Worksheet#6

Jan James Lennox Tahumar

2022-11-24

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

```
library(ggplot2)
```

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

```
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
```

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

```
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:

```
#dodge has 37 modelss totalno <-
mpg %>% group_by(manufacturer)
%>% tally(sort = TRUE)
```

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

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

```
## # 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                                                                          3

## 7 chevrolet malibu                                                                                   3

## 8 dodge caravan 2wd 9## 9 dodge dakota pickup 4wd                                                    3


## 10 dodge             durango 4wd                   6

## # ... with 28 more rows
```

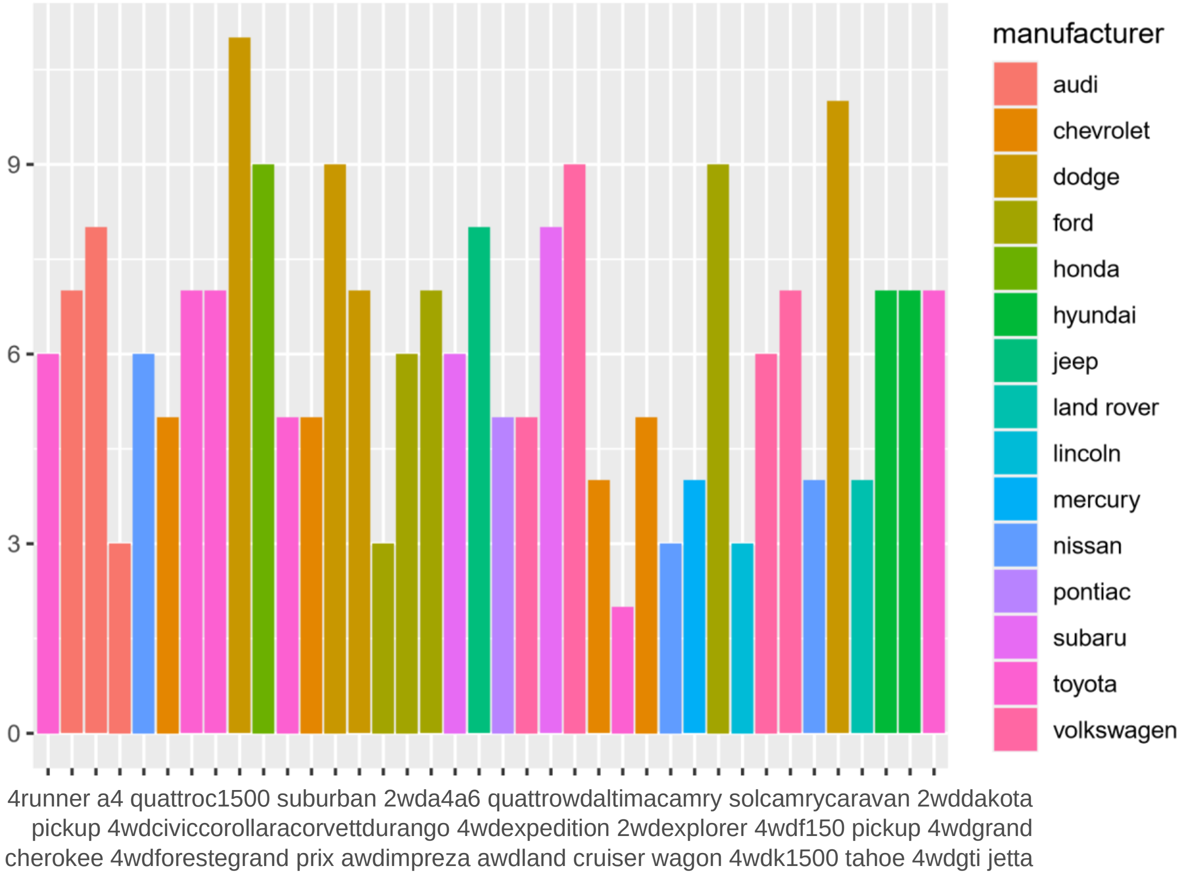
```
colnames(uniqMods) <- c("Manufacturer", "Model","Counts") uniqMods
```

```
## # 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)
```

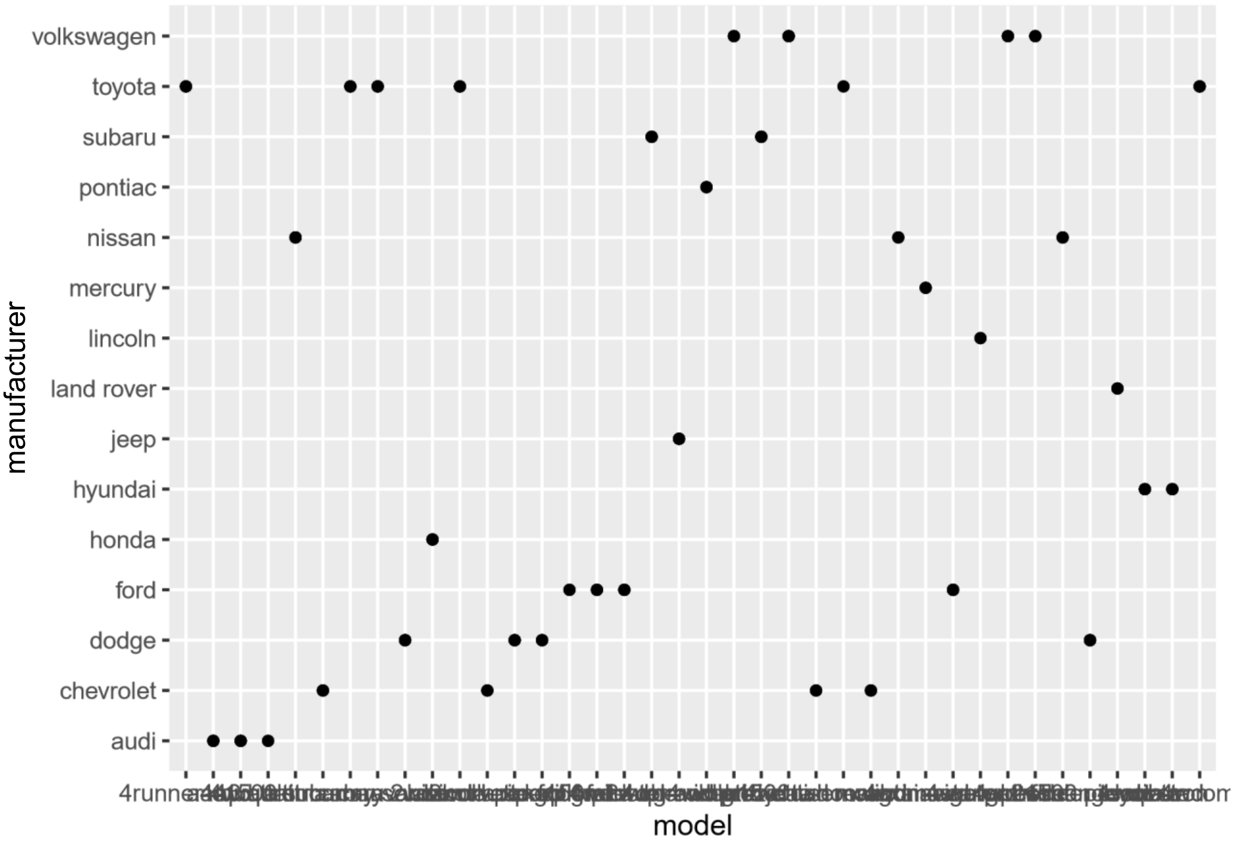
```
## Warning: ‘qplot()‘ was deprecated in ggplot2 3.4.0.
```



mountaineer 4wdmalibumaximanavigatormustnew beetleram 1500 pickup 4wdpathfinder
4wdngpassat2wdrange rovertoyota tacoma 4wdsonatatiburon

model

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



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

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

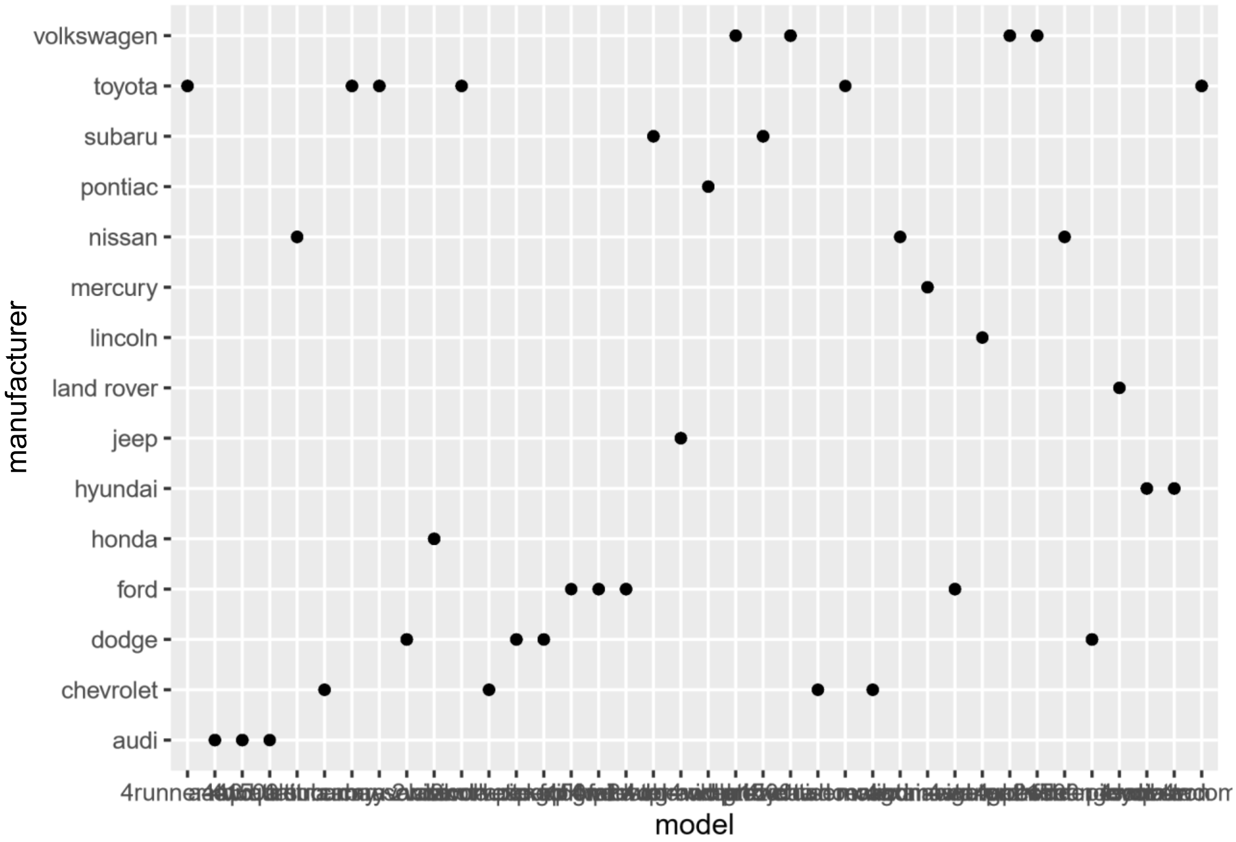
```
## # A tibble: 38 x 3
## # Groups:        manufacturer, model [38]
##     manufacturer model                  n
##    <chr>         <chr>                <int>
## 1 audi           a4                     7
## 2 audi            a4 quattro            8
## 3 audi            a6 quattro            3
##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

```
## # 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?

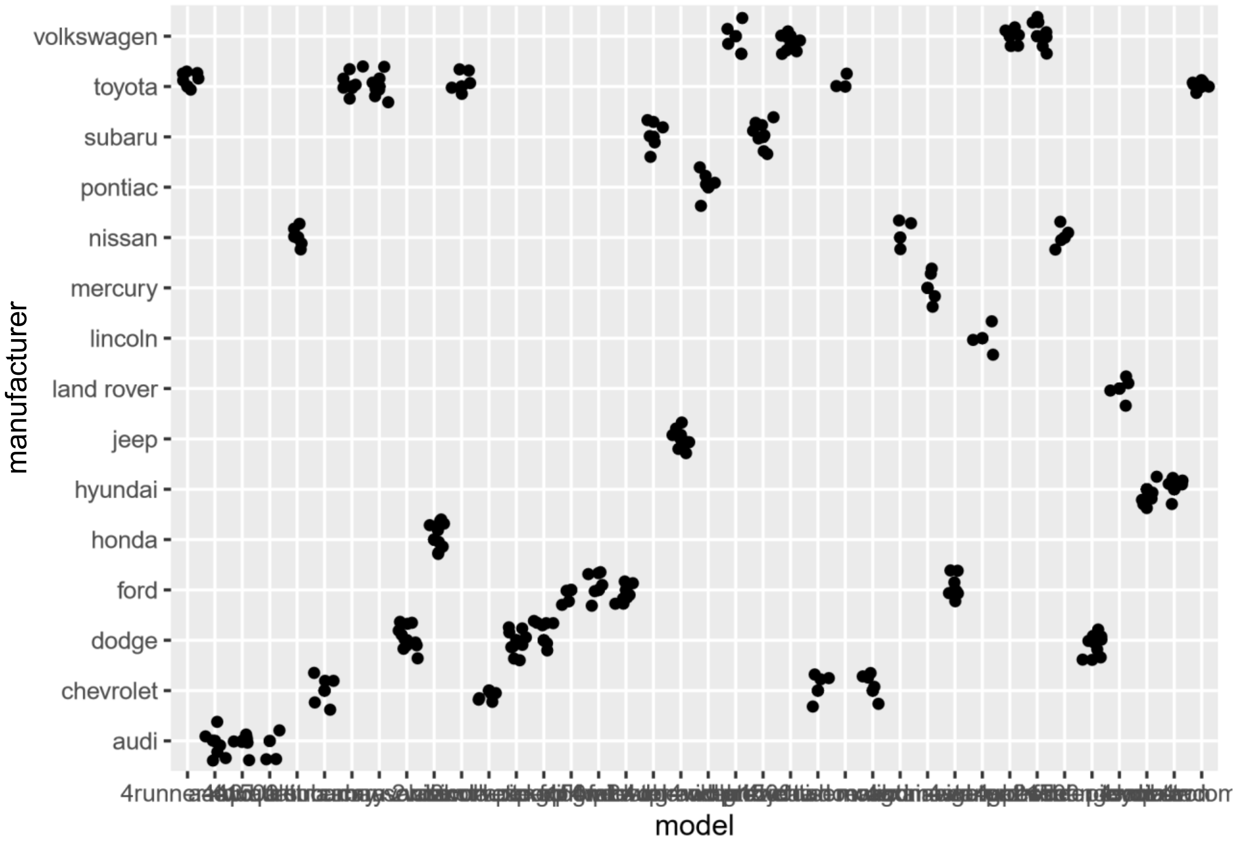
```
ggplot(mpg, aes(model, manufacturer)) + geom_point()
```



```
#geometric point graph of mpg(model and manufacturer)
```

#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 could trackdown the data of each model of the manufacturer

```
#to modify the data: ggplot(mpg,
aes(model, manufacturer)) +
geom_point() + geom_jitter()
```



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

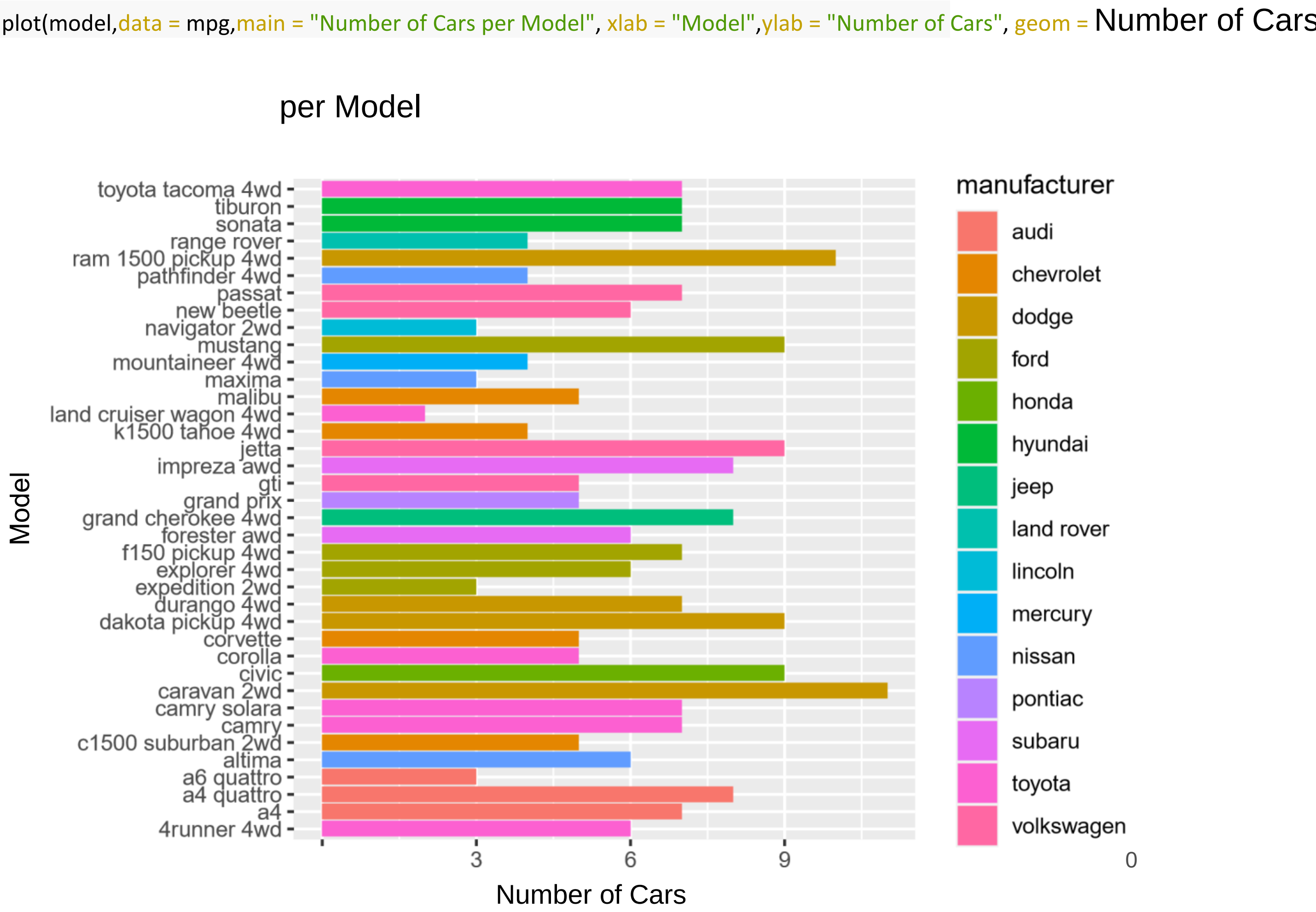
```
library(ggplot2) library(dplyr)

datampg <- uniqMods %>% group_by(Model) %>% count() datampg
```

```
## # 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(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 =
```

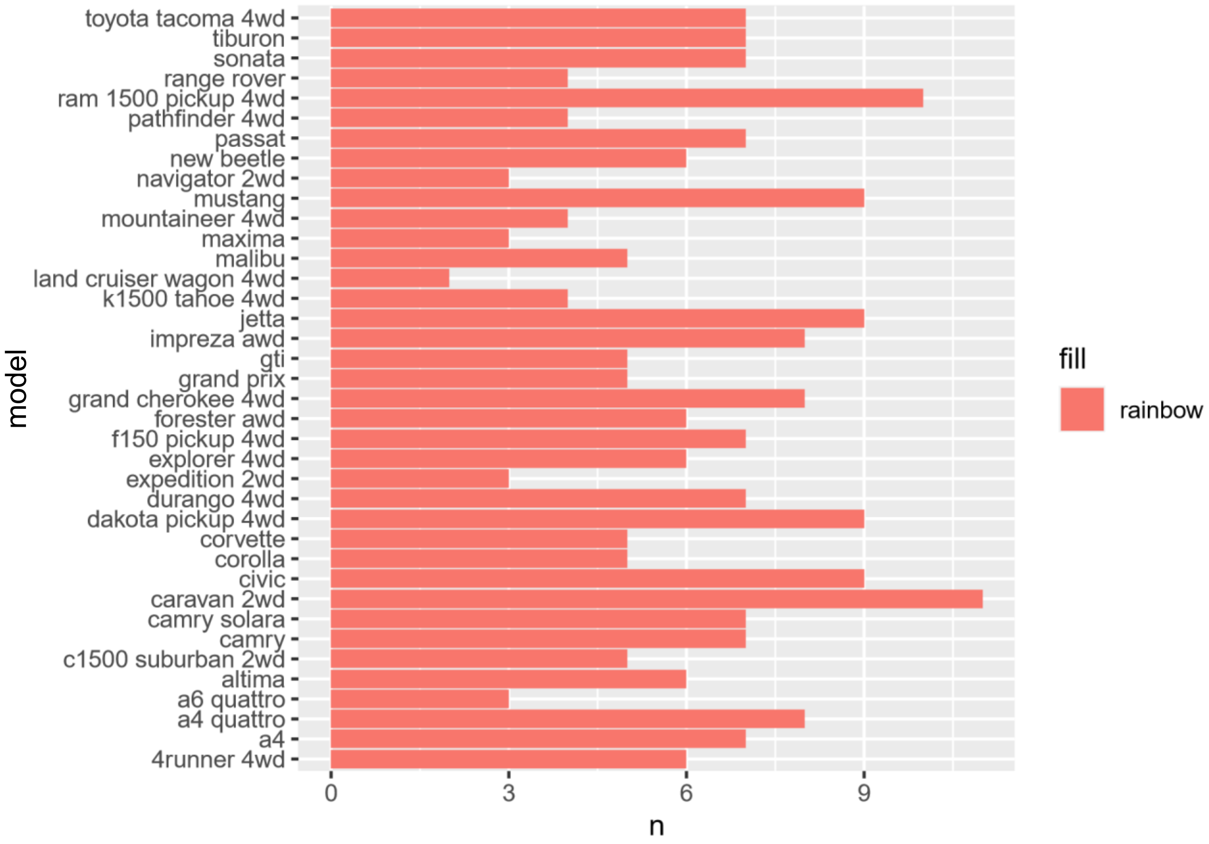


#b. Use only the top 20 observations. Show code and results.

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

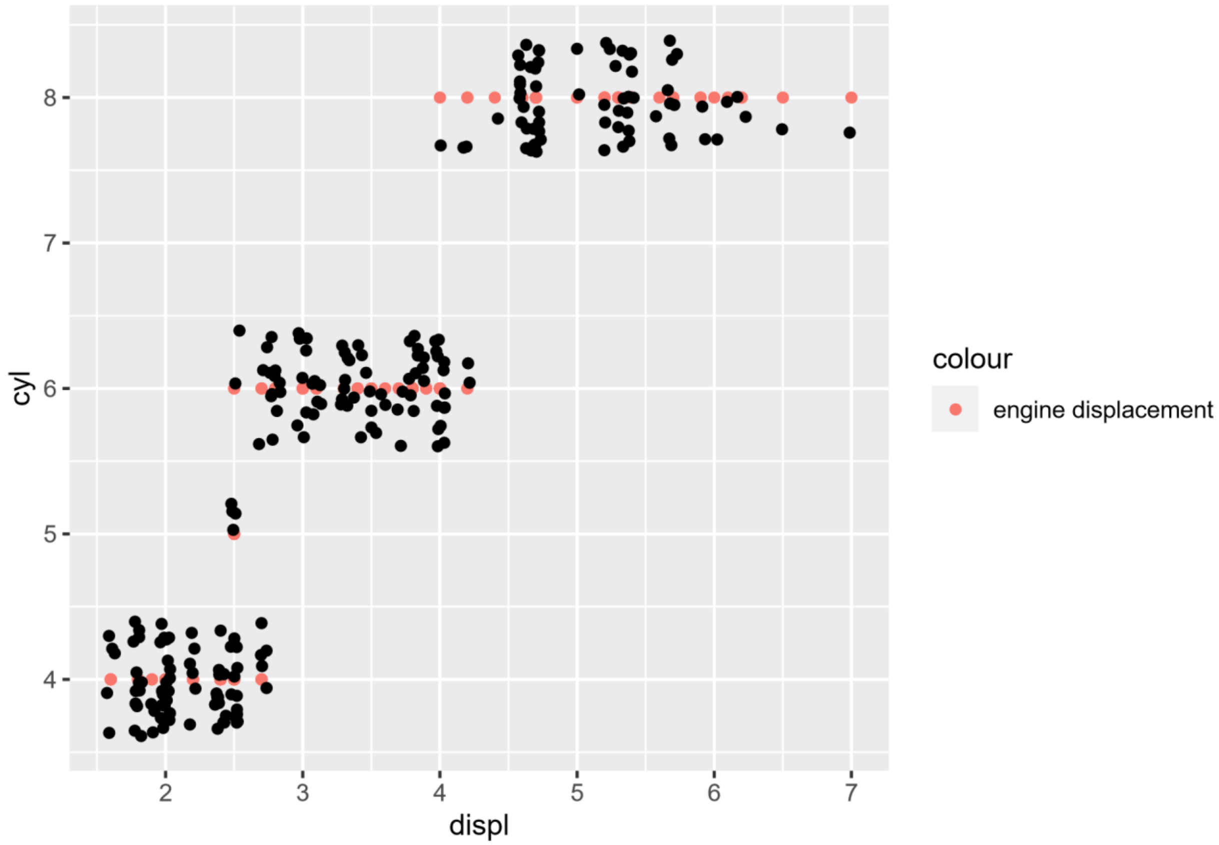
```
## # A tibble: 38 x 2
##        model                     n
##        <chr>                 <int>
## 1 caravan 2wd               11
## 2 ram 1500 pickup 4wd       10
## 3 civic                      9
## 4 dakota pickup 4wd          9
## 5 jetta                      9
## 6 mustang                    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 E
```

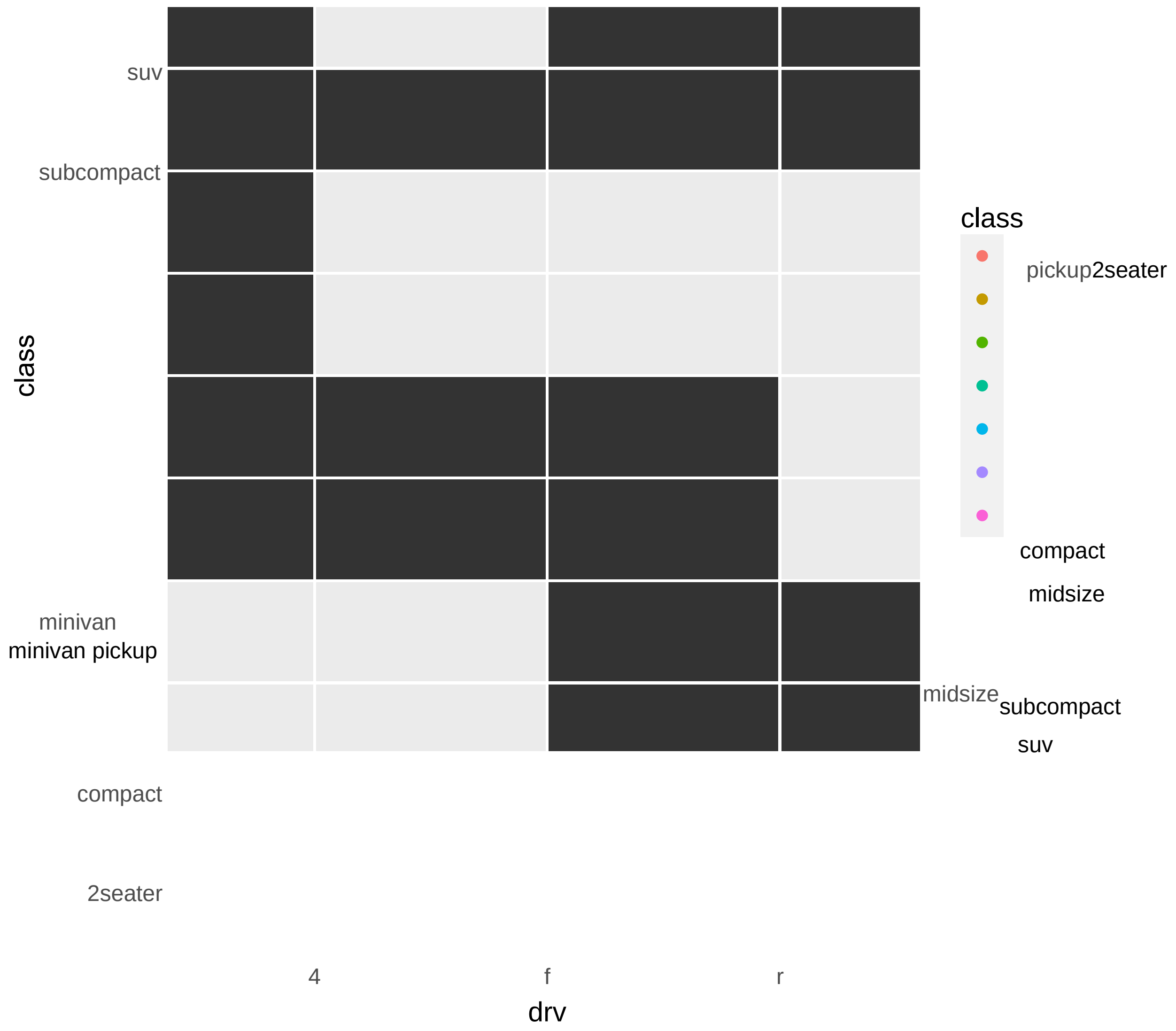


#I would say according to my data of making cyl the y, the graph is jittered. the pink color indicates the engine displacement and you can see that it is in a straight horizontal position.

#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= geom_tile()
```

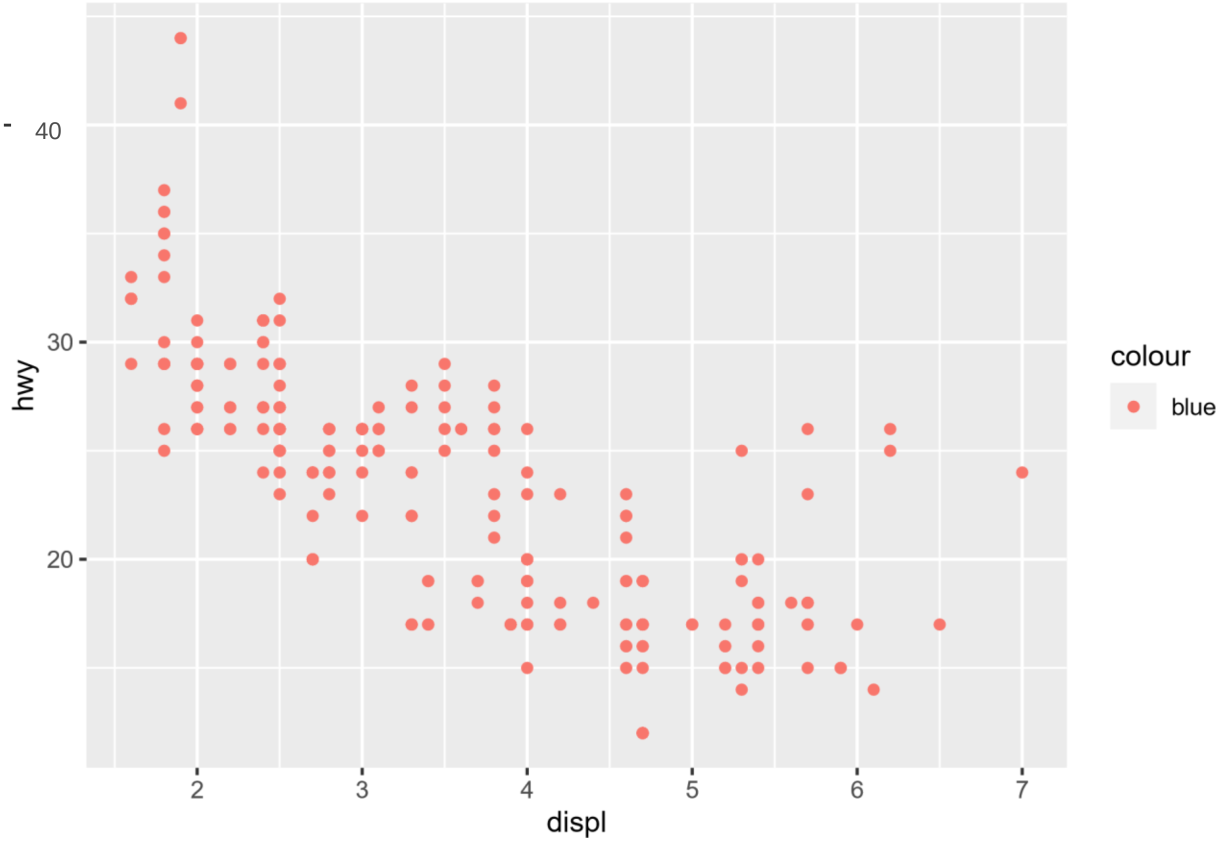
class)) +

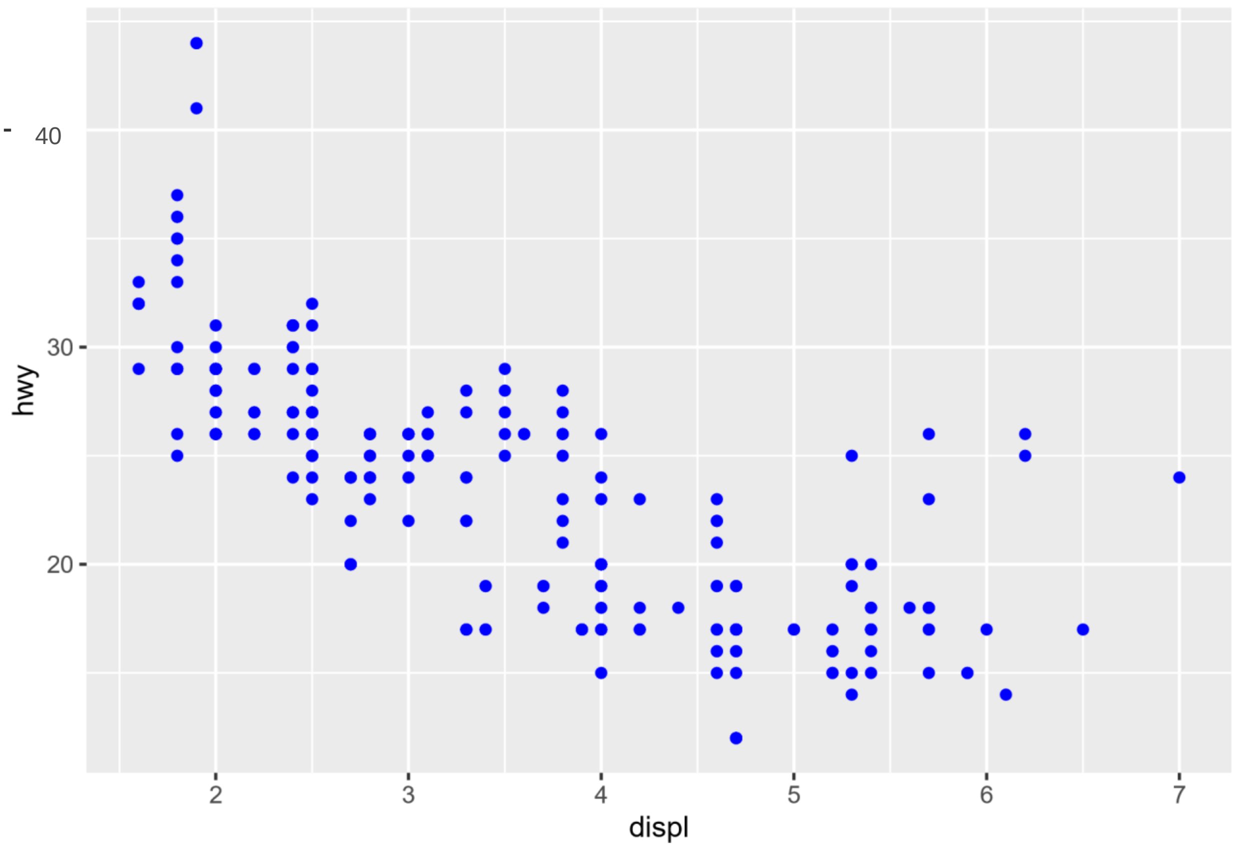


#b. Interpret the result: Areas covered with black are “mapped” using the mapping geometric point graph. y as class and x as drv.

#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"))
```


40
30
20
hwy
2
3
4
5
6
7
displ
colour
● blue



```
#+ Code #2 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
```

#a. Which variables from mpg dataset are categorical?

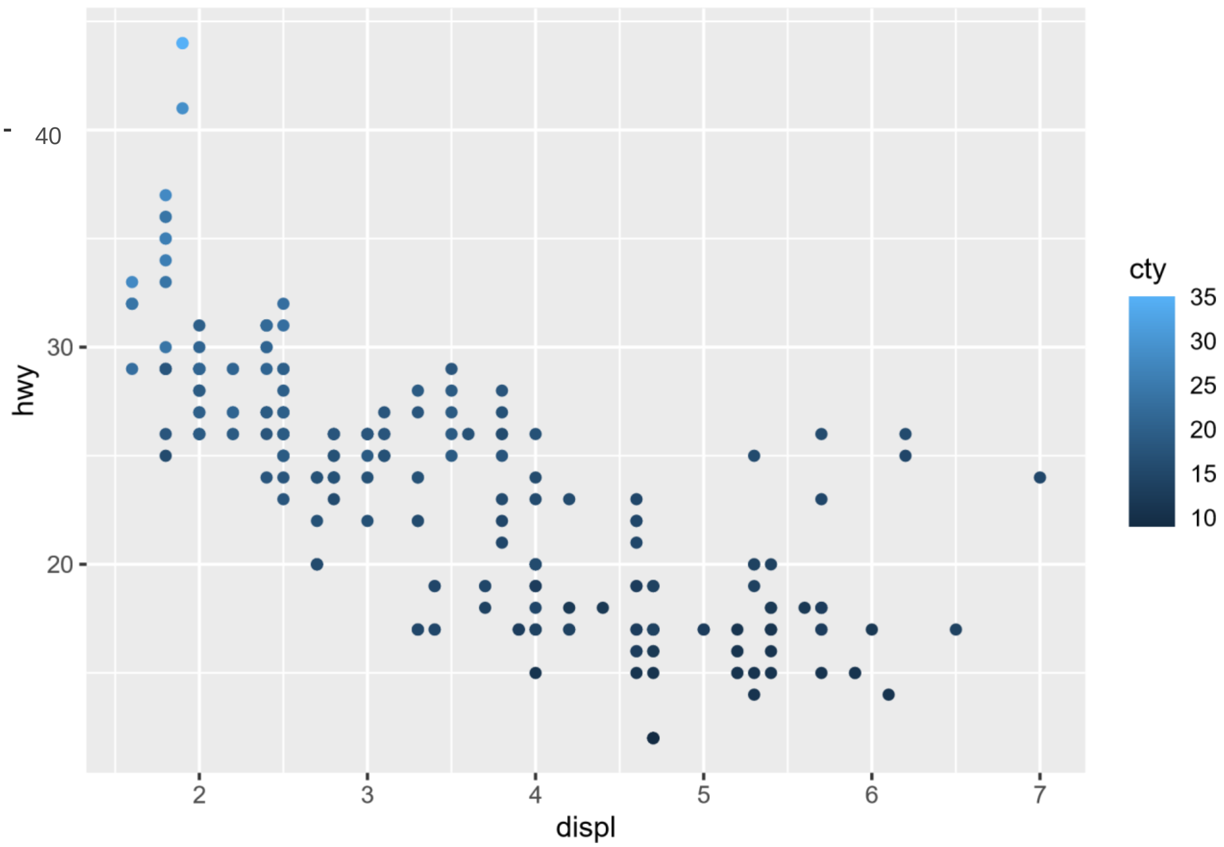
```
#Categorical variables in mpg include: manufacturer, model, trans (type of transmission), drv (front-whe
```

#b. Which are continuous variables?

```
#Continuous varibles in R are called 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? : data tracks the cty by placing cty(city miles per ga

#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”.

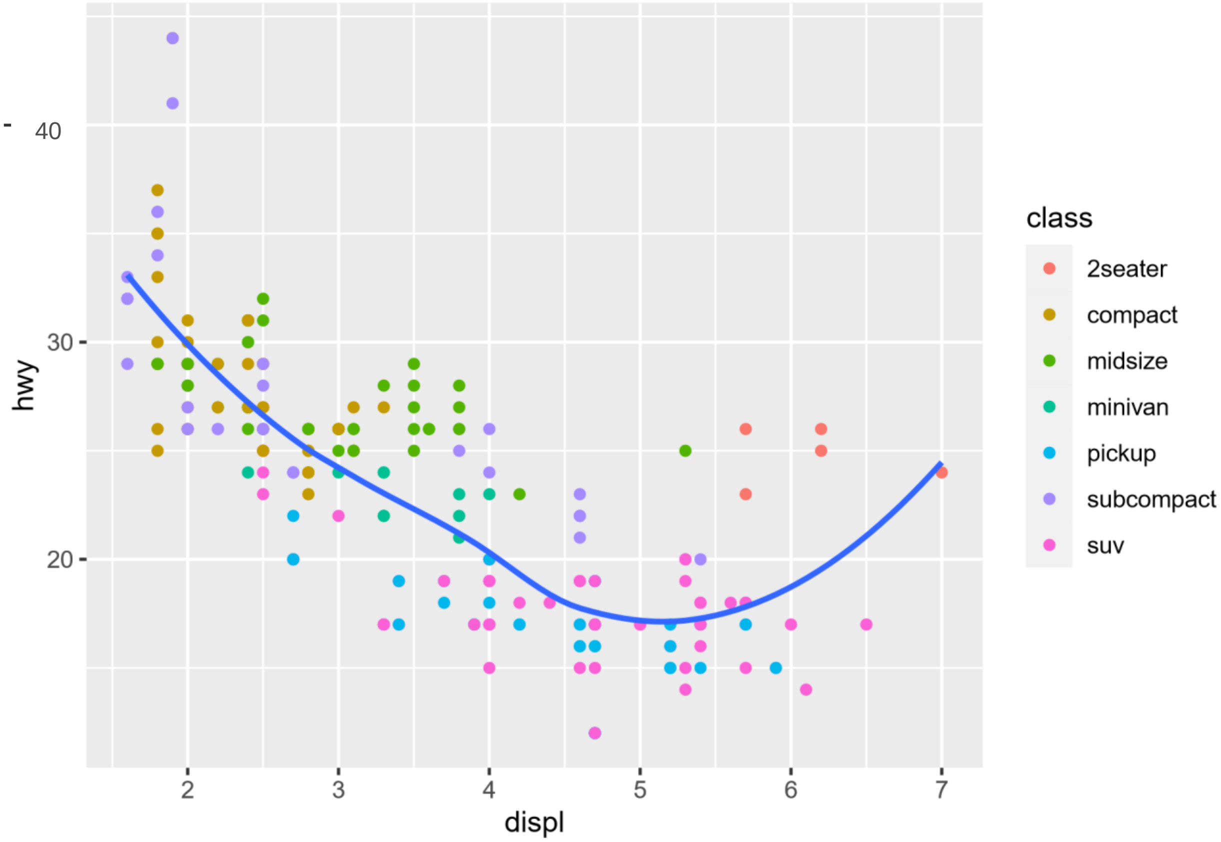
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_point(mapping=aes(color=class)) +
   geom_smooth(se = FALSE)
```

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

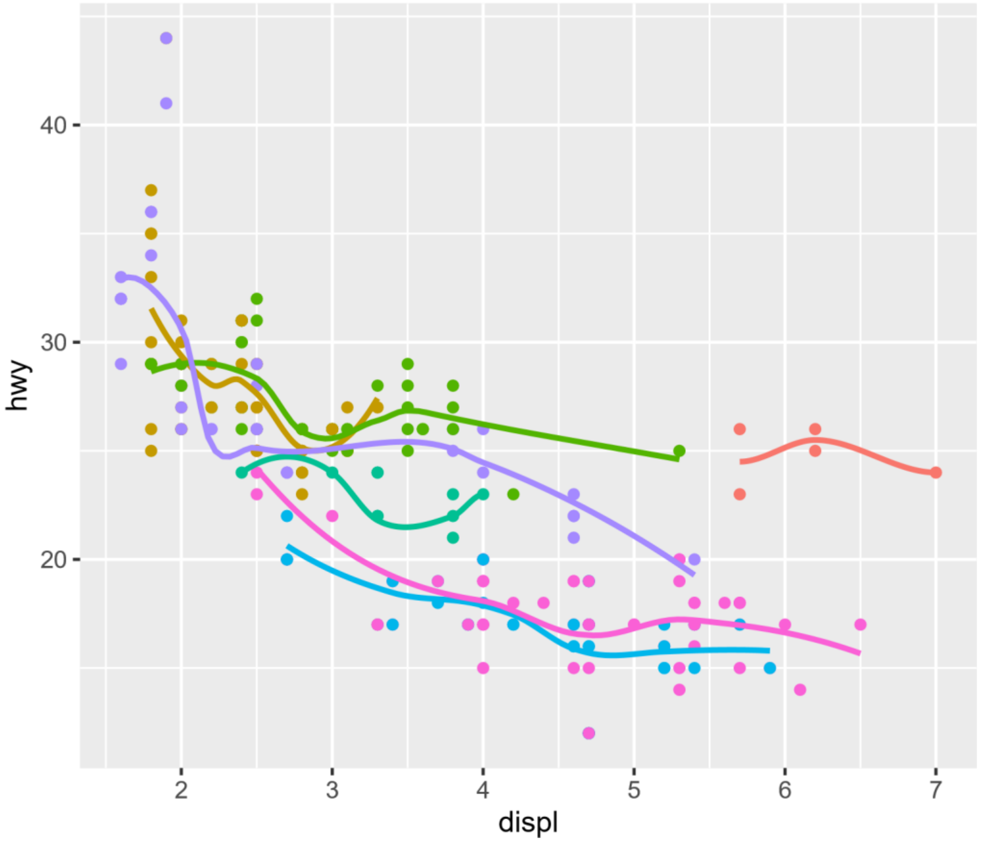
#10.Using the relationship of displ and hwy, add a trend line over existing plot. Set the se = FALSE to remove the confidence interval and method = lm to check for linear modeling

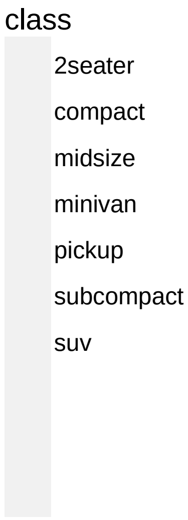
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color =
class)) +
geom_point() + geom_smooth(se = FALSE)
```

```
## ‘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
```




class
2seater
compact
midsize
minivan
pickup
subcompact
suv