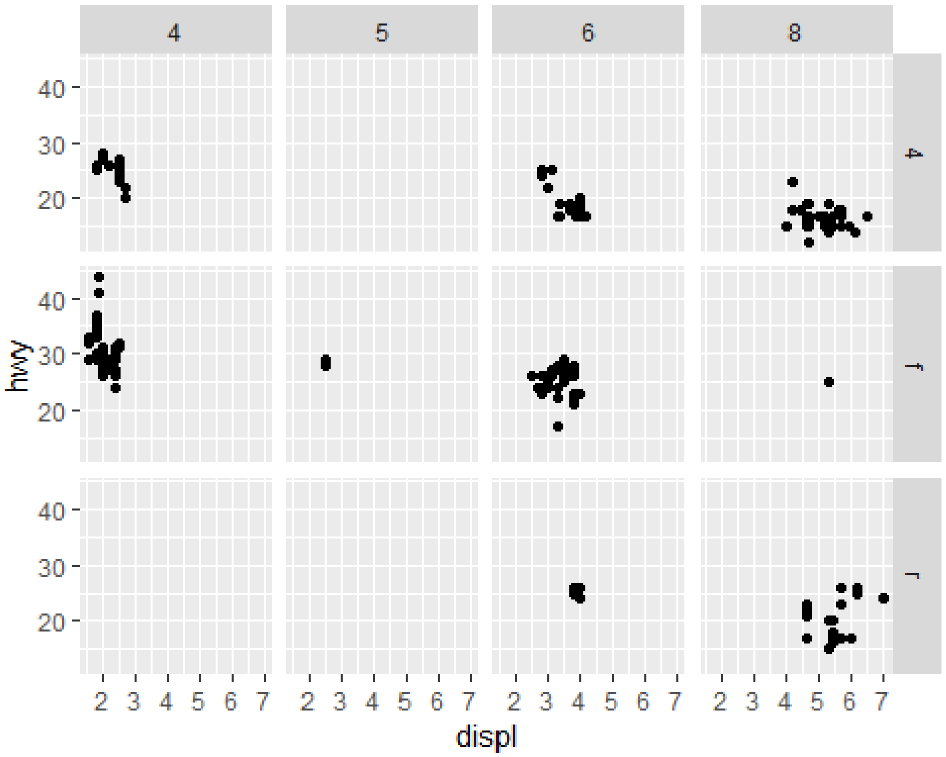
Continuous variables now treated as categorical/discrete

```
ggplot(data = mpg)+
  geom_point(aes(displ, hwy))+
  facet_wrap(~ cty)
```



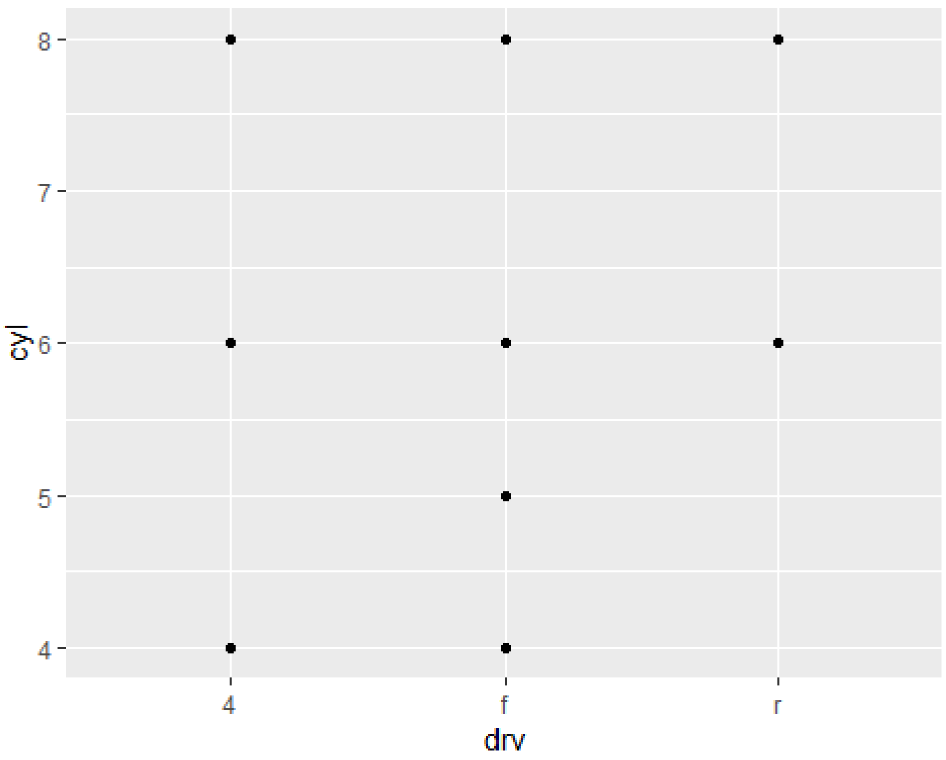
Why the empty cells in the code below?

```
ggplot(mpg, aes(displ, hwy))+
  geom_point()+
  facet_grid(drv~cyl)
```



No values for rear wheel - 4 and 5 cylinders

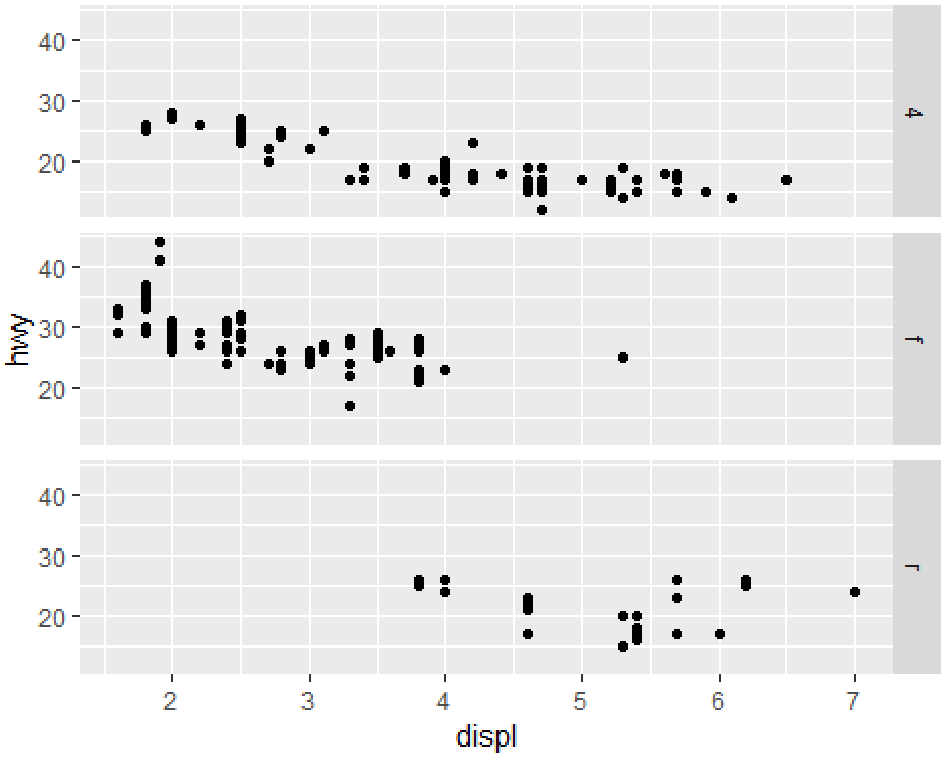
```
ggplot(data = mpg)+
  geom_point(mapping = aes(x = drv, y = cyl))
```



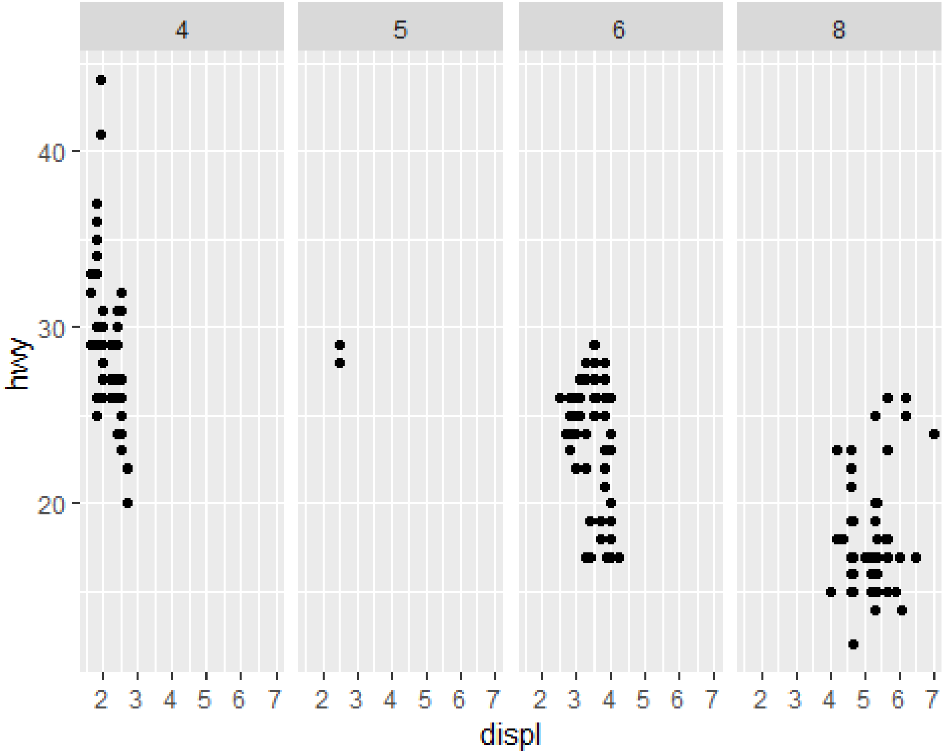
The first plot is categorised rowwise/ y-axis(drv). The . doesn’t define the column.

The second plot is categorised columnwise(cyl). The . doesn’t define the row.

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



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



Advantages

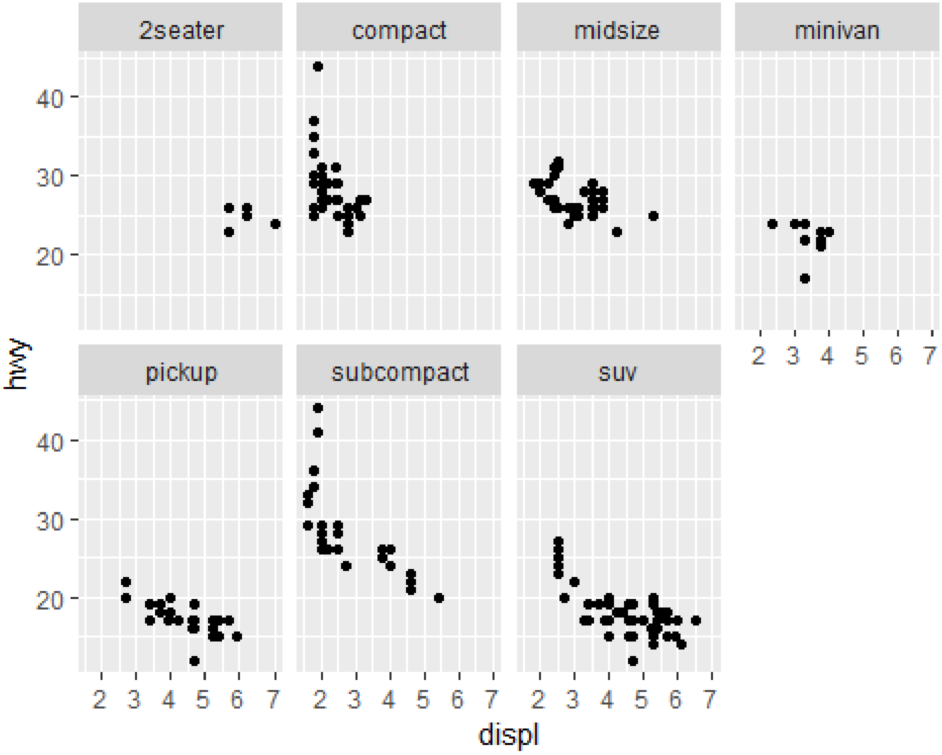
1. Once can easily identify the relationship between engine size and fuel usage for each class of car model

Disadvantage

1.One can't easily identify the outliers that is cars that use the most fuel and their engine size.

With a larger data set, there would probably be more variation of the points.

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



nrow = controls the number of rows for the faceting variables

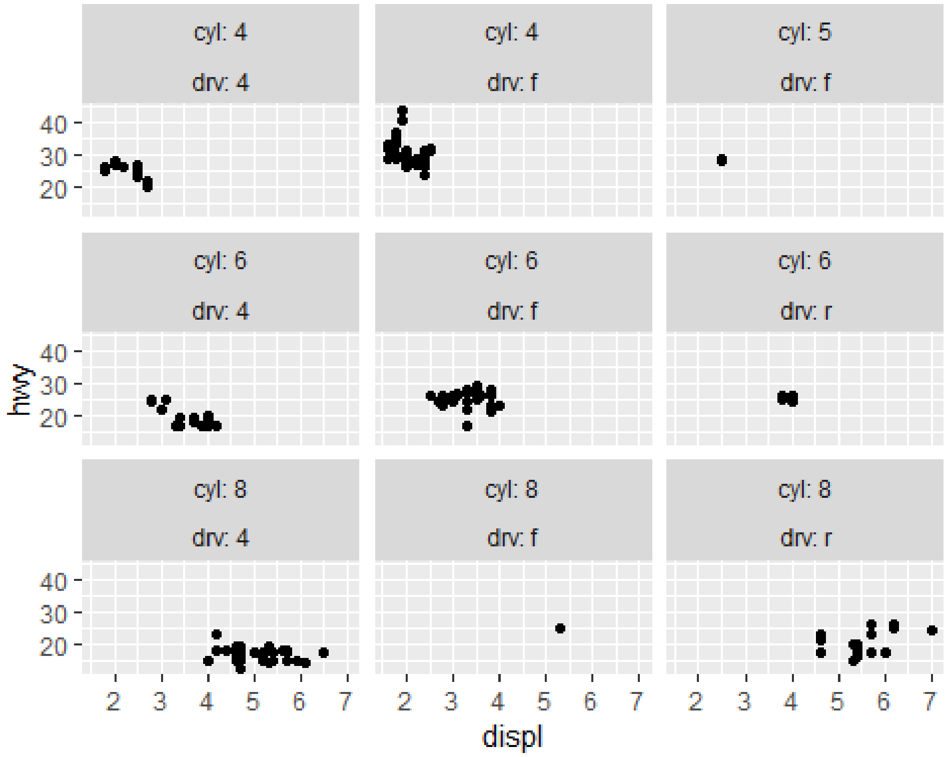
ncol = controls the number of columns for the faceting variables

Why doesn't facet_grid() have nrow and ncol arguments? facet_grid() forms a matrix of panels defined by row and column faceting variables

Other options that control output of facet_wrap? vars, labeller, scales

```
?facet_wrap
```

```
ggplot(mpg, aes(displ, hwy))+
  geom_point()+
  facet_wrap(c("cyl", "drv"), labeller = "label_both")
```



When using facet_grid() you should usually put the variable with more unique levels in the columns. Why? It is easier to read columnwise than rowwise. If the more unique levels were in rows, you would probably have to tilt your head to check comparisons.

3.6.1 Exercises

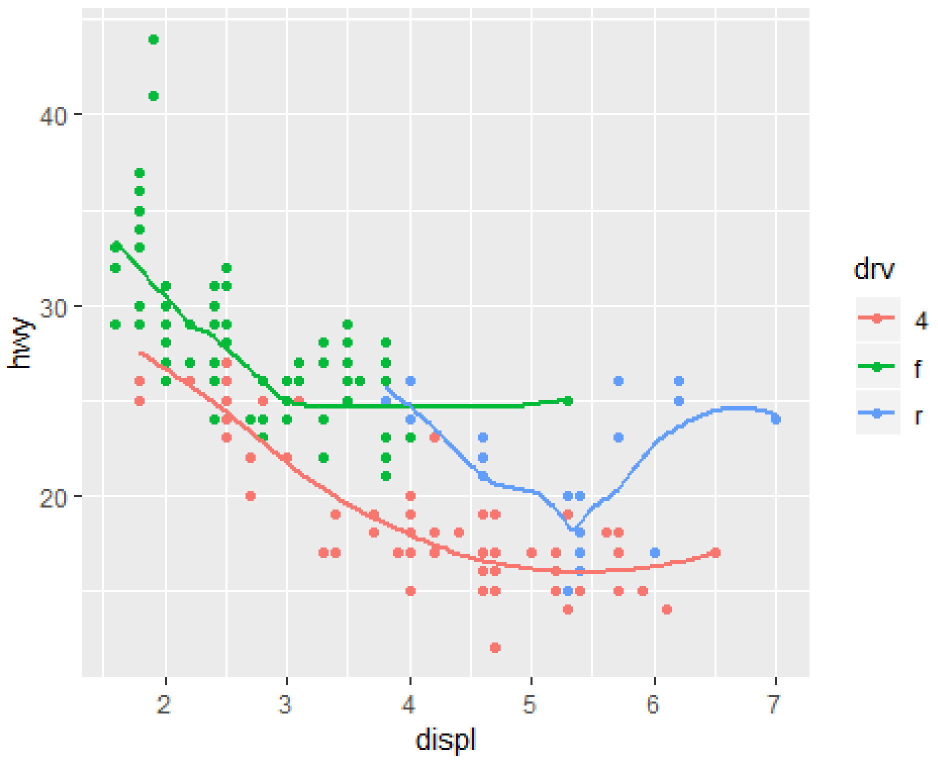
geom_line - A line chart

geom_boxplot - A boxplot

geom_histogram - A histogram

geom_area - Area chart

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

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



show.legend = FALSE doesn’t include the legend. when you remove it, R assigns a default legend to your plot based on what you are classifying, and if any aesthetics are mapped.

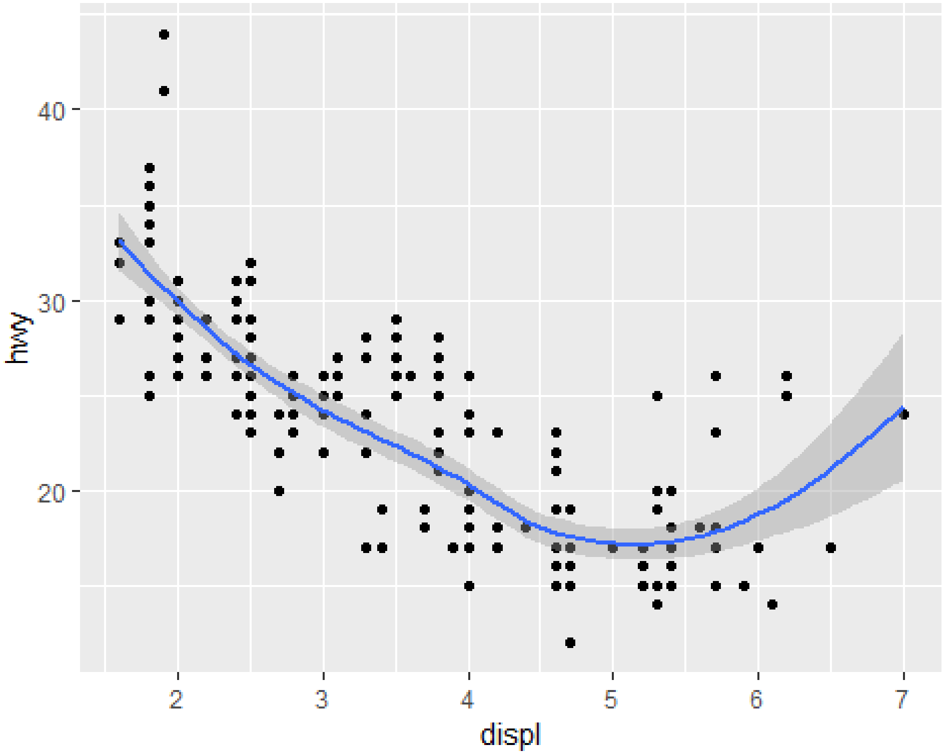
se(standard error) - Displays confidence interval around smooth

The below two codes will give similar graphs;Why?

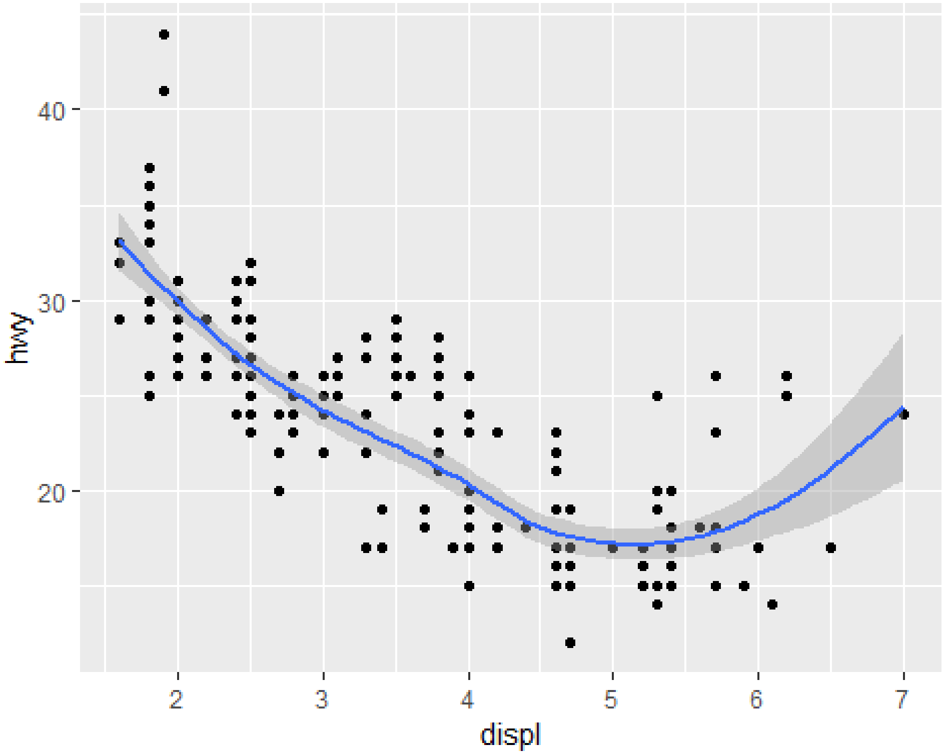
The first code - treats mappings as global mappings that apply to each geom in the graph.

Second code - duplication of code passed on to each geom. These mapping are treated as local mappings for that specific layer only.

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

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

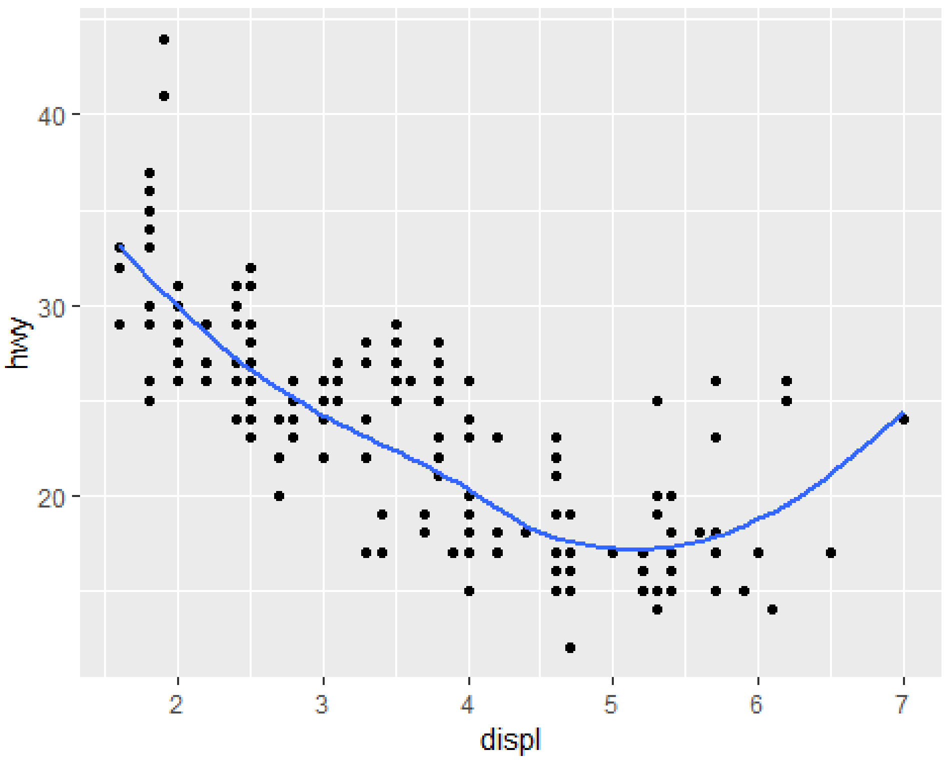


```
ggplot()+
  geom_point(data = mpg, mapping = aes(x = displ, y = hwy))+
  geom_smooth(data = mpg, mapping = aes(x = displ, y = hwy))

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

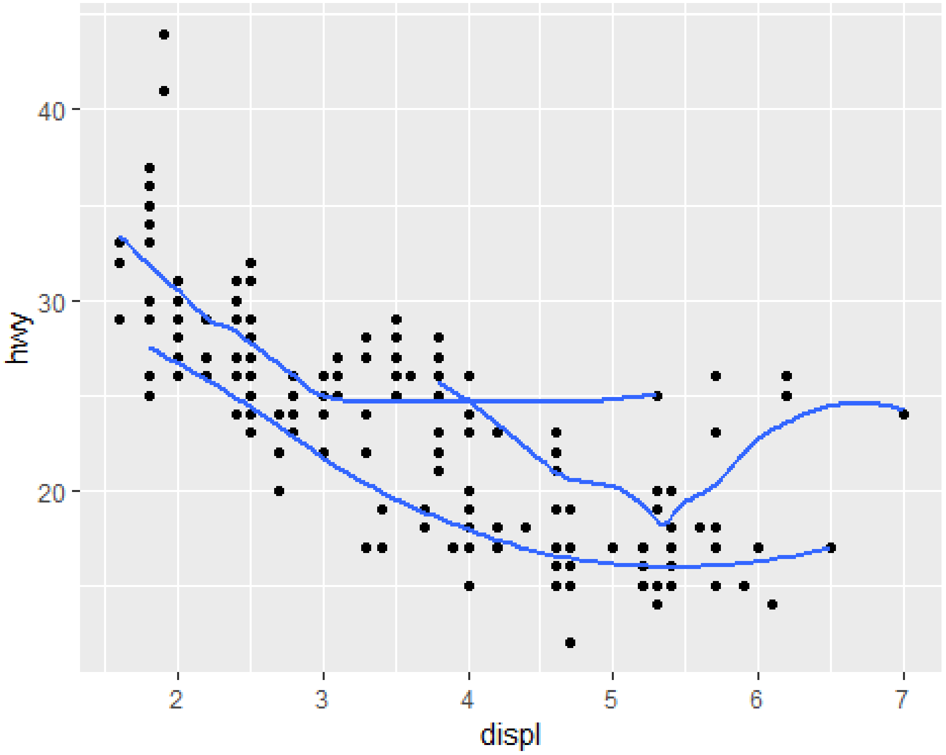


Recreating the necessary R codes;

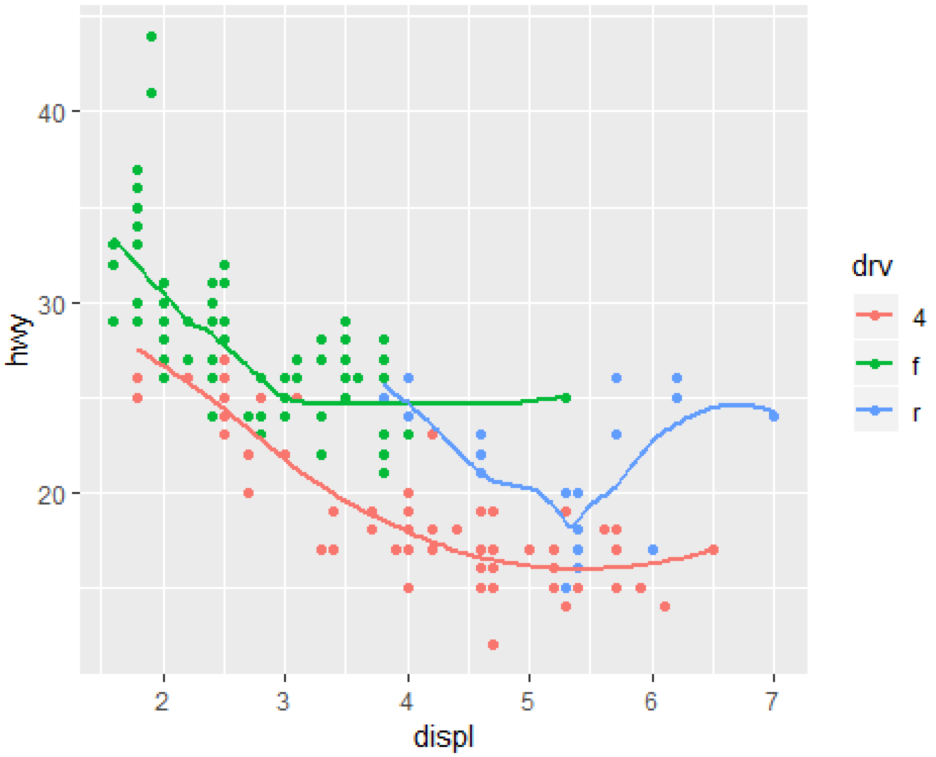
```
ggplot(mpg, aes(displ, hwy))+
  geom_point()+
  geom_smooth(se = FALSE)

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



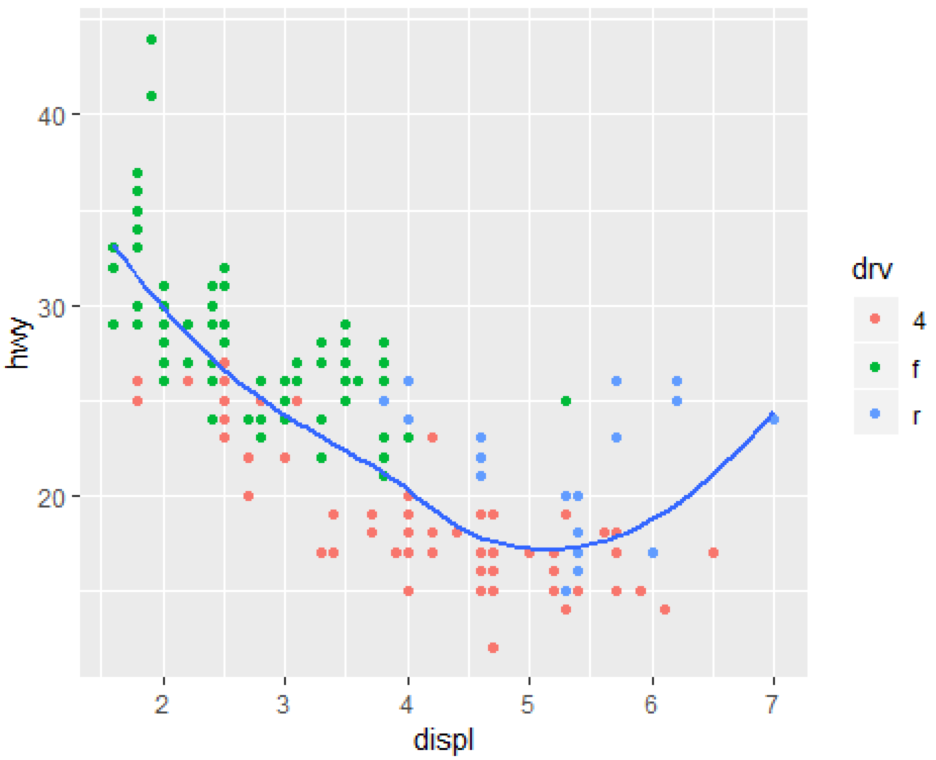
```
ggplot(mpg, aes(displ, hwy))+
  geom_point()+
  geom_smooth(aes(group = drv), se = FALSE)

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



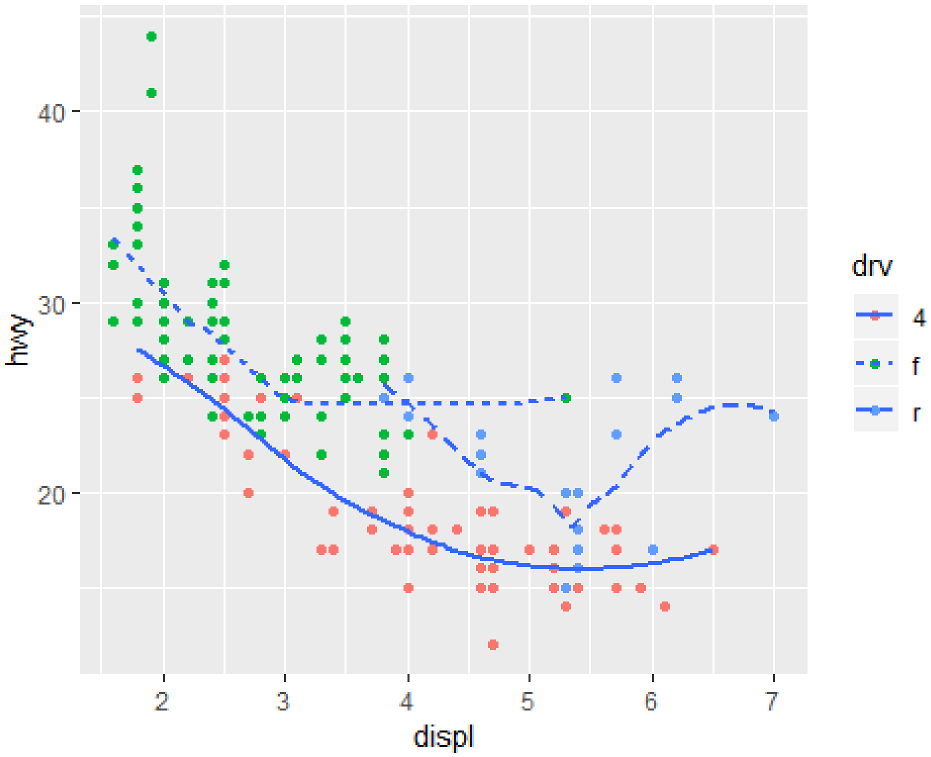
```
ggplot(mpg, aes(displ, hwy, color = drv))+
  geom_point()+
  geom_smooth(se = FALSE)

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



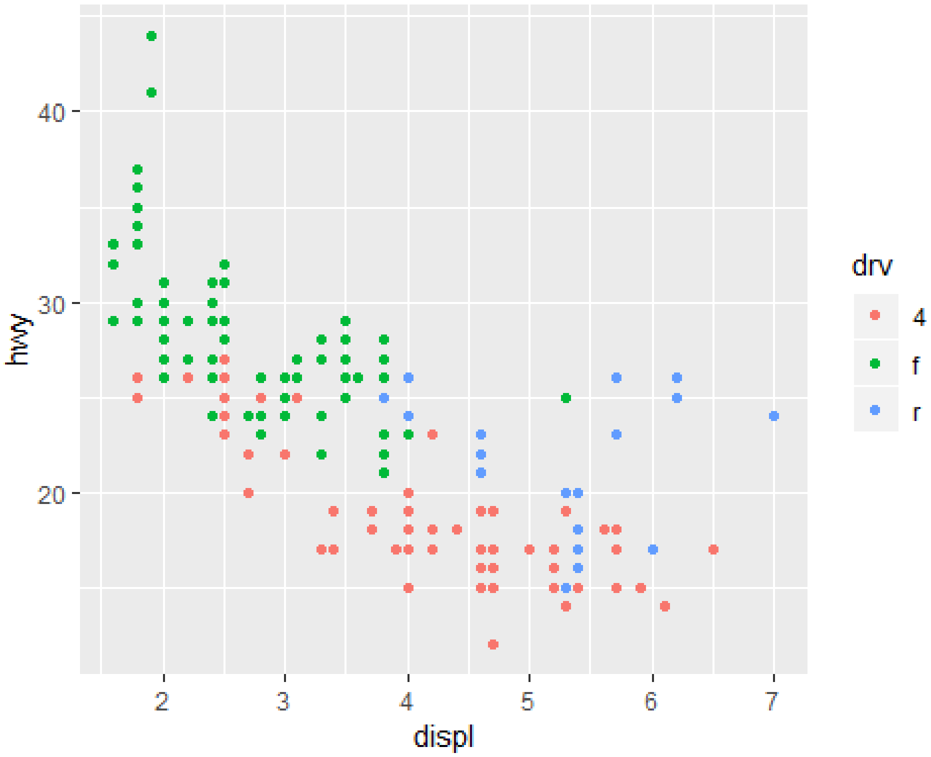
```
ggplot(mpg, aes(displ, hwy))+
  geom_point(aes(color = drv))+
  geom_smooth(se = FALSE)

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



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

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



```
ggplot(mpg, aes(displ, hwy))+
  geom_point(aes(color = drv))
```



Statistical Transformations

3.7.1 Exercises

geom_col() - heights of the bars represent values in the data

geom_bar() - makes the height of the bar proportional to the number of cases in each group

geom_bar() uses stat_count()

geom_col() uses stat_identity(): it leaves the data as is

stat_smooth() - Aids the eye in seeing patterns in the presence of overplotting. Similar to geom_smooth()

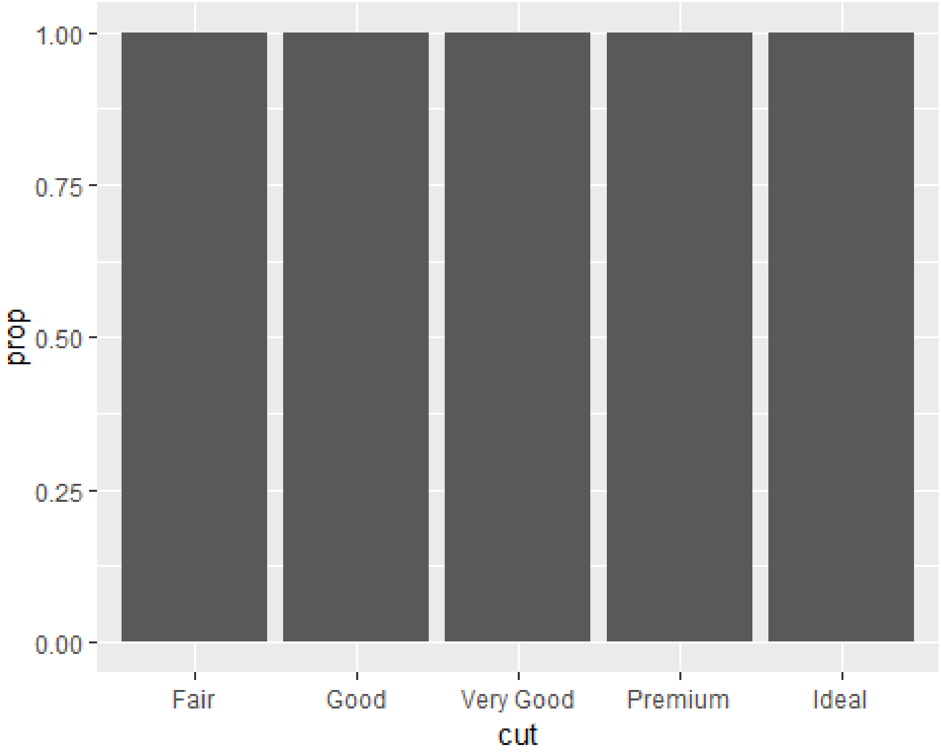
stat_summary() is similar to geom_point for discrete variables

What parameters control its behaviour? (x, y; mapping),se, formula(y~x)

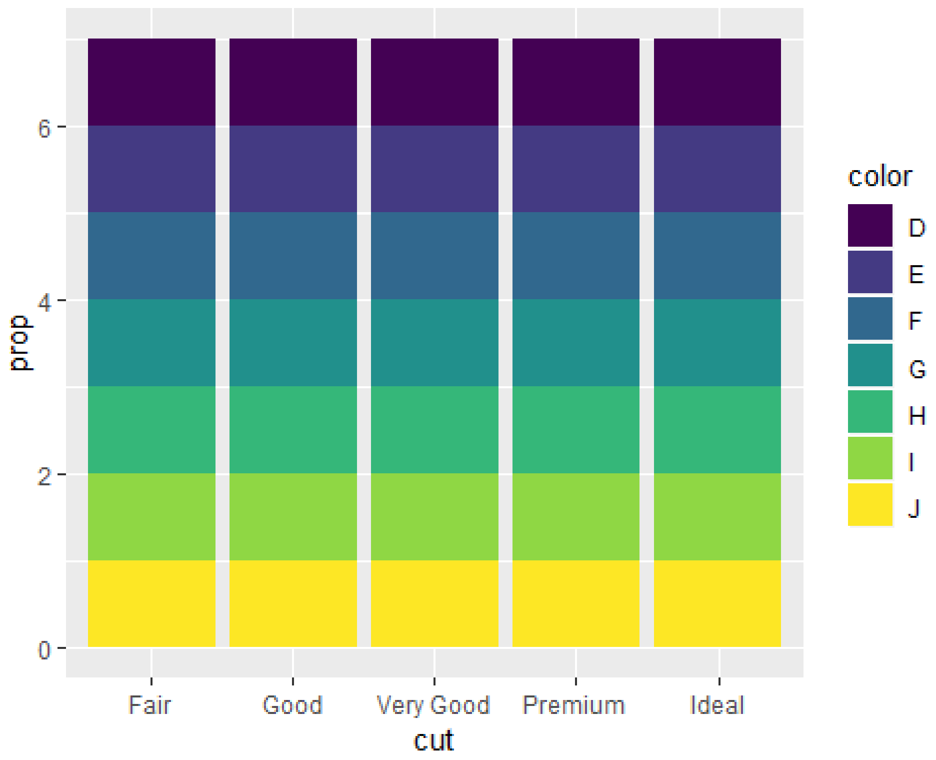
Why do we set group = 1?

What is wrong with these two graphs?

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

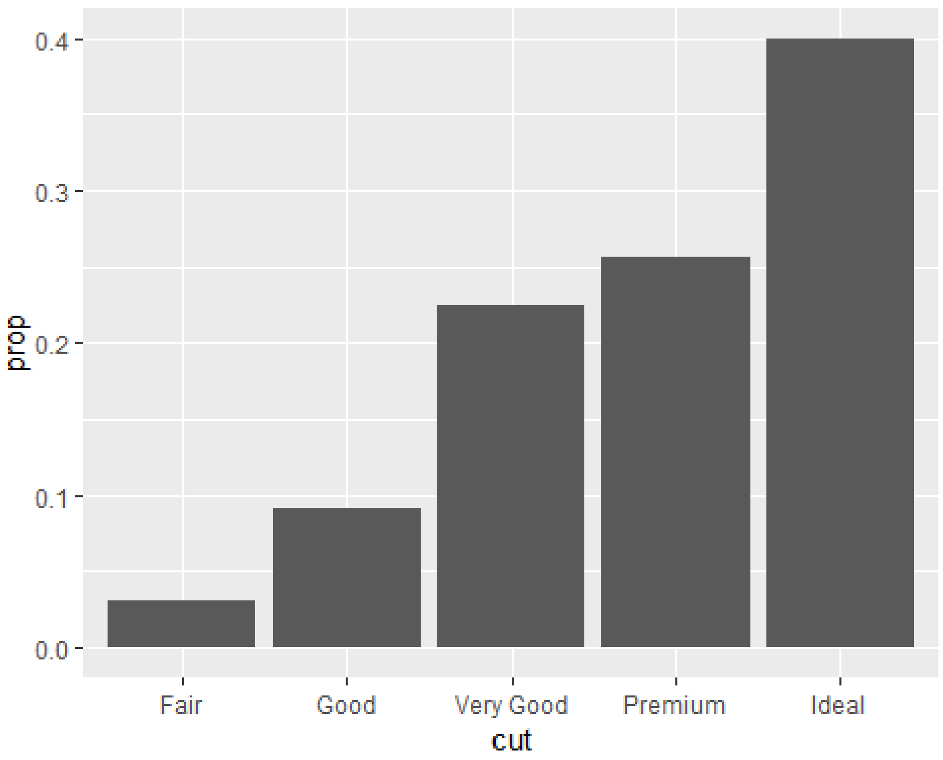


```
ggplot(data = diamonds)+
  geom_bar(mapping = aes(x = cut, fill =color, y = ..prop..))
```

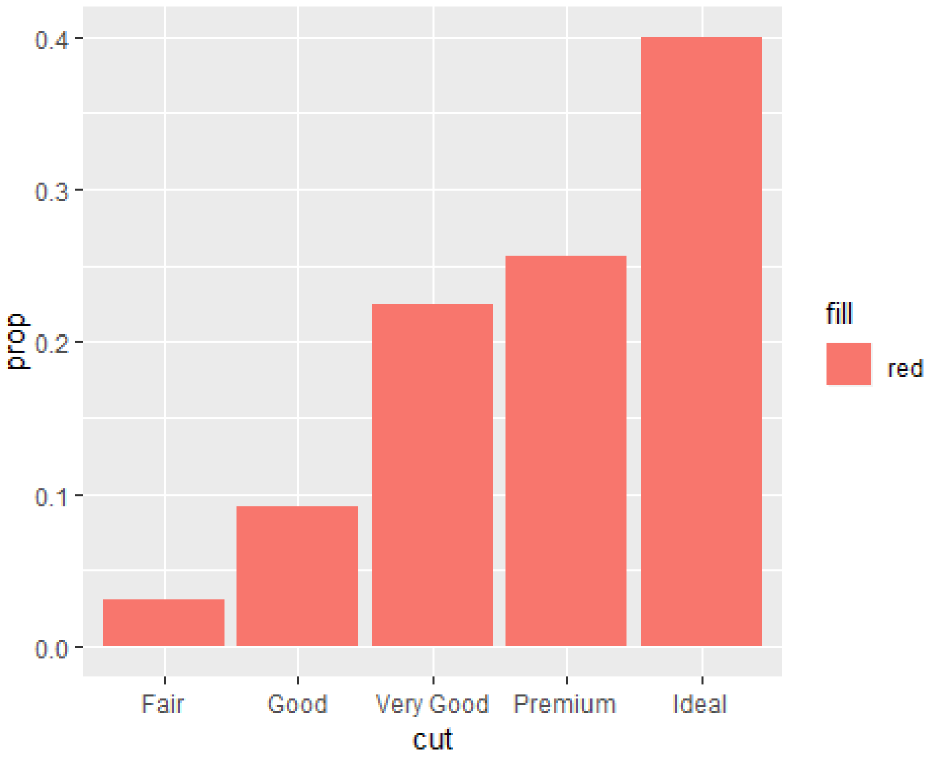


Edit the above graphs with group = 1

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



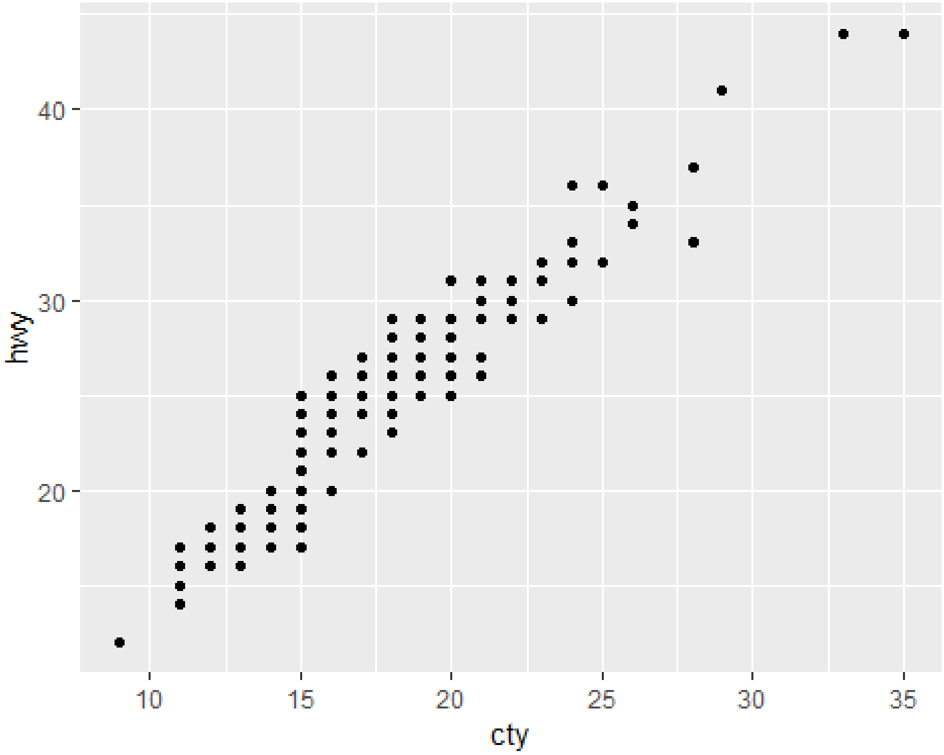
```r
ggplot(data = diamonds)+
  geom_bar(mapping = aes(x = cut, fill= "red", y = ..prop.., group = 1))
```



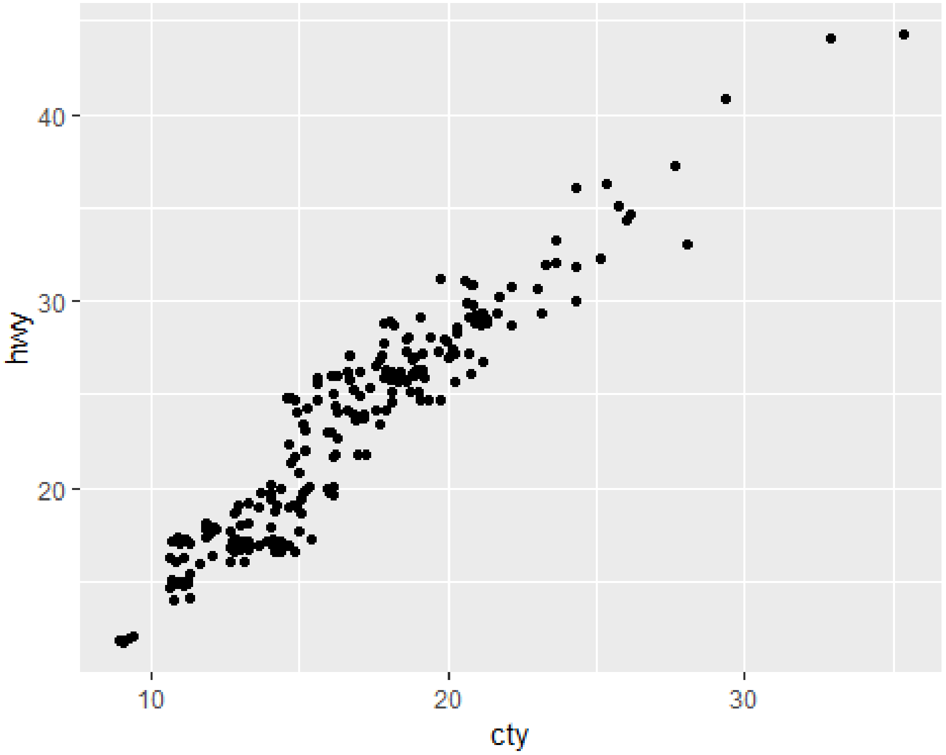
3.8.1 Exercises

Problem with the below plot? Solution?

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



```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy))+
  geom_point(position = "jitter")
```

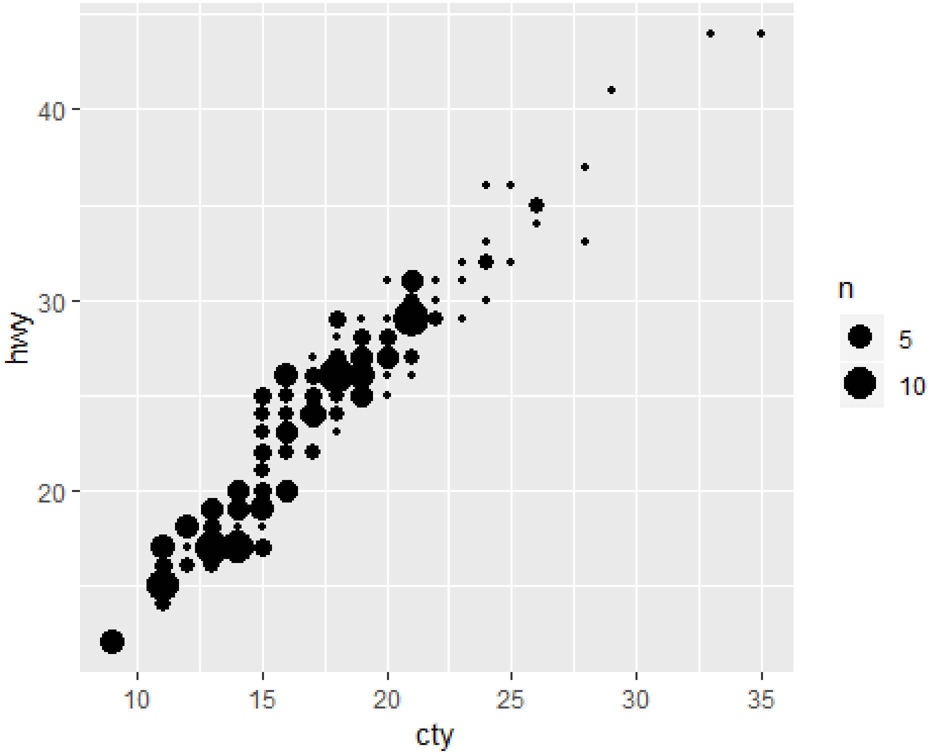


Parameters that control the amount of jittering? mapping

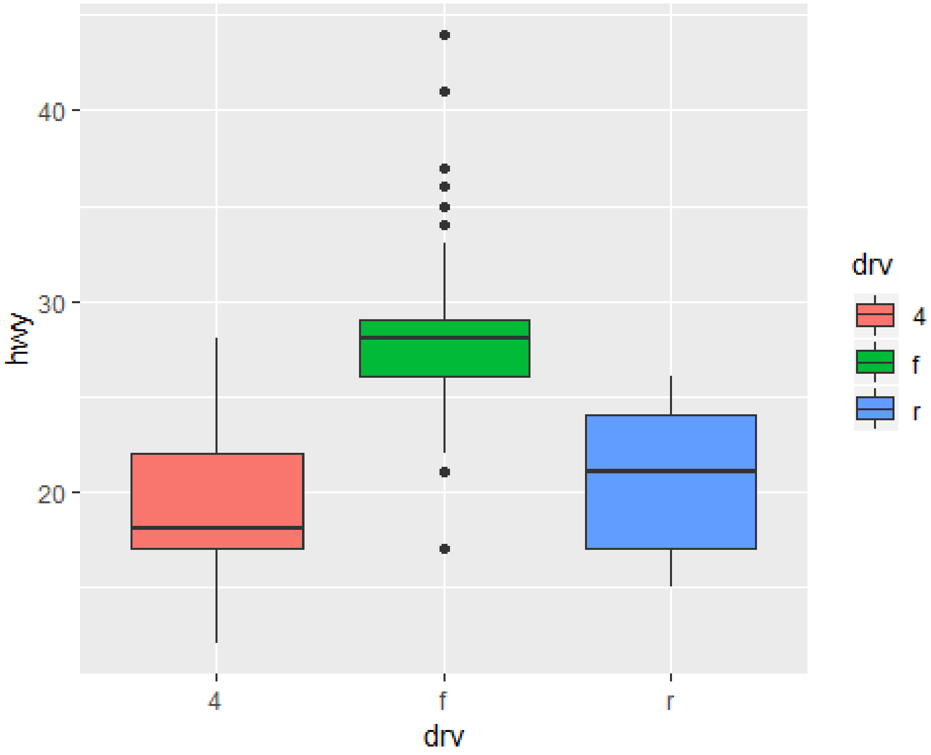
geom_count() - counts the number of observations at each location,then maps the count to point area. It useful when you have discrete data and overplotting.

geom_jitter() - random noise

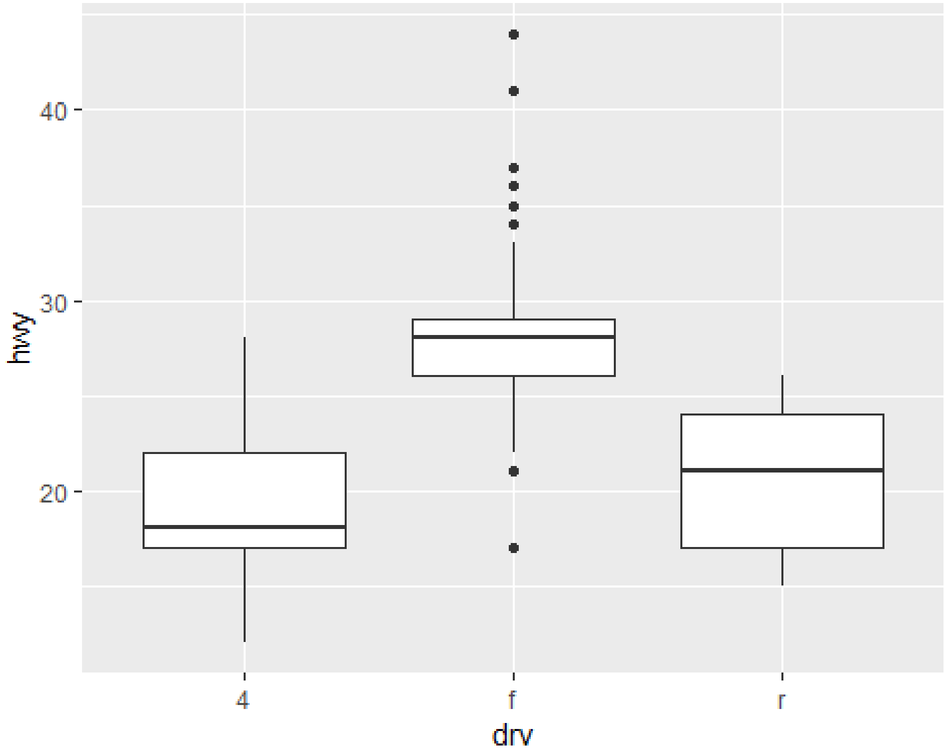
```
ggplot(mpg, aes(cty, hwy))+
  geom_count()
```



```r
ggplot(mpg, aes(x = drv, y = hwy, fill = drv))+
  geom_boxplot()
```



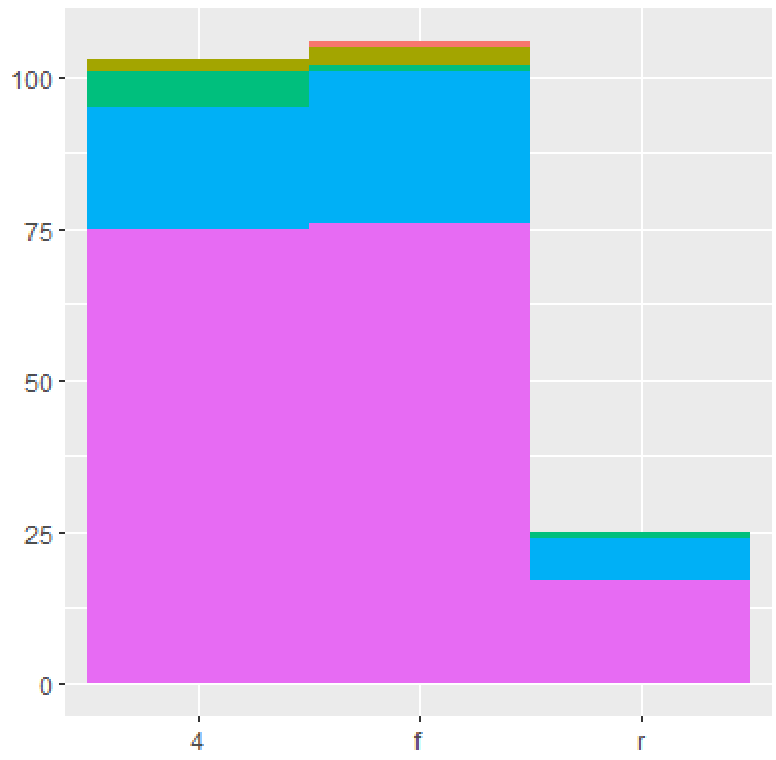
```r
ggplot(mpg, aes(drv, hwy))+
  stat_boxplot()
```



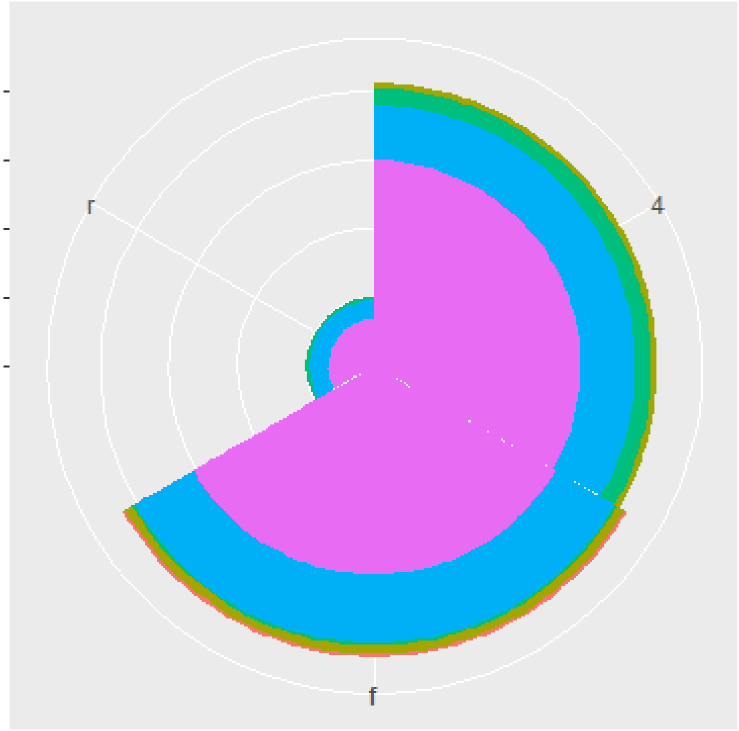
3.9.1 Exercises

Turn a stacked bar chart into a pie chart using coord_polar()

```r
stbar<-ggplot(data = mpg)+
  geom_bar(aes(x = drv, fill = fl),
           show.legend = FALSE,
           width = 1)+
  theme(aspect.ratio = 1)+
  labs(x = NULL, y = NULL)
stbar
```



```
stbar+coord_polar()
```



coord_map() - projects a portion of the earth, which is approximately spherical coord_quickmap() - a quick approximation that does preserve straight lines

There is a positive relationship between city miles per gallon and the fuel efficiency.

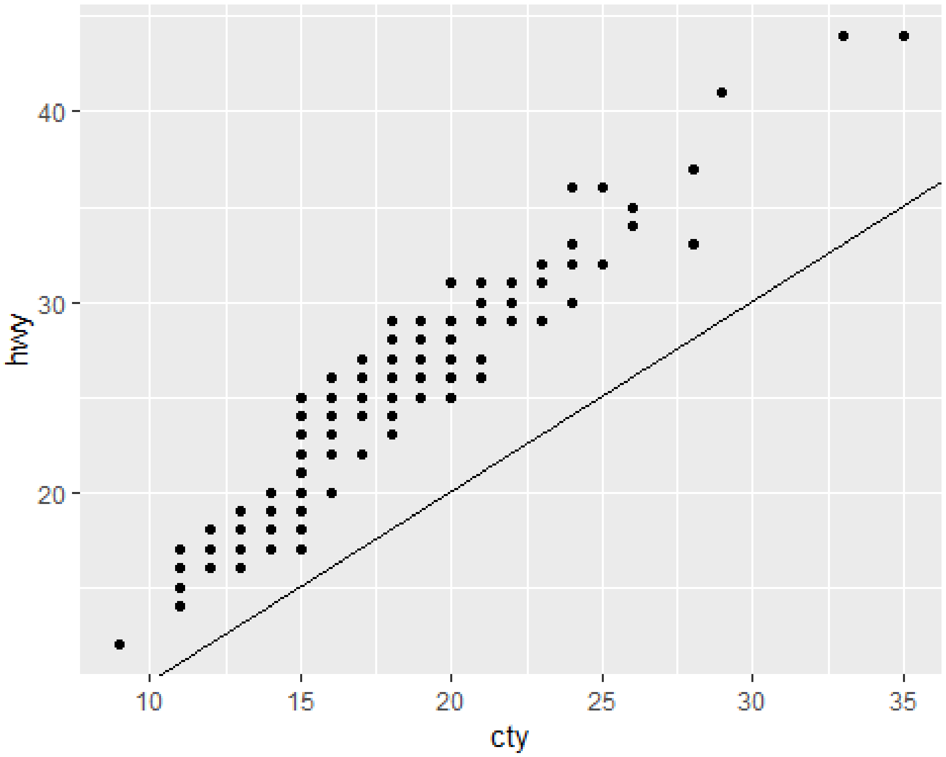
NOTICE: coord_fixed() - forces a specified ratio between the physical representation of data units on the axes.

The default, ratio = 1, ensures that one unit on the x-axis is the same length as one unit on the y-axis.

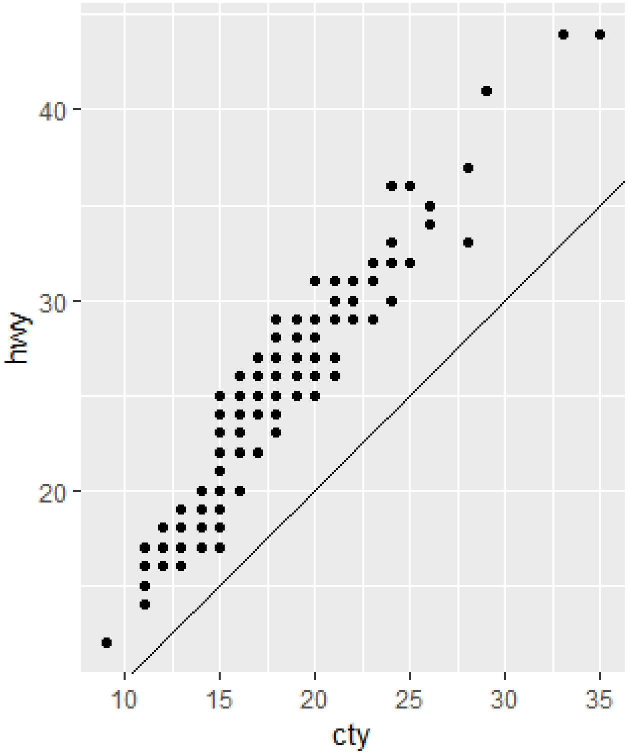
Ratios higher than one make units on the y axis longer than units on the x-axis, and vice versa.

geom_abline(): slope and intercept

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



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


40
30
20
hwy
10
15
20
25
30
35
cty