

# RworkSheet - 6

Mamon, Jasmin Mae G.

2022-11-24

## R Markdown

### Worksheet for R Programming

#### Instructions:

- Use RStudio or the RStudio Cloud accomplish this worksheet.
- Save the R script as RWorksheet\_lastname#6.R.
- On your own GitHub repository, push the R script, the Rmd file, as well as this pdf worksheet to the repo you have created before.
- Do not forget to comment your Git repo on our VLE
- Accomplish this worksheet by answering the questions being asked and writing the code manually.

```
library(ggplot2)
```

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

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

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

```
data(mpg)
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
```

```
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)
```

```
## 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)
```

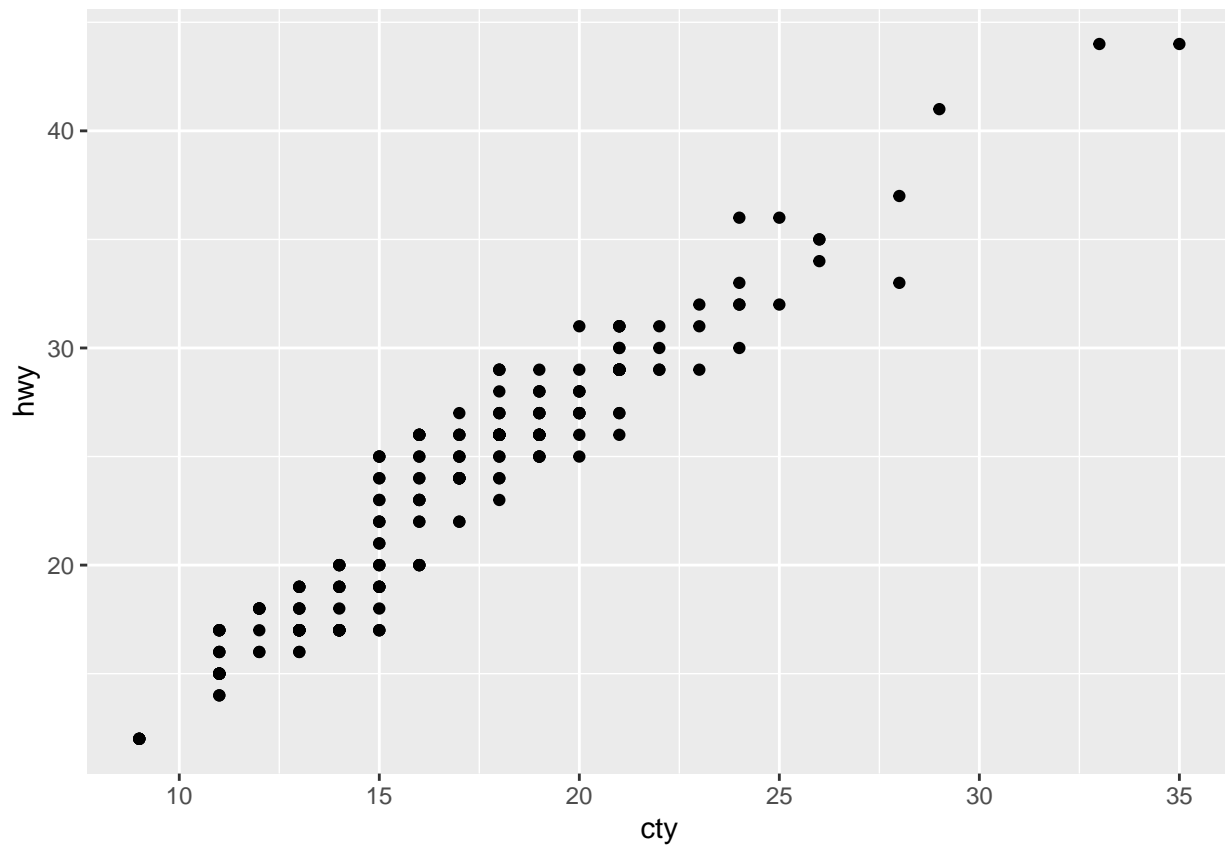
```
## Rows: 234
```

```
## 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", "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", "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 dataset? How about the number of rows? Show the codes and its result. There are 11 columns and 234 rows in the mpg data frame.

```
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", "p~  
## $ class       <chr> "compact", "compact", "compact", "compact", "compact", "c~
```

```
nrow(mpg)
```

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

```
ncol(mpg)
```

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

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

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

Answer: dodge has 37 models

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

```
data_mpg <- mpg  
two_a <- data_mpg %>% group_by(manufacturer, model) %>%  
  distinct() %>% count()  
two_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
```

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

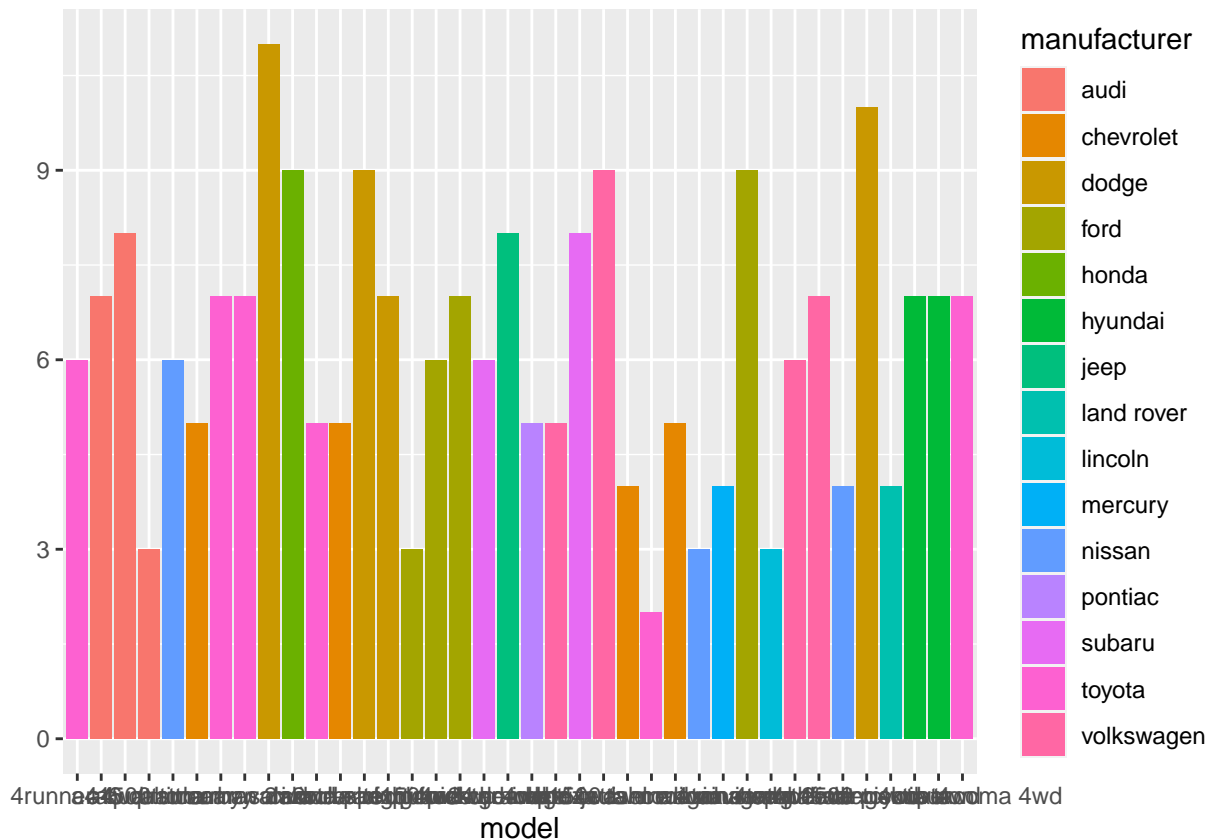
```
## # A tibble: 38 x 3  
## # Groups:   Manufacturer, Model [38]  
##   Manufacturer Model      Counts  
##   <chr>         <chr>    <int>  
## 1 audi         a4          7  
## 2 audi         a4 quattro    8
```

```
## 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
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()
```

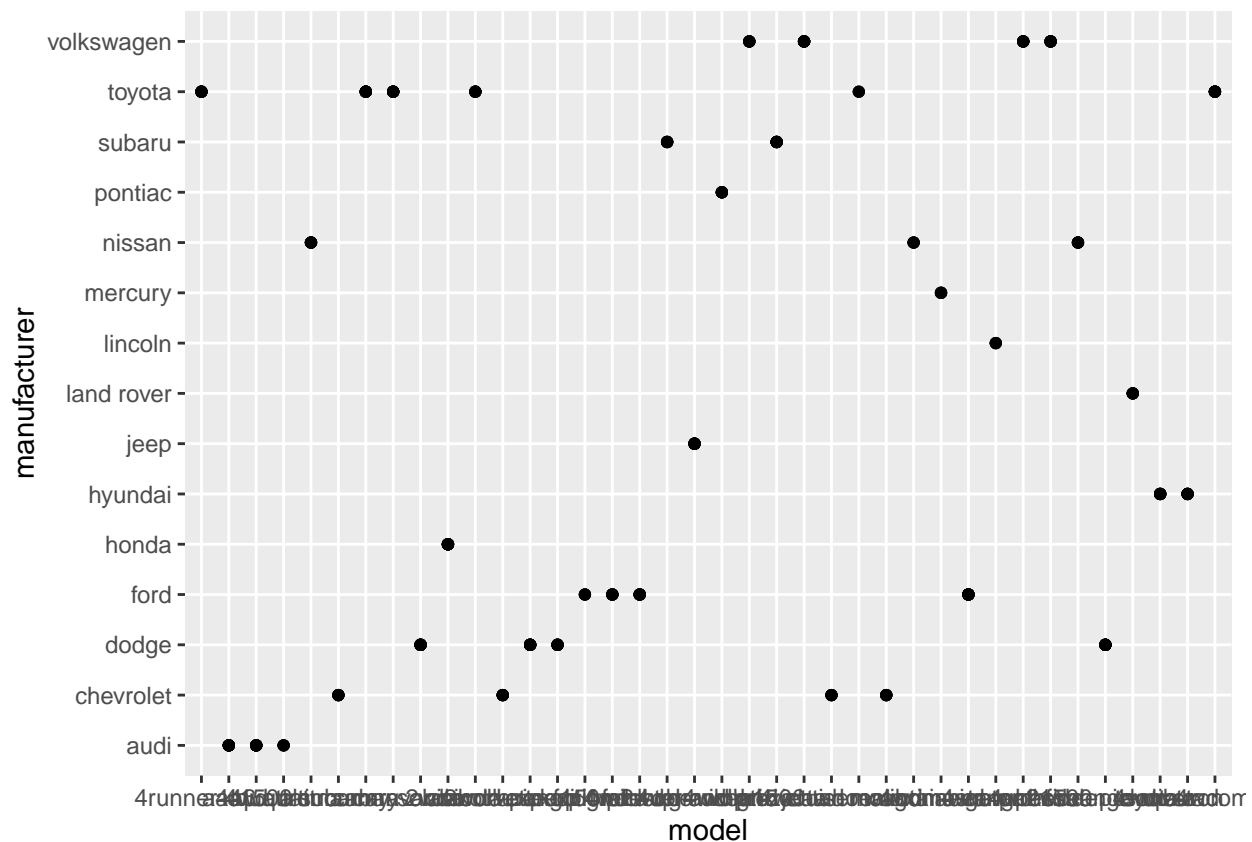


```
colnames(model_fact) <- c("Manufacturer", "Model")
model_fact
```

```
## # 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()
```



Answer: The plot shows the geometric point graph of mpg's model and manufacturer.



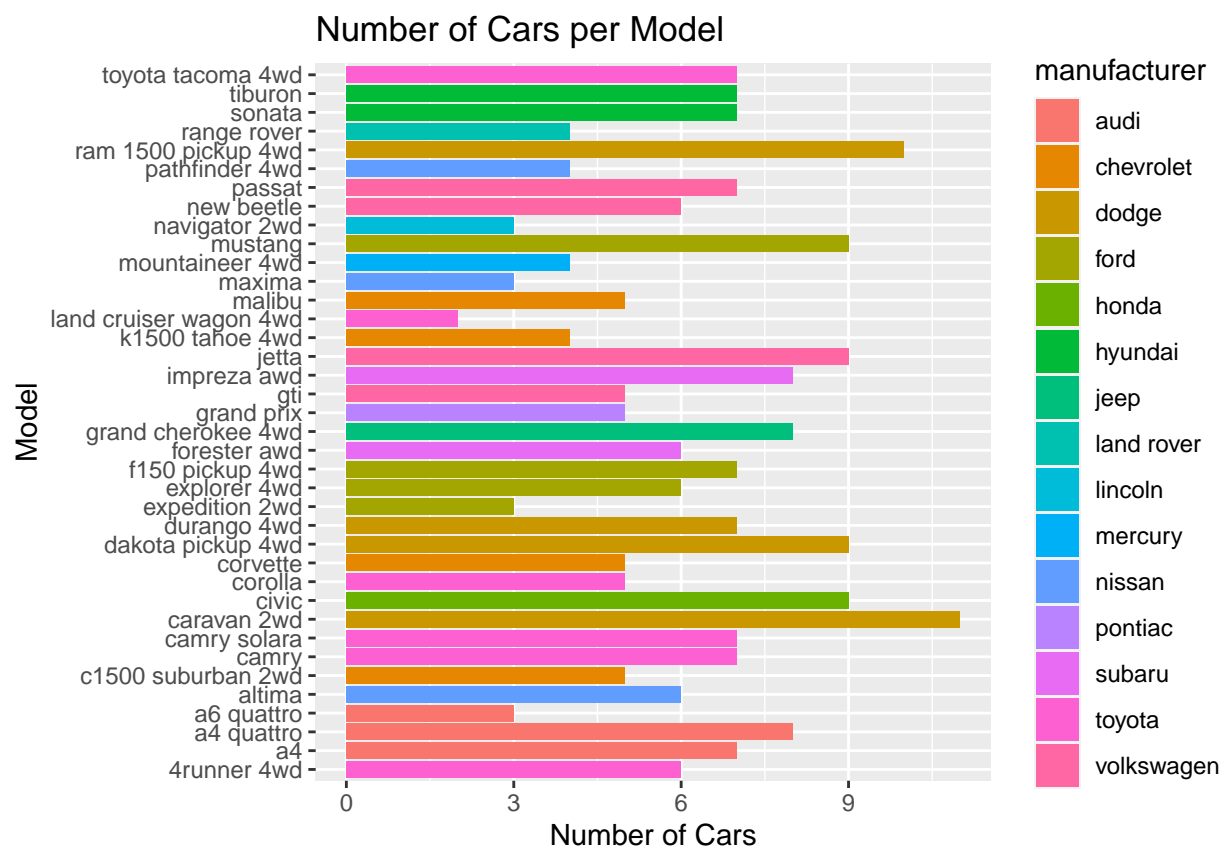


```
## 7 camry 1
## 8 camry solara 1
## 9 caravan 2wd 1
## 10 civic 1
## # ... with 28 more rows
```

```
colnames(datampg) <- 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 =
```



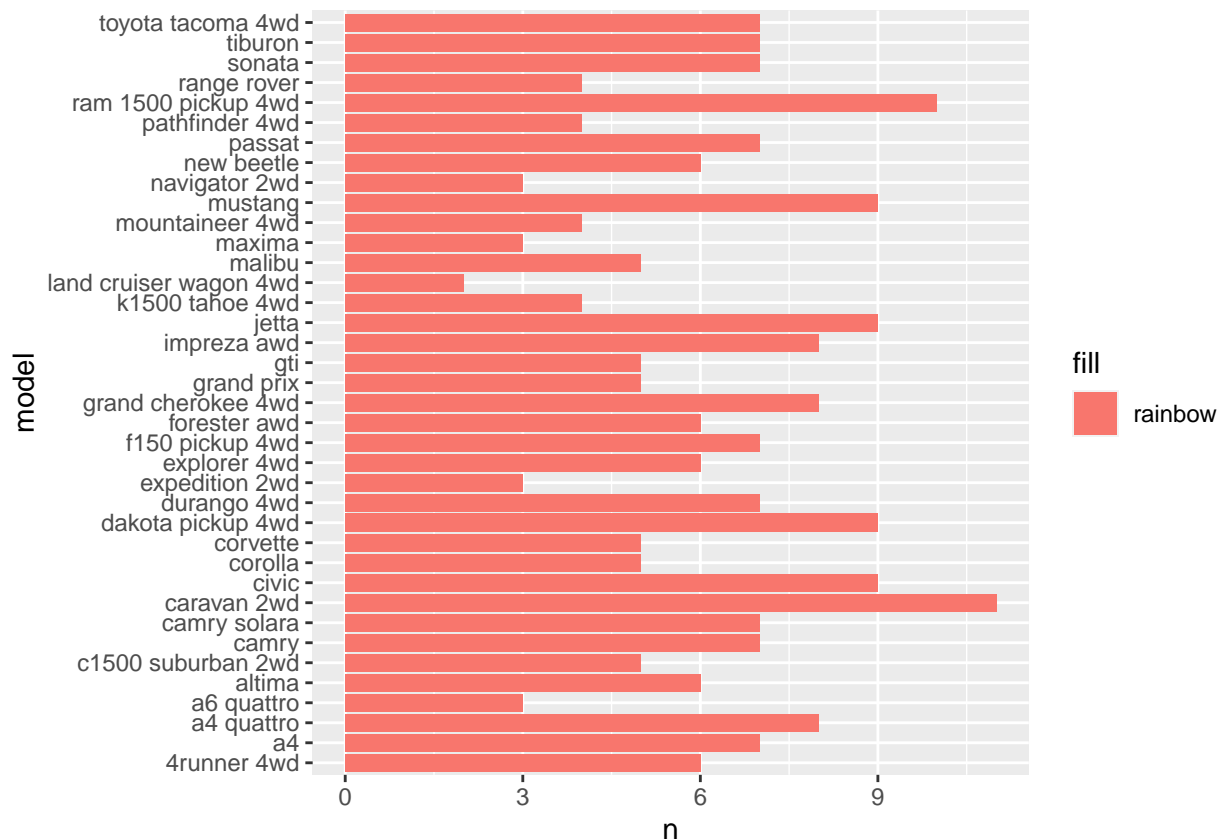
- b. Use only the top 20 observations. Show code and results.

```
cars_Model <- mpg %>%
  group_by(model) %>%
  tally(sort = TRUE)
cars_Model
```

```
## # A tibble: 38 x 2
##   model          n
##   <chr>        <int>
```

```
## 1 caravan 2wd      11
## 2 ram 1500 pickup 4wd 10
## 3 civic            9
## 4 dakota pickup 4wd 9
## 5 jetta            9
## 6 mustang          9
## 7 a4 quattro       8
## 8 grand cherokee 4wd 8
## 9 impreza awd      8
## 10 a4              7
## # ... with 28 more rows
```

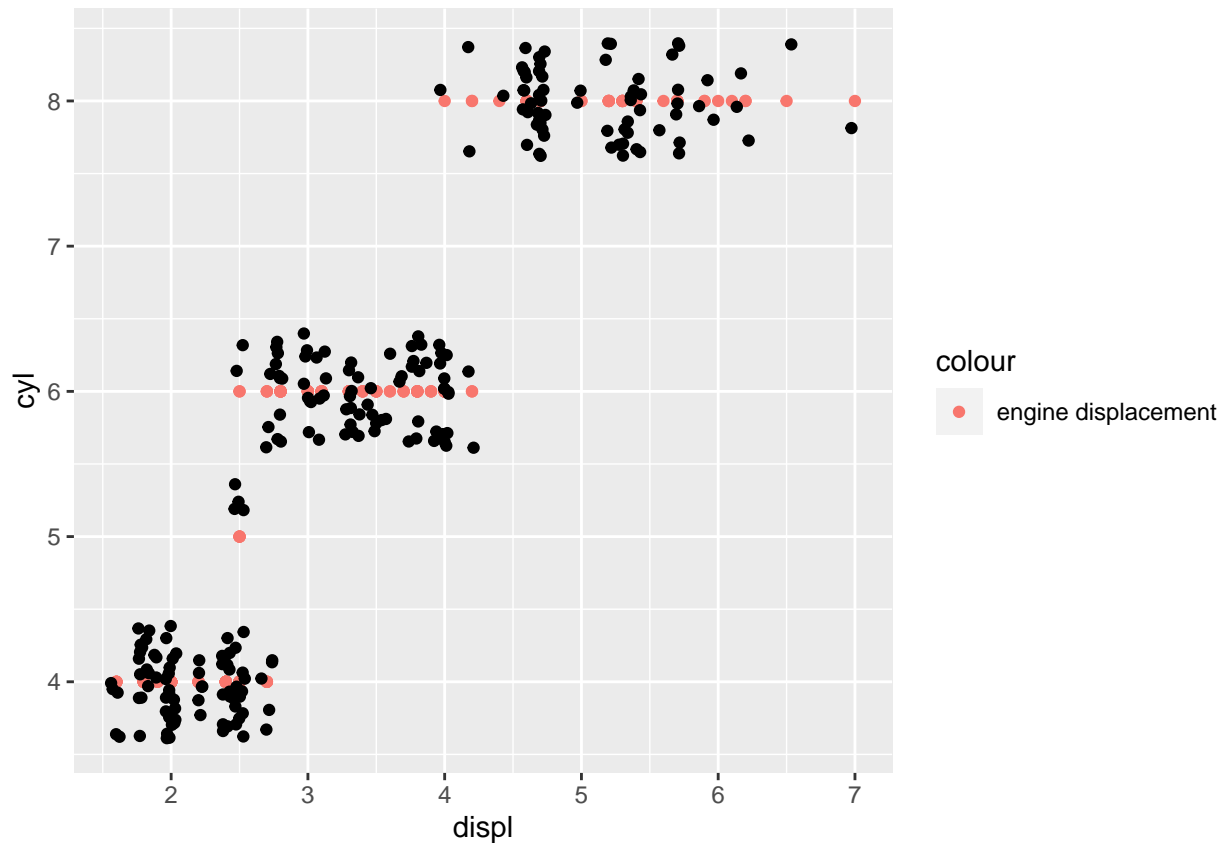
```
ggplot(cars_Model, aes(x = model, y = n, fill = "rainbow")) +
  geom_bar(stat = "identity") + coord_flip()
```



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 = 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()
```

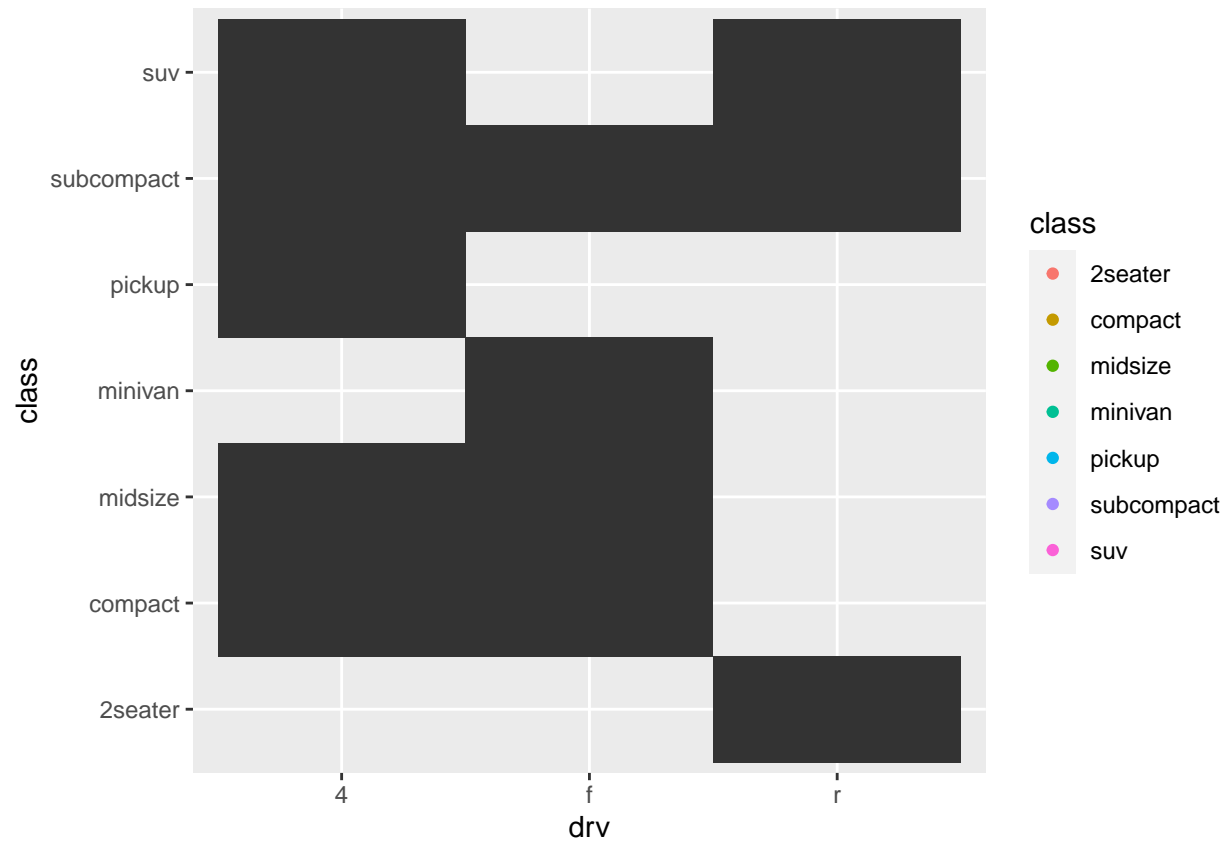


b. How would you describe its relationship? According to my data of making cyl the y, the jittered graph displays the pink and black color. The pink indicates the engine displacement. The dots indicates the number of cylinder which is 4,5,6,8.

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()
```

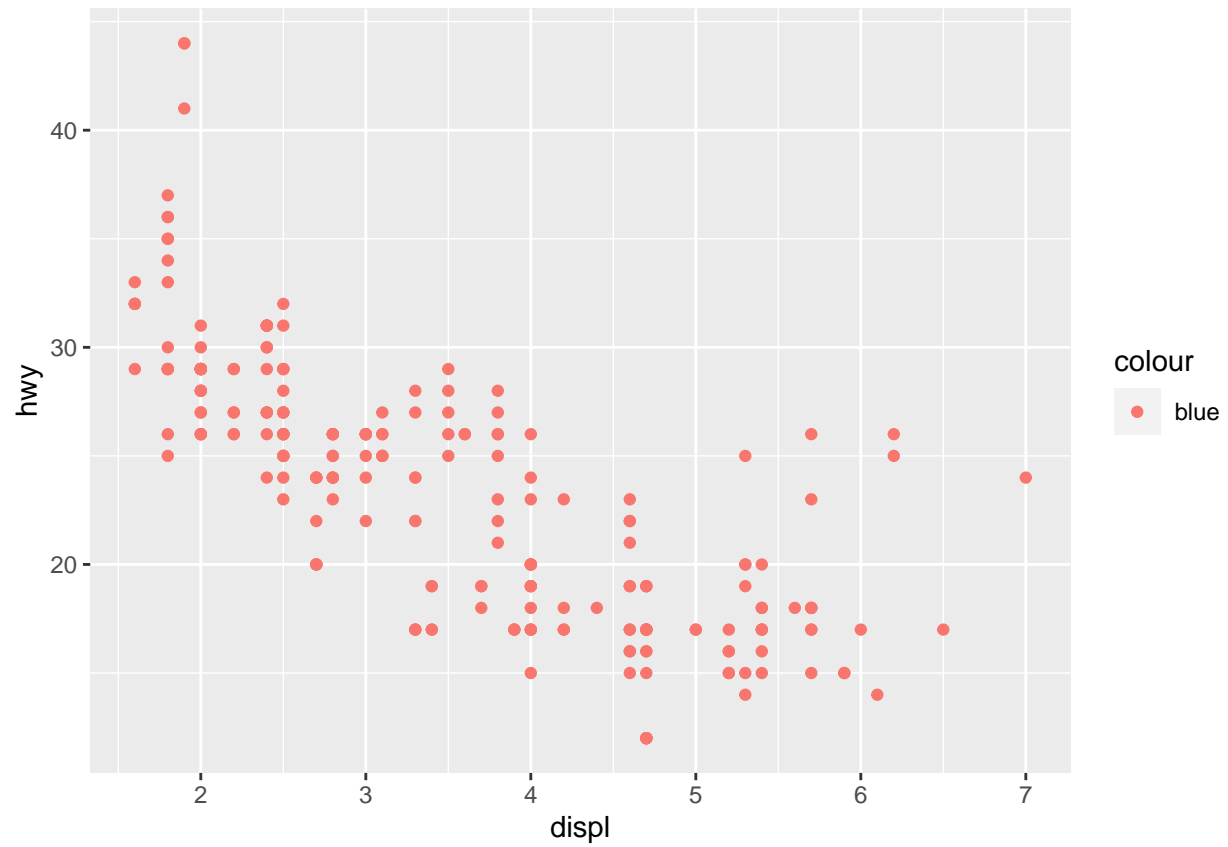


b. Interpret the result.

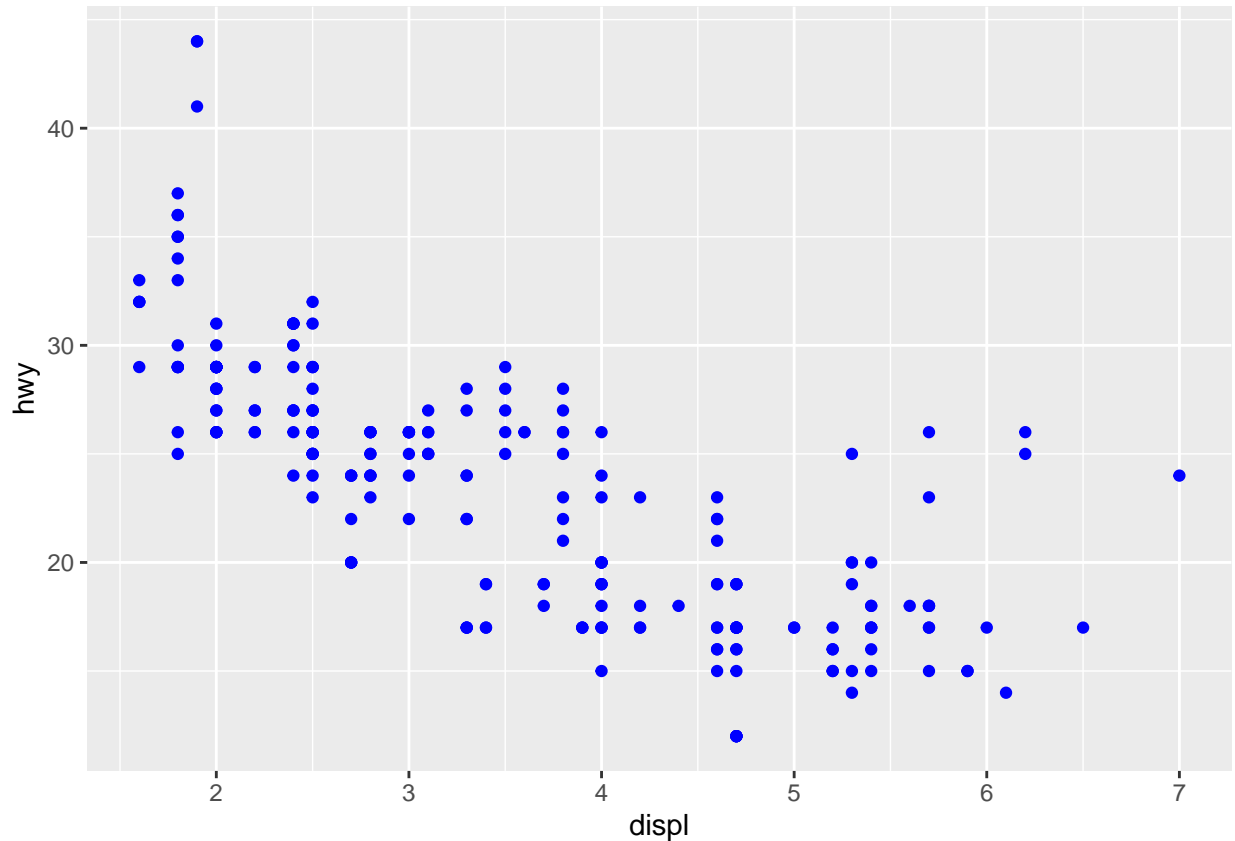
Answer: The areas that have covered with tile are mapped using the geometric point graph, y as class and x as drv.

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"))
```



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



The difference between the code #1 and #2 is that in #2 the aes or the x and y was inside the close-open parenthesis which separated with the color, blue. While the code #1 the x and y was not enclosed that result the mix of color and since the color red is the color for engine displacement it automatically display the red color which is the engine displacement.

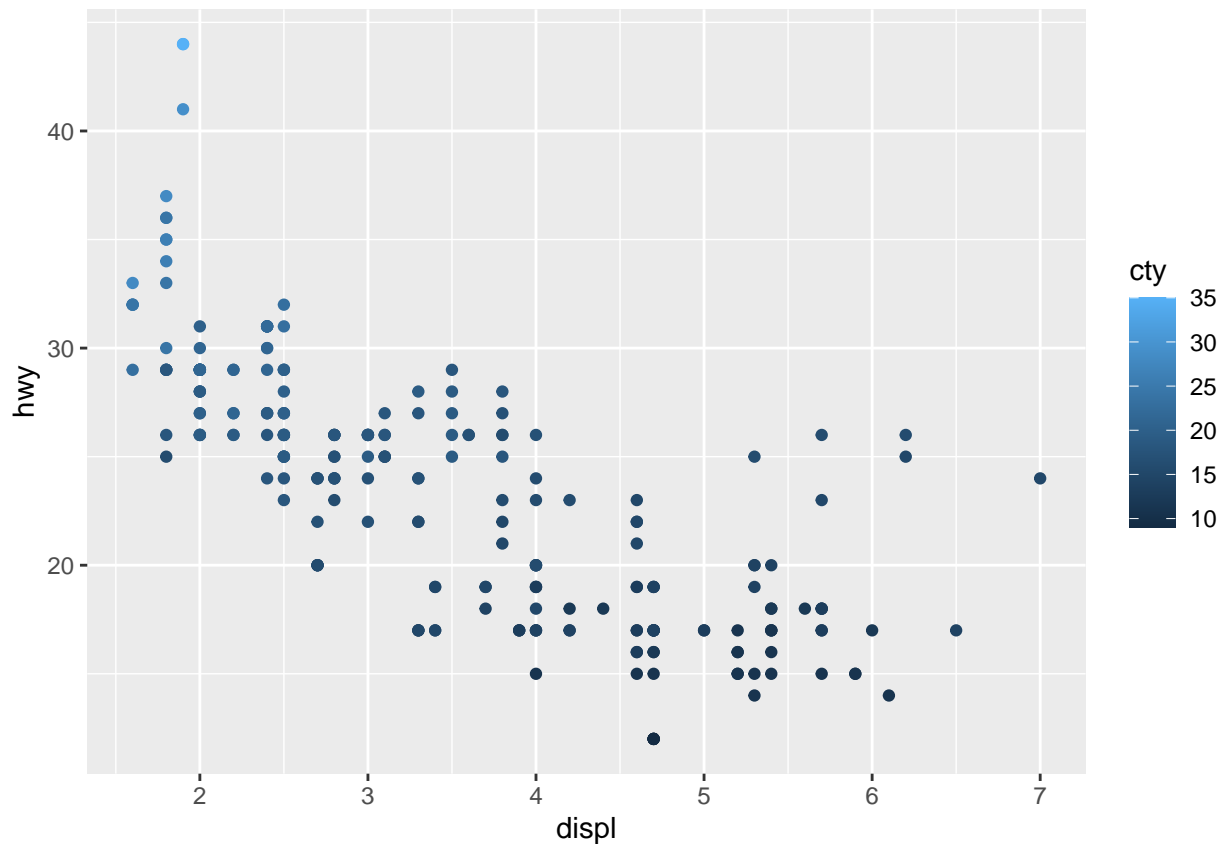
8. Try to run the command `?mpg`. What is the result of this command?

```
?mpg
```

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

- Which variables from mpg dataset are categorical? Answer: Categorical variables in mpg include: manufacturer, model, trans (type of transmission), drv (front-wheel drive, rear-wheel, 4wd), fl (fuel type), and class (type of car).
- Which are continuous variables? Answer: Continuous variables in mpg include: displ (engine displacement in litres), cyl (number of cylinders), cty (city miles/gallon), and hwy (highway gallons/mile).
- 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(displ, hwy, colour = cty)) + geom_point()
```

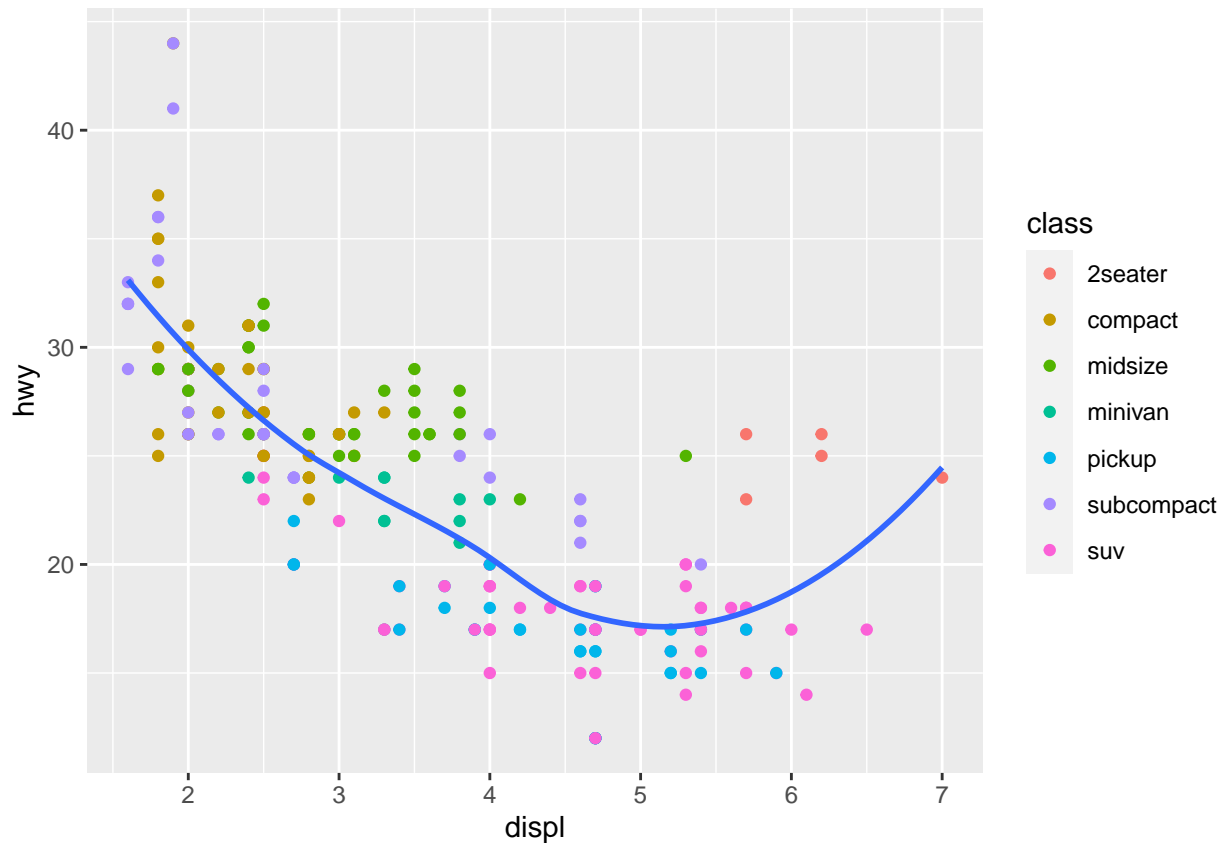


What is its result? Why it produced such output? Answer: The data tracks the cty(city miles per gallon), it also displays the engine displacement the high way gallons/miles and the number of cylinders in the hue of blue.

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(displ, hwy)) +  
  geom_point(mapping=aes(color=class)) +  
  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, color = class)) +
  geom_point() +
  geom_smooth(se = FALSE)
```

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

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 5.6935
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.5065
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.65044
```



```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : pseudoinverse used at 4.008
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : neighborhood radius 0.708
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : There are other near singularities as well. 0.25
```

