

RWorksheet_salve#6

Ben Robert T. Salve.

2022-11-24

Worksheet 6

Ben Robert T. Salve. BSIT2-A

1)

```
library(ggplot2)
data(mpg)
nrow(mpg)
```

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

```
ncol(mpg)
```

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

Answer: The number of columns in a mpg dataset is 11 while the number of rows is 234.

2)

Code:

```
library(dplyr) mpg print(mpg) View(mpg)
most_model <- mpg %>% group_by(manufacturer) %>% tally(sort = TRUE) most_model
most_variations <- mpg %>% group_by(model) %>% tally(sort = TRUE) most_variations
```

Output:

A tibble: 15 × 2

manufacturer n

1 dodge 37

2 toyota 34

3 volkswagen 27

4 ford 25

5 chevrolet 19

6 audi 18

7 hyundai 14

8 subaru 14

9 nissan 13

10 honda 9

11 jeep 8

12 pontiac 5

13 land rover 4

14 mercury 4

15 lincoln 3

Dodge has the most models in this data set and the model that has the most variations is caravan 2wd.

a.

Code:

```
data1 <- mpg
u_models <- data1 %>% group_by(manufacturer, model) %>% distinct() %>% count()
u_models colnames(u_models) <- c("Manufacturer", "Model", "Counts")
u_models
```

Output:

A tibble: 38 × 3

Groups: Manufacturer, Model [38]

Manufacturer Model Counts

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

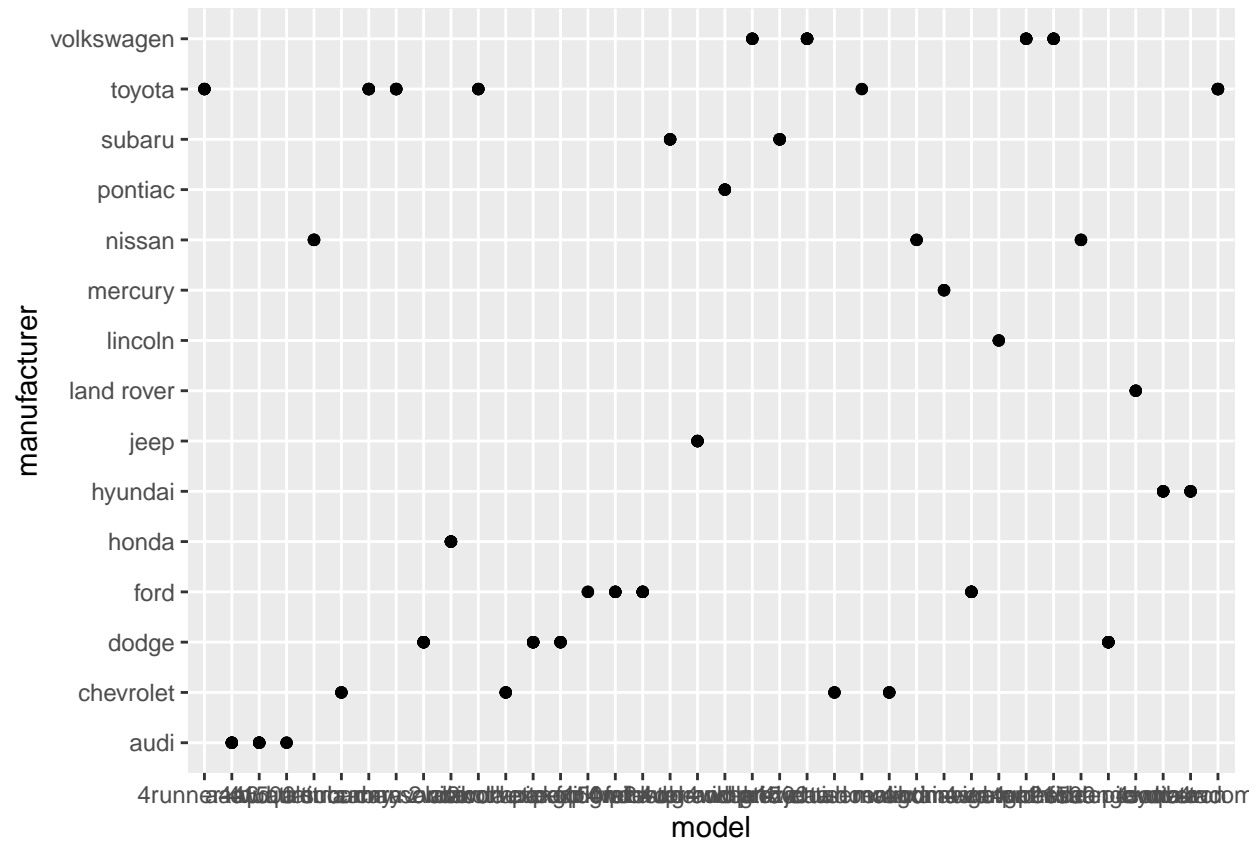
10 dodge durango 4wd 6

... with 28 more rows

b.

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

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

3)

Code:

```
data1 <- mpg
u_models <- data1 %>% group_by(manufacturer, model) %>% distinct() %>% count()
u_models$colnames(u_models) <- c("Manufacturer", "Model", "Counts")
u_models
```

Output:

A tibble: 38 × 3

Groups: Manufacturer, Model [38]

Manufacturer Model Counts

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

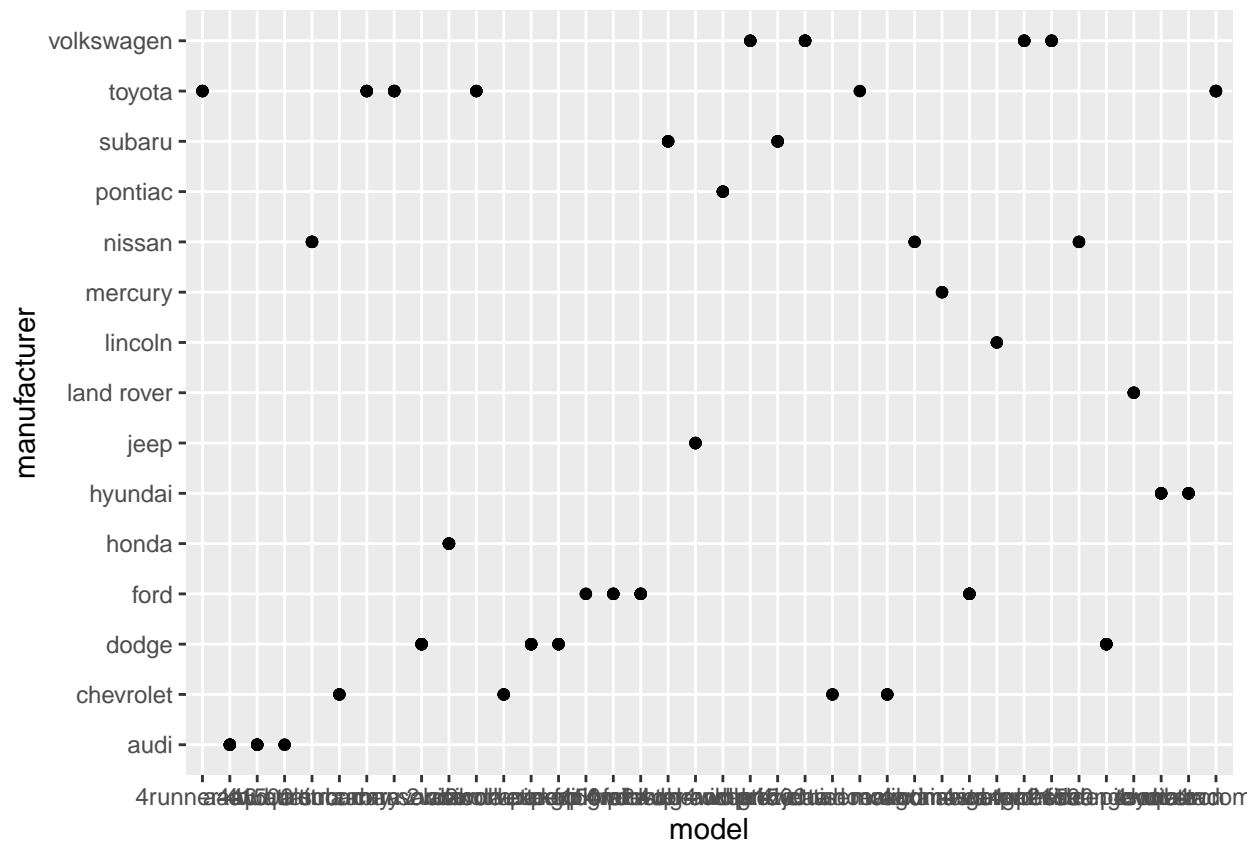
9 dodge dakota pickup 4wd 8

10 dodge durango 4wd 6

... with 28 more rows

a.

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



Answer: This code shows a geometric point graph of model and manufacturer of the mpg data set.

b.

Answer: For me, the plot is already useful if you want to find a specific model and what

manufacturer made it, then this plot can really help you find the information that you need.

4)

Code:

```
data2 <- u_models %>% group_by(Model) %>% count() colnames(data2) <- c("Model","Counts") data2
```

Output:

A tibble: 38 × 2

Groups: Model [38]

Model Counts

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

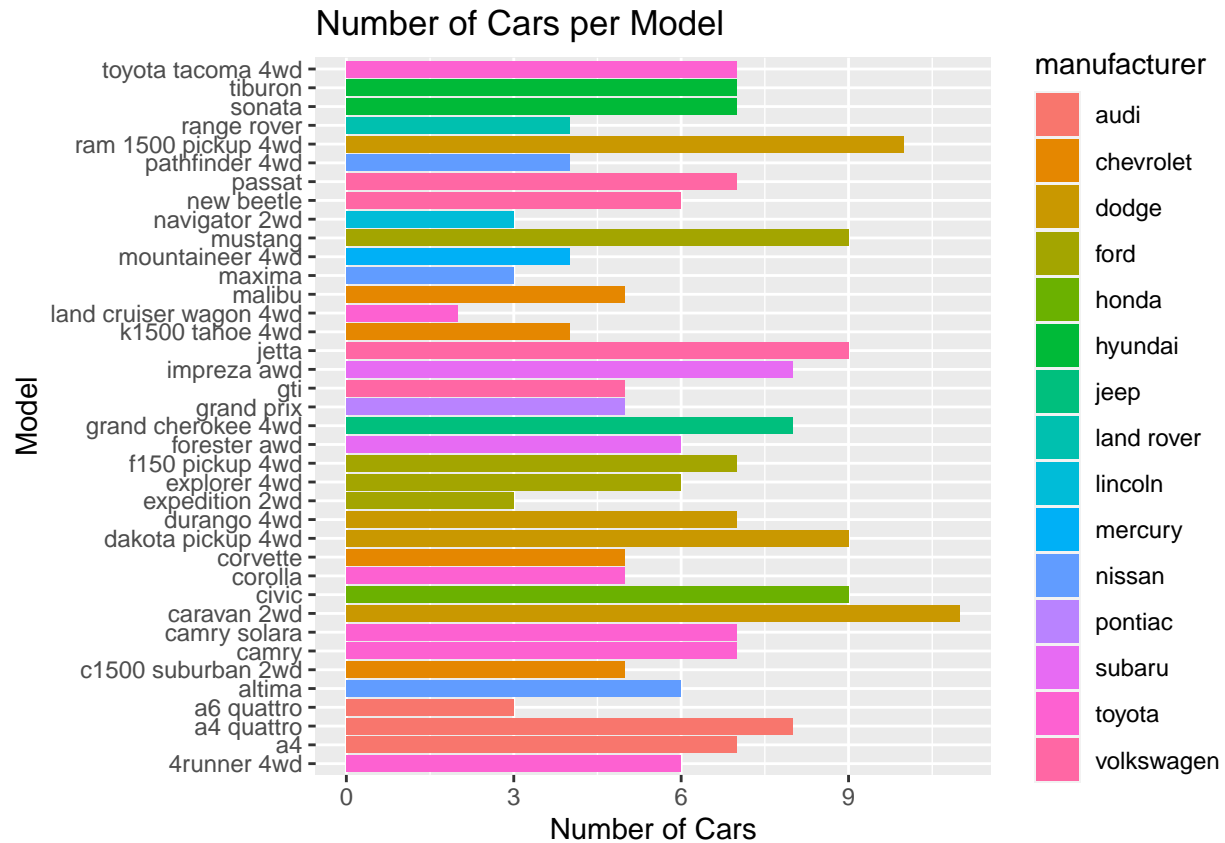
9 caravan 2wd 1

10 civic 1

... with 28 more rows

a.

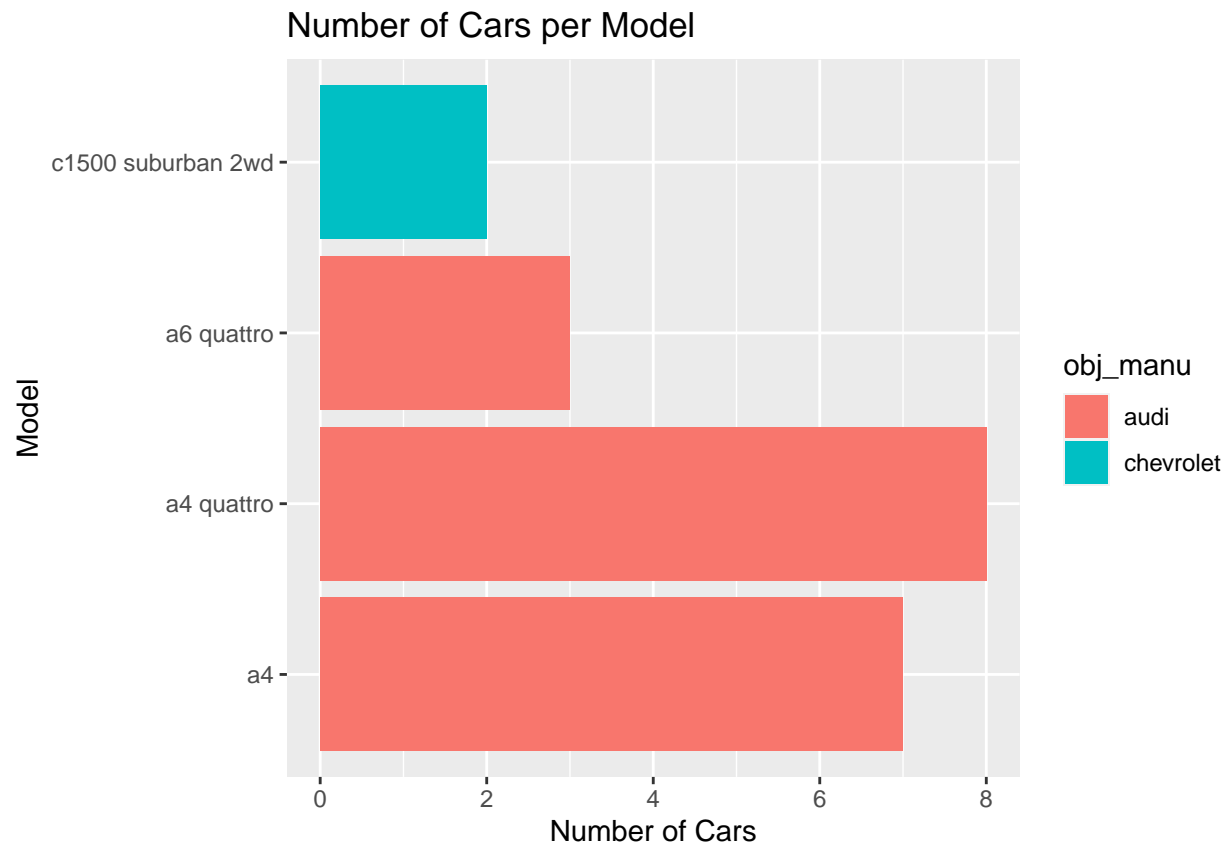
```
qplot(model, data= mpg, main= "Number of Cars per Model",  
      xlab= "Model", ylab= "Number of Cars", geom= "bar", fill= manufacturer) + coord_flip()
```

b.

```
data1 <- subset(mpg[c(1:20),c(1:11)])
obj_manu <- mpg$manufacturer[1:20]
obj_mode <- mpg$model[1:20]

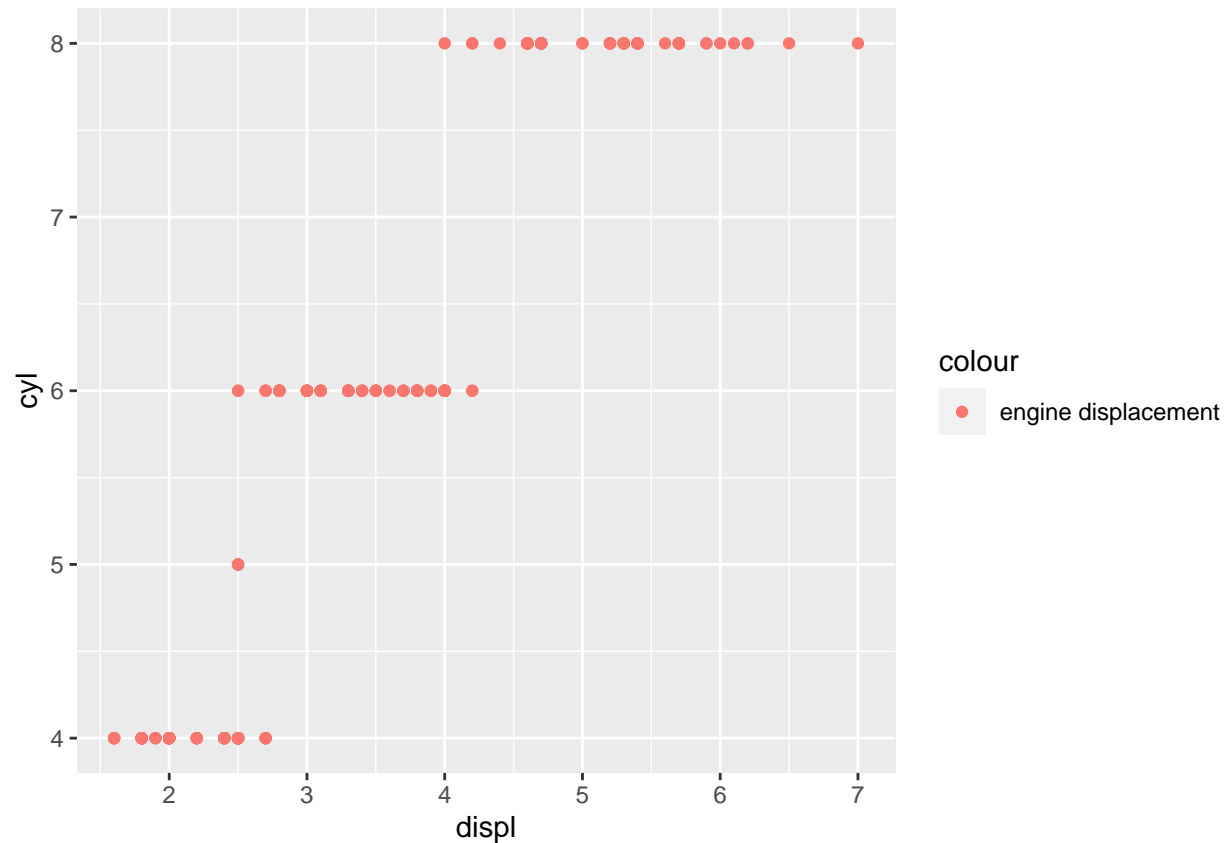
qplot(obj_mode, data = data1, main = "Number of Cars per Model",
       xlab = "Model", ylab = "Number of Cars", geom = "bar", fill = obj_manu) + coord_flip()
```



5)

a.

```
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"))
```



b.

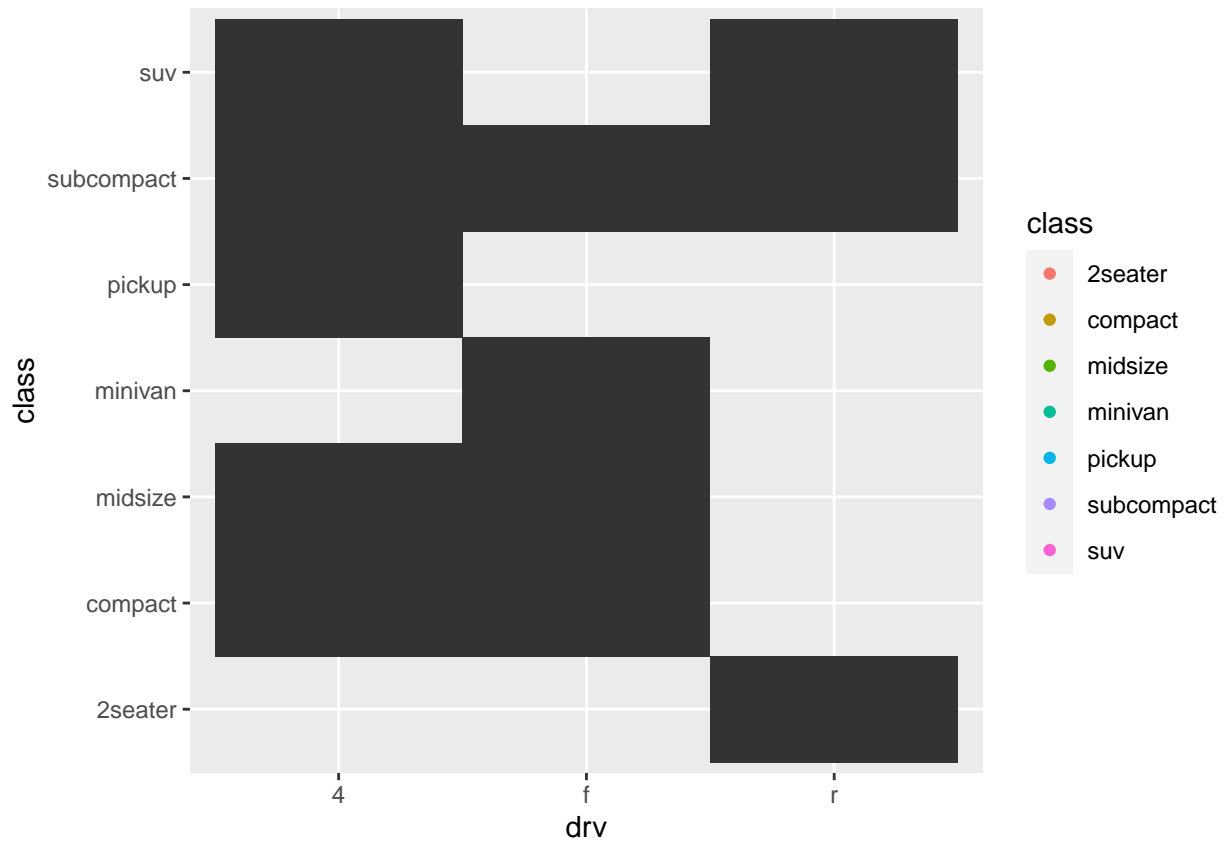
Answer: In the y axis we have the number of cylinders and in the x axis we have the displacement,

the red dots represents the engine displacement. The plot is about the cylinders by displacement.

6)

a.

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



b.

Answer: It is class by the type of drive train, where f = front-wheel drive,

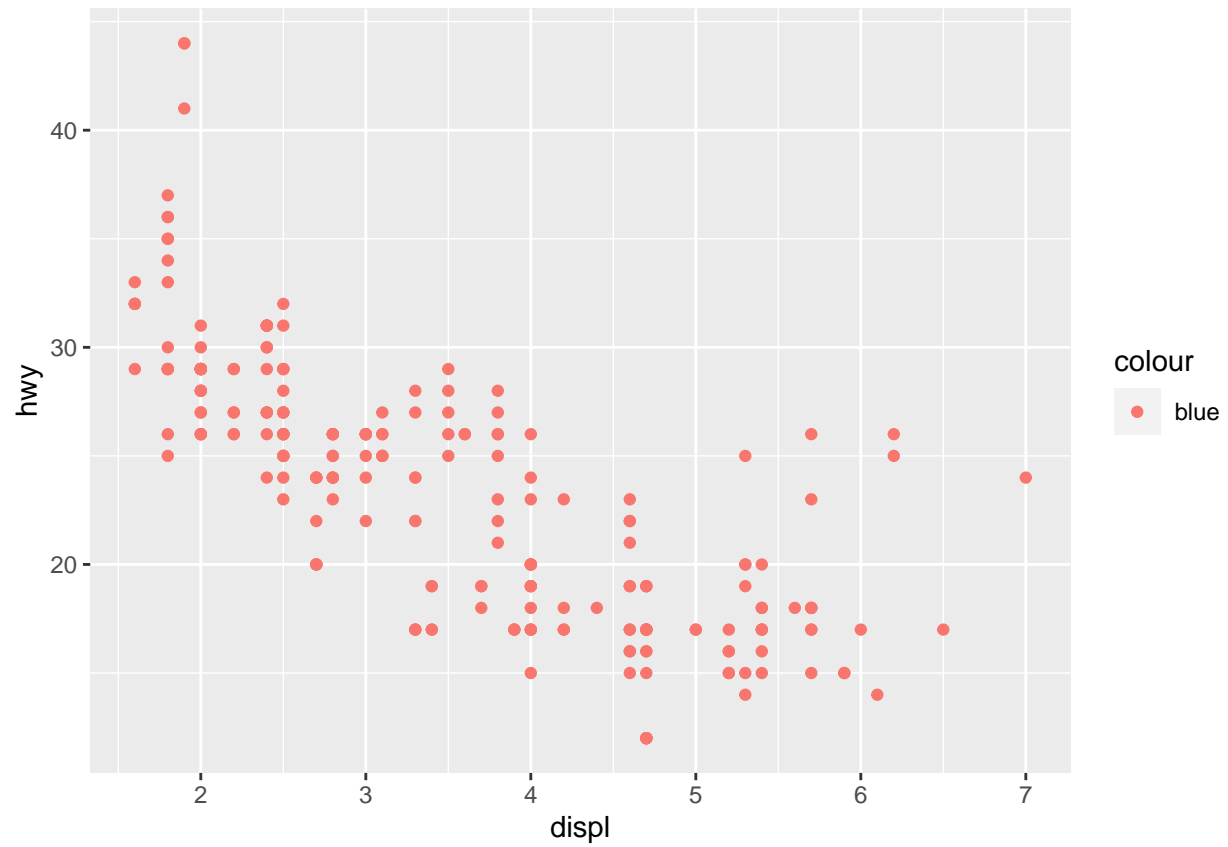
r = rear wheel drive, 4 = 4wd. The plot looks like a heatmap but the color is black.

The class or the legend has color representations but the plot doesn't have any color.

7)

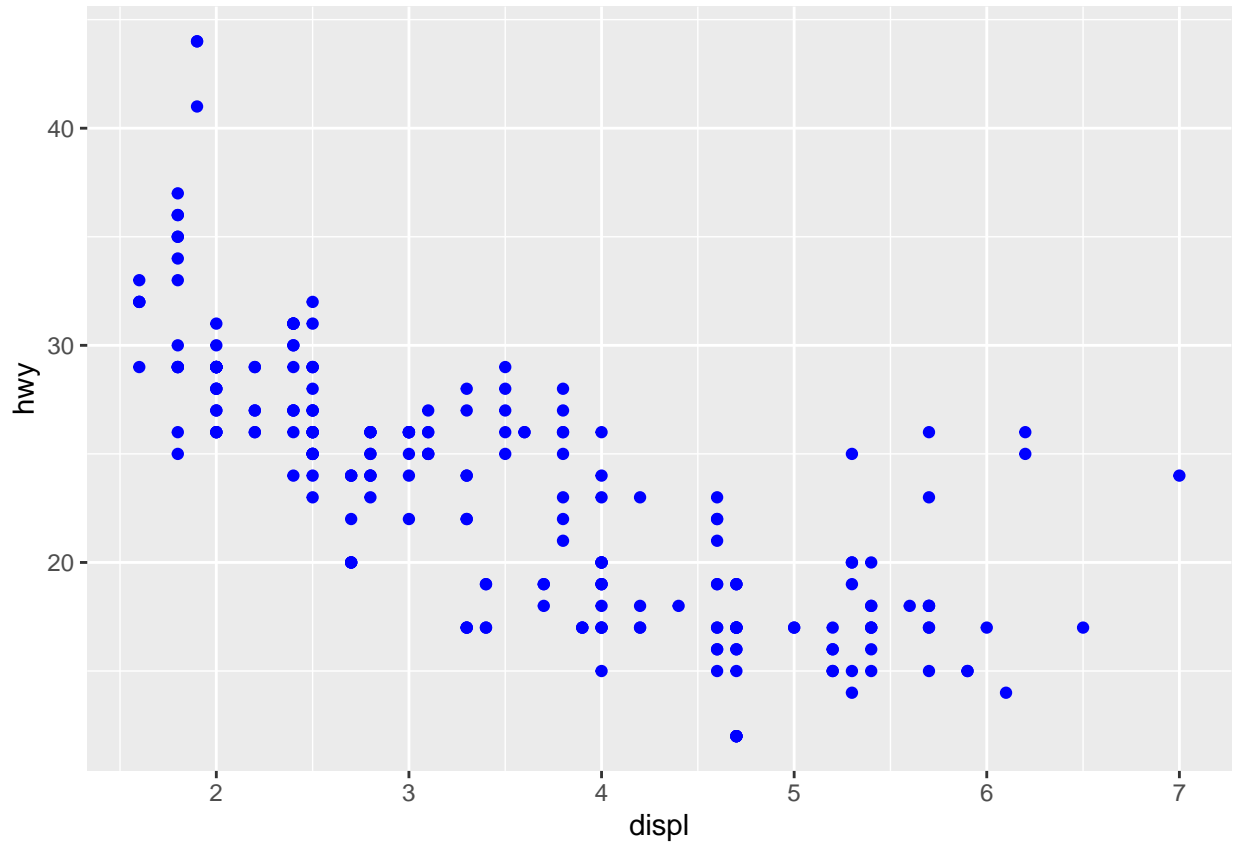
Code 1.

```
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")
```



Answer: The difference between code 1 and code 2 is that in code 1 colour is for representing what the dot is. It is basically a legend, while in code 2 it is for changing the color of the dots.

8)

?mpg

Answer: The result of the command is that it opened up the help tab. It gave the title

Fuel economy data from 1999 to 2008 for 38 popular models of cars, it gave a short description,

usage, and format.

a.

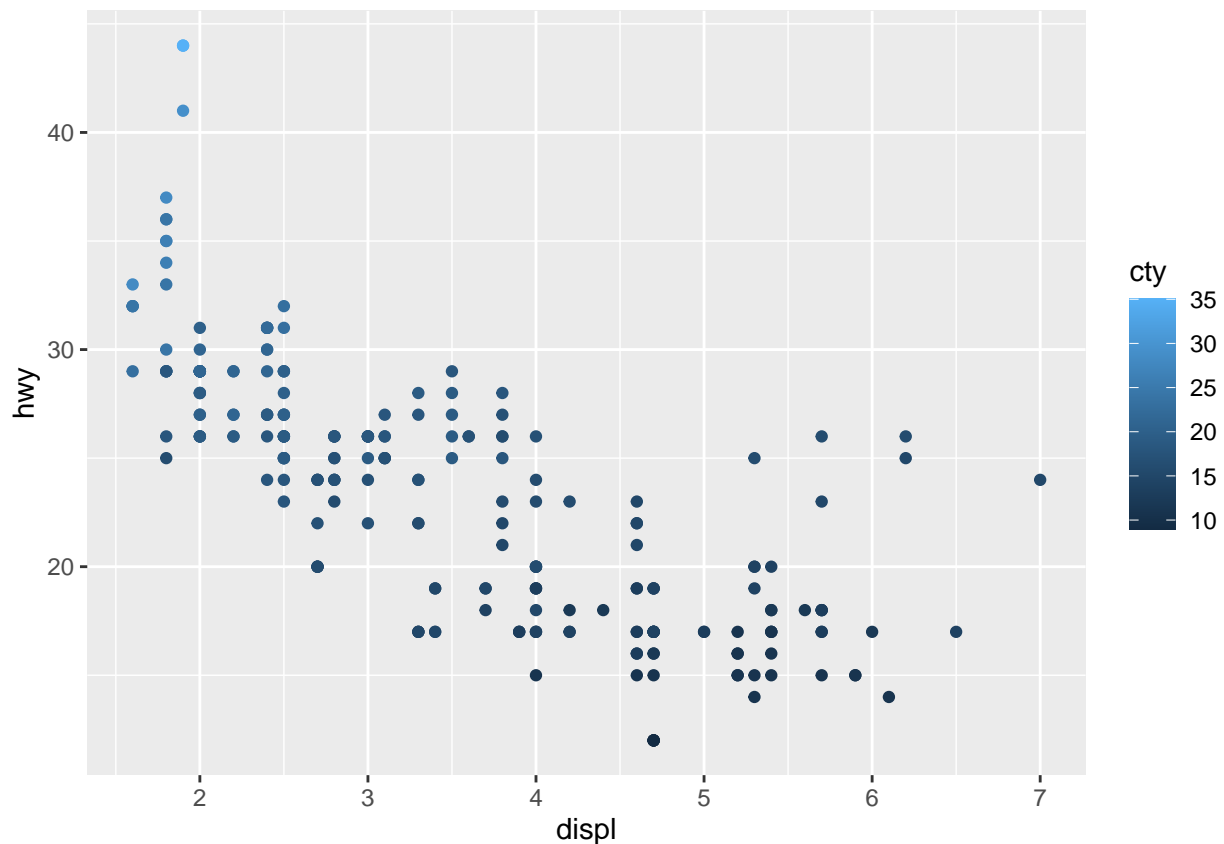
Answer: The variables that are categorical in the mpg data set are manufacturer, model, trans, drv, fl, and class.

b.

Answer: The continuous variables in the mpg data set are displ, year, cyl.

c.

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

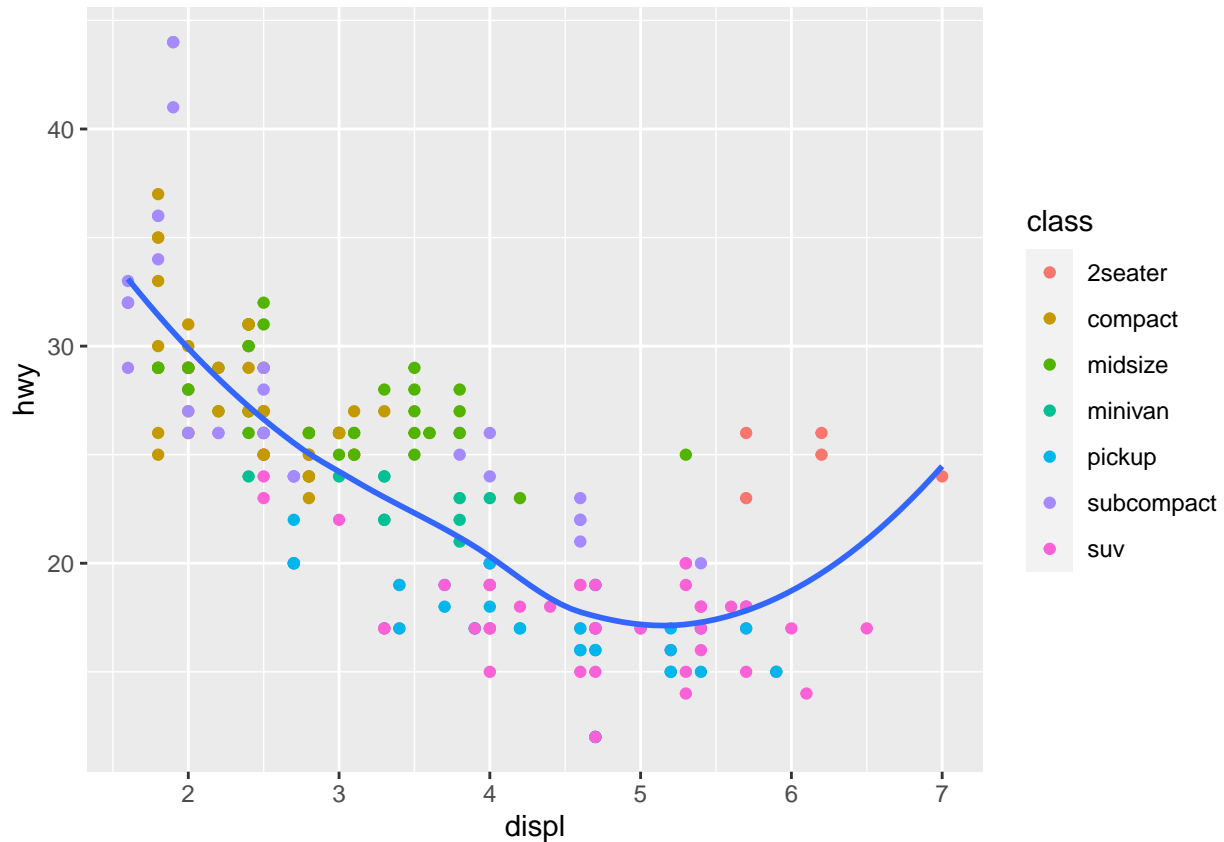


Answer: The plot is keeping track of the cty by having different hues of color blue. ## It produced this output because of the ggplot code.

9)

```
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)

```
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
```

