RWorksheet_Mirabuena#6

Jessa Mae Mirabuena

2022-11-25

```
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)
data(mpg)
```

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

```
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: #most models = dodge 34

#most unique “a4”

```
unique(mpg$model)
```

```
##  [1] "a4"                     "a4 quattro"             "a6 quattro"
##  [4] "c1500 suburban 2wd"     "corvette"               "k1500 tahoe 4wd"
##  [7] "malibu"                 "caravan 2wd"            "dakota pickup 4wd"
## [10] "durango 4wd"            "ram 1500 pickup 4wd"    "expedition 2wd"
## [13] "explorer 4wd"           "f150 pickup 4wd"        "mustang"
## [16] "civic"                  "sonata"                 "tiburon"
## [19] "grand cherokee 4wd"     "range rover"            "navigator 2wd"
## [22] "mountaineer 4wd"        "altima"                 "maxima"
## [25] "pathfinder 4wd"         "grand prix"             "forester awd"
## [28] "impreza awd"            "4runner 4wd"            "camry"
## [31] "camry solara"           "corolla"                "land cruiser wagon 4wd"
## [34] "toyota tacoma 4wd"      "gti"                    "jetta"
## [37] "new beetle"             "passat"
```

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

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

```
## # 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
```

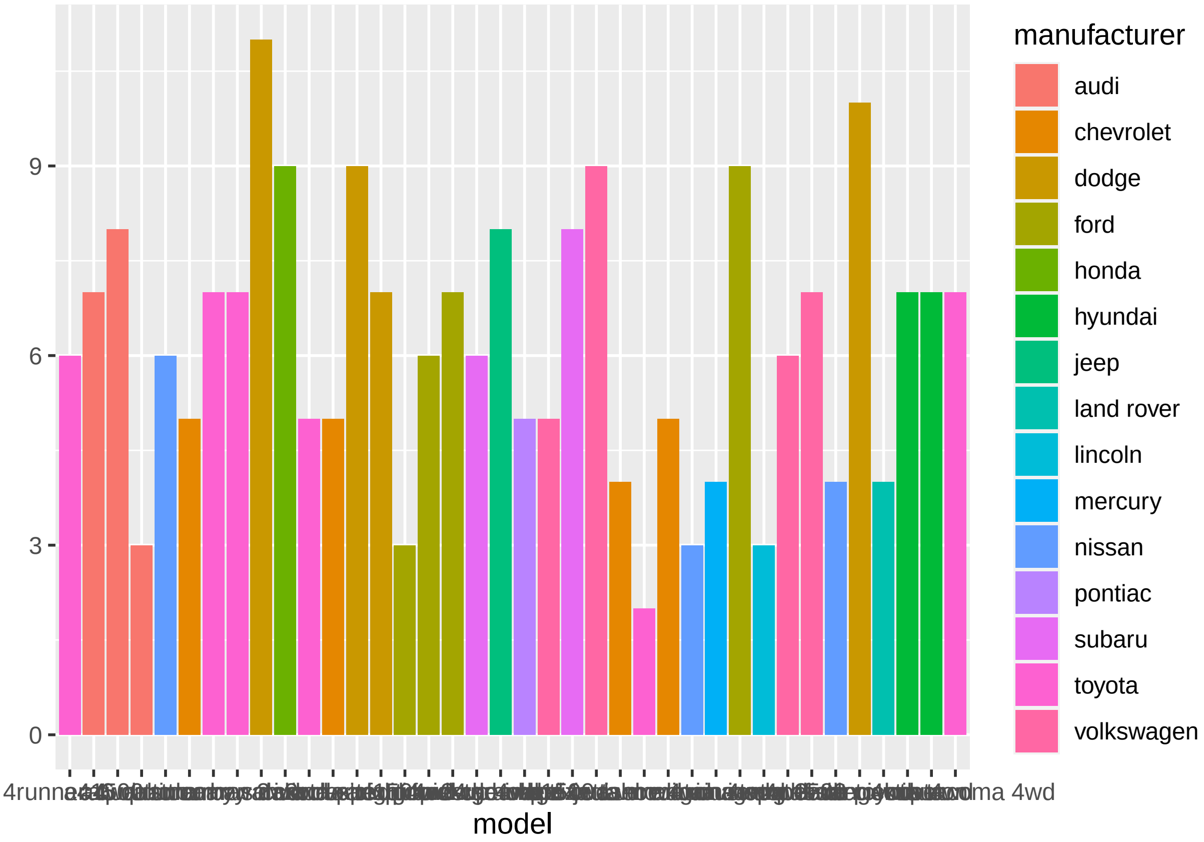
```
colnames(datax) <- c("Manufacturer", "Model","Counts")
datax
```

```
## # 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
```

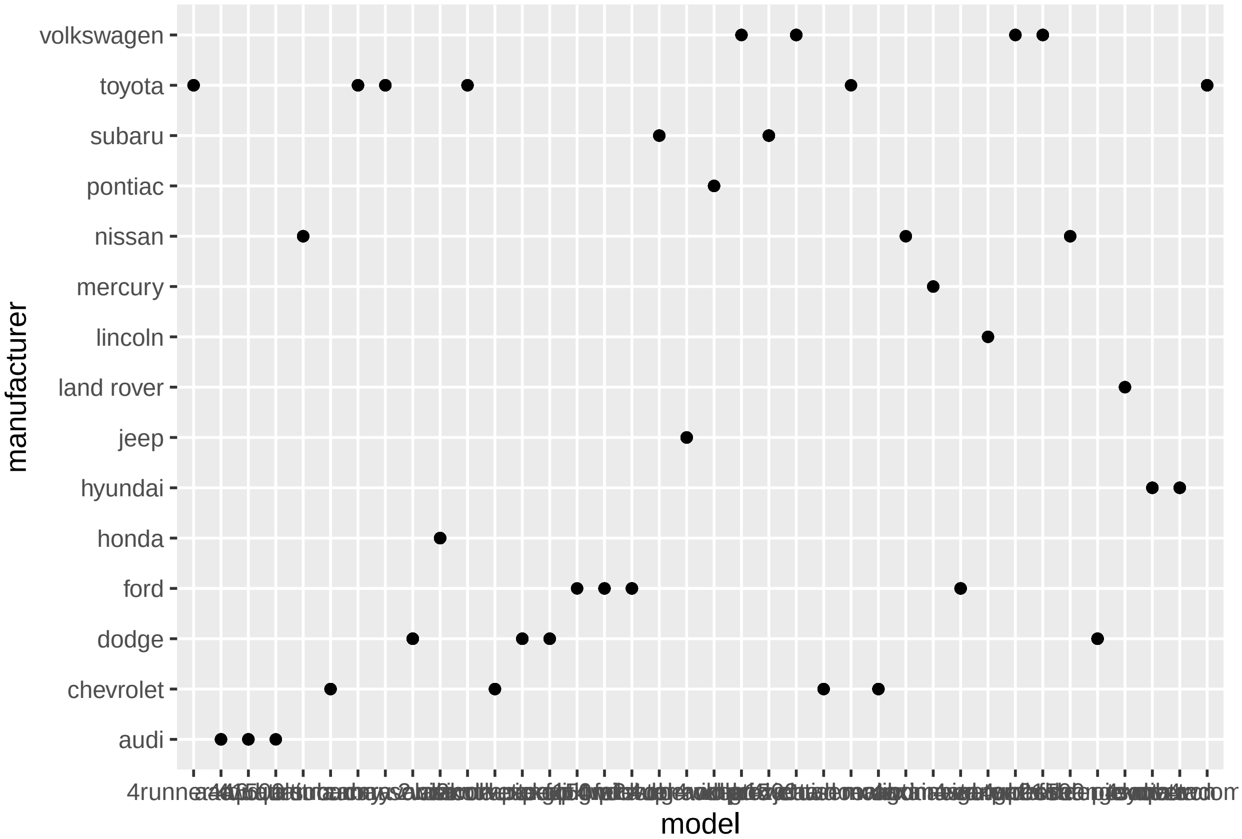
#b. Graph the result by using plot() and ggplot(). Write the codes and its result.

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

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



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



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

```
datampg <- mpg
data2 <- datampg %>% group_by(manufacturer, model) %>%
  distinct() %>% count()
data2
```

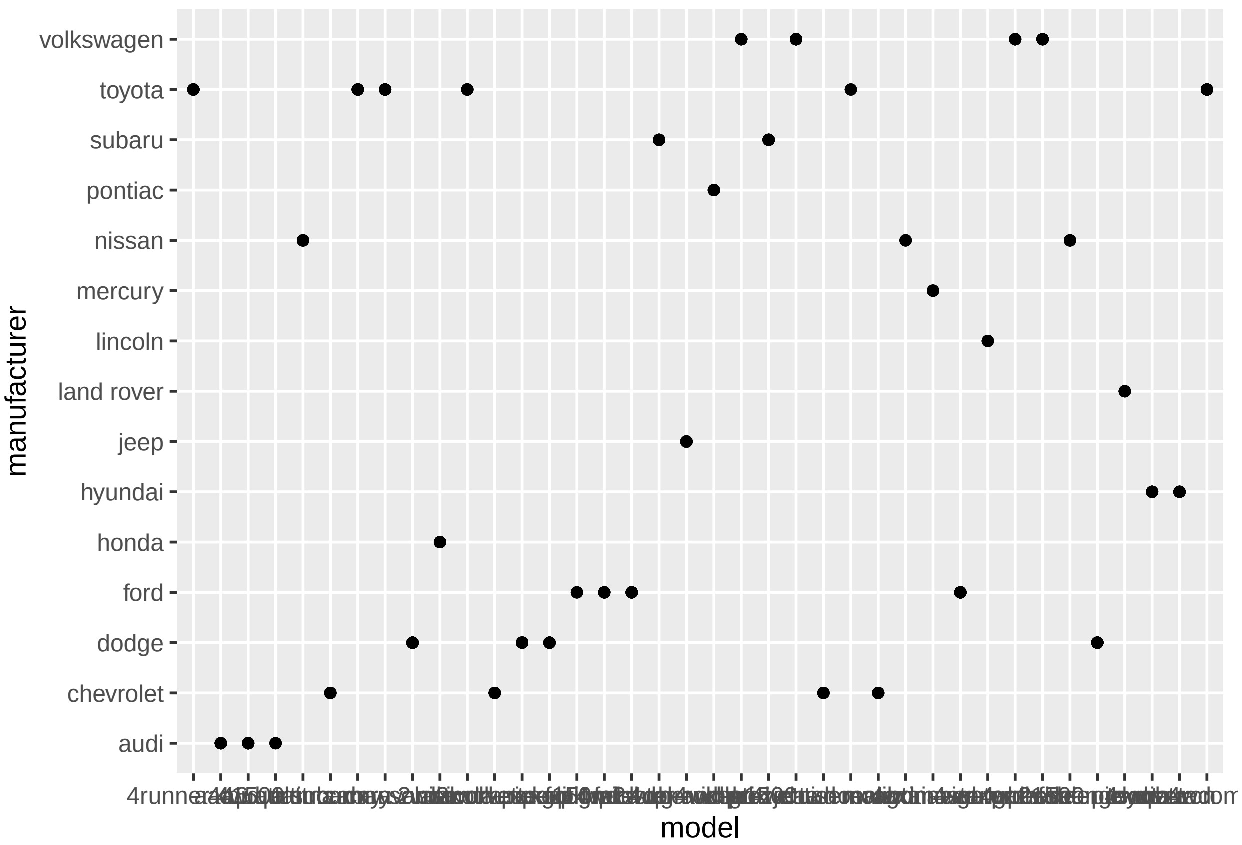
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet    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(data2) <- c("Manufacturer", "Model")
data2
```

```
## # 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()
```



# It shows the model scater plot of the dataset

#b. For you, is it useful? If not, how could you modify the data to make it more #informative? #Yes it is useful it can be use to identify the different variation of data.

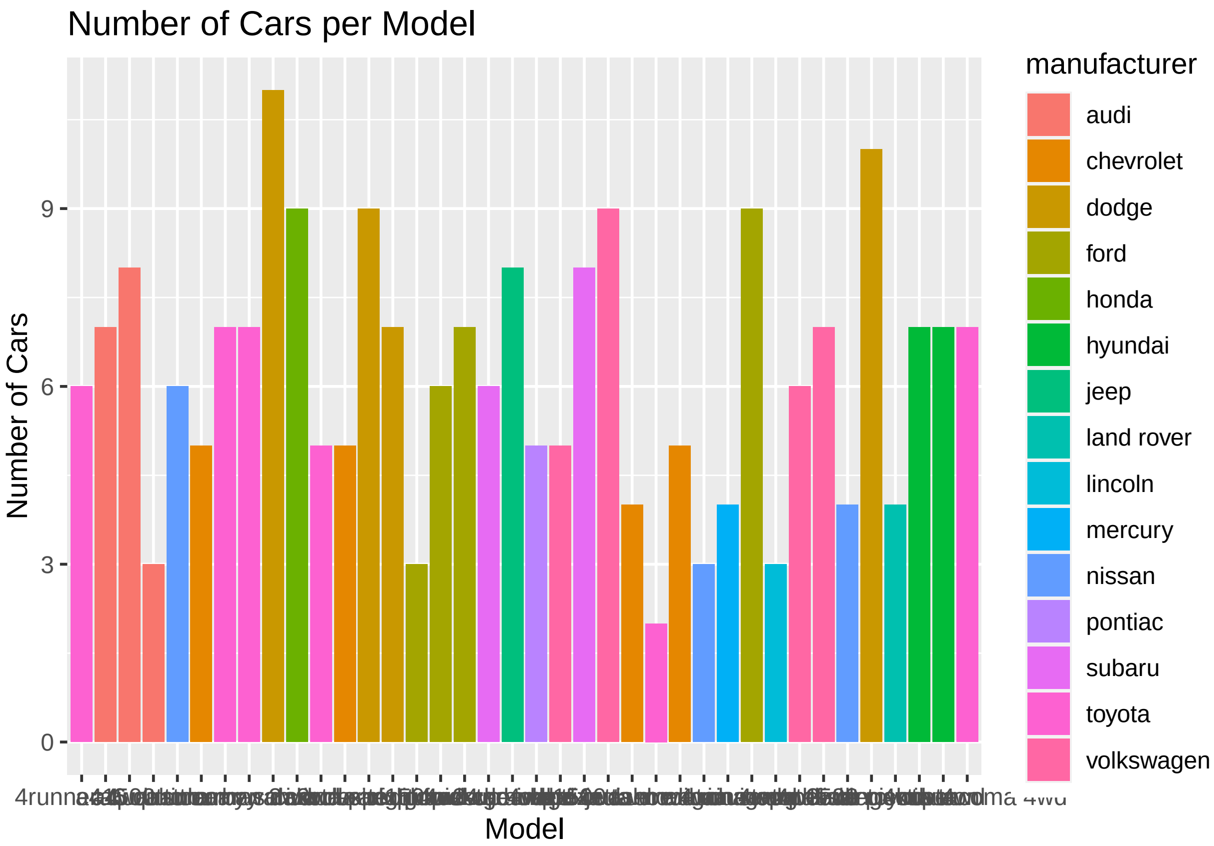
#4. Using the pipe (%>%), group the model and get the number of cars per model. Show #codes and its result.

```
data3 <- datax %>% group_by(Model) %>% count()
data3
```

```
## # 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
```

#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>
```

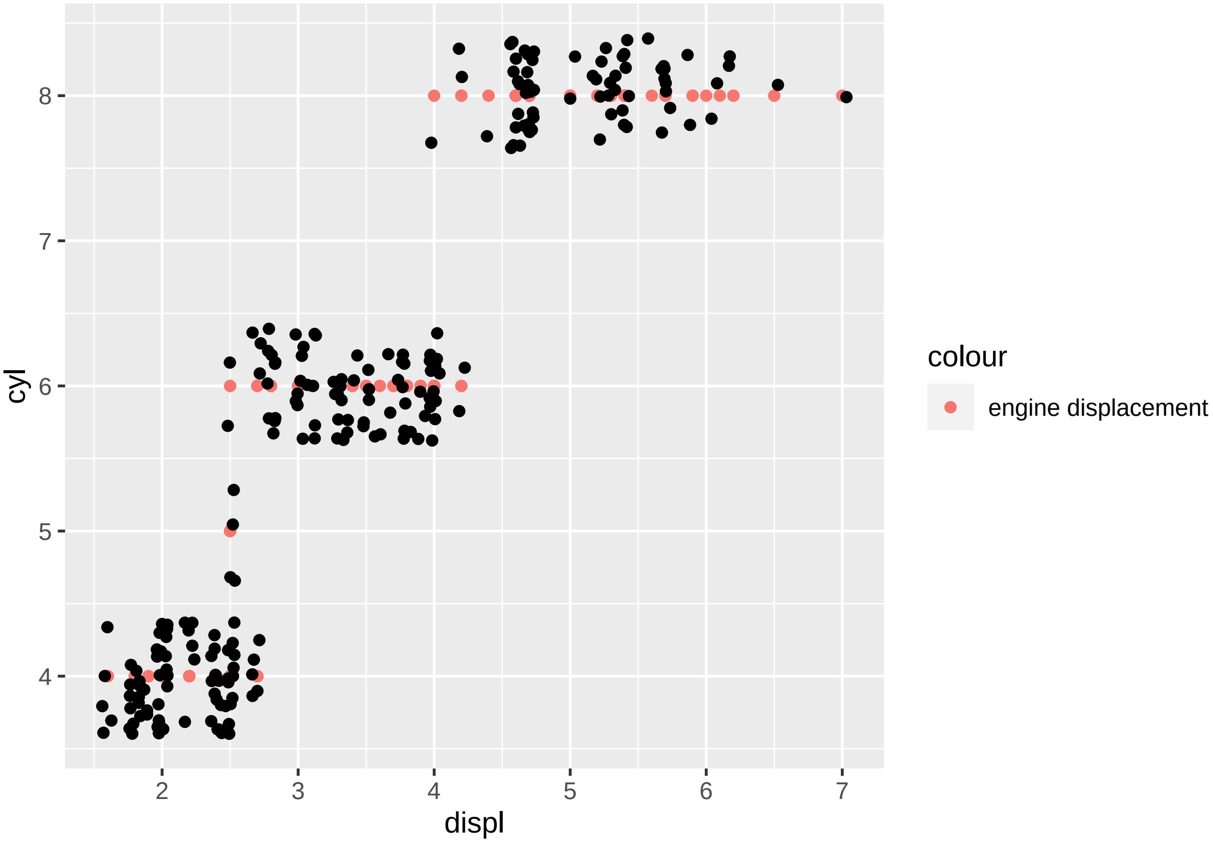
b. Use only the top 20 observations. Show code and results.

```
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". #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()
```

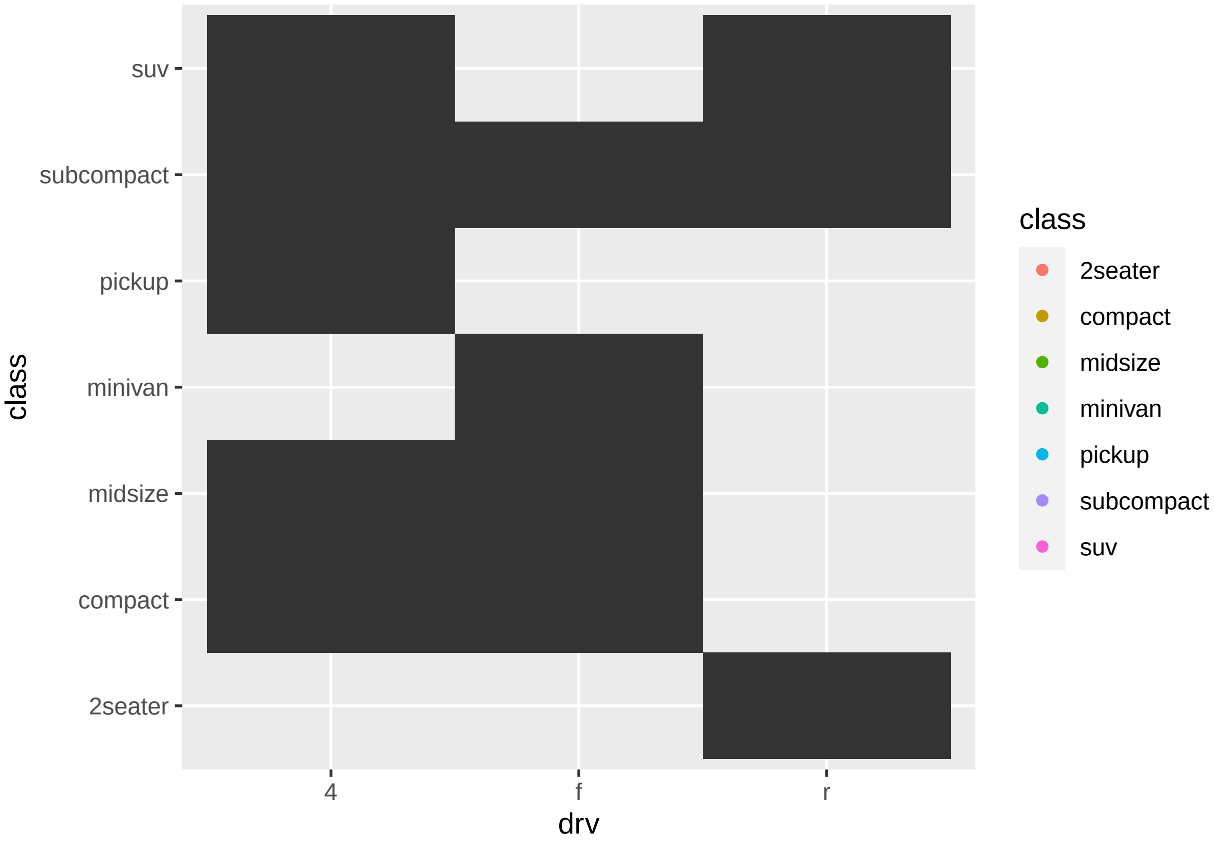


#b. How would you describe its relationship? #Using the geometric point it shows the engine displacement with legend that is color #pink

#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.

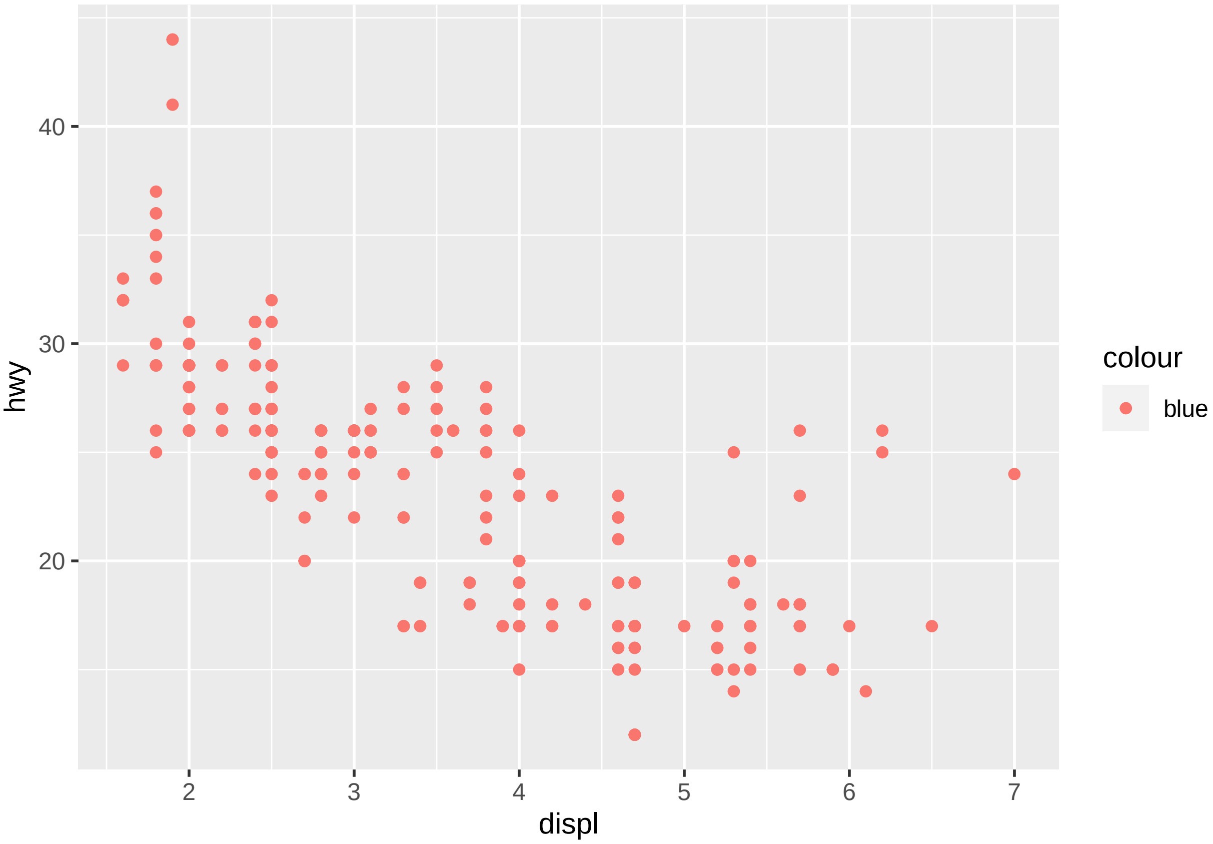
```
ggplot(data = mpg, mapping = aes(x = drv, y = class)) +
 geom_point(mapping=aes(color=class)) +
 geom_tile()
```



#b. Interpret the result. #These tiles represent unobserved combinations of class and drv values.

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

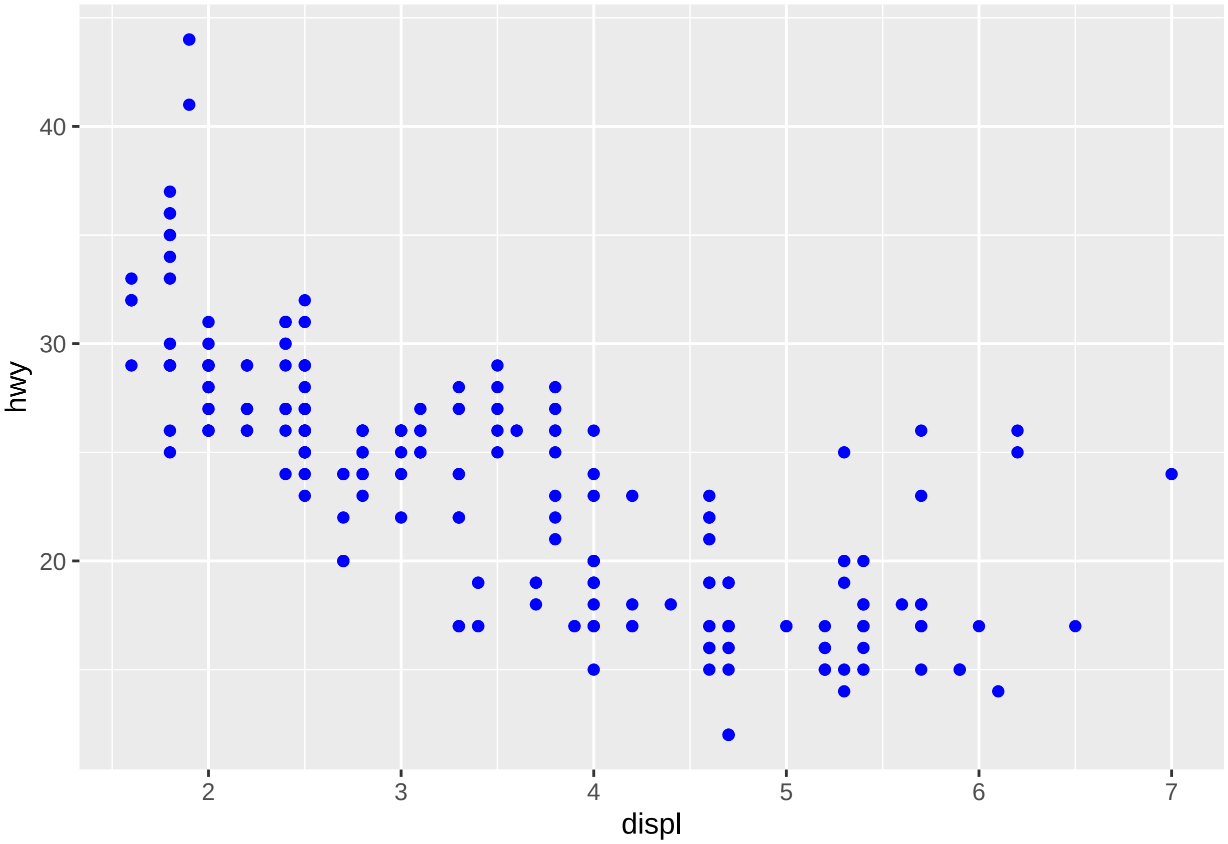
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, colour = "blue"))
```



#

Code #2

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



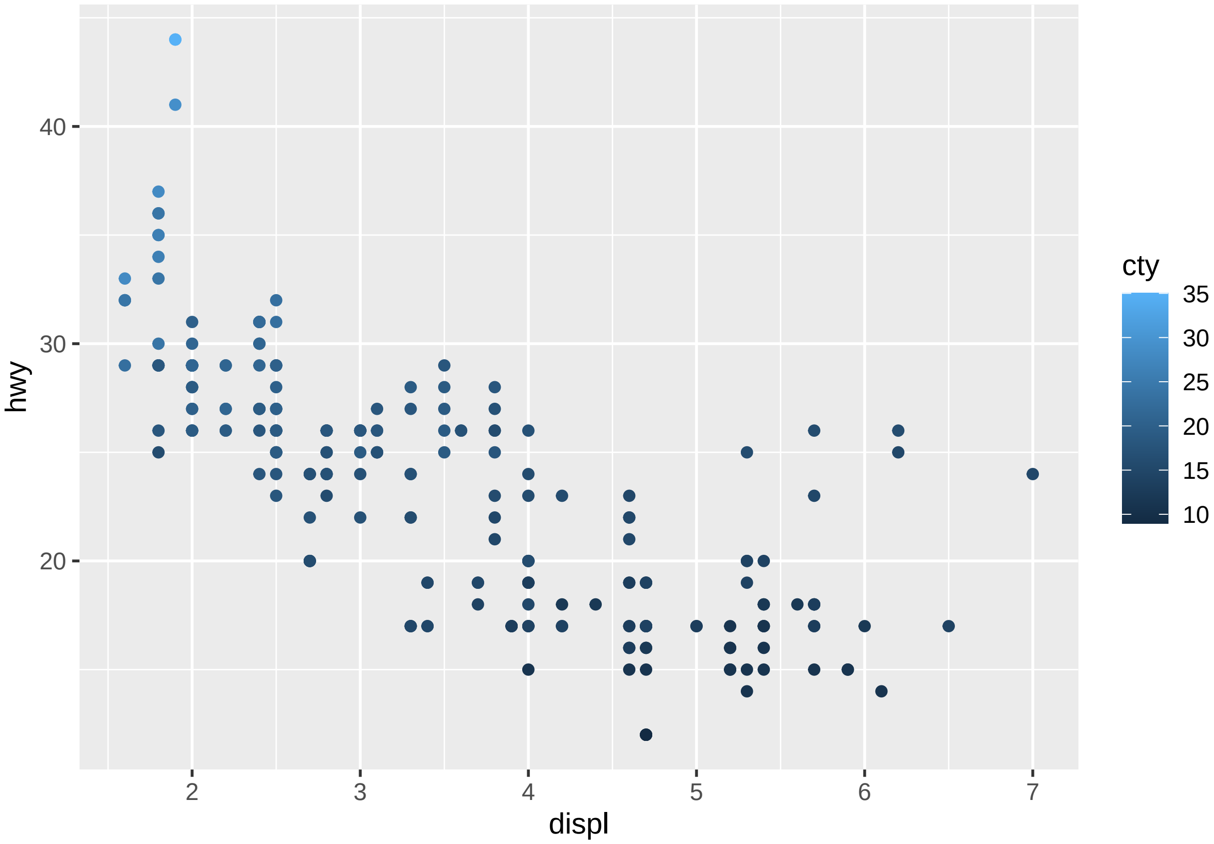
#The first code shows pink even the color is blue because The mapping argument, which is a mapping between a variable and a value, #has the argumentcolor = "blue," which is handled as an aesthetic as a result.

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

# It shows the cars dataset #a. Which variables from mpg dataset are categorical? #Categorical variables in mpg which include: #the manufacturer, model, trans (type of transmission), #drv (front-wheel drive, rear-wheel, 4wd), fl (fuel type), #and class (type of car).

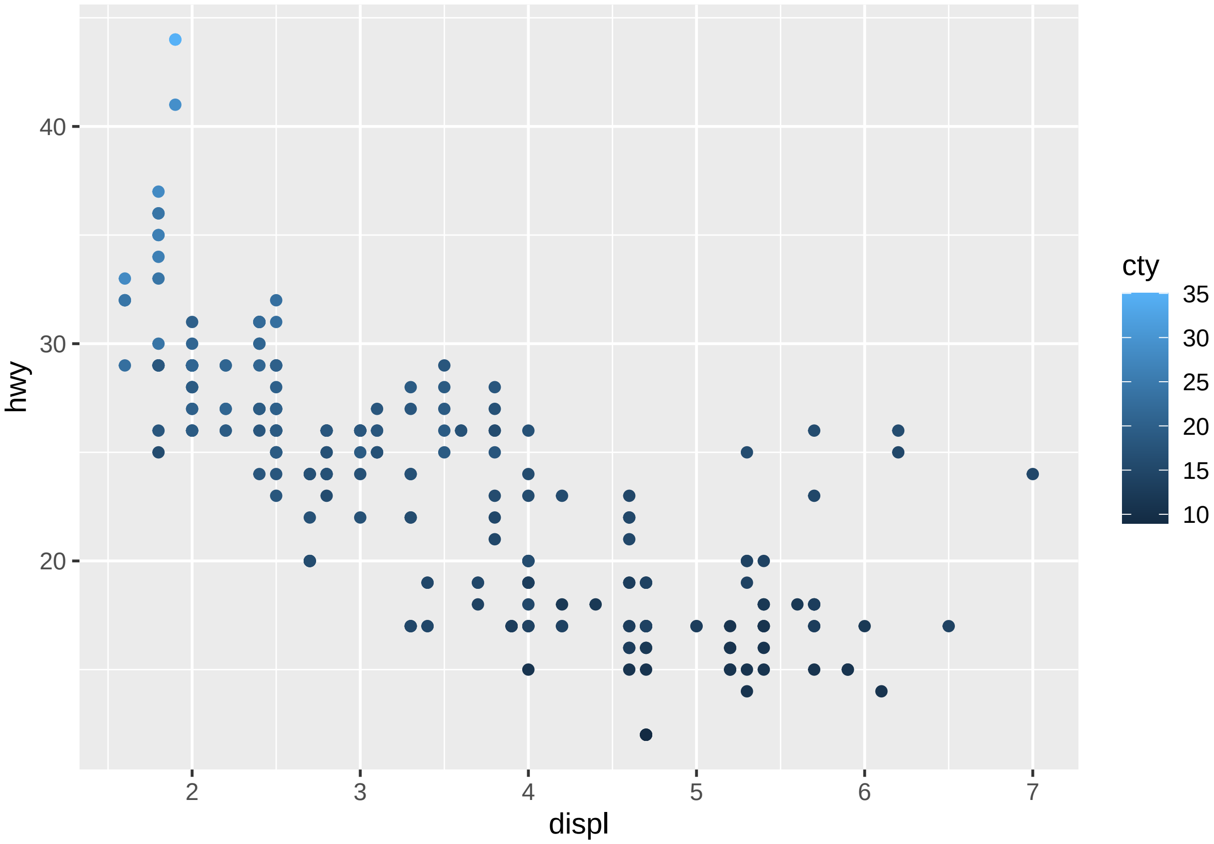
#b. Which are continuous variables? #The variable cty , city highway miles per gallon, is a continuous variable

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



#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 itsresult? Why it produced such output?

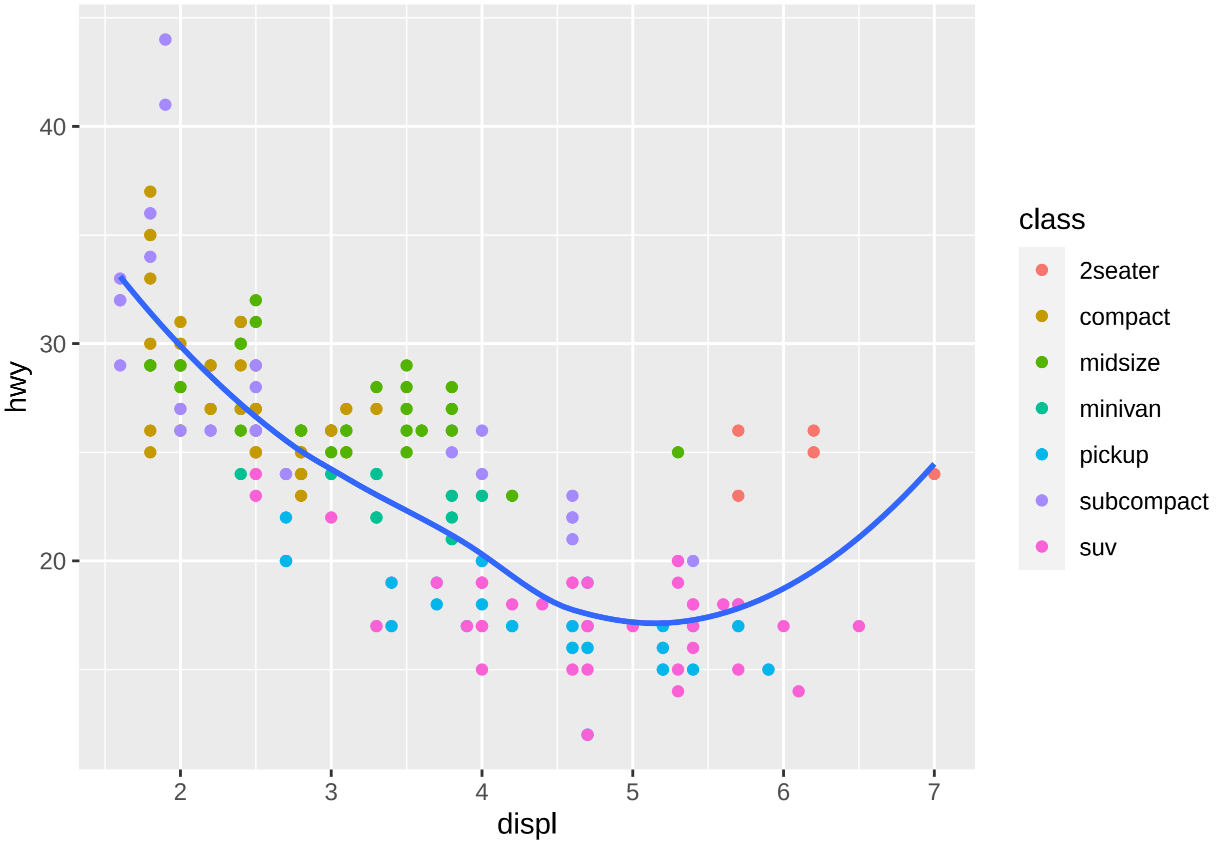
```
ggplot(mpg, aes(x = displ, y = hwy, colour = cty)) + geom_point()
```



#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=class)) +
  geom_smooth(se = FALSE)
```

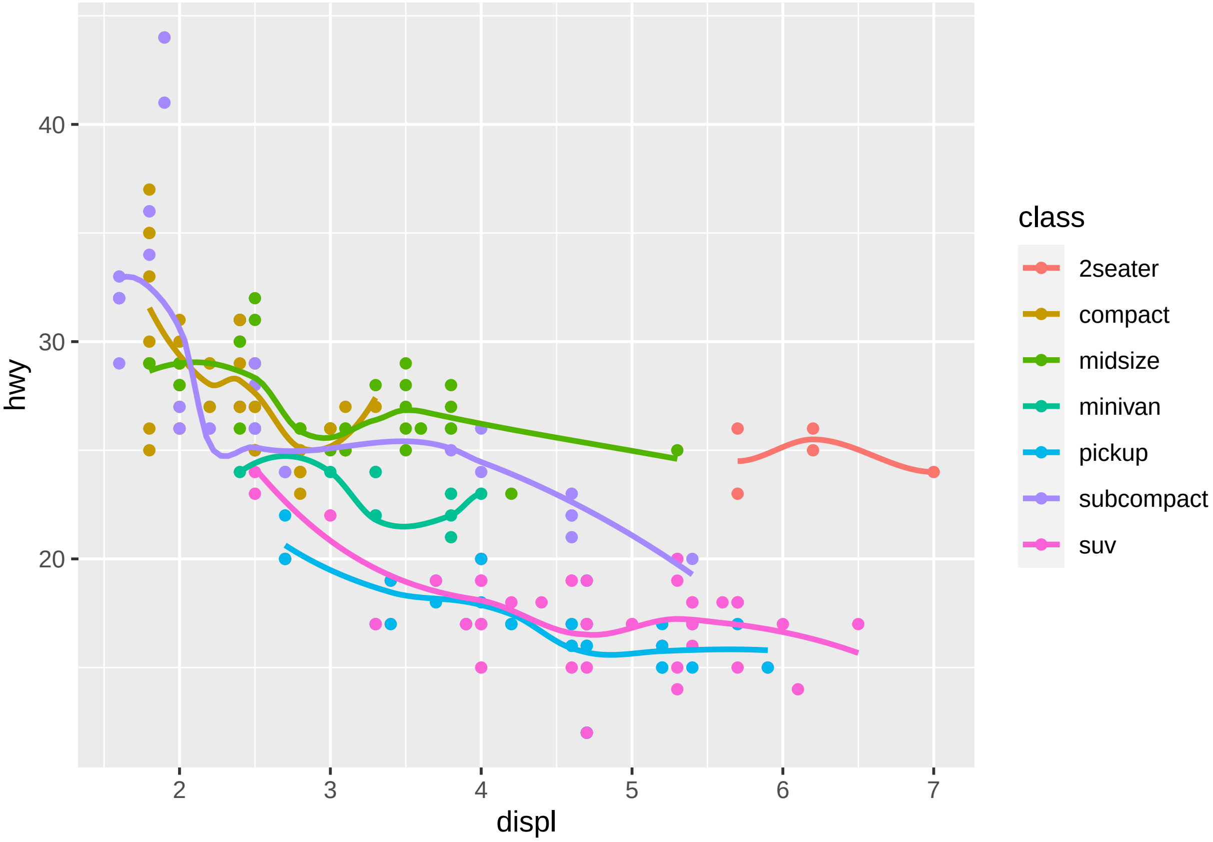
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



#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
```


hwy
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv