RWorksheet_Sicabalo#6

Mark Lexter Sicabalo

2022-12-05

#Worksheet#6 #Mark Lexter Sicabalo BSIT 2-A

```
library(ggplot2)
data(mpg)
as.data.frame(data(mpg))
```

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

```
"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" ...
```

```
library(dplyr)
```

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

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

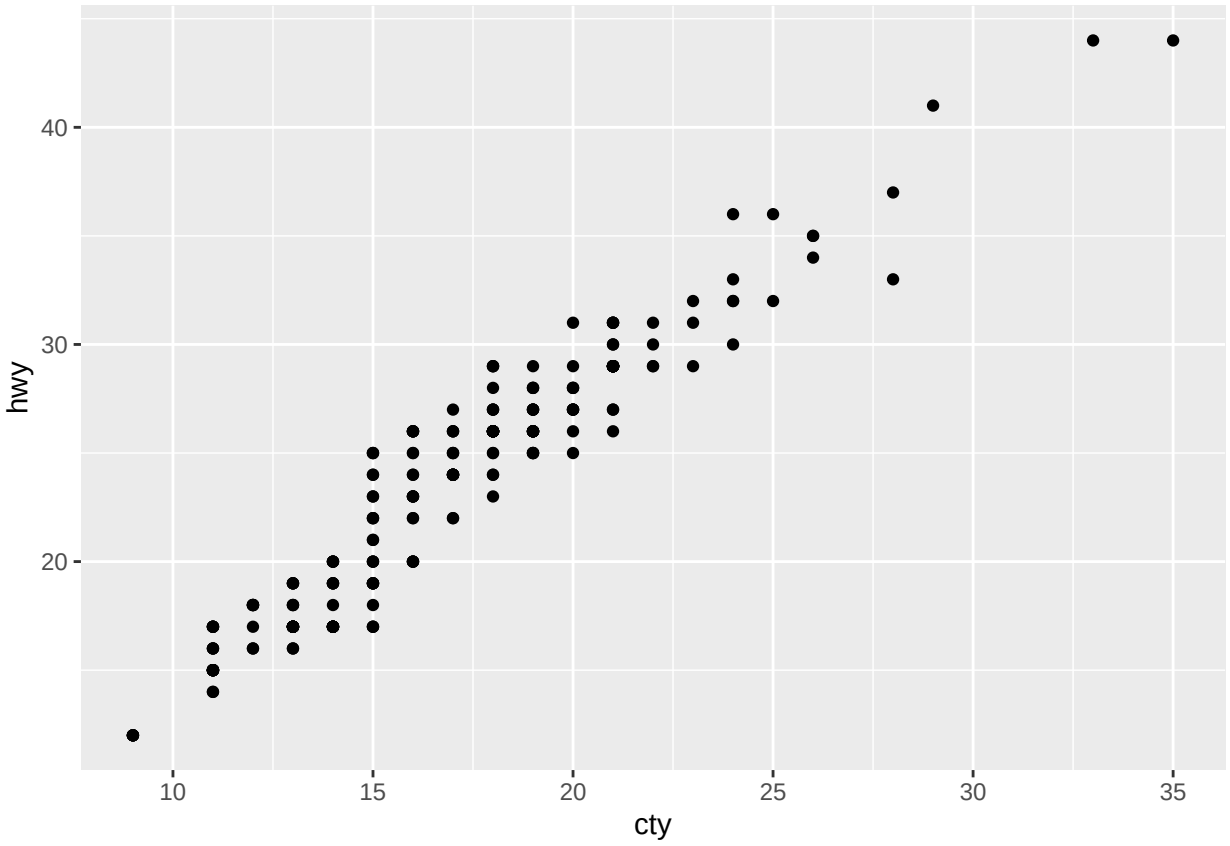
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
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~
```

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



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

```
data_mpg <- 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~
```

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

#Answer: There are 234 rows and have a 11 columns.

#2. Which manufacturer has the most models in this data set? Which model has the most variations? #Ans: Dodge, because it has 37 models.

```
brand_count <-  data_mpg %>% group_by(manufacturer,model) %>% count()
brand_count
```

```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet    c1500 suburban 2wd     5
##  5 chevrolet    corvette               5
##  6 chevrolet    k1500 tahoe 4wd        4
##  7 chevrolet    malibu                 5
##  8 dodge        caravan 2wd           11
##  9 dodge        dakota pickup 4wd      9
## 10 dodge        durango 4wd            7
## # ... with 28 more rows
```

```
colnames(brand_count) <- c("Manufacturer","Model","Counts")
```

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

```
unique_models <- data_mpg %>% group_by(manufacturer,model) %>% distinct() %>% count()
unique_models
```

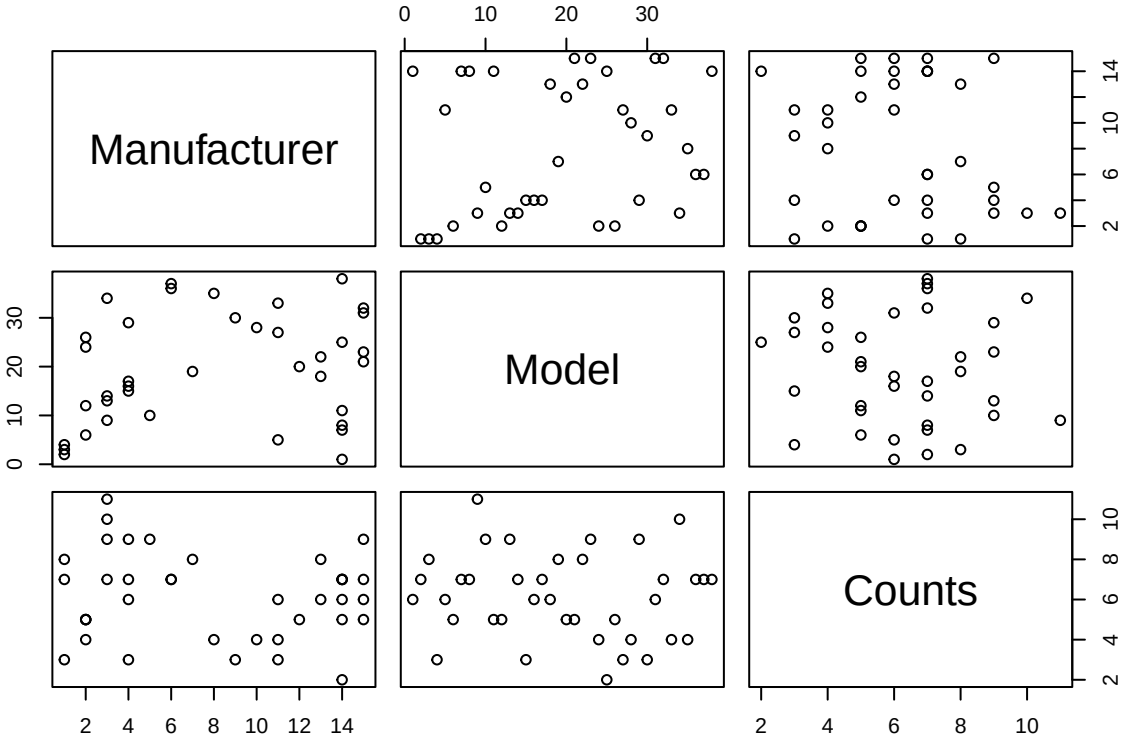
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet    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
```

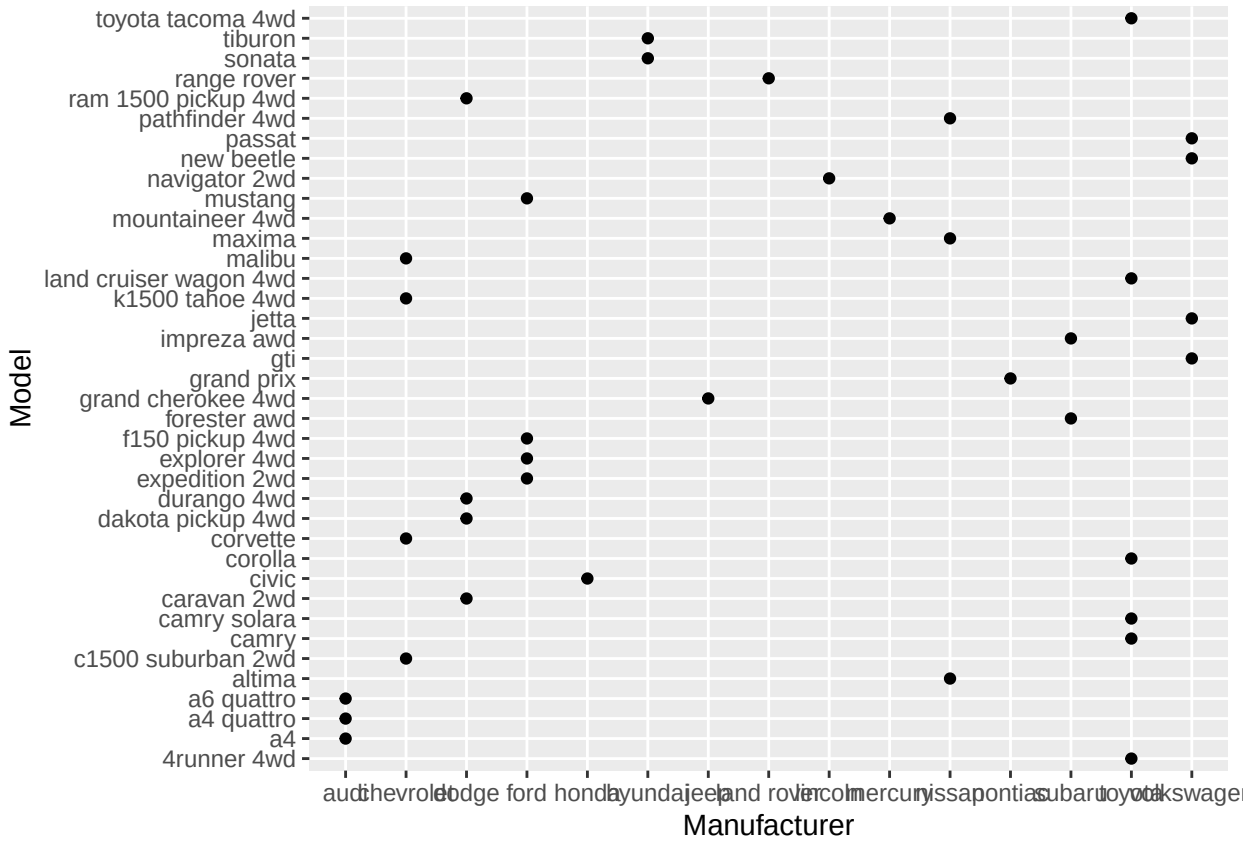
```
colnames(unique_models) <- c("Manufacturer","Model","Counts")
```

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

```
plot(brand_count)
```

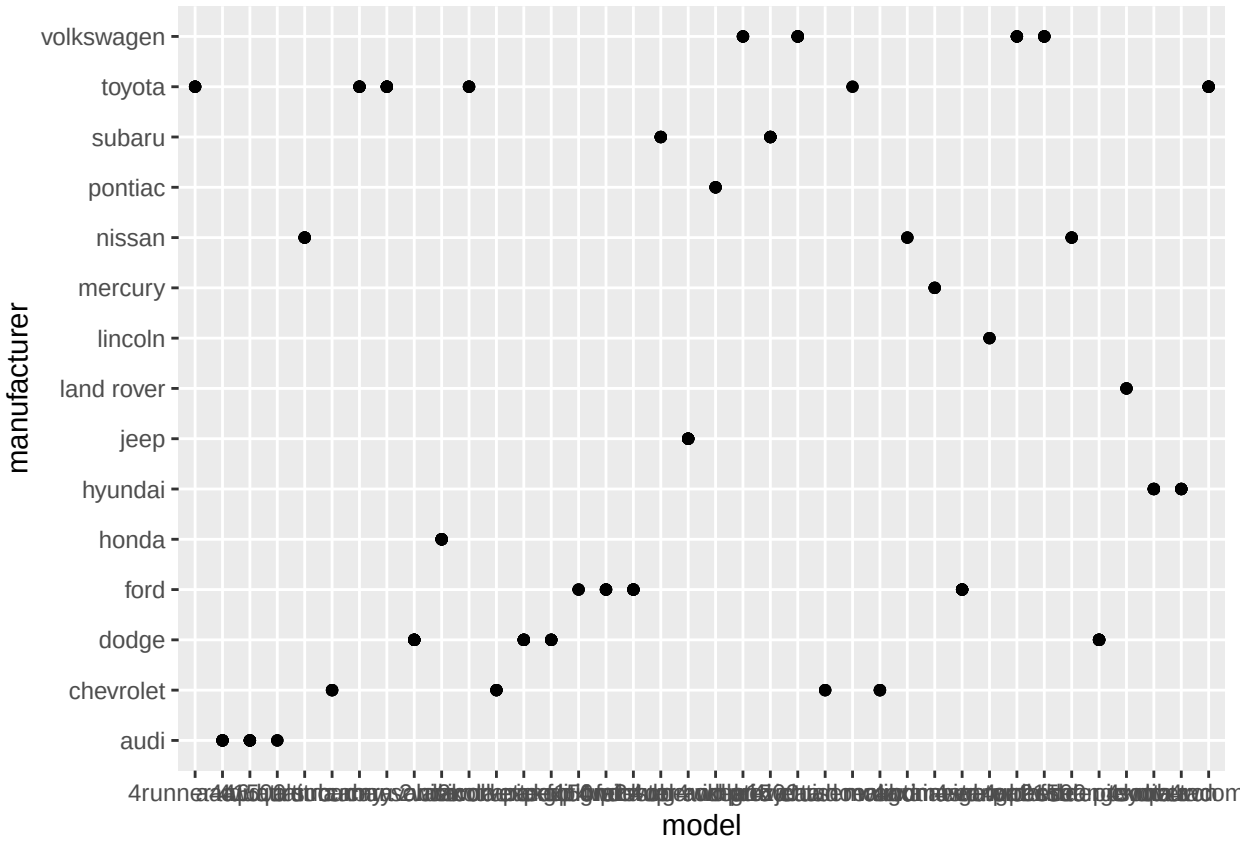



```
ggplot(brand_count, aes(Manufacturer,Model)) +geom_point()
```



#3. Same dataset will be used. You are going to show the relationship of the modeland the manufacturer.
#a. What does ggplot(mpg, aes(model, manufacturer)) + geom_point() show?

```
ggplot(data_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? #Answer: No, it is not useful because the data is already organized, but it can be improved to look more informative by including a legend to help users understand the data from the scatter plot.

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

```
car_data <- data_mpg %>% group_by(model) %>% count()
car_data
```

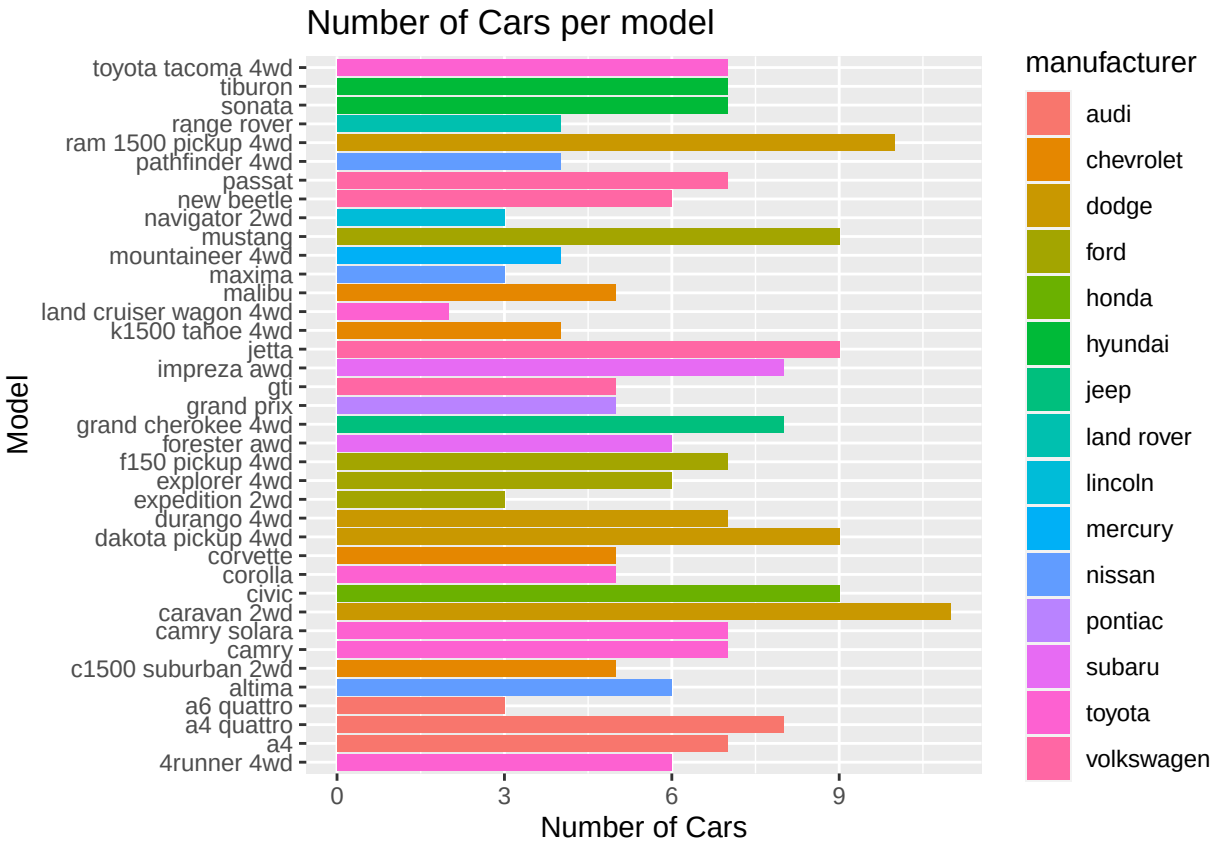
```
## # A tibble: 38 x 2
## # Groups:   model [38]
##    model                 n
##    <chr>             <int>
##  1 4runner 4wd           6
##  2 a4                    7
##  3 a4 quattro            8
##  4 a6 quattro            3
##  5 altima                6
##  6 c1500 suburban 2wd    5
##  7 camry                 7
##  8 camry solara          7
##  9 caravan 2wd          11
## 10 civic                 9
## # ... with 28 more rows
```

```
colnames(car_data) <- c("Model","Count")
```

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

```
qplot(model,data = 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.

```
toptwenty_data <- car_data[1:20,] %>% top_n(2)
```

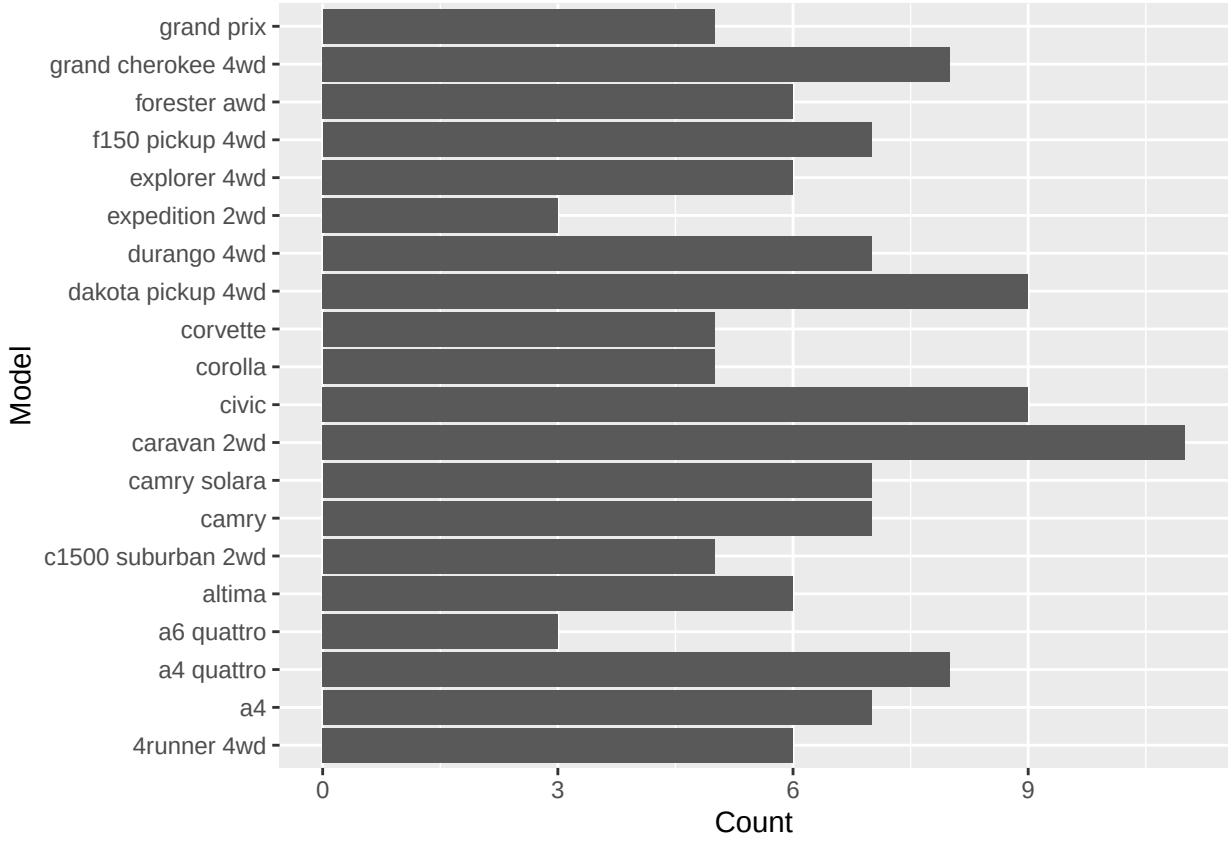
```
## Selecting by Count
```

```
toptwenty_data
```

```
## # A tibble: 20 x 2
## # Groups:   Model [20]
##    Model              Count
##    <chr>              <int>
##  1 4runner 4wd            6
##  2 a4                     7
##  3 a4 quattro             8
##  4 a6 quattro             3
##  5 altima                 6
##  6 c1500 suburban 2wd     5
##  7 camry                  7
##  8 camry solara           7
##  9 caravan 2wd           11
## 10 civic                  9
```

```
## 11 corolla                5
## 12 corvette               5
## 13 dakota pickup 4wd      9
## 14 durango 4wd            7
## 15 expedition 2wd         3
## 16 explorer 4wd           6
## 17 f150 pickup 4wd        7
## 18 forester awd           6
## 19 grand cherokee 4wd     8
## 20 grand prix             5
```

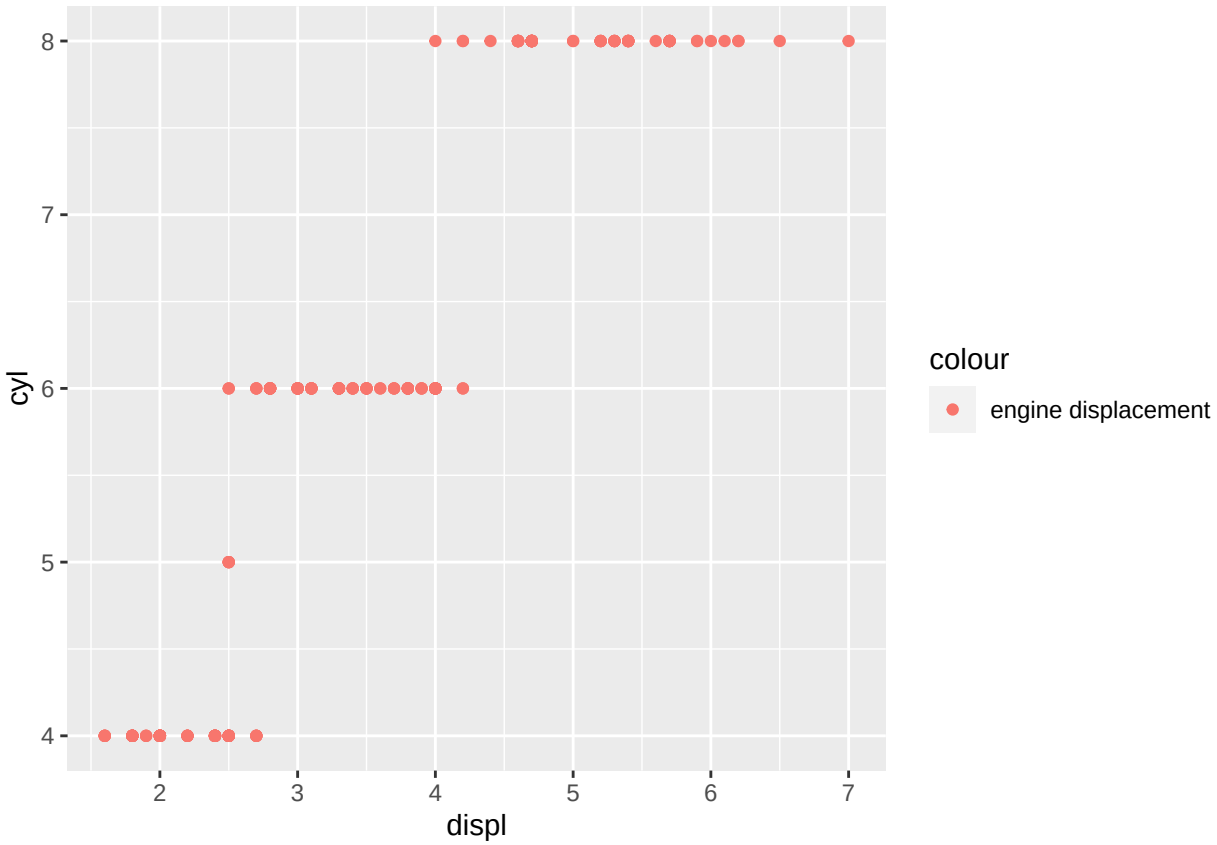
```
ggplot(toptwenty_data, aes(x = Model, y = Count)) + geom_bar(stat = "Identity") + coord_flip()
```




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

#5. Plot the relationship between cyl - number of cylinders and displ - engine displacement 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.

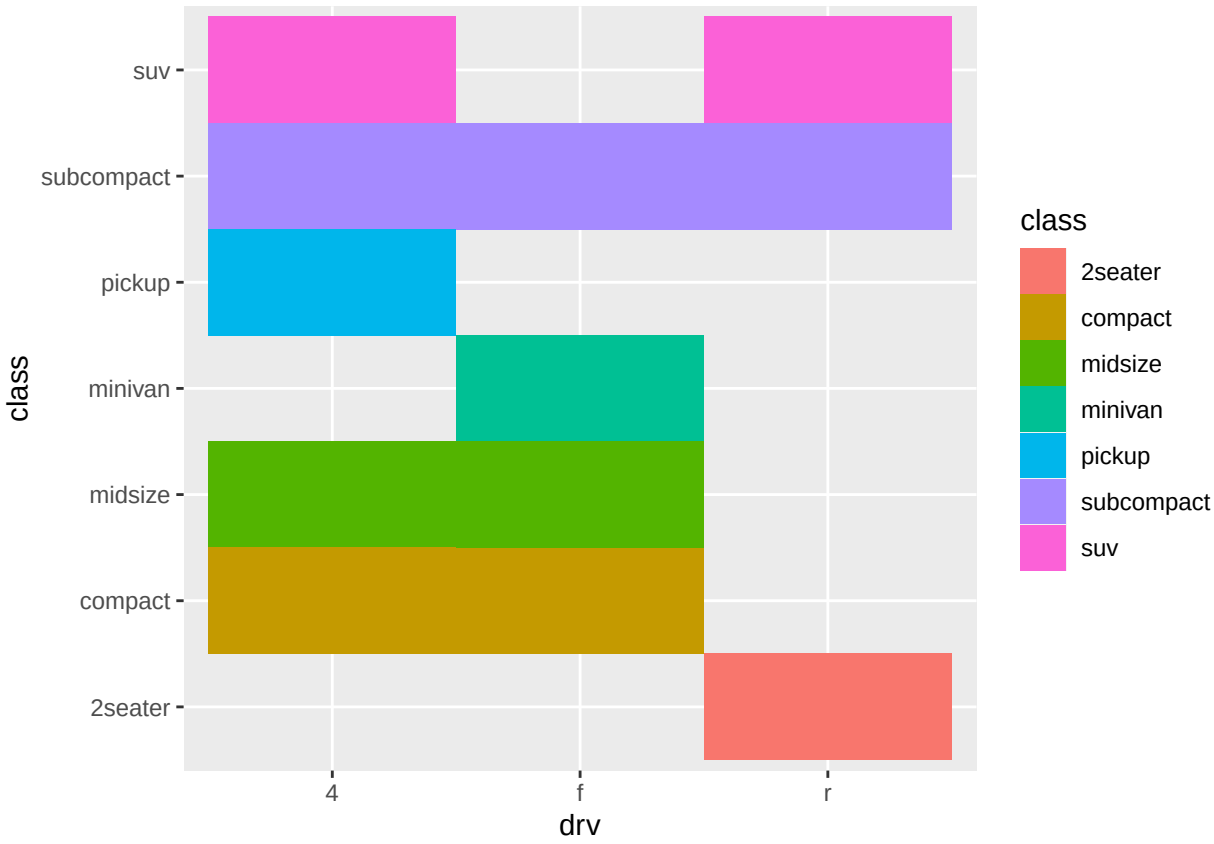
```
ggplot(data = data_mpg ,
       mapping = aes(x = displ, y = cyl,
                     main = "Relationship between No of Cylinders and Engine Displacement")) + geom_poi
```



#b. How would you describe its relationship? #Answer: The relationship between cyl - number of cylinders and displ - engine displacement is they are proportional with each other because if cyl increases also the displ increases.

#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.

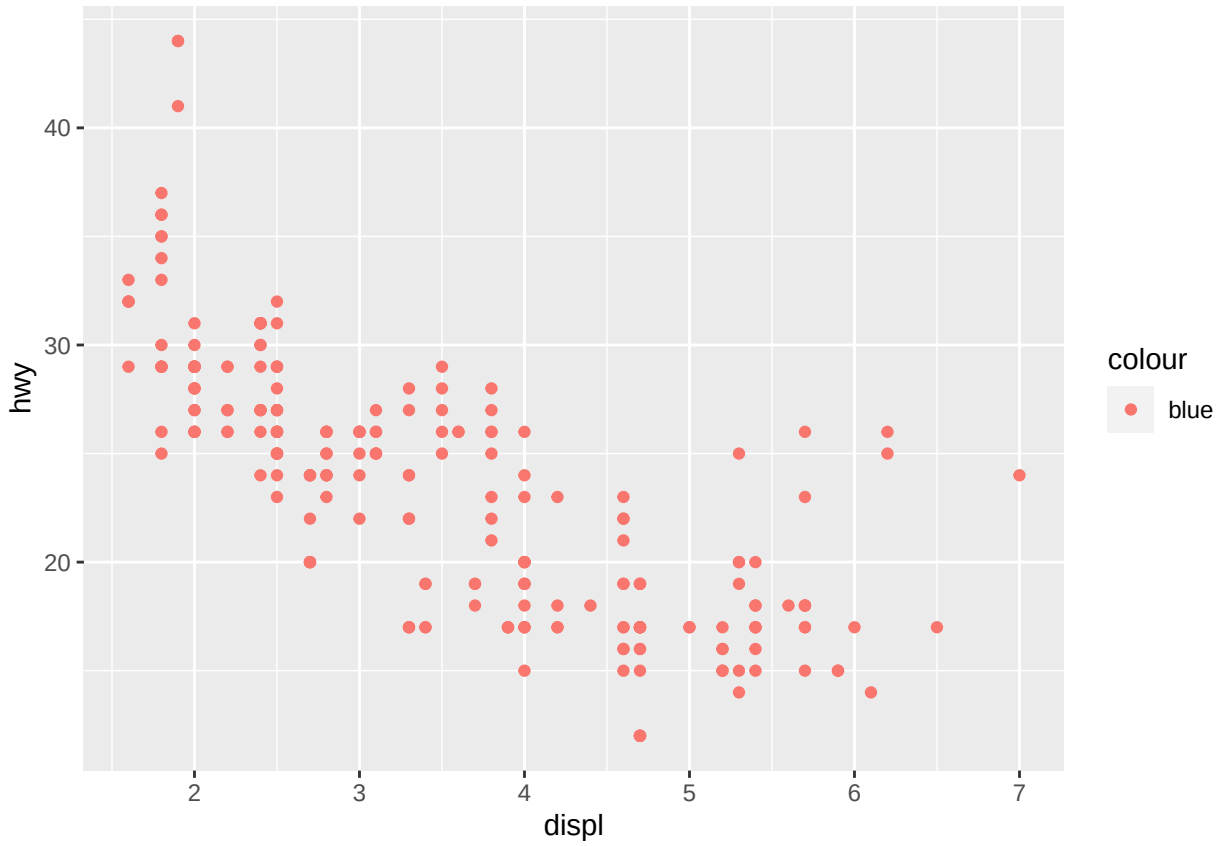
```
ggplot(data = data_mpg, mapping = aes(x = drv, y = class)) +geom_tile(aes(fill=class))
```



#b. Interpret the result. #Answer: When mapping a geomatric tile, it graphs the data and fill a random different colors depends on its class, drv is the x axis while class is the y axis.

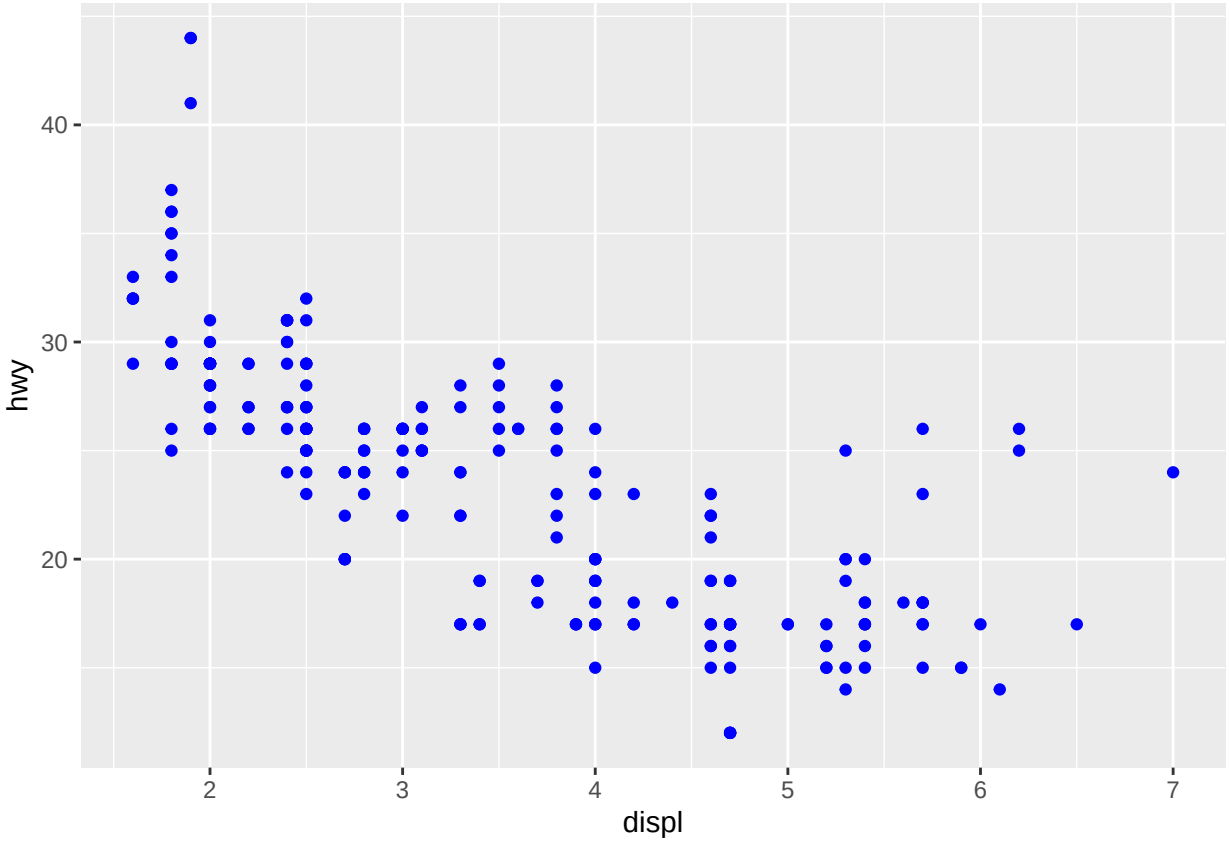
#7. Discuss the difference between these codes. Its outputs for each are shown below. #Code #1

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



#& Code #2

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



#Answer: If the x and y axis is with the colour inside the parenthesis, it graph with legend but the color is red. But when the x and y axis is separated with colour it only graph with a color of blue.

#8. Try to run the command ?m pg. What is the result of this command?

```
?mpg
```

#a. Which variables from mpg dataset are categorical? #Answer: The manufacturer, model, trans, drv, fl, class are the categorical variables from the data-set of mpg.

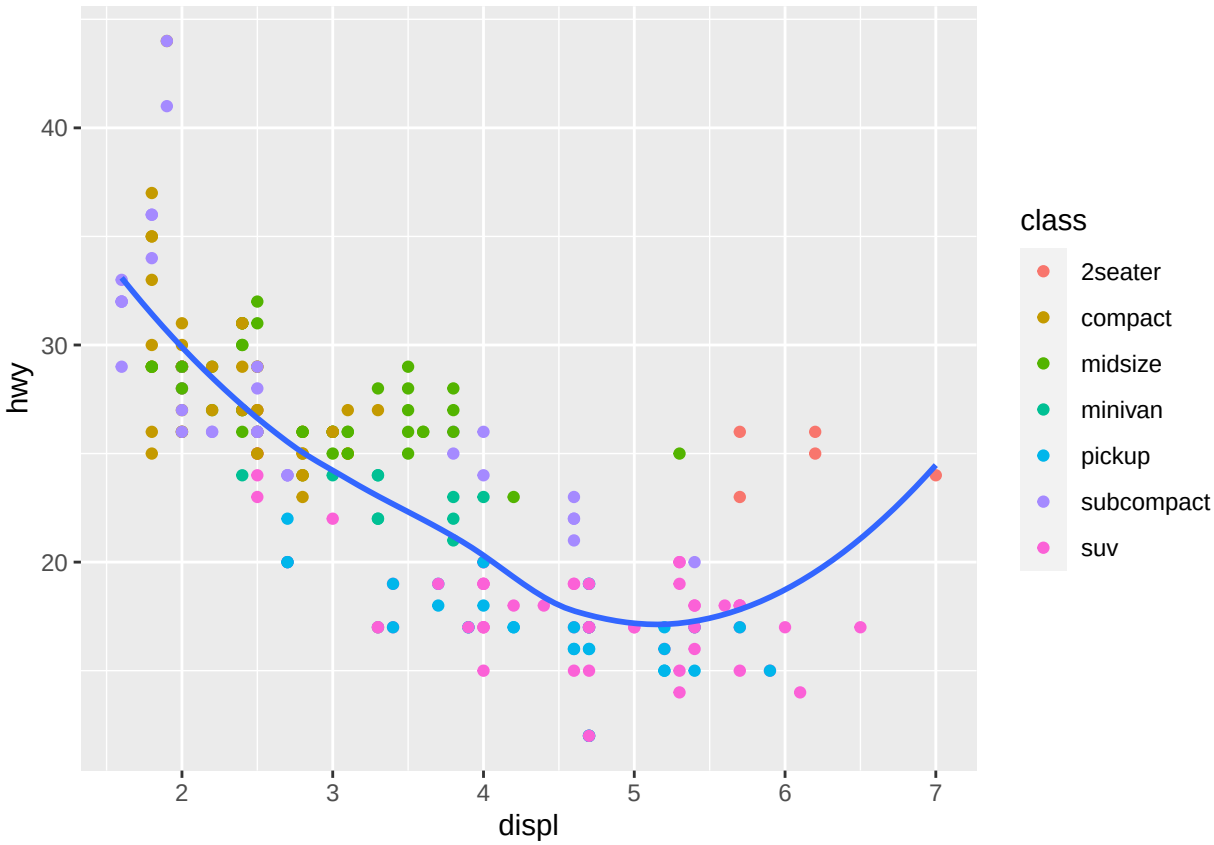
#b. Which are continuous variables? #Answee: The continuous variable of the mpg data-set are the dsipl, year, cyl, cty, and hwy.

#c. Plot the relationship between displ (engine displacement) and hwy(highway mile 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 = cty, y = hwy, colour = displ)) + 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 usingeom_smooth() with se = FALSE. Default method is "loess". per gallon) using geom_point(). Add a trend line over the existing plot usimgeom_smooth() with se = FALSE. Default method is "loess".uigeom_smooth() with se = FALSE. Default method is "loess".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_smoothl
```

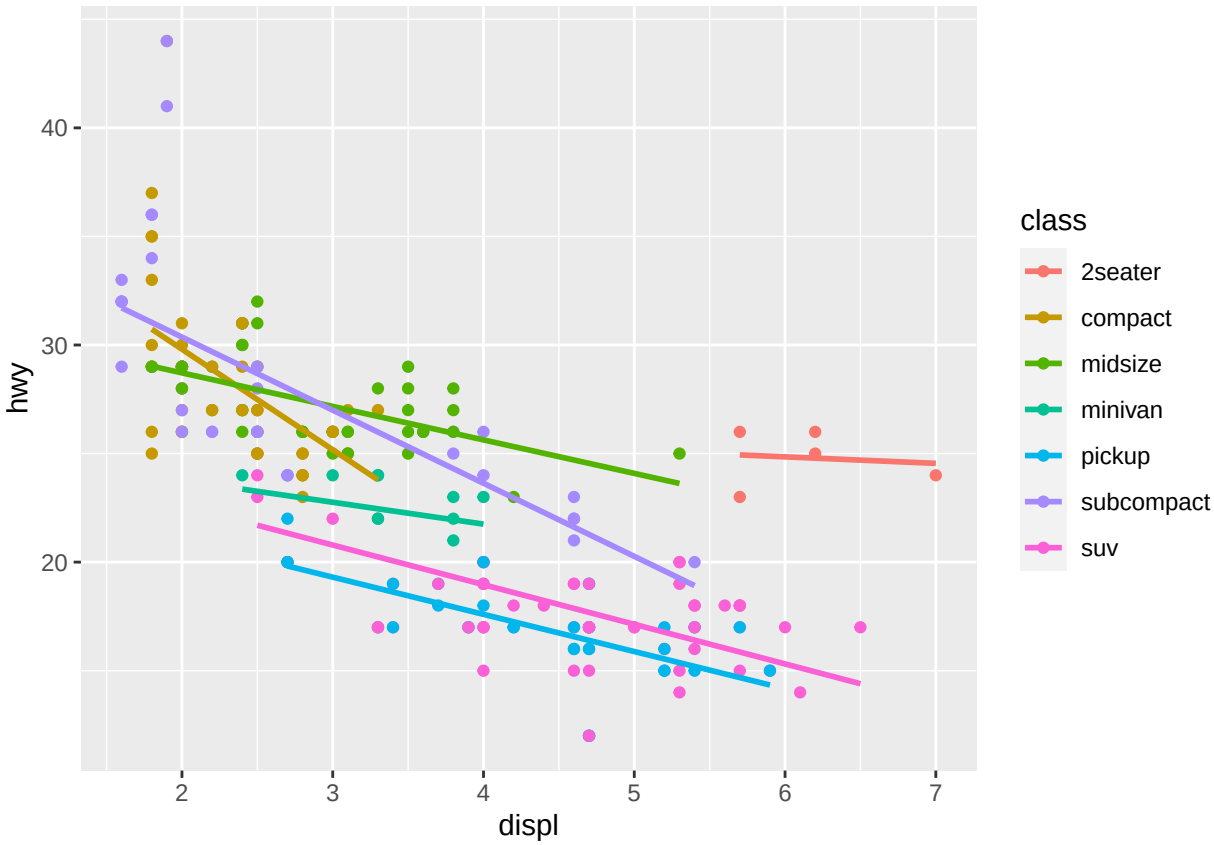
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



#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 = FA
```

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


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