RWorksheet_CAHUYA#6

CAHUYA, CARLO J'NAED LYTON BSIT-2A

2022-11-23

```
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(ggplot2)
```

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

```
data(mpg)
mpg_data <- 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~
```

```
mpg_data
```

```
## # 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
```

```
nrow(mpg)
```

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

```
ncol(mpg)
```

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

```
# The mpg data set have 11 columns and 234 rows.
```

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

```
most_mods <- mpg_data %>% group_by(manufacturer) %>% count()
most_mods
```

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

```r
colnames(most_mods) <- c("Manufacturer","Counts")
most_mods
```

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

```r
#  Dodge has the most number of models. It has 37 models.
```

Which model has the most variations?

```r
most_var<- mpg_data %>% group_by(model) %>% count()
most_var
```

```
## # 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
```

```r
colnames(most_var) <- c("Model","Counts")
most_var
```

```
## # A tibble: 38 x 2
## # Groups:   Model [38]
##    Model              Counts
##    <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
```

```
# Caravan 2wd model has the most variations. It has 11 variations.
```

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

```
unique_mods<- mpg_data %>% group_by(manufacturer, model) %>% distinct() %>% count()
unique_mods
```

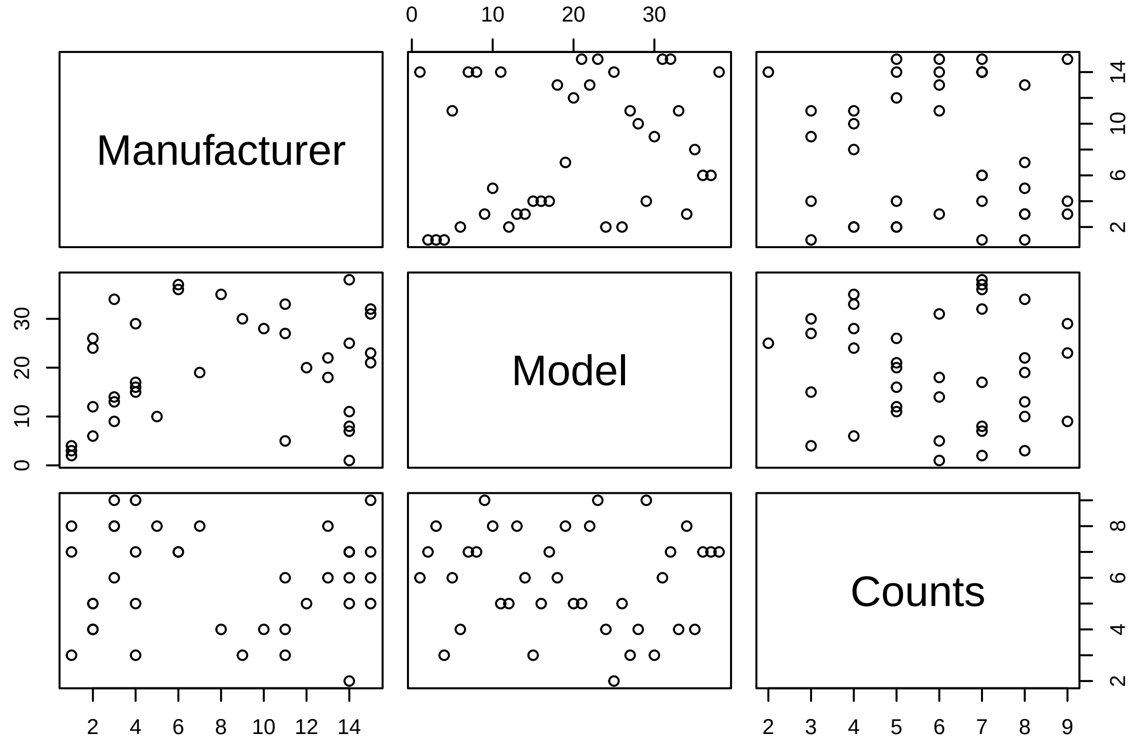
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet    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_mods) <- c("Manufacturer", "Model","Counts")
unique_mods
```

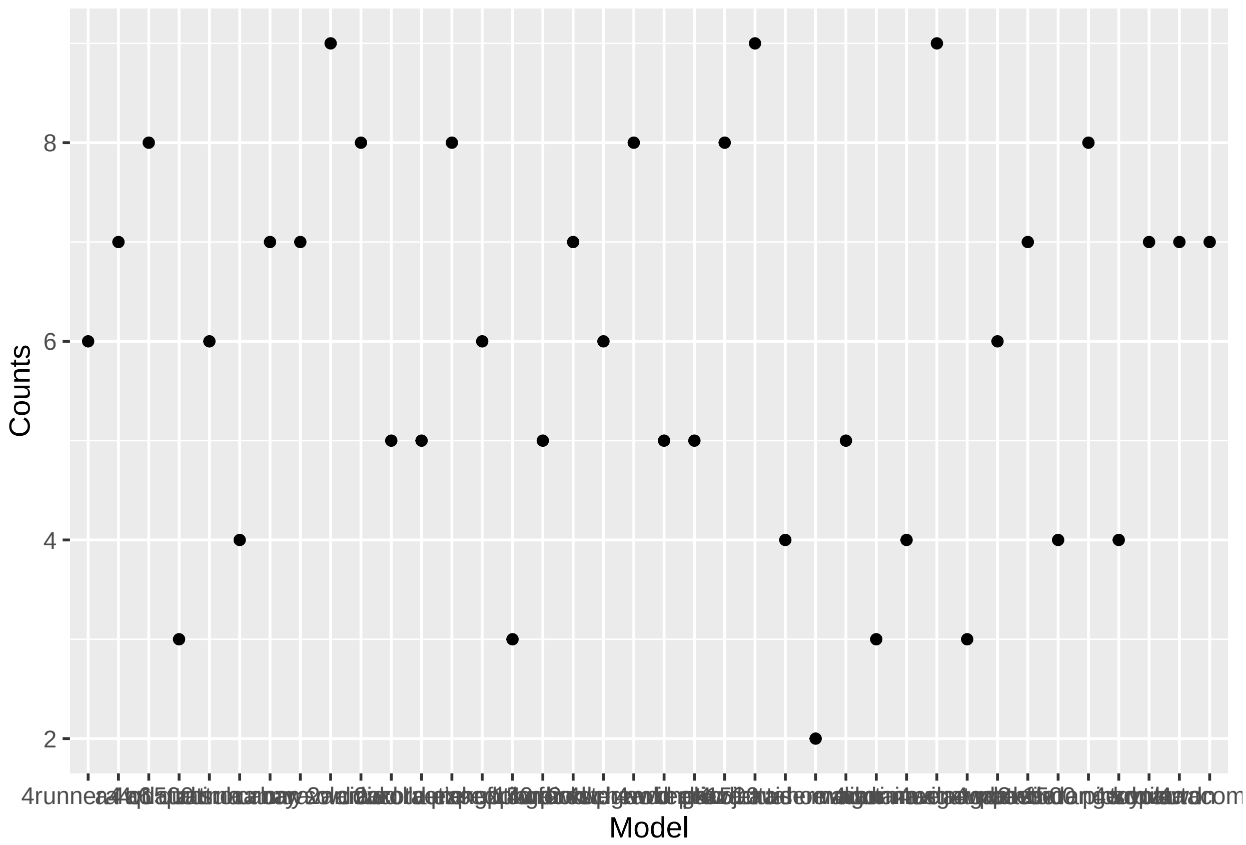
```
## # A tibble: 38 x 3
## # Groups:   Manufacturer, Model [38]
##    Manufacturer Model              Counts
##    <chr>        <chr>               <int>
##  1 audi         a4                      7
##  2 audi         a4 quattro              8
##  3 audi         a6 quattro              3
##  4 chevrolet    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(unique_mods)
```

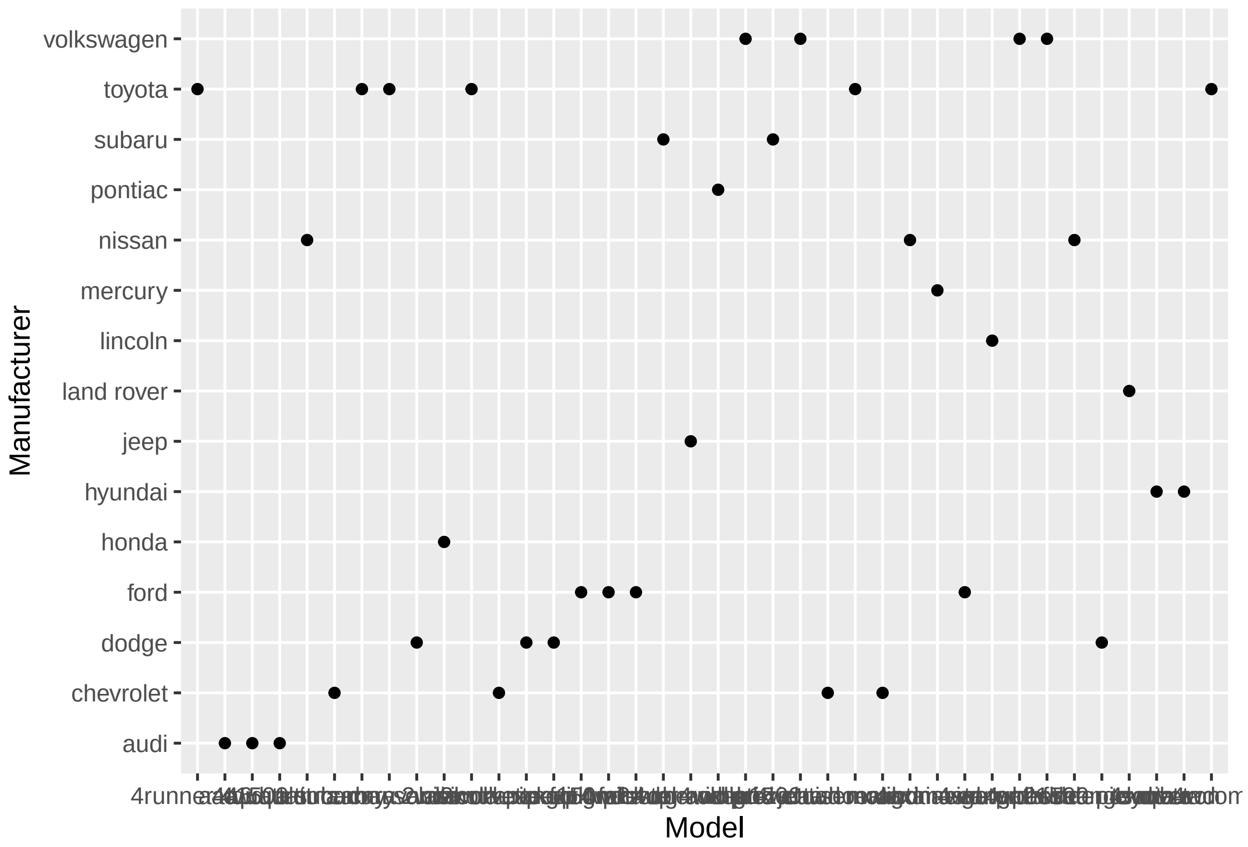


```
ggplot(unique_mods, aes(x = Model, y = Counts )) + geom_point(color='black')
```



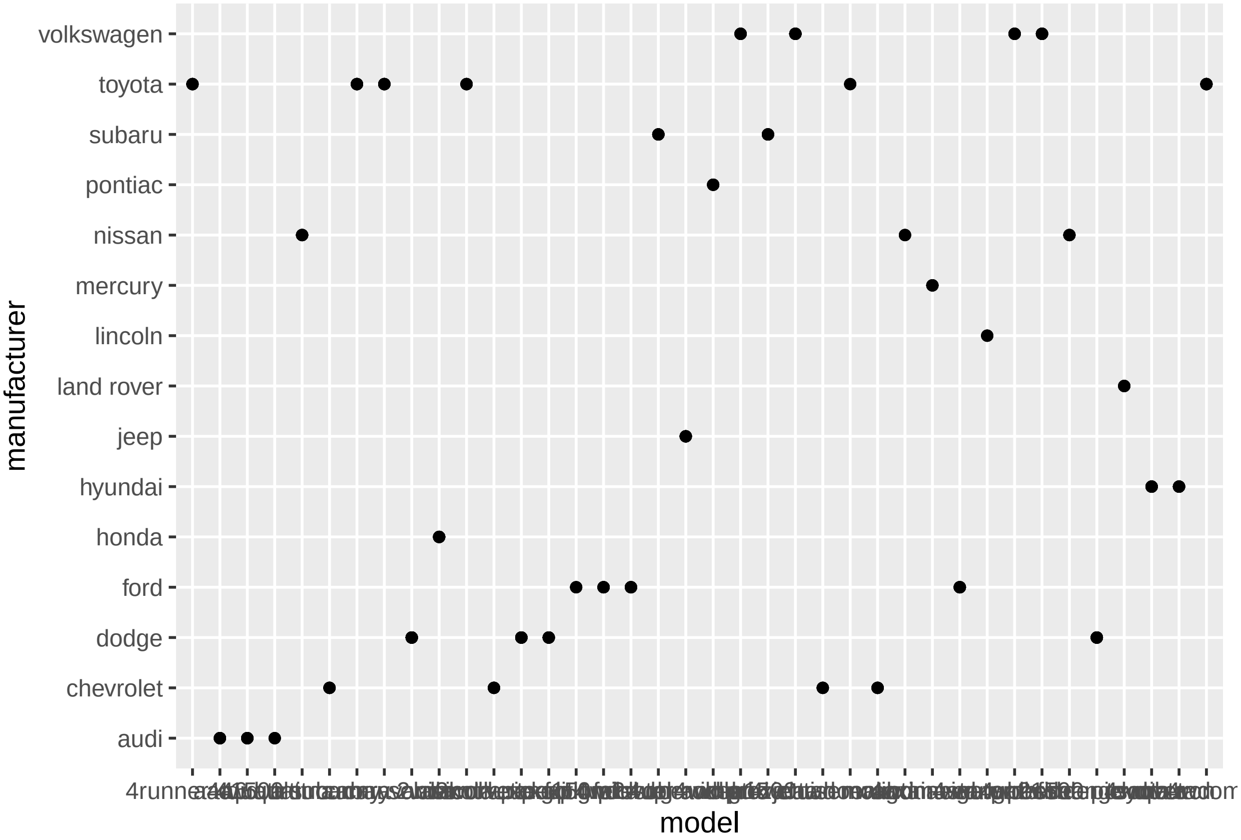
3. Same dataset will be used. You are going to show the relationship of the model and the manufacturer.

```
ggplot(unique_mods, aes(x = Model, y = Manufacturer )) + geom_point(color='black')
```



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

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



```
# The output of the code displays a plot with black points in it.
```

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

```
# Yes, it is helpful since you can trace down the data for each model from the manufacturers.
```

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

```
car_mods <- mpg_data %>% group_by(model) %>% count()
car_mods
```

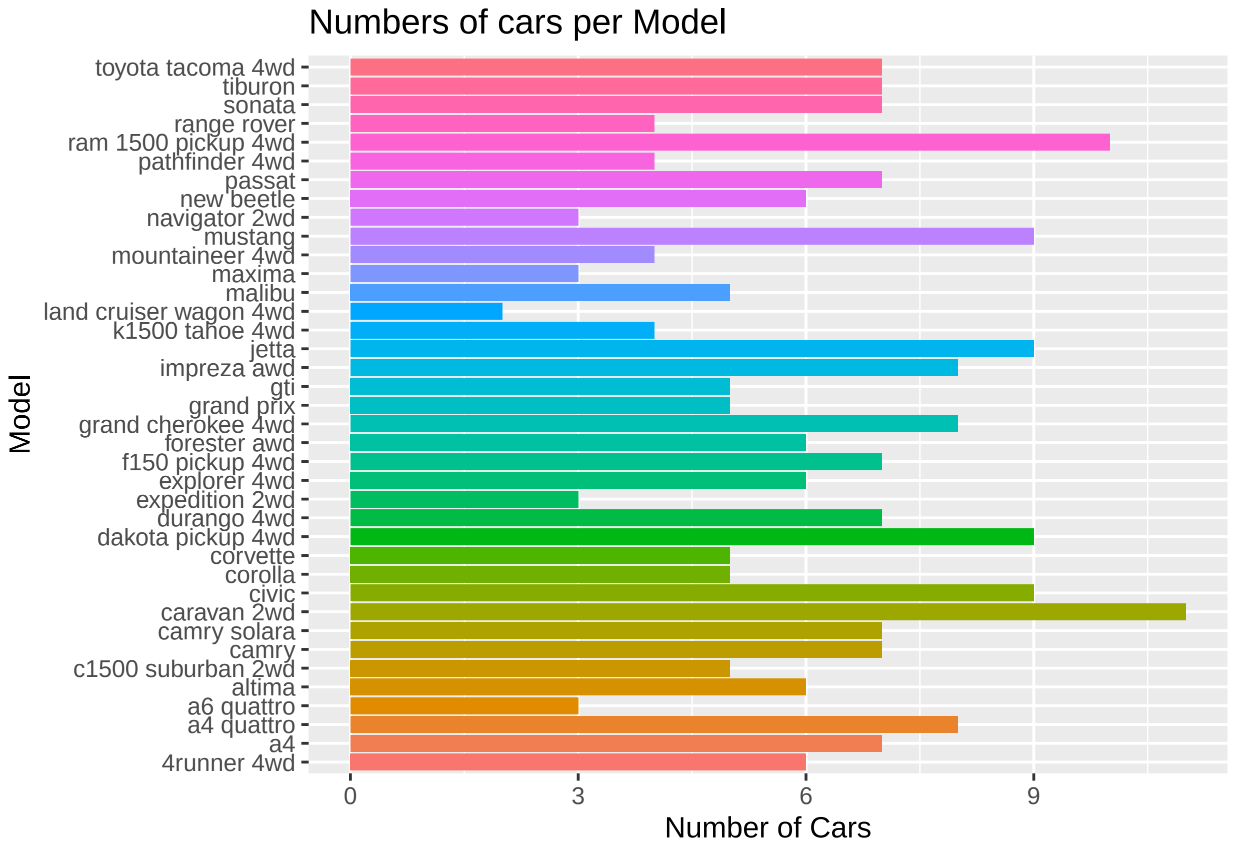
```
## # A tibble: 38 x 2
## # Groups:   model [38]
##    model                  n
##    <chr>              <int>
##  1 4runner 4wd            6
##  2 a4                     7
##  3 a4 quattro             8
##  4 a6 quattro             3
##  5 altima                 6
##  6 c1500 suburban 2wd     5
##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

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

```
## # A tibble: 38 x 2
## # Groups:   Model [38]
##    Model              Counts
##    <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
```

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

```
bar_graph <- ggplot(car_mods, aes( x = Model, y = Counts, fill = Model)) +
  labs(title = "Numbers of cars per Model", y = "Number of Cars", x = "Model")  +
  geom_bar(stat = "identity") + theme(legend.position = "none")
bar_graph +
  coord_flip()
```



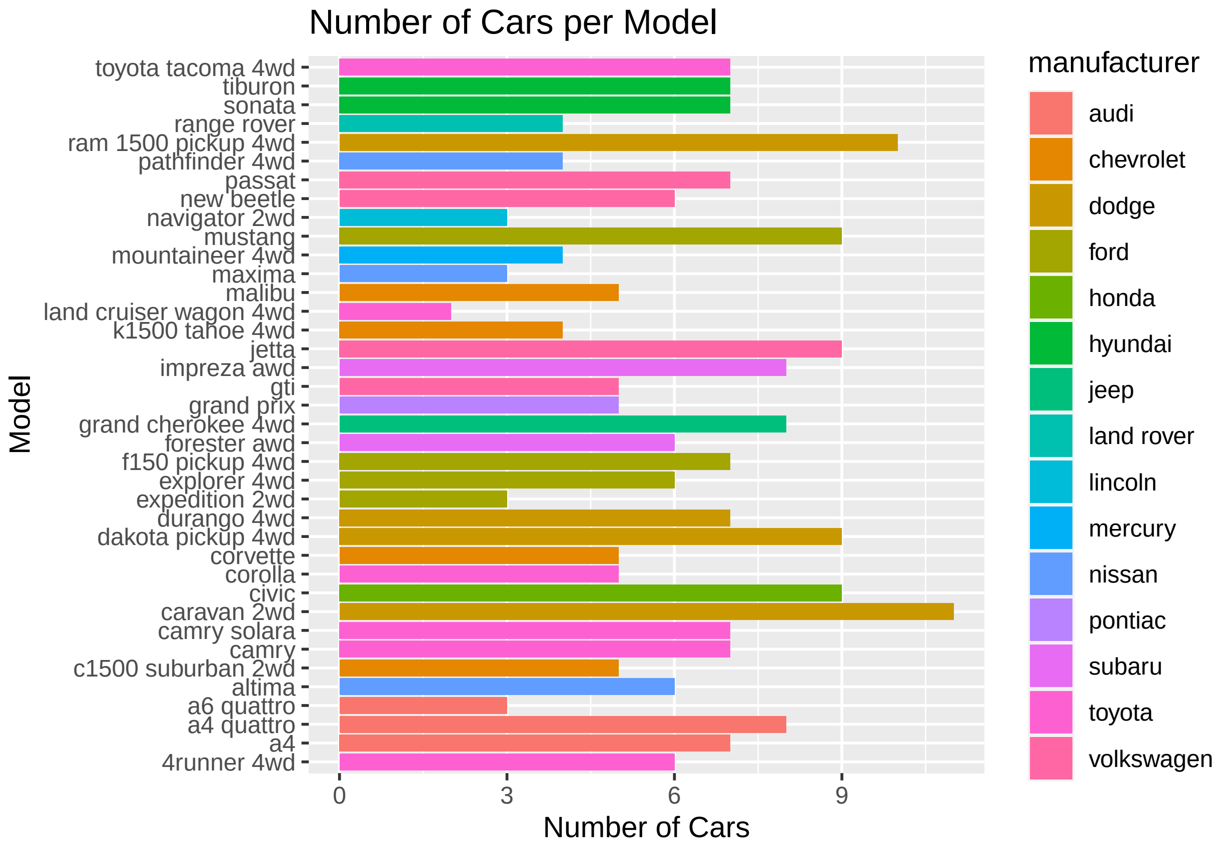
b. Use only the top 20 observations. Show code and results.

```
head(car_mods, n = 20)
```

```
## # A tibble: 20 x 2
## # Groups:   Model [20]
##    Model                 Counts
##    <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
```

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.

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

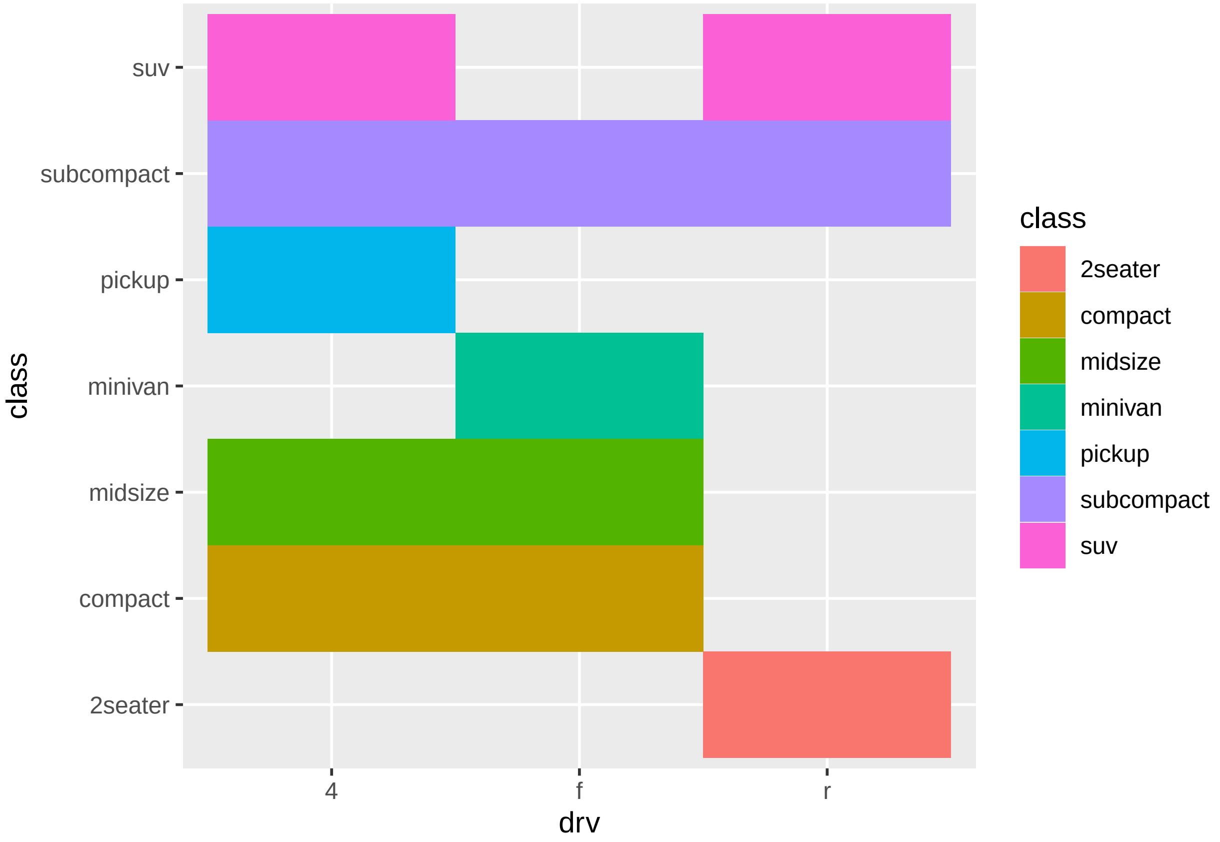


b. How would you describe its relationship?

```
# Based on the output of the code, I can say that their connection is very consistent or stable.
```

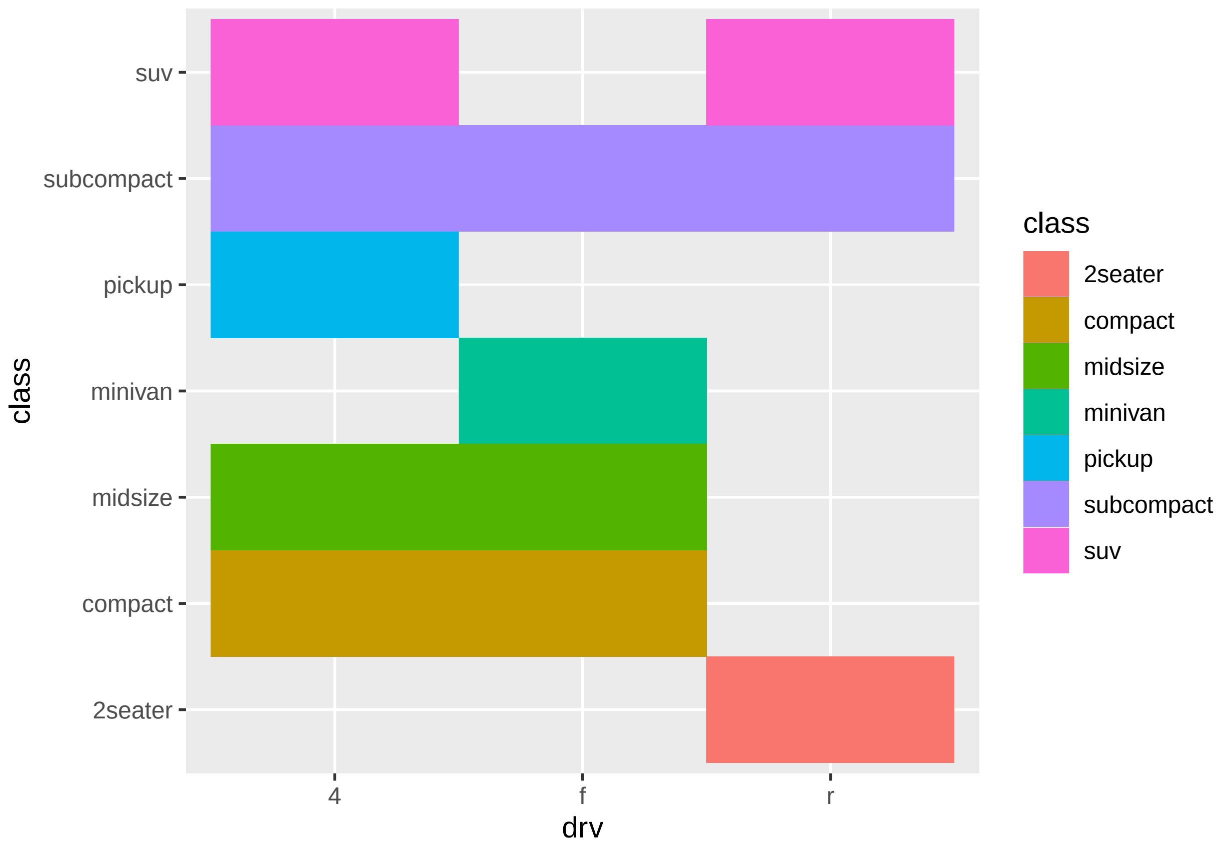
6. Get the total number of observations for drv - type of drive train (f = front-wheel drive, r = rear wheel drive, 4 = 4wd) and class - type of class (Example: suv, 2seater, etc.).

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



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(mpg, aes(drv, class)) +
  geom_tile (aes(fill = class))
```

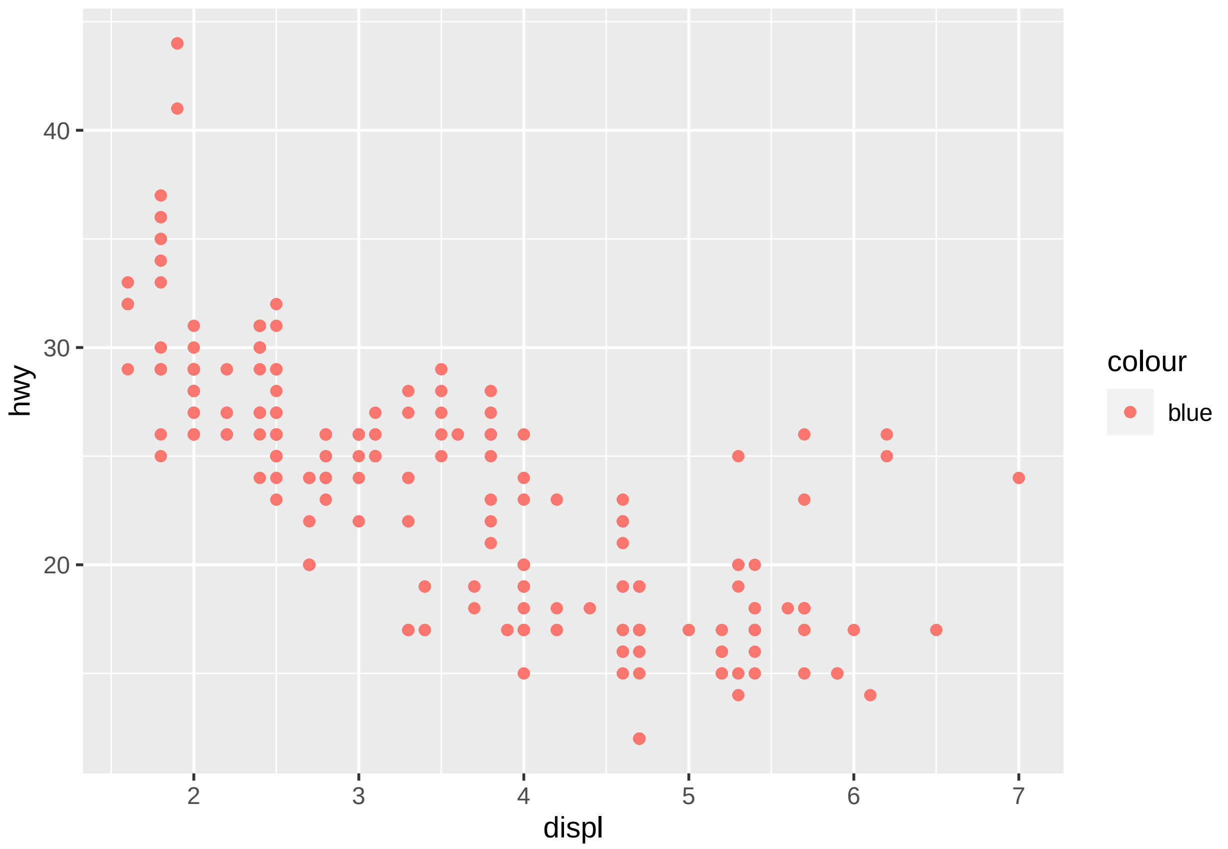


b. Interpret the result.

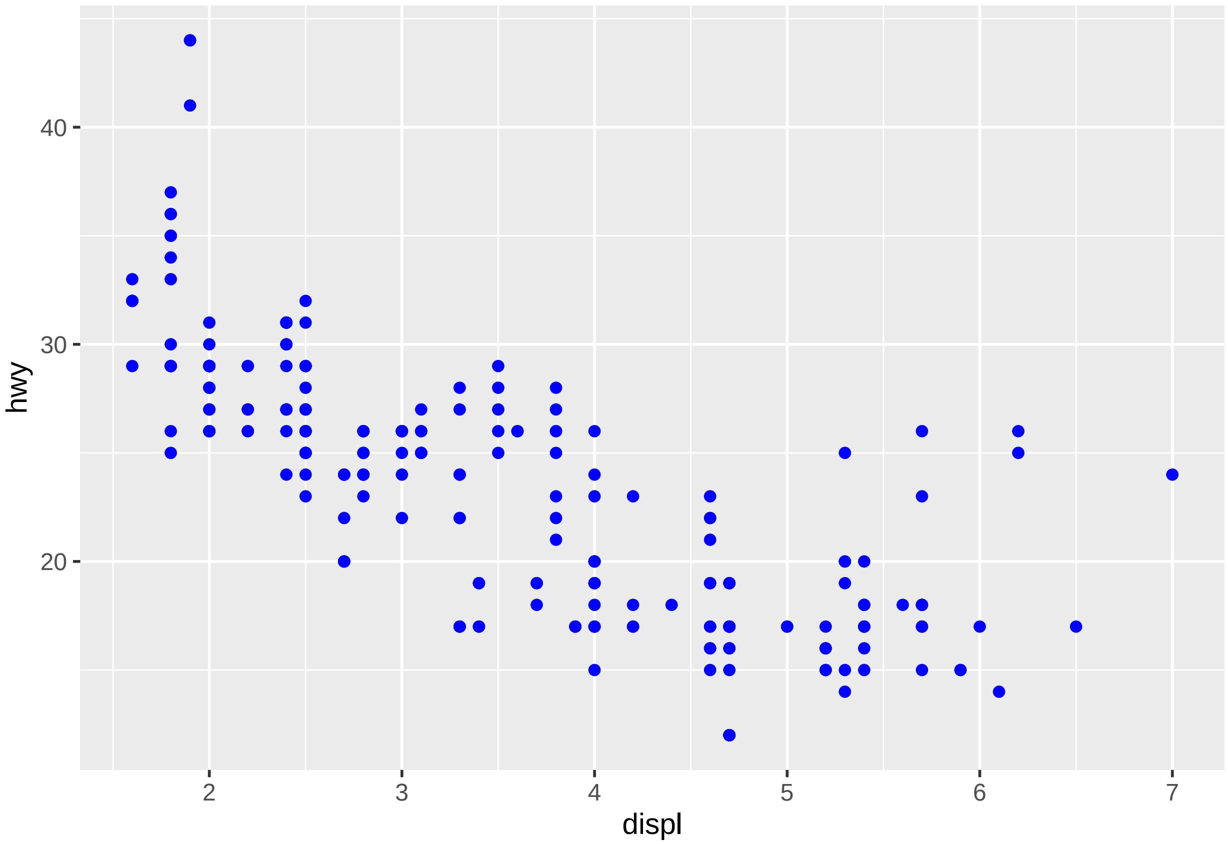
```
# The mapping geometric point graph is used to "map" the black areas.
# The output shows that if there is a relationship between a class and drv, a tile is created.
```

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")
```



```
# The "colour = blue" code in the first code was inside the function aes(), so it failed to add
# black dots or points. The second code, on the other hand, was performed and was in its right
# location or outside the aes() method, and the plot was shown accordingly.
```

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

```
?mpg
```

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

```
# It shows "Fuel economy data from 1999 to 2008 for 38 popular models of cars" in the help panel.
```

a. Which variables from mpg data set are categorical?

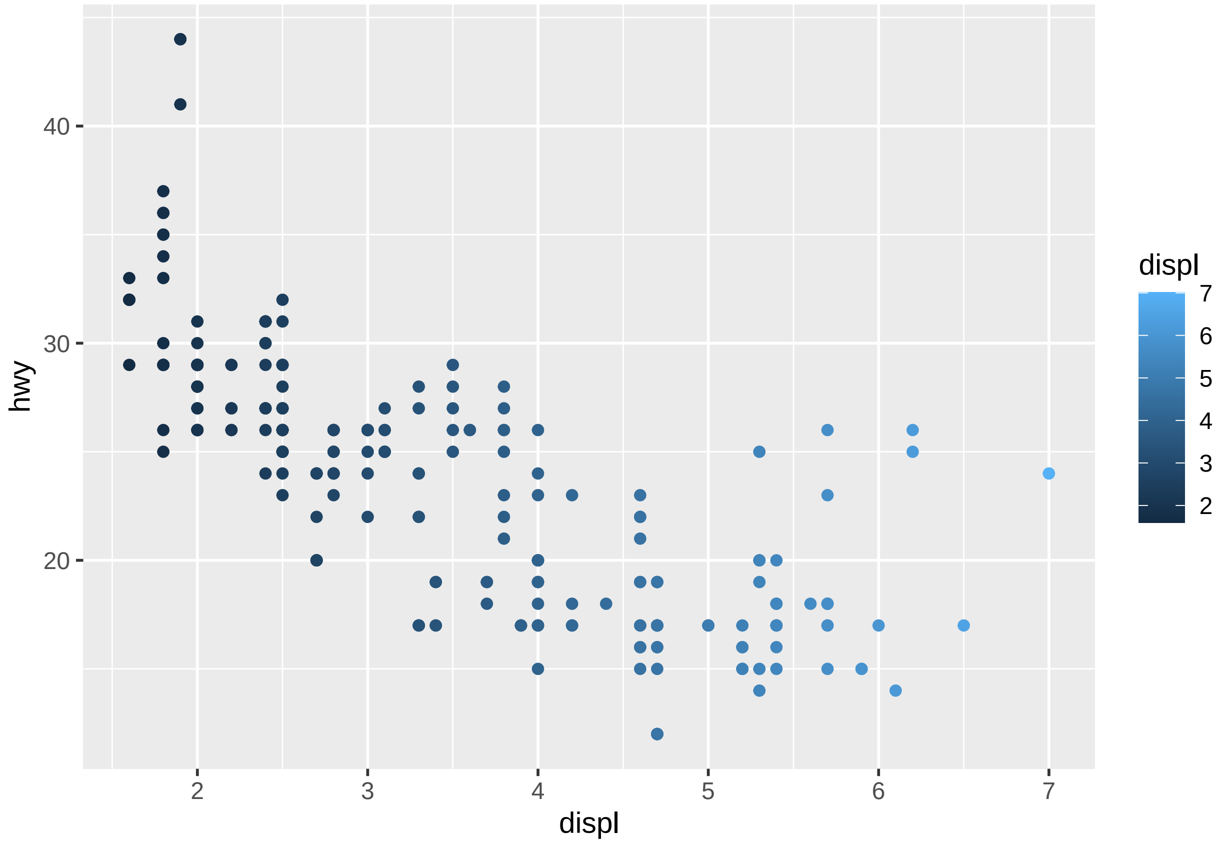
- Categorical variables in mpg include: manufacturer, model, trans (type of transmission), drv (front-wheel drive, rear-wheel, 4wd), fi (fuel type), and class (type of car)

b. Which are continuous variables?

- Continuous variables in mpg include: displ (engine displacement in litres), cyl (number of cylinders), cty (city miles/gallon), and hwy (highway gallons/mile)

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

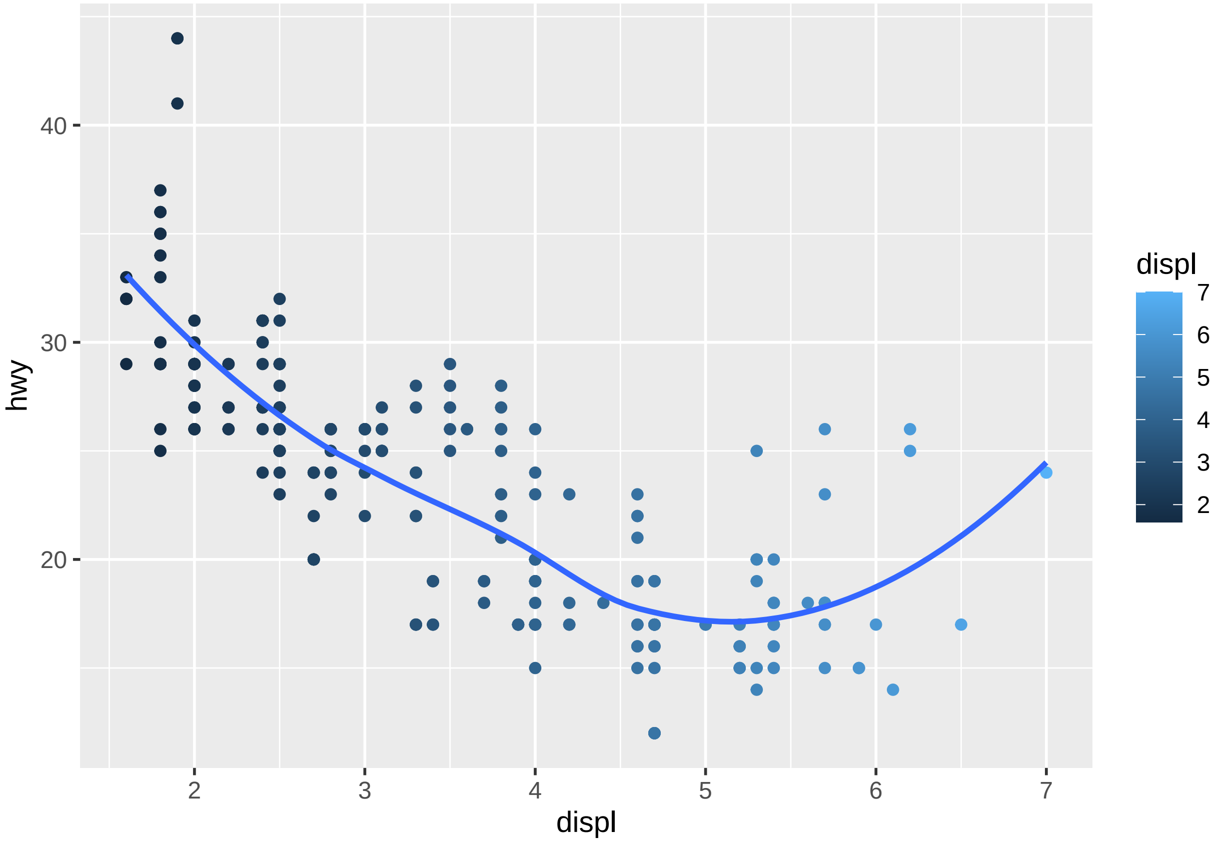


```
# It produced such output because we plot the relationship between the displ and hwy and its geom_point
```

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=displ)) +
  geom_smooth(se =FALSE)
```

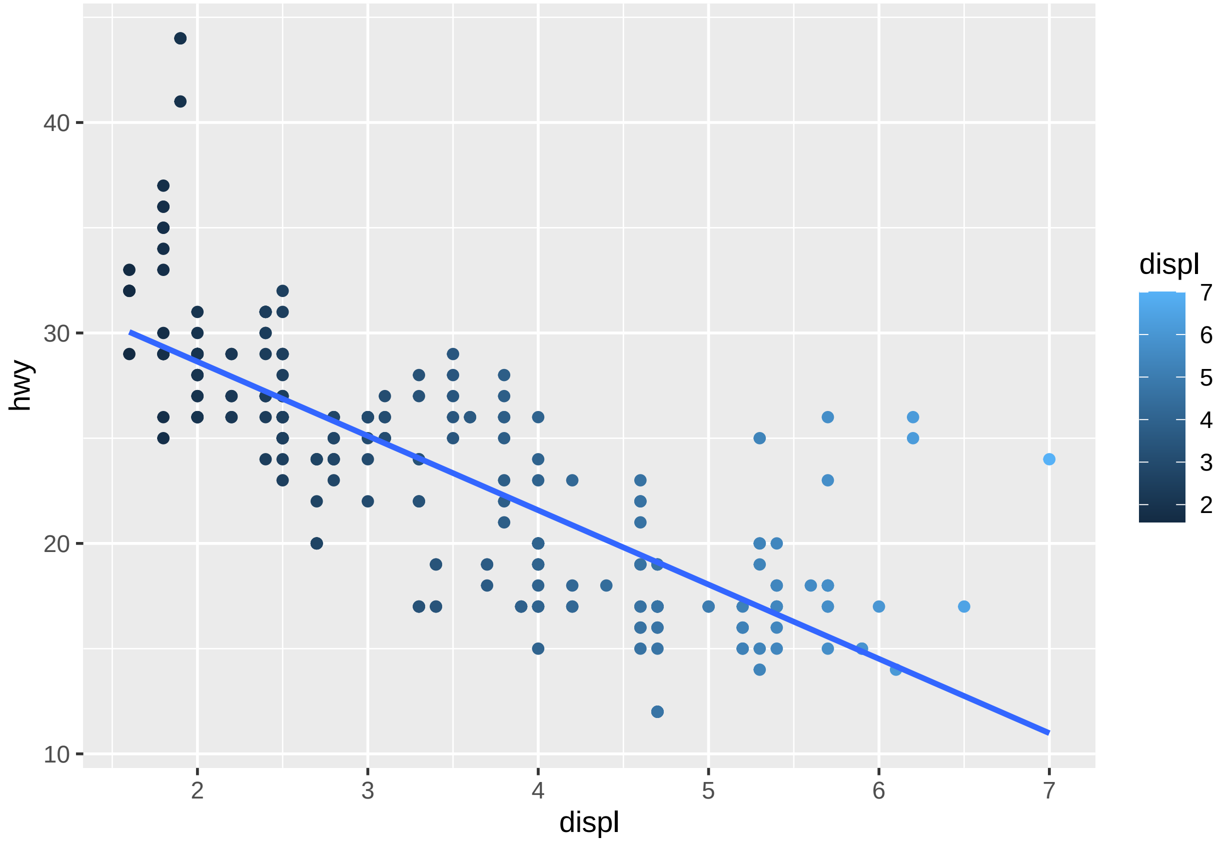
```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



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)) +
  geom_point(mapping=aes(color=displ)) +
  geom_smooth(se =FALSE,method = lm)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```


hwy
40
30
20
10
2
3
4
5
6
7
displ
displ
7
6
5
4
3
2