Worksheet-6 in R

Drake Francis M. Jaculina

2022-11-25

Use the dataset mpg

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v tibble  3.1.8     v purrr   0.3.5
## v tidyr   1.2.1     v stringr 1.4.1
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
data(mpg)
as.data.frame(data(mpg))
```

```
##   data(mpg)
## 1       mpg
```

```
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

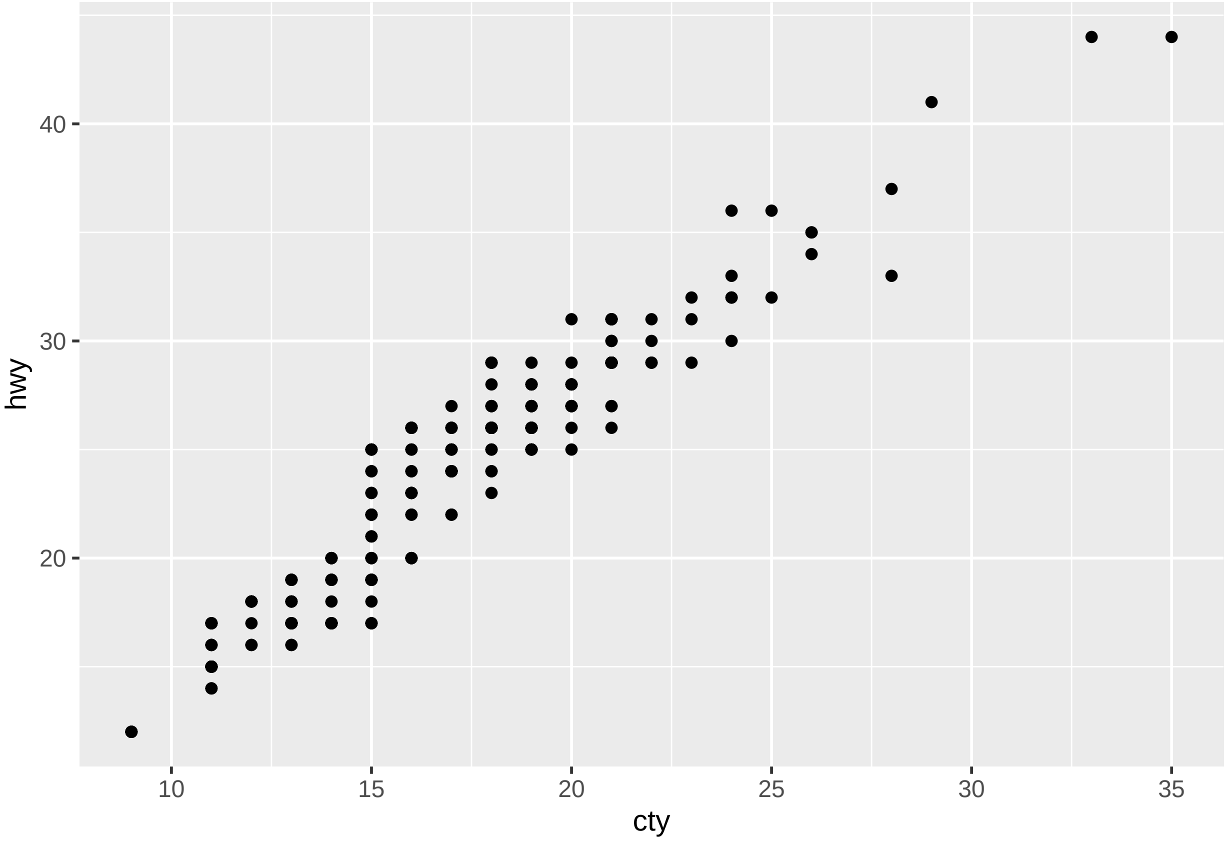
Use of glimpse() - much tidier compared to str()

```
glimpse(mpg)
```

```
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "~
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "~
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.~
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200~
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, ~
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto~
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4~
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1~
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2~
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p~
## $ class        <chr> "compact", "compact", "compact", "compact", "compact", "c~
```

Example. graph using ggplot()

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



1. How many columns are in mpg data set? How about the number of rows? Show the codes and its result.

```
#Number of columns in mpg data set
mpg_col <- ncol(mpg)
mpg_col
```

```
## [1] 11
```

```
#Number of rows in mpg data set
mpg_row <- nrow(mpg)
mpg_row
```

```
## [1] 234
```

```
#Ans: There are 11 columns and 234 rows in the mpg data set.
```

2. Which manufacturer has the most models in this data set? Which model has the most variations?

```
manuf_count <- mpg %>% group_by(manufacturer) %>% tally (sort = TRUE)
manuf_count
```

```
## # A tibble: 15 x 2
##    manufacturer     n
##    <chr>        <int>
##  1 dodge           37
##  2 toyota          34
##  3 volkswagen      27
##  4 ford            25
##  5 chevrolet       19
##  6 audi            18
##  7 hyundai         14
##  8 subaru          14
##  9 nissan          13
## 10 honda            9
## 11 jeep             8
## 12 pontiac          5
## 13 land rover       4
## 14 mercury          4
## 15 lincoln          3
```

```
colnames(manuf_count)<-c("Manufacturer", "Counts")

#Ans: dodge is the manufacturer has the most models in this data set which has 37 models.
```

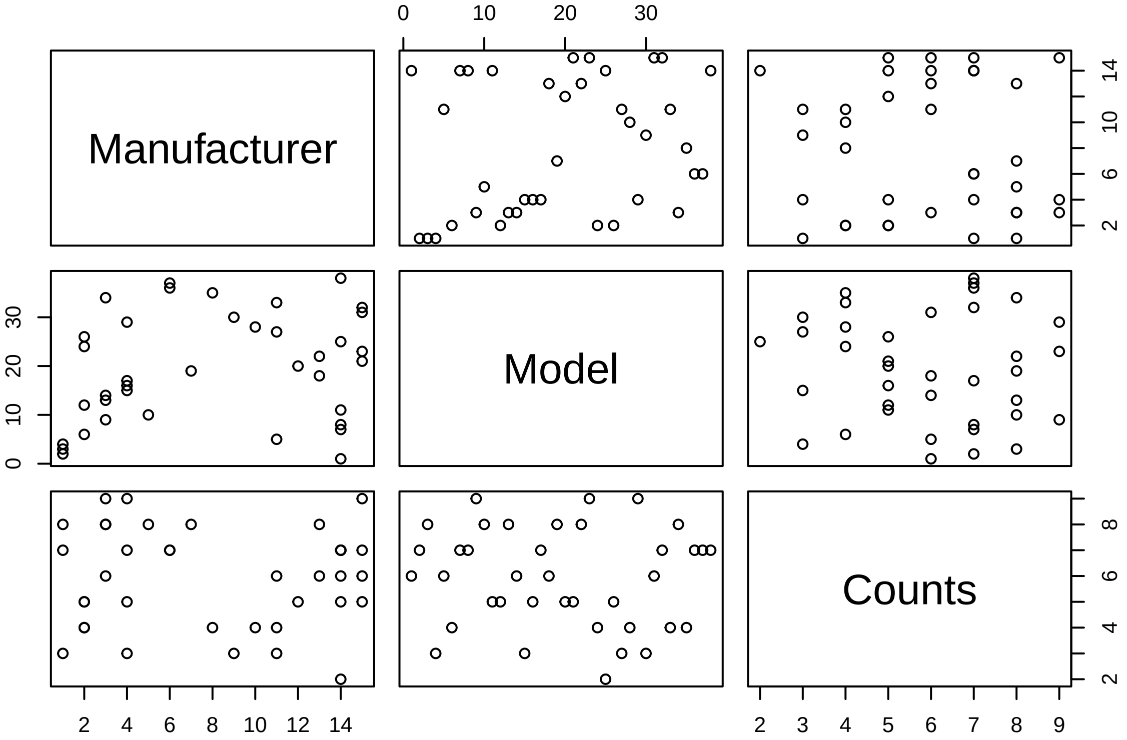
a. Group the manufacturers and find the unique models. Copy the codes and result.

```
unique_manuf <- mpg %>% group_by(manufacturer, model) %>%
  distinct() %>% count()
colnames(unique_manuf) <- c("Manufacturer", "Model","Counts")
unique_manuf
```

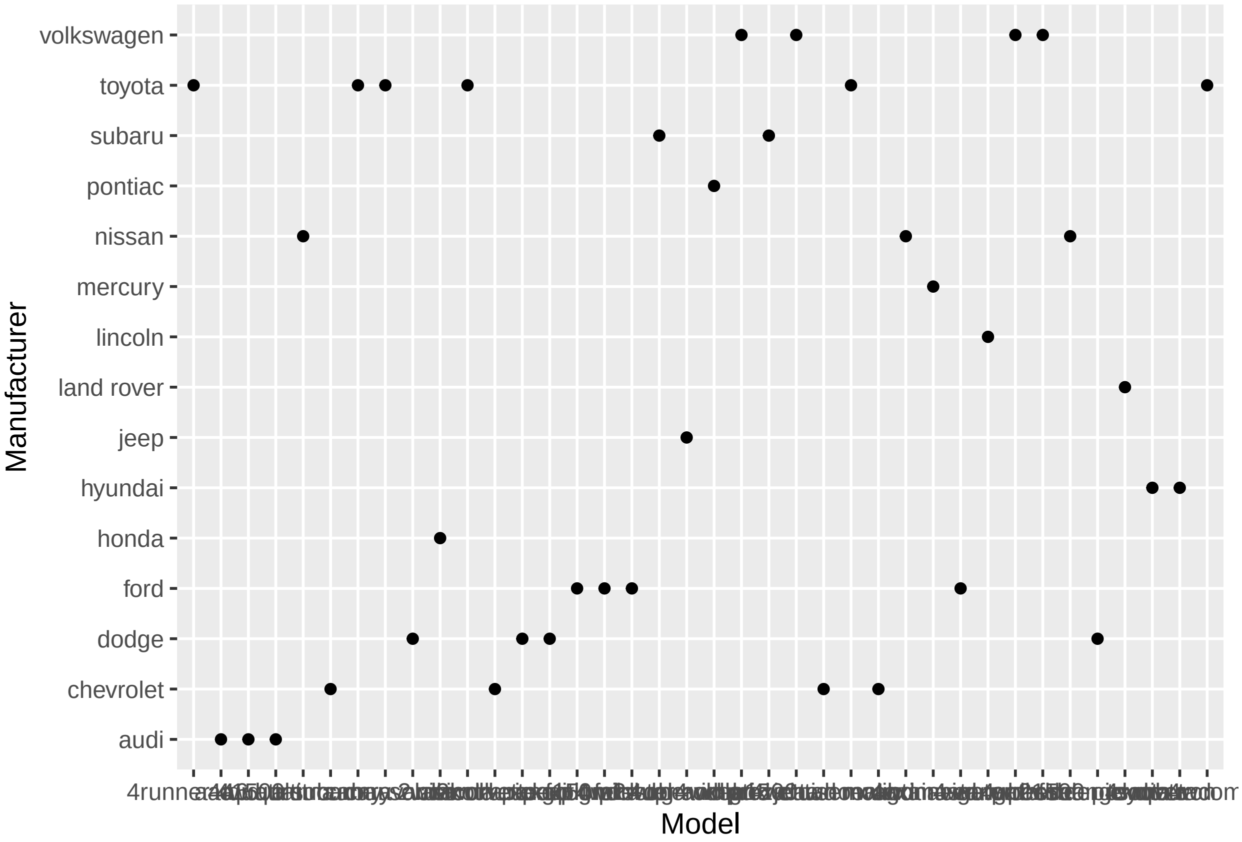
```
## # A tibble: 38 x 3
## # Groups:   Manufacturer, Model [38]
##    Manufacturer Model              Counts
##    <chr>        <chr>                <int>
##  1 audi         a4                       7
##  2 audi         a4 quattro               8
##  3 audi         a6 quattro               3
##  4 chevrolet    c1500 suburban 2wd       4
##  5 chevrolet    corvette                 5
##  6 chevrolet    k1500 tahoe 4wd          4
##  7 chevrolet    malibu                   5
##  8 dodge        caravan 2wd              9
##  9 dodge        dakota pickup 4wd        8
## 10 dodge        durango 4wd              6
## # ... with 28 more rows
```

b. Graph the result by using plot() and ggplot(). Write the codes and its result.

```
#plot()
plot(unique_manuf)
```



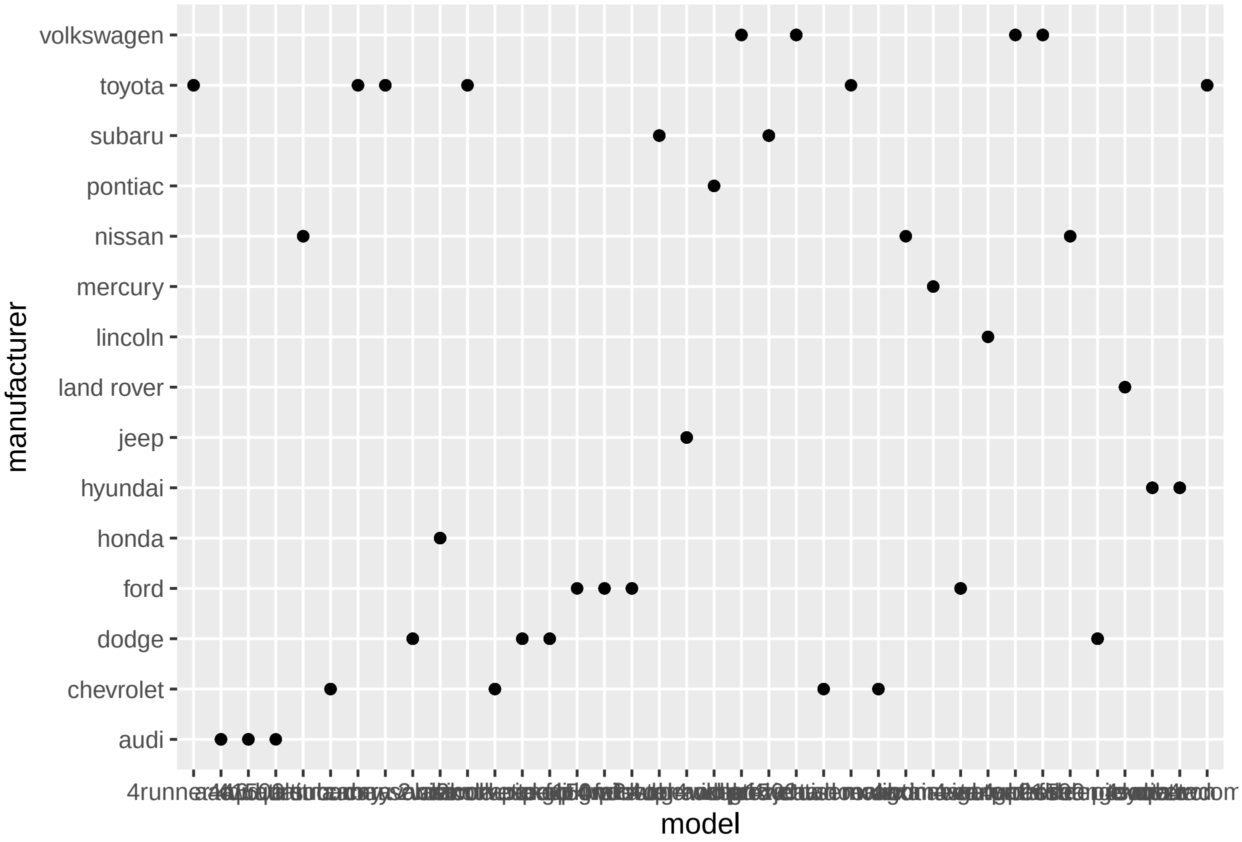
```
#ggplot()
ggplot(unique_manuf, aes(Model, Manufacturer)) + geom_point()
```



3. Same dataset will be used. You are going to show the relationship of the model and the manufacturer.

a. What does ggplot(mpg, aes(model, manufacturer)) + geom_point() show?

```
ggplot(mpg, aes(model, manufacturer)) + geom_point()
```



b. For you, is it useful? If not, how could you modify the data to make it more informative?

```
#Ans: It is not useful, It's already informative but you can't see the ranking or the hierarchy of the
```

4. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its result.

```
group_model <- unique_manuf %>% group_by(Model) %>% count()
group_model
```

```
## # A tibble: 38 x 2
## # Groups:   Model [38]
##    Model                  n
##    <chr>              <int>
##  1 4runner 4wd            1
##  2 a4                     1
##  3 a4 quattro             1
##  4 a6 quattro             1
##  5 altima                 1
##  6 c1500 suburban 2wd     1
##  7 camry                  1
##  8 camry solara           1
##  9 caravan 2wd            1
## 10 civic                  1
## # ... with 28 more rows
```

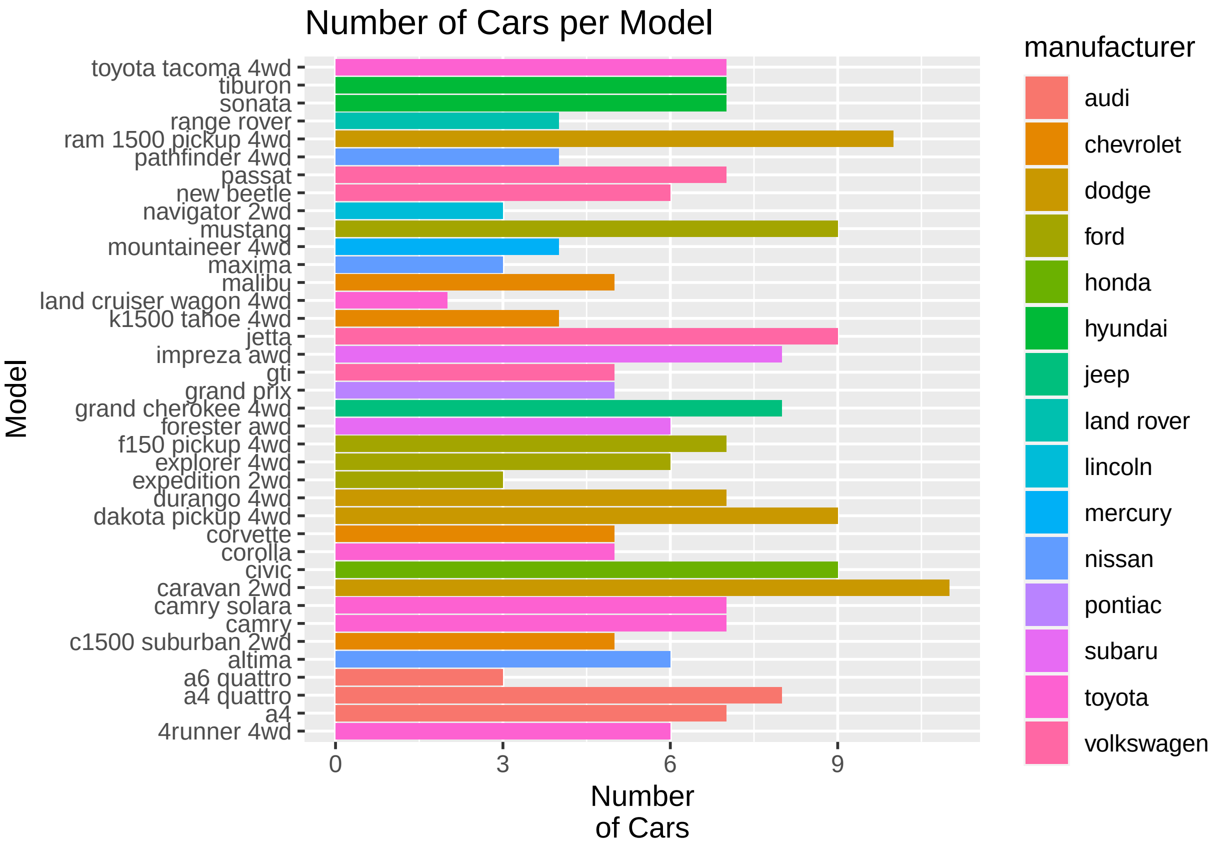
```
colnames(group_model) <- c("Model","Counts")
group_model
```

```
## # A tibble: 38 x 2
## # Groups:   Model [38]
##    Model                Counts
##    <chr>                 <int>
##  1 4runner 4wd               1
##  2 a4                        1
##  3 a4 quattro                1
##  4 a6 quattro                1
##  5 altima                    1
##  6 c1500 suburban 2wd        1
##  7 camry                     1
##  8 camry solara              1
##  9 caravan 2wd               1
## 10 civic                     1
## # ... with 28 more rows
```

a. Plot using the geom_bar() + coord_flip() just like what is shown below. Show codes and its result.

```
qplot(model,data = mpg,main = "Number of Cars per Model", xlab = "Model",ylab = "Number
of Cars", geom = "bar", fill = manufacturer) + coord_flip()
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```



b. Use only the top 20 observations. Show code and results.

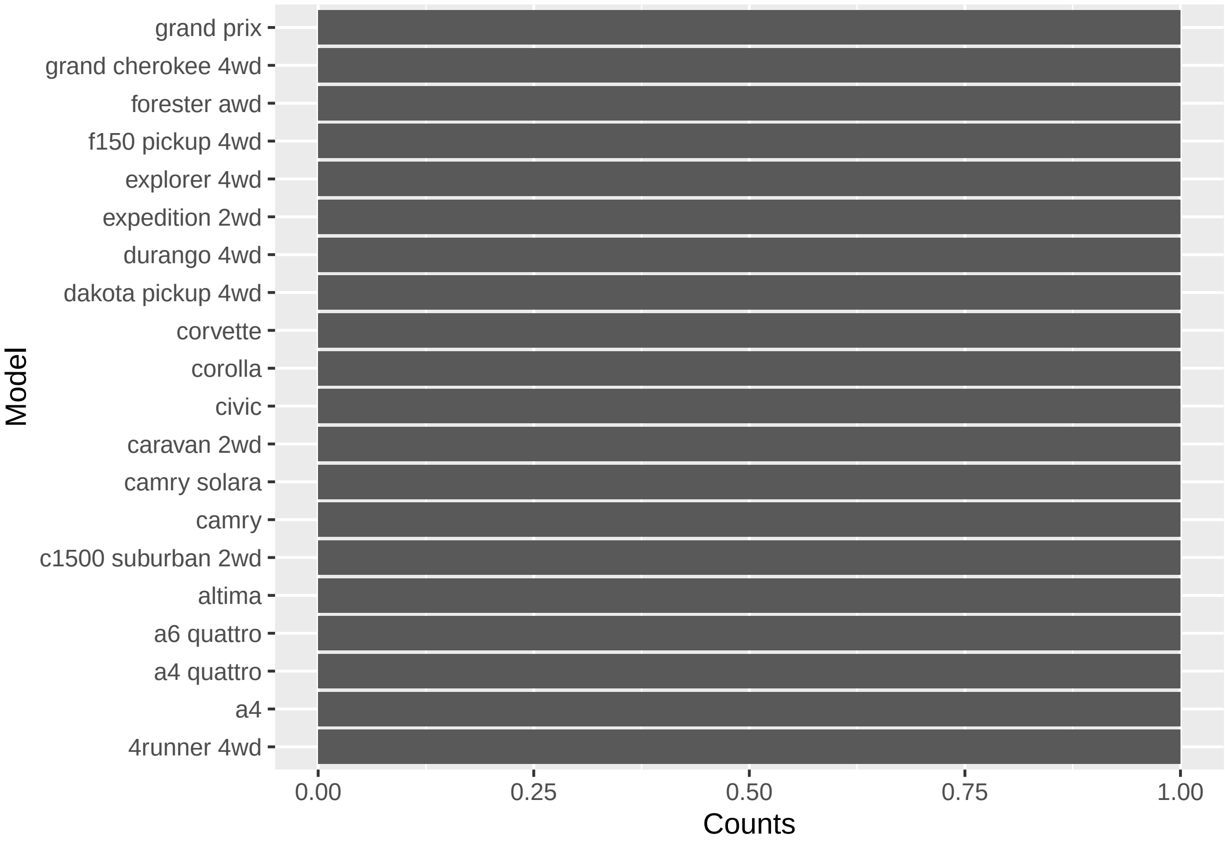
```
twenty_obser <- group_model[1:20,] %>% top_n(2)
```

```
## Selecting by Counts
```

```
twenty_obser
```

```
## # A tibble: 20 x 2
## # Groups:   Model [20]
##    Model              Counts
##    <chr>               <int>
##  1 4runner 4wd             1
##  2 a4                      1
##  3 a4 quattro              1
##  4 a6 quattro              1
##  5 altima                  1
##  6 c1500 suburban 2wd      1
##  7 camry                   1
##  8 camry solara            1
##  9 caravan 2wd             1
## 10 civic                   1
## 11 corolla                 1
## 12 corvette                1
## 13 dakota pickup 4wd       1
## 14 durango 4wd             1
## 15 expedition 2wd          1
## 16 explorer 4wd            1
## 17 f150 pickup 4wd         1
## 18 forester awd            1
## 19 grand cherokee 4wd      1
## 20 grand prix              1
```

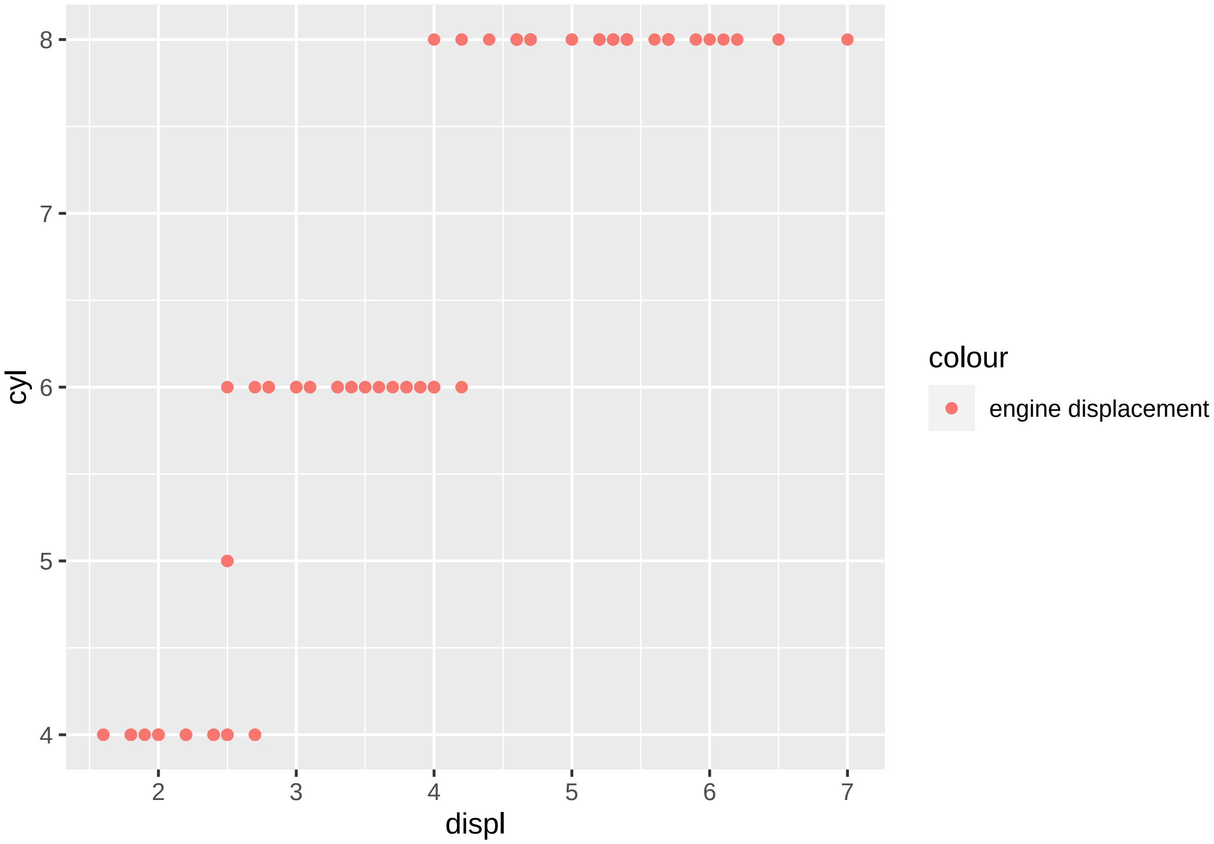
```
#Using ggplot()
ggplot(twenty_obser,aes(x = Model, y = Counts)) + geom_bar(stat = "Identity") +coord_flip()
```



5. Plot the relationship between cyl - number of cylinders and displ - engine displace-ment using geom_point with aesthetic colour = engine displacement. Title should be "Relationship between No. of Cylinders and Engine Displacement".

a. Show the codes and its result.

```
ggplot(data = mpg, mapping = aes(x = displ, y = cyl, main = "Relationship between No. of Cylinders and
  geom_point(mapping=aes(colour = "engine displacement"))
```



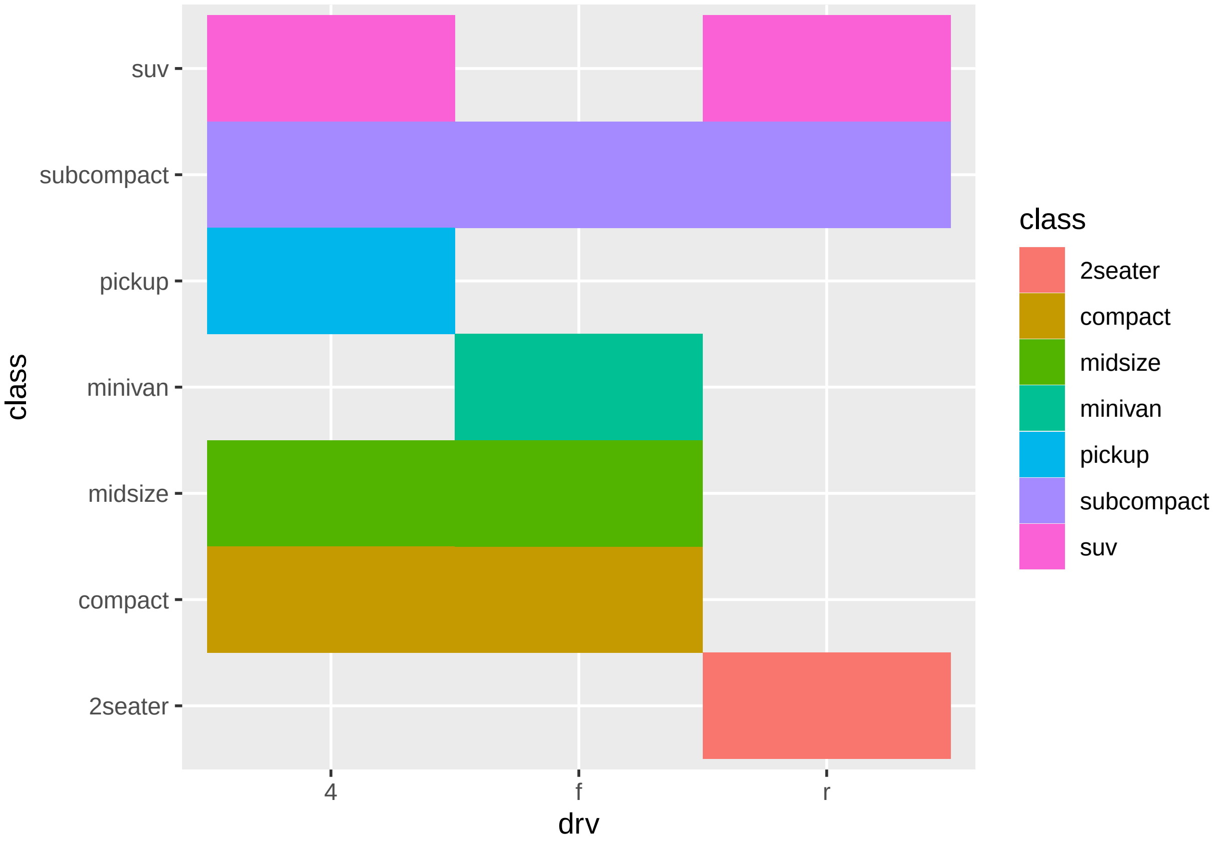
b. How would you describe its relationship?

```
#Ans: We can see that the the scatter plot it has 3 clustered data, the more number of cylinder increas
```

6. Get the total number of observations for drv - type of drive train (f = front-wheel drive, r = rear wheel drive, 4 = 4wd) and class - type of class (Example: suv, 2seater, etc.). Plot using the geom_tile() where the number of observations for class be used as a fill for aesthetics.

a. Show the codes and its result for the narrative in #6.

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

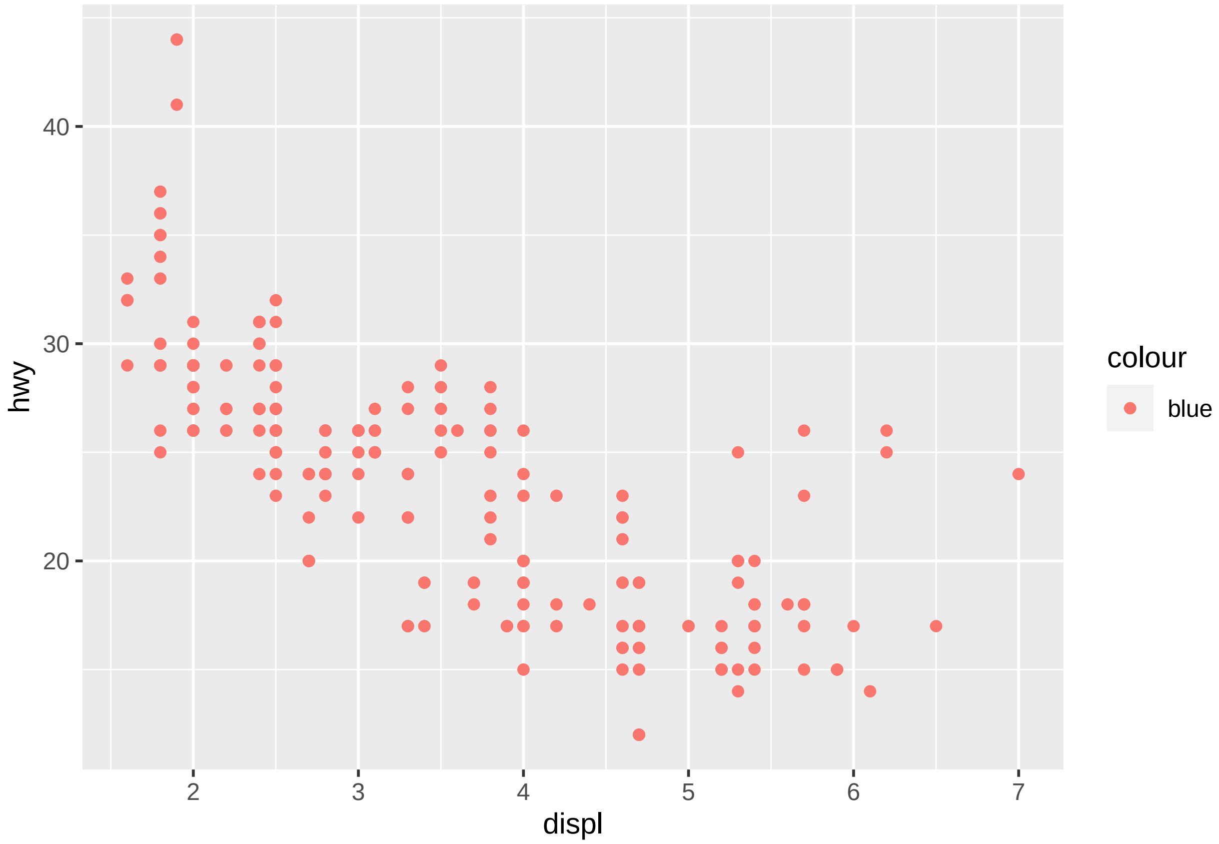


b. Interpret the result.

```
#Ans: The result of given data is that it shows the class and drv. subcompact has all the drv which inc
```

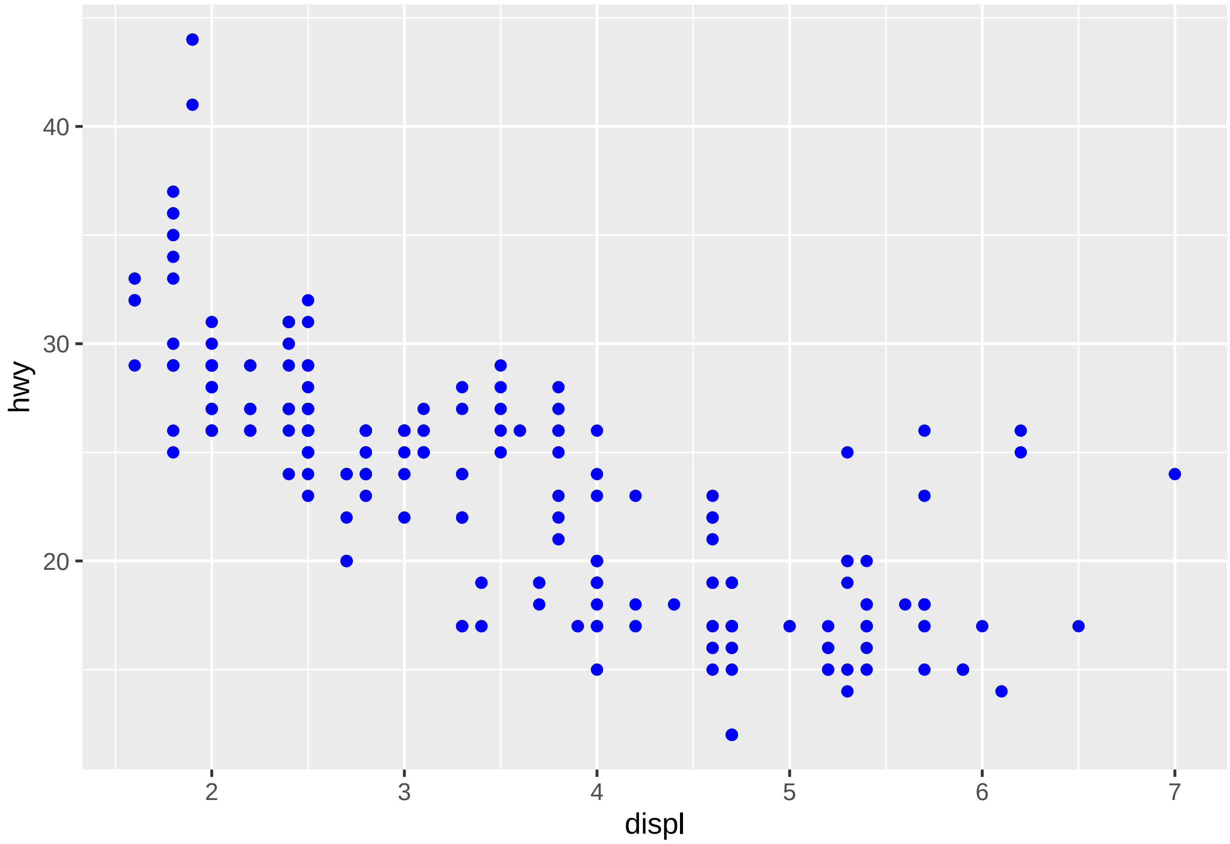
7. Discuss the difference between these codes. Its outputs for each are shown below.

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



Code #1

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



Code #2

```
#Ans: The given codes are mostly the same but it has slight differences. The code #1, colour = "blue" i
```

8. Try to run the command ?mpg. What is the result of this command?

```
?mpg
#Ans: The result of this command it will go to the Help pane and it will allow us to see or access the
```

a. Which variables from mpg dataset are categorical?

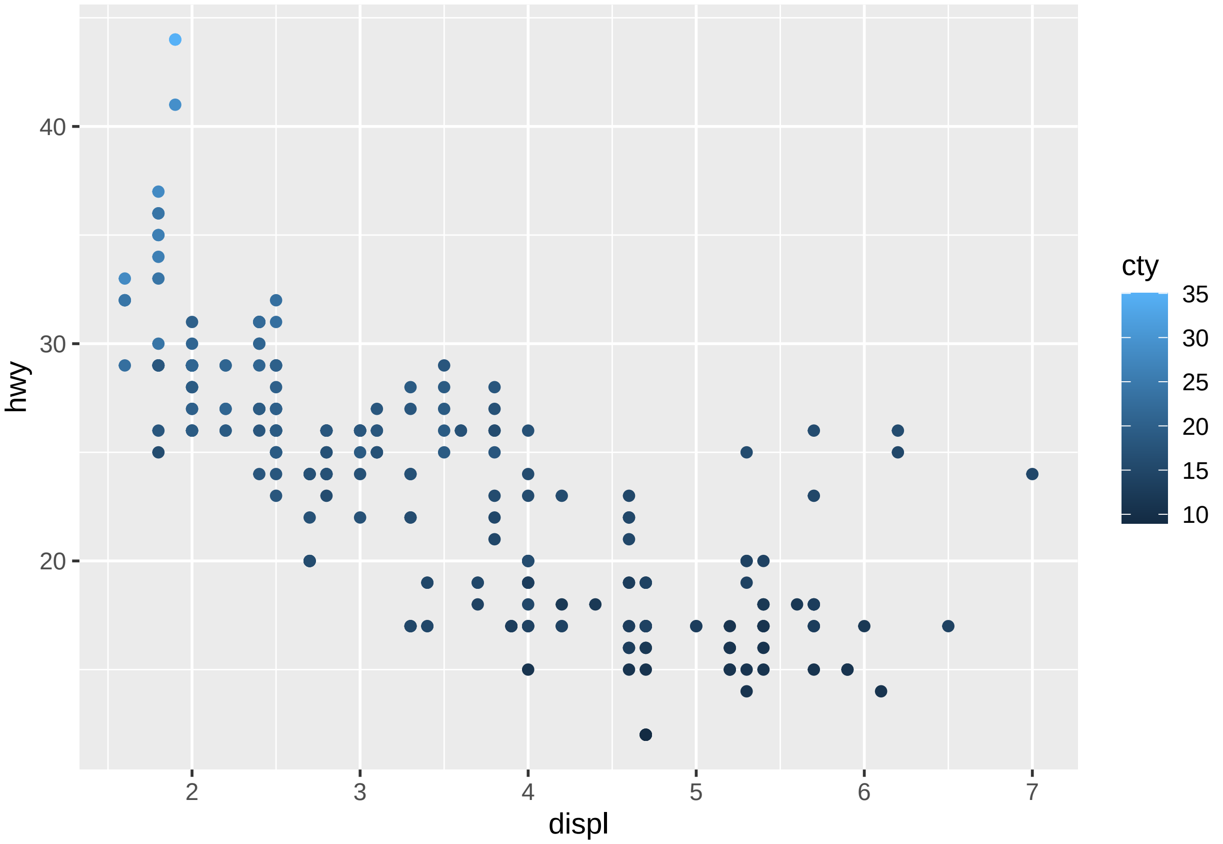
```
#Ans: manufacturer, model, trans, drv, fl, and class.
```

b. Which are continuous variables?

```
#Ans: displ, year, cyl, cty, and hwy.
```

c. Plot the relationship between displ (engine displacement) and hwy(highway miles per gallon). Mapped it with a continuous variable you have identified in #5-b. What is its result? Why it produced such output?

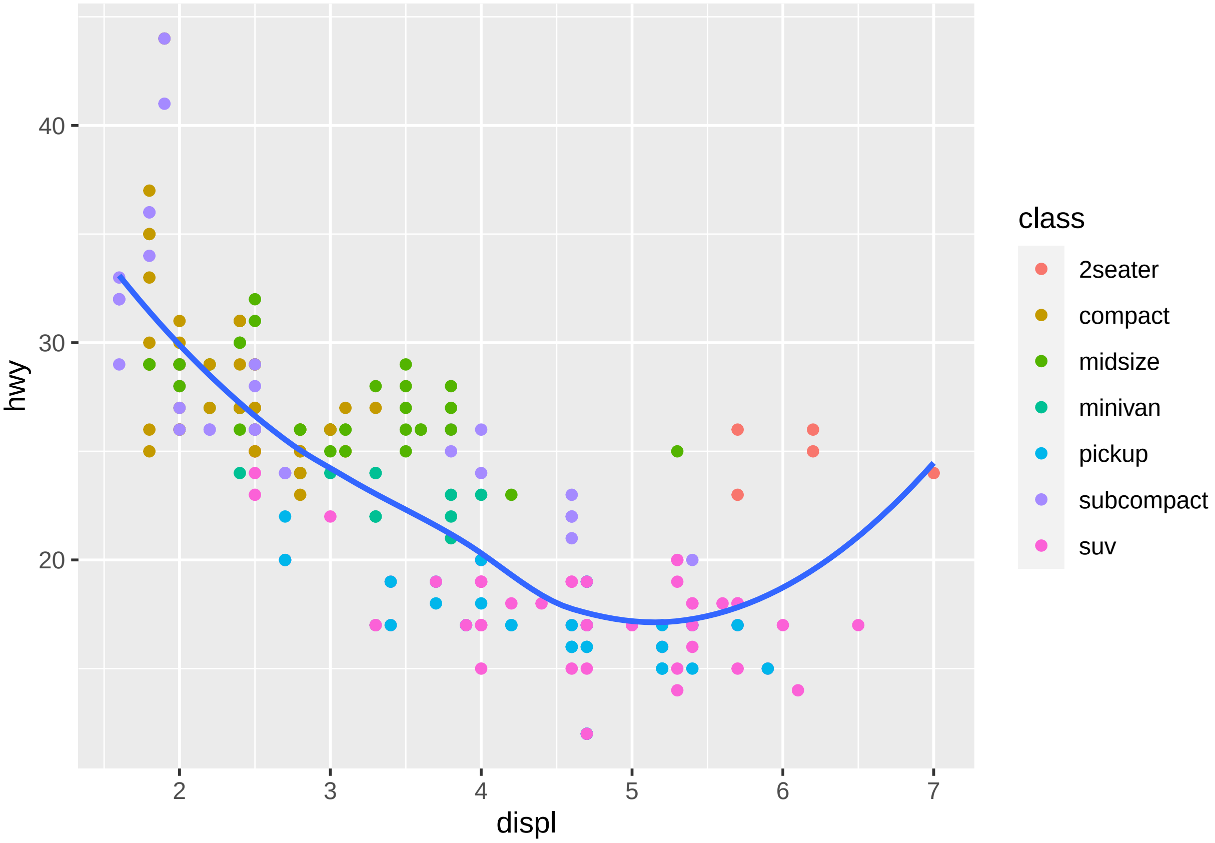
```
ggplot(mpg, aes(x = displ, y = hwy, colour = cty)) + geom_point()
```



9. Plot the relationship between displ (engine displacement) and hwy(highway miles per gallon) using geom_point(). Add a trend line over the existing plot using geom_smooth() with se = FALSE. Default method is "loess".

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

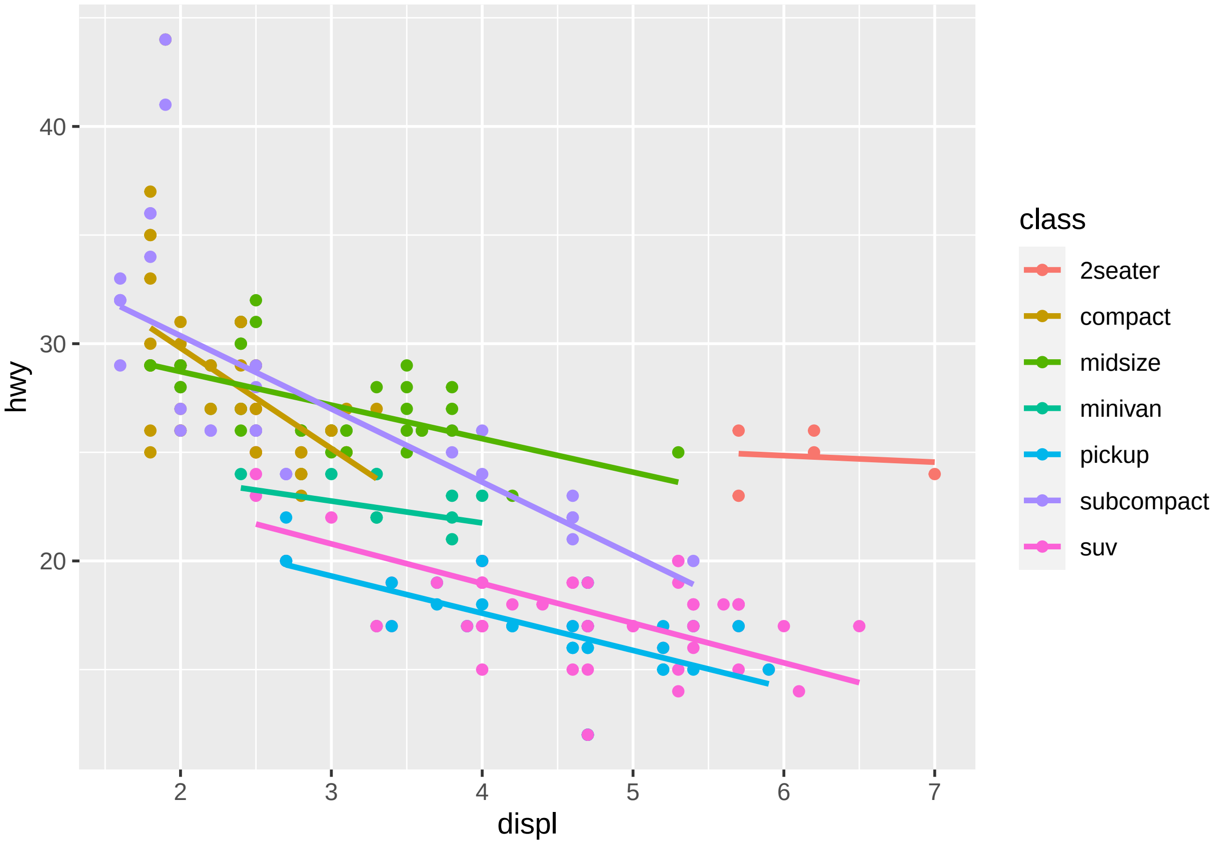
```
## `geom_smooth()` using formula = 'y ~ x'
```



10. Using the relationship of displ and hwy, add a trend line over existing plot. Set the se = FALSE to remove the confidence interval and method = lm to check for linear modeling.

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

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


40
30
20
hwy
2
3
4
5
6
7
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv