RWorksheet_Taltal#6

Mike Anthony Taltal

2022-11-25

#Use the dataset mpg

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
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" ...
```

```
library(tinytex)
```

```
data("mpg")
str("mpg")
```

```
##  chr "mpg"
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

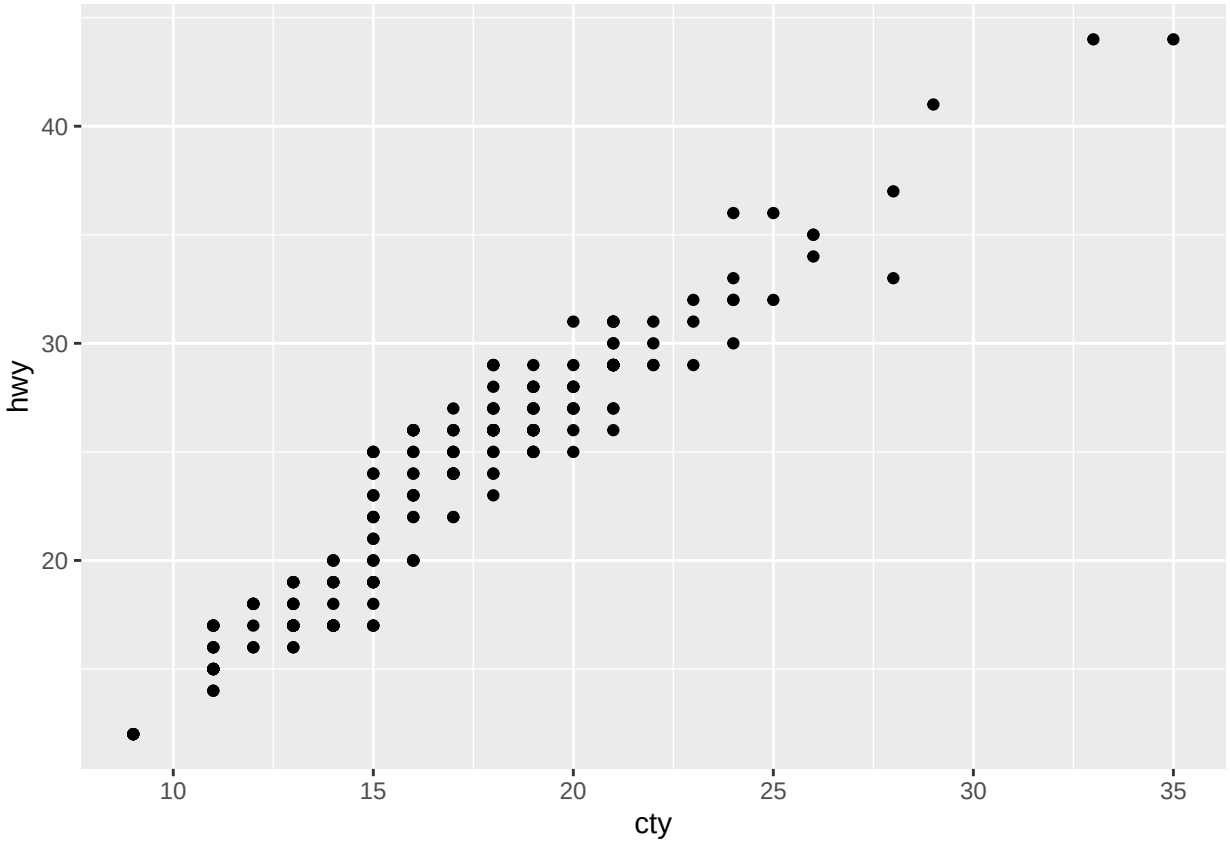
```
##
## Attaching package: 'dplyr'

## 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~
```

#to get the mpg dataset, load the ggplot package first data(mpg) #as.data.frame(data(mpg)) #converting from list to data frame

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



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

```
ncol(mpg)
```

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

```
nrow(mpg)
```

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

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

```
datampg <- mpg
num2a <- datampg %>% group_by(manufacturer, model) %>%
  distinct() %>% count()
num2a
```

```
## # 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(num2a) <- c("Manufacturer", "Model","Counts")
num2a
```

```
## # 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
```

the most models in data sets is dodge which consists of 37 variations

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

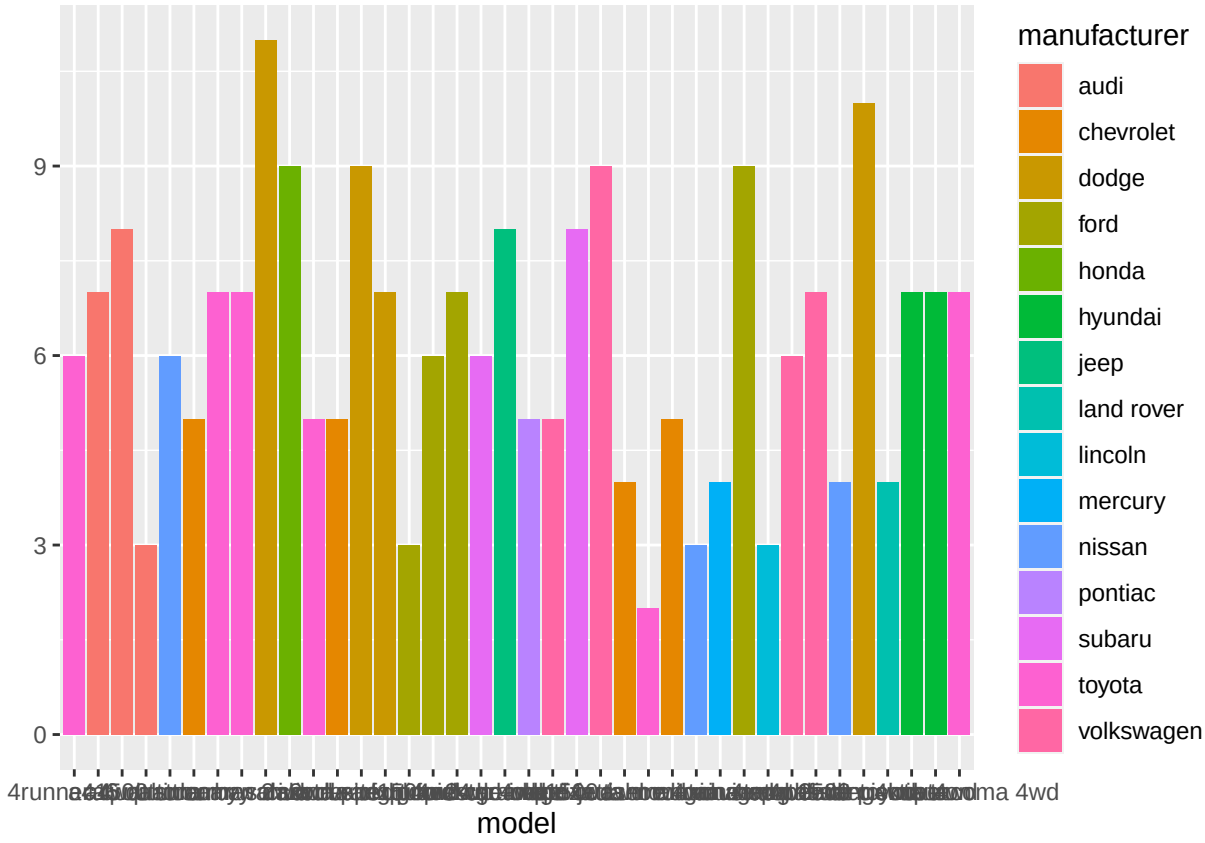
```
datampg <- mpg
a <- datampg %>% group_by(manufacturer, model) %>% distinct() %>% count()
a
```

```
## # 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
```

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

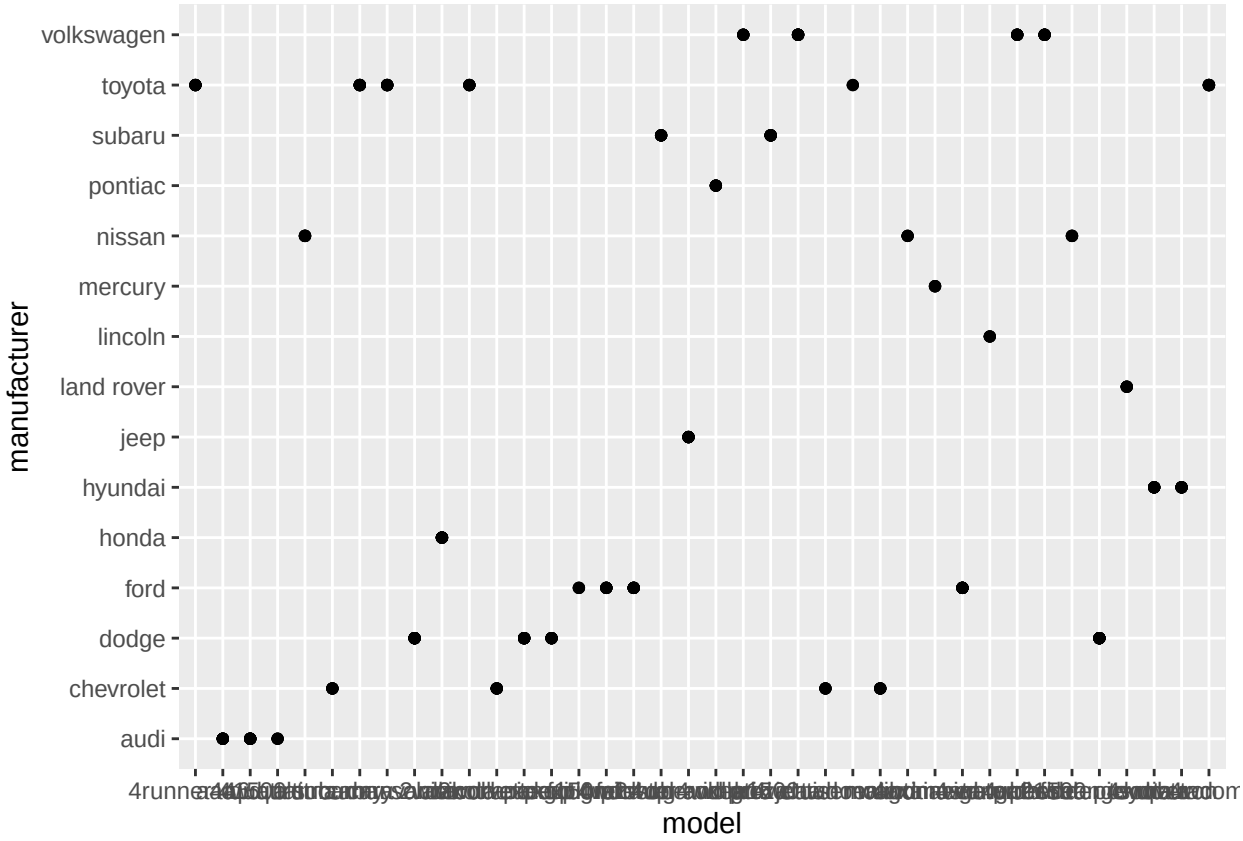
```
qplot(model, data = mpg,geom = "bar", fill=manufacturer)
```

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



ggplot

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



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

```
datampg <- mpg
b <- datampg %>% group_by(manufacturer, model) %>% distinct() %>% count()
b
```

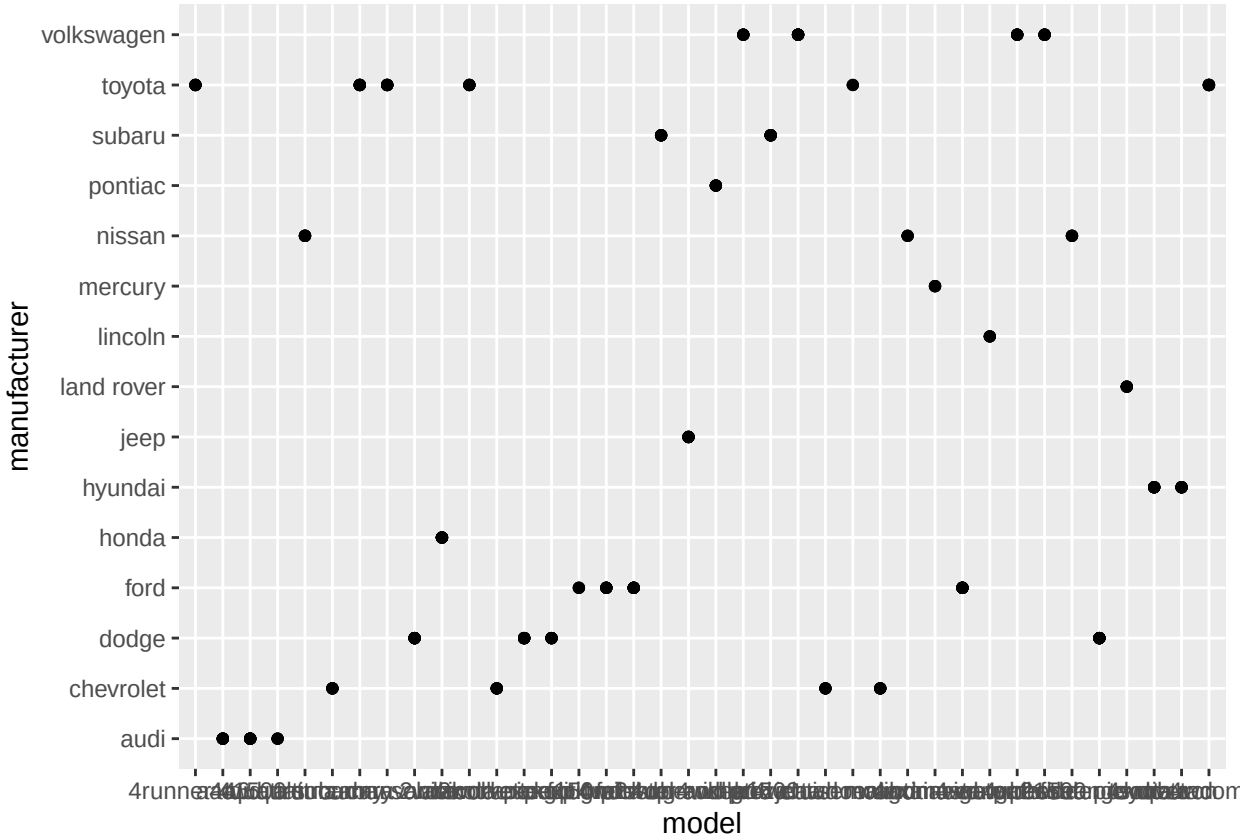
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet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

```
## # A tibble: 38 x 3
## # Groups:   Manufacturer, Model [38]
##    Manufacturer Model                ``
##    <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
```

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?

Yes, it is useful because you can track down the data for each model of the manufacturer and modify it.

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

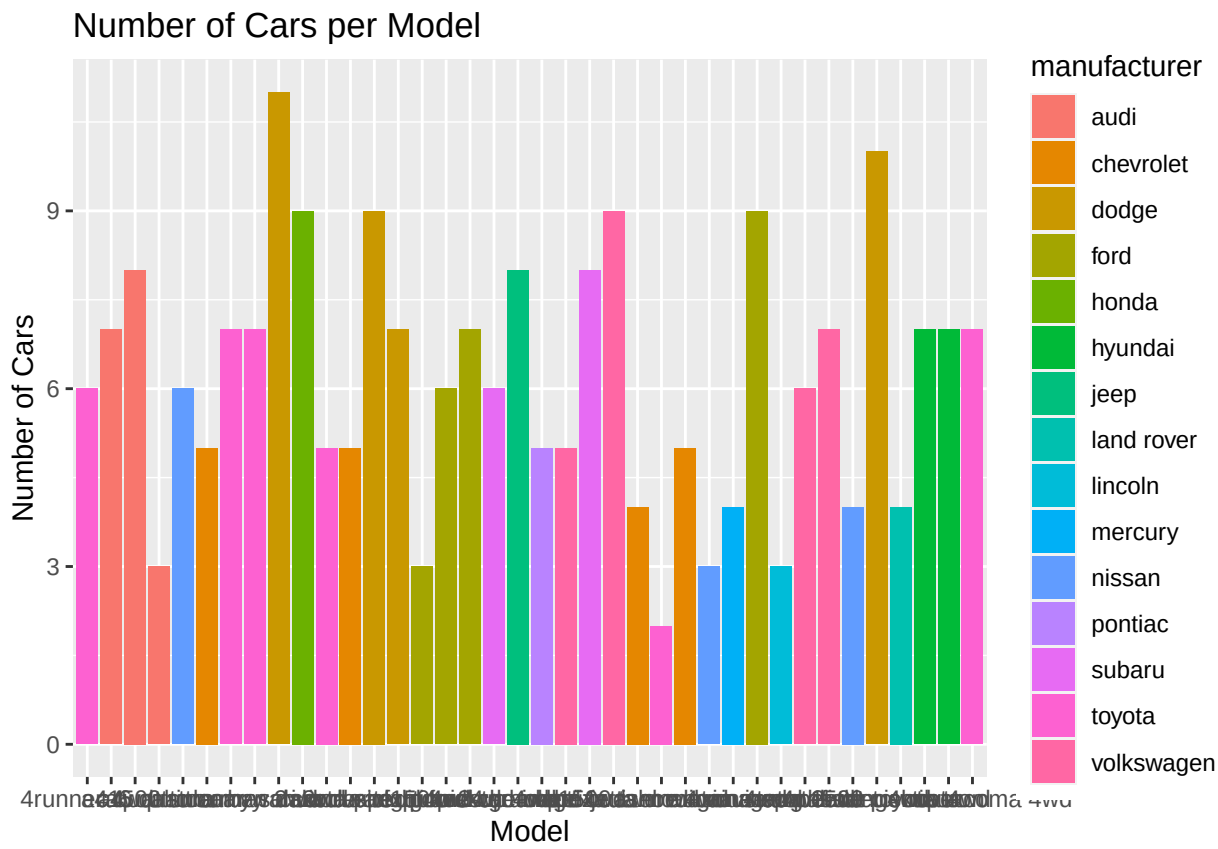
```
e <- num2a %>% group_by(Model) %>% count()
e
```

```
## # 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(e) <- c("Model","Counts")
```

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()
```

```
## <ggproto object: Class CoordFlip, CoordCartesian, Coord, gg>
##     aspect: function
##     backtransform_range: function
##     clip: on
##     default: FALSE
##     distance: function
##     expand: TRUE
##     is_free: function
##     is_linear: function
##     labels: function
##     limits: list
##     modify_scales: function
##     range: function
##     render_axis_h: function
##     render_axis_v: function
##     render_bg: function
##     render_fg: function
##     setup_data: function
##     setup_layout: function
##     setup_panel_guides: function
##     setup_panel_params: function
##     setup_params: function
##     train_panel_guides: function
```

```
##     transform: function
##     super:  <ggproto object: Class CoordFlip, CoordCartesian, Coord, gg>
```

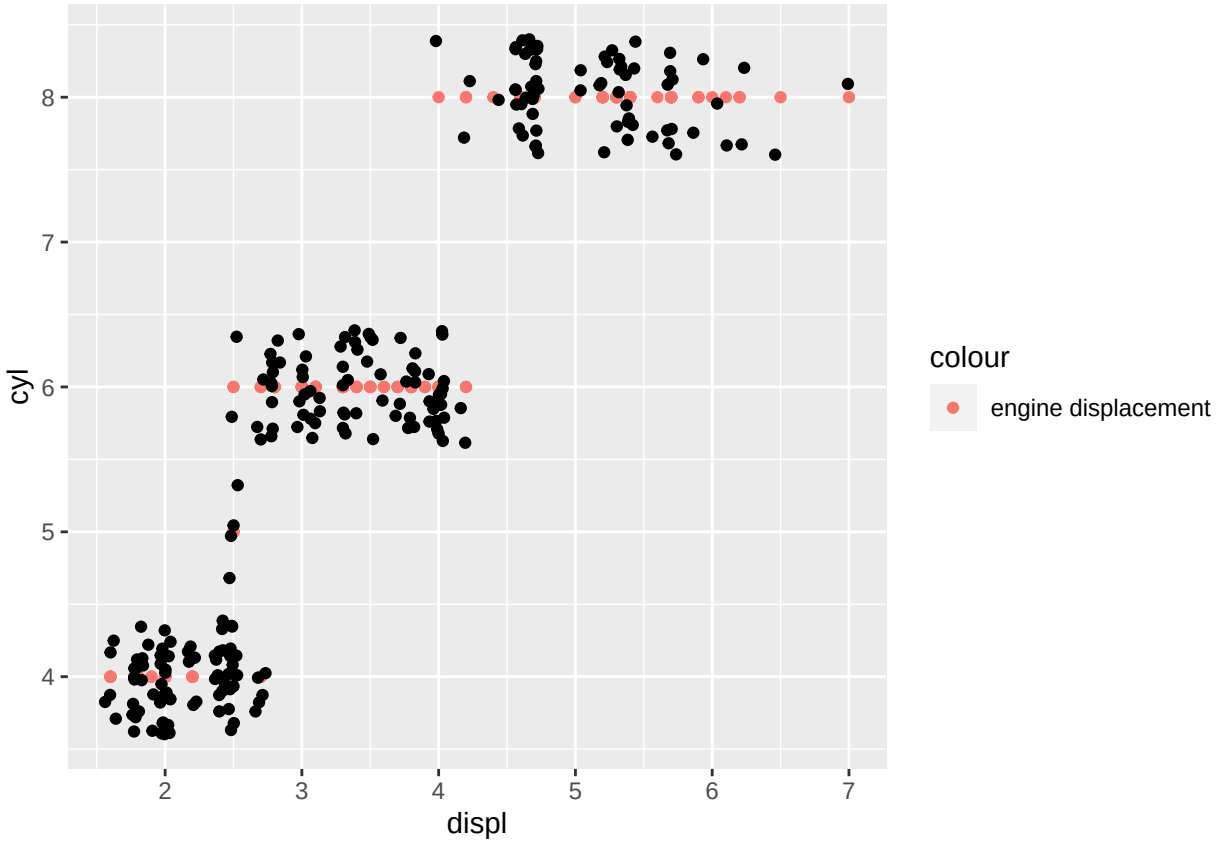
b. Use only the top 20 observations. Show code and results.

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

```
## # A tibble: 20 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~
## 11 audi         a4 quattro   2    2008     4 auto~ 4        19    27 p     comp~
## 12 audi         a4 quattro   2.8  1999     6 auto~ 4        15    25 p     comp~
## 13 audi         a4 quattro   2.8  1999     6 manu~ 4        17    25 p     comp~
## 14 audi         a4 quattro   3.1  2008     6 auto~ 4        17    25 p     comp~
## 15 audi         a4 quattro   3.1  2008     6 manu~ 4        15    25 p     comp~
## 16 audi         a6 quattro   2.8  1999     6 auto~ 4        15    24 p     mids~
## 17 audi         a6 quattro   3.1  2008     6 auto~ 4        17    25 p     mids~
## 18 audi         a6 quattro   4.2  2008     8 auto~ 4        16    23 p     mids~
## 19 chevrolet    c1500 sub~   5.3  2008     8 auto~ r        14    20 r     suv
## 20 chevrolet    c1500 sub~   5.3  2008     8 auto~ r        11    15 e     suv
```

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

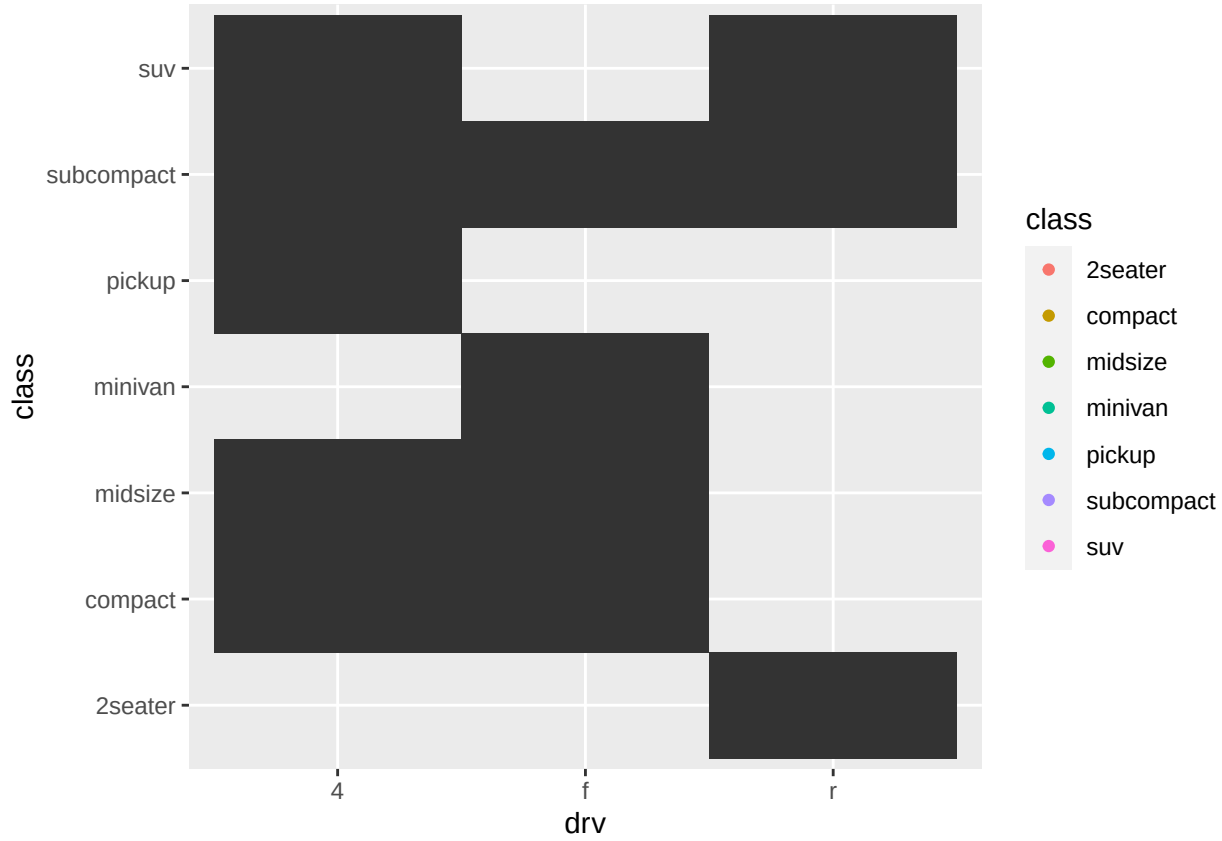
```
ggplot(data = mpg , mapping = aes(x = displ, y = cyl,
main = "Relationship between No of Cylinders and Engine Displacement")) +
geom_point(mapping=aes(colour = "engine displacement")) + geom_jitter()
```



b. How would you describe its relationship? The relationship between data is makking cyl, and y jittered, and the pink color indicates engine displacement.

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_point(mapping=aes(color=class)) +
  geom_tile()
```

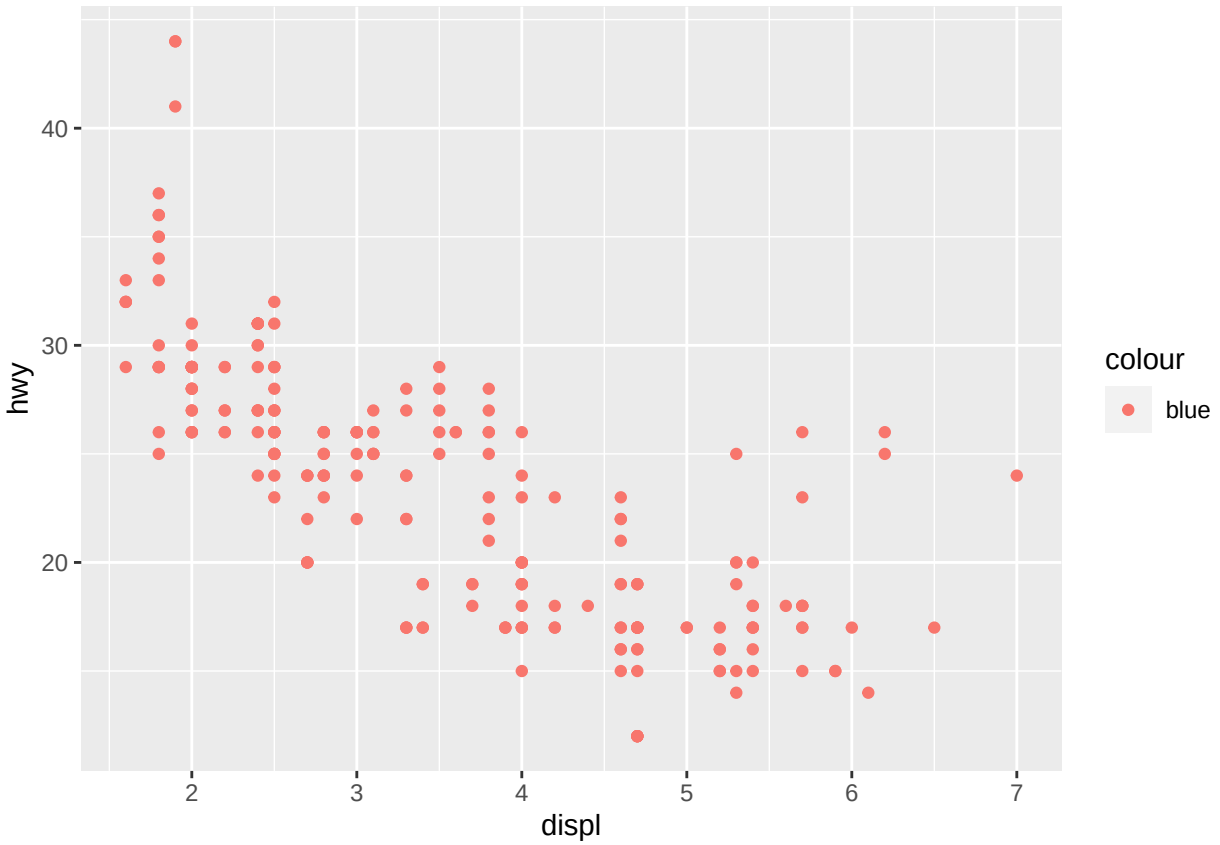


b. Interpret the result: The mapping geometric point graph is used to "map" the areas that are covered in black. x as drv and y as class.

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

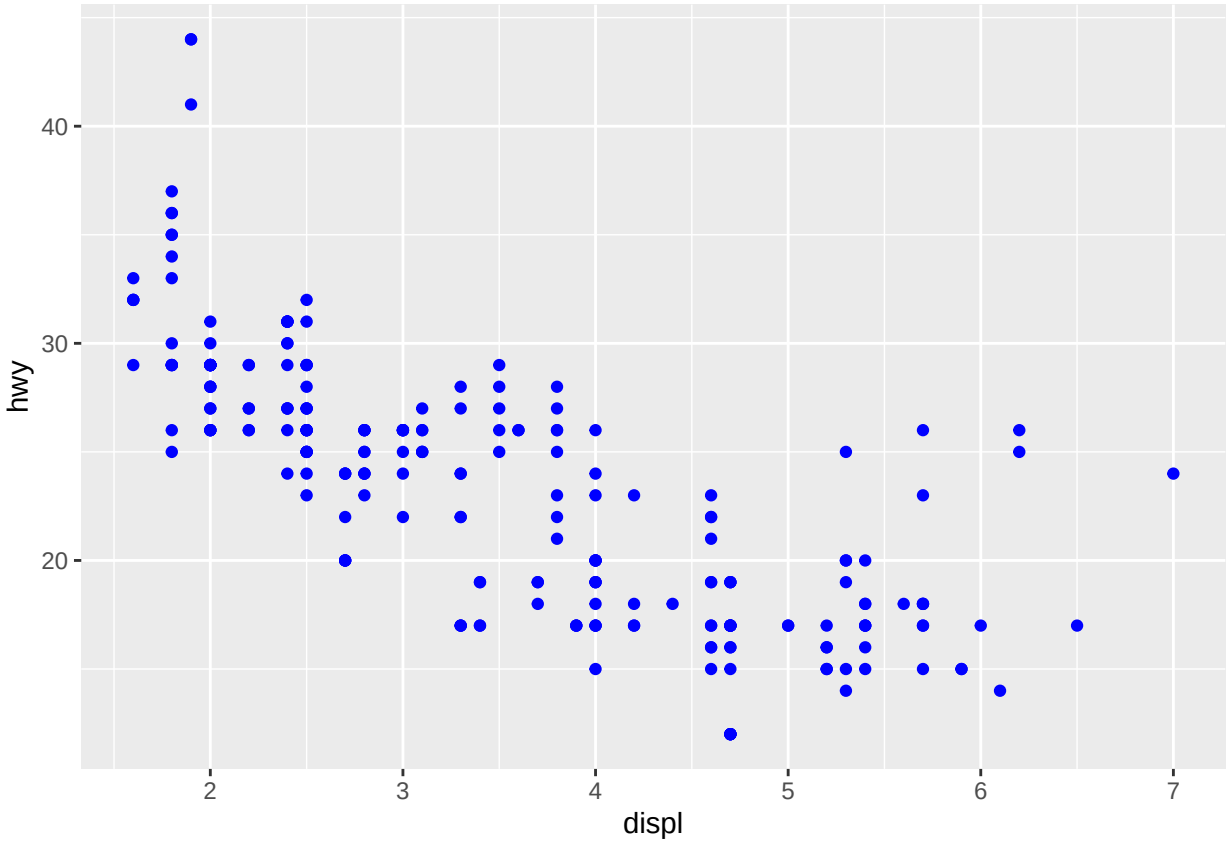
#Code1

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



#Code2

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



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

```
?mpg
```

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

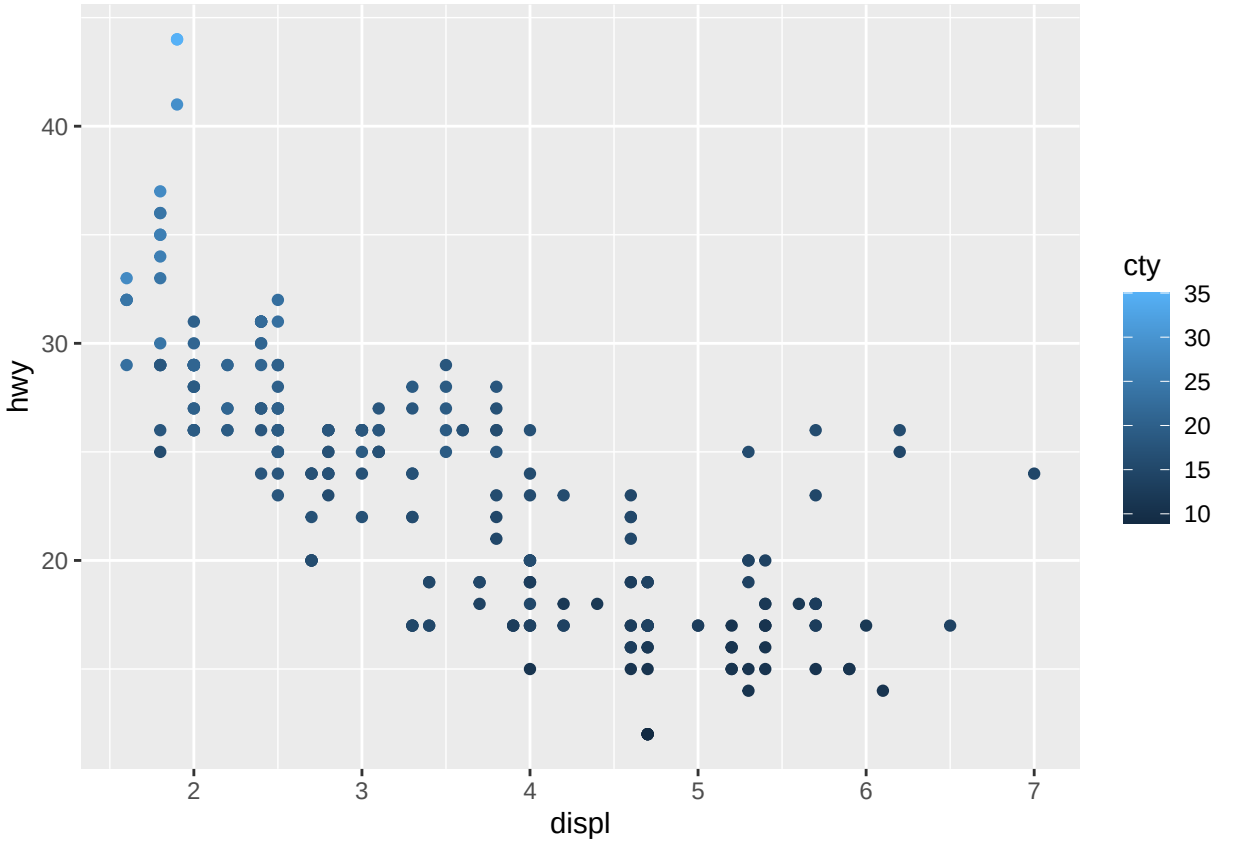
The result of this command mpg it shows the company data from 1999 to 2008 for #38 models of cars.

a. Which variables from mpg dataset are categorical? The data set are categorized according to their manufacturers name, model name,engine displacement, in litres, year of manufacture, number of cylinders Type of transmission, he type of drive train, where f = front-wheel drive, r = rear wheel drive, 4 = 4wd, city miles per gallon, highway miles per gallon highway miles per gallon, and type of car.

b. Which are continuous variables? Continuous varibles in R was also known as doubles or integers.

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.

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

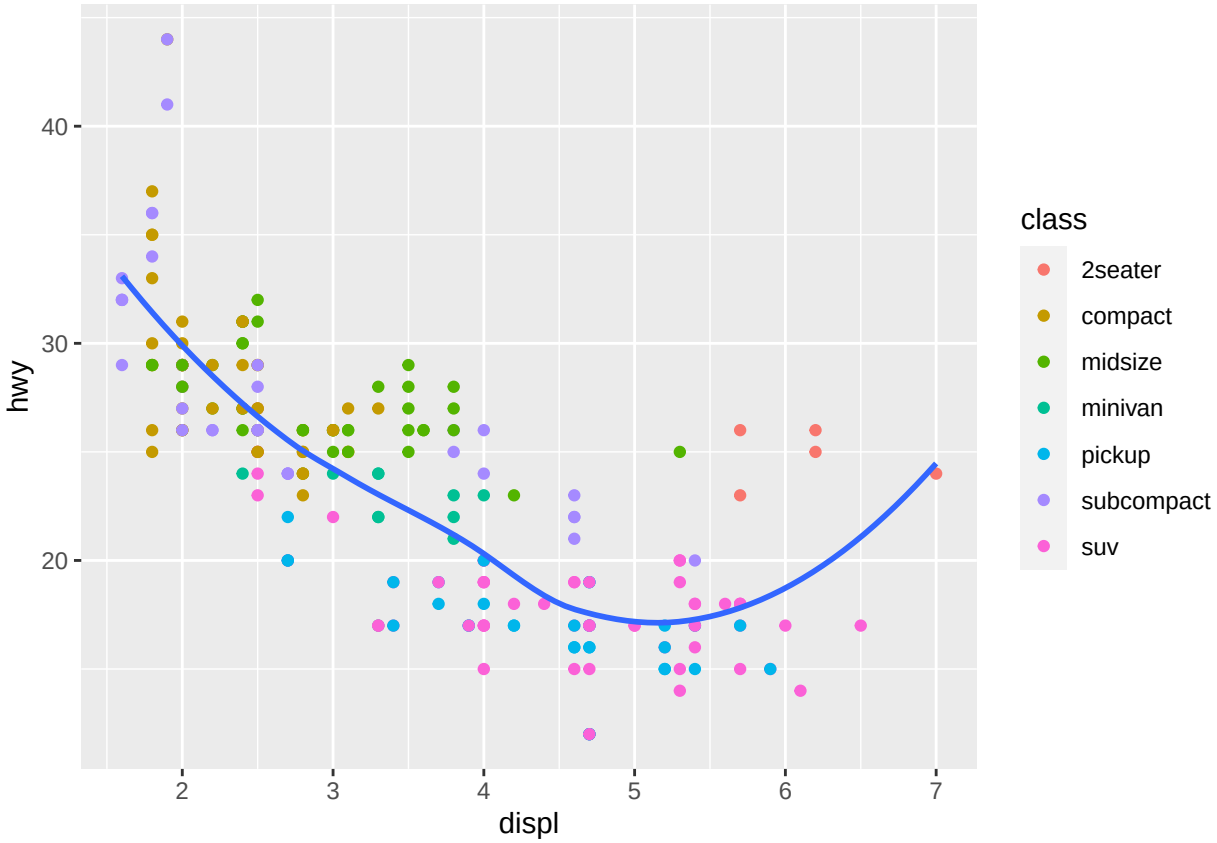


What is its result? Why it produced such output? The ata tracks the cty by placing cty(city miles per gallon) at color having a variation or hues of blue.

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)
```

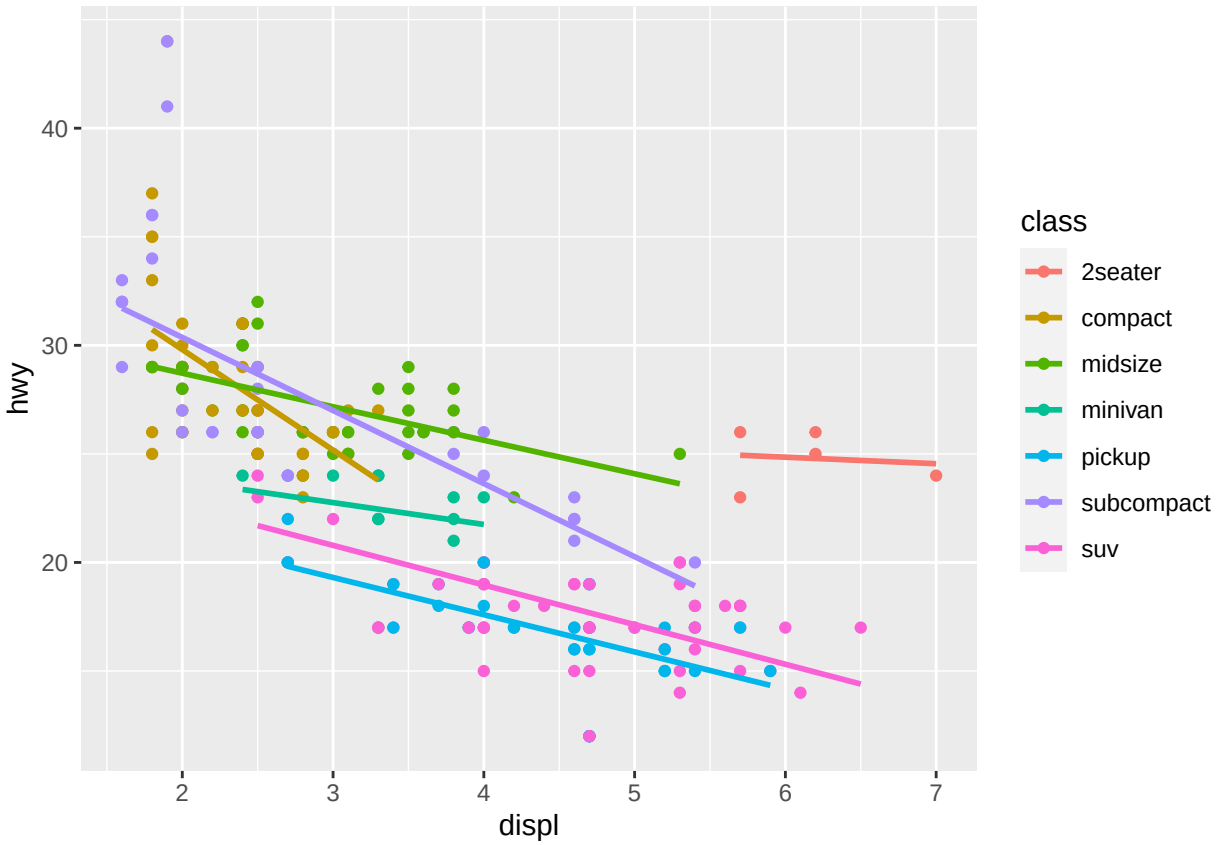
```
## `geom_smooth()` using formula = 'y ~ x'
```



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'
```


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