R for data science

Maruf Ahmed Bhuiyan

5/1/2020

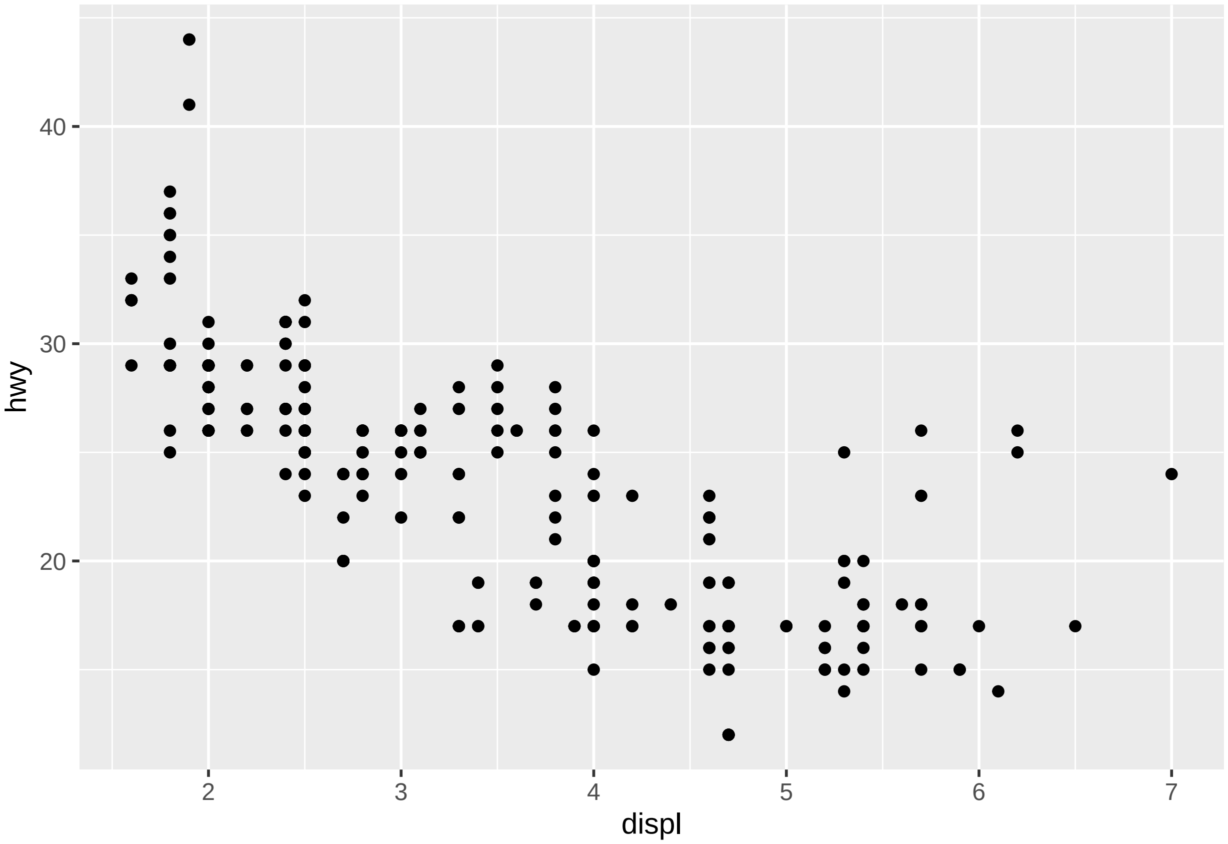
```
library(tidyverse)
```

```
## -- Attaching packages ------------------

## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts -- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

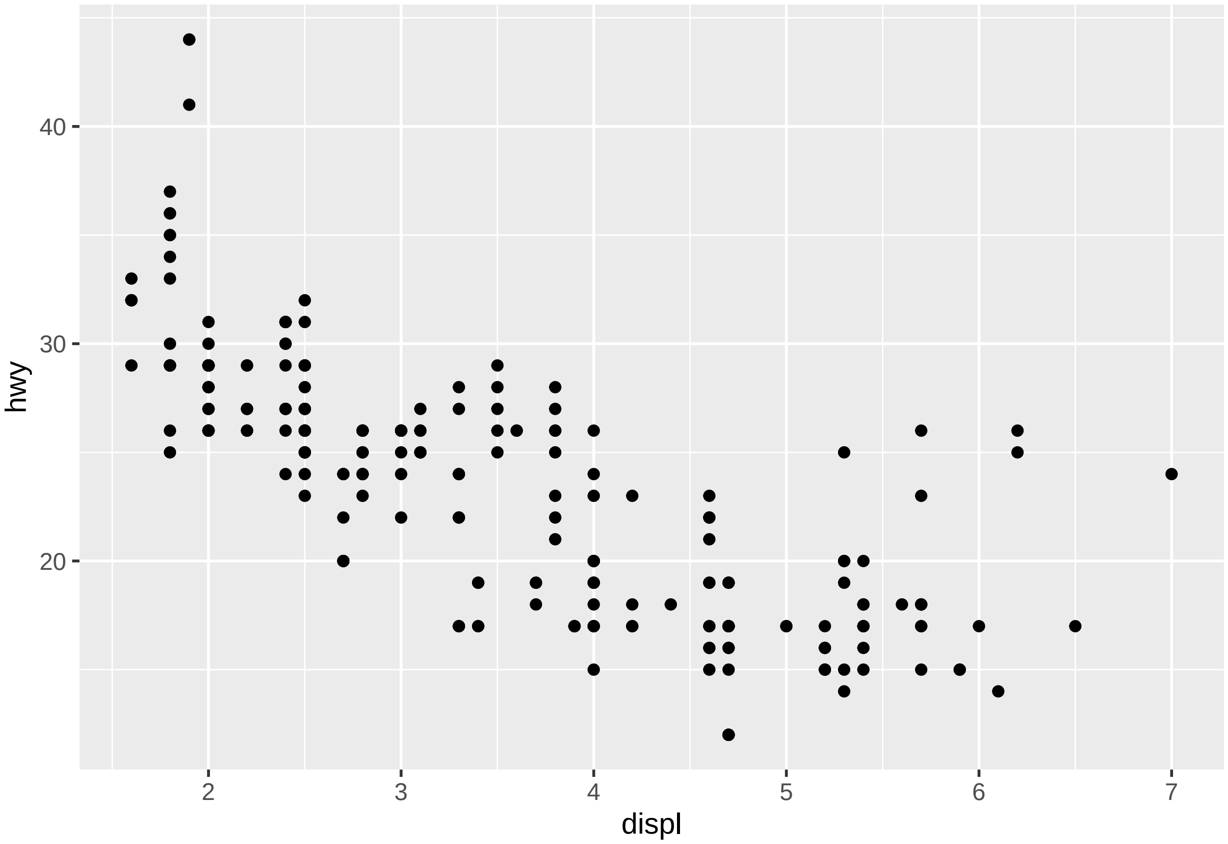
```
mpg
```

```
## # 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(l~ f        18    29 p     comp~
##  2 audi         a4         1.8  1999     4 manual~ f        21    29 p     comp~
##  3 audi         a4         2    2008     4 manual~ f        20    31 p     comp~
##  4 audi         a4         2    2008     4 auto(a~ f        21    30 p     comp~
##  5 audi         a4         2.8  1999     6 auto(l~ f        16    26 p     comp~
##  6 audi         a4         2.8  1999     6 manual~ f        18    26 p     comp~
##  7 audi         a4         3.1  2008     6 auto(a~ f        18    27 p     comp~
##  8 audi         a4 quat~   1.8  1999     4 manual~ 4        18    26 p     comp~
##  9 audi         a4 quat~   1.8  1999     4 auto(l~ 4        16    25 p     comp~
## 10 audi         a4 quat~   2    2008     4 manual~ 4        20    28 p     comp~
## # ... with 224 more rows
```

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



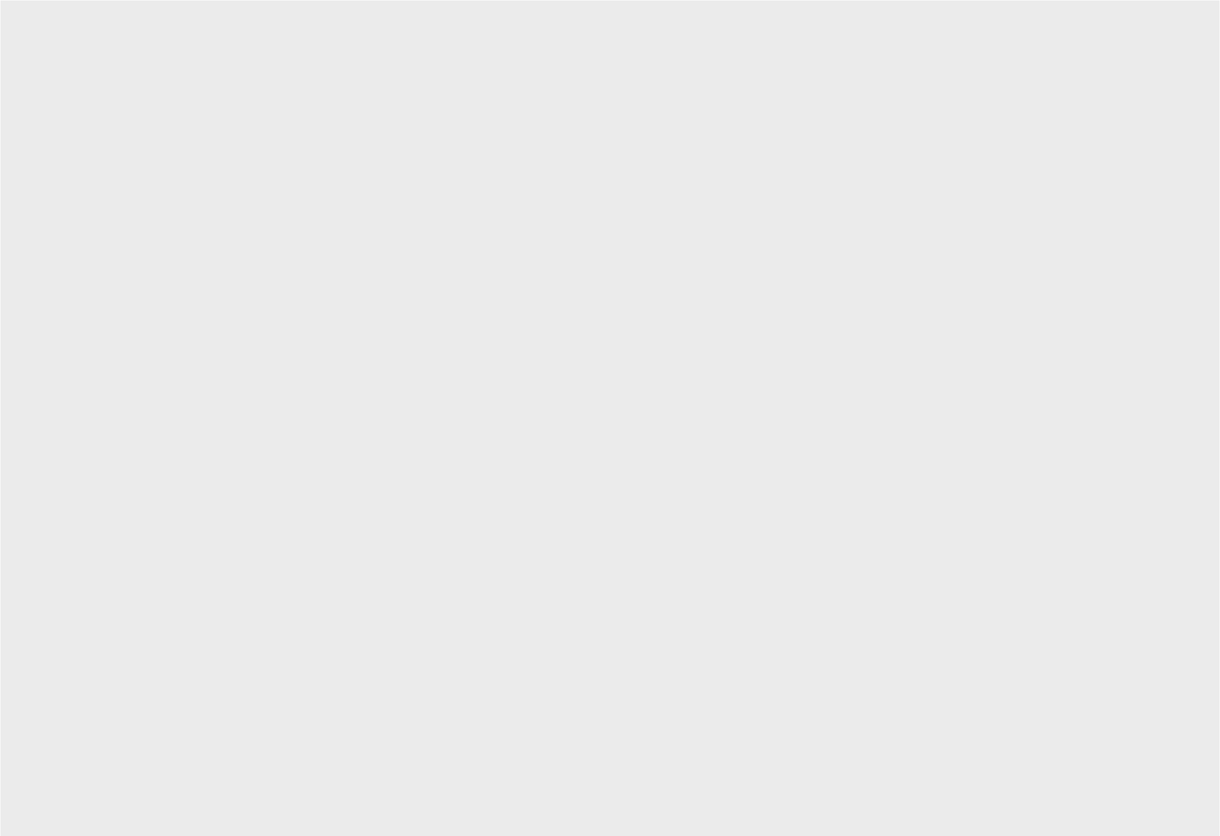
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



3.2.4 Exercises

- Run ggplot(data = mpg). What do you see?

```
ggplot(data = mpg)
```



- How many rows are in mpg? How many columns?

```
nrow(mpg)
```

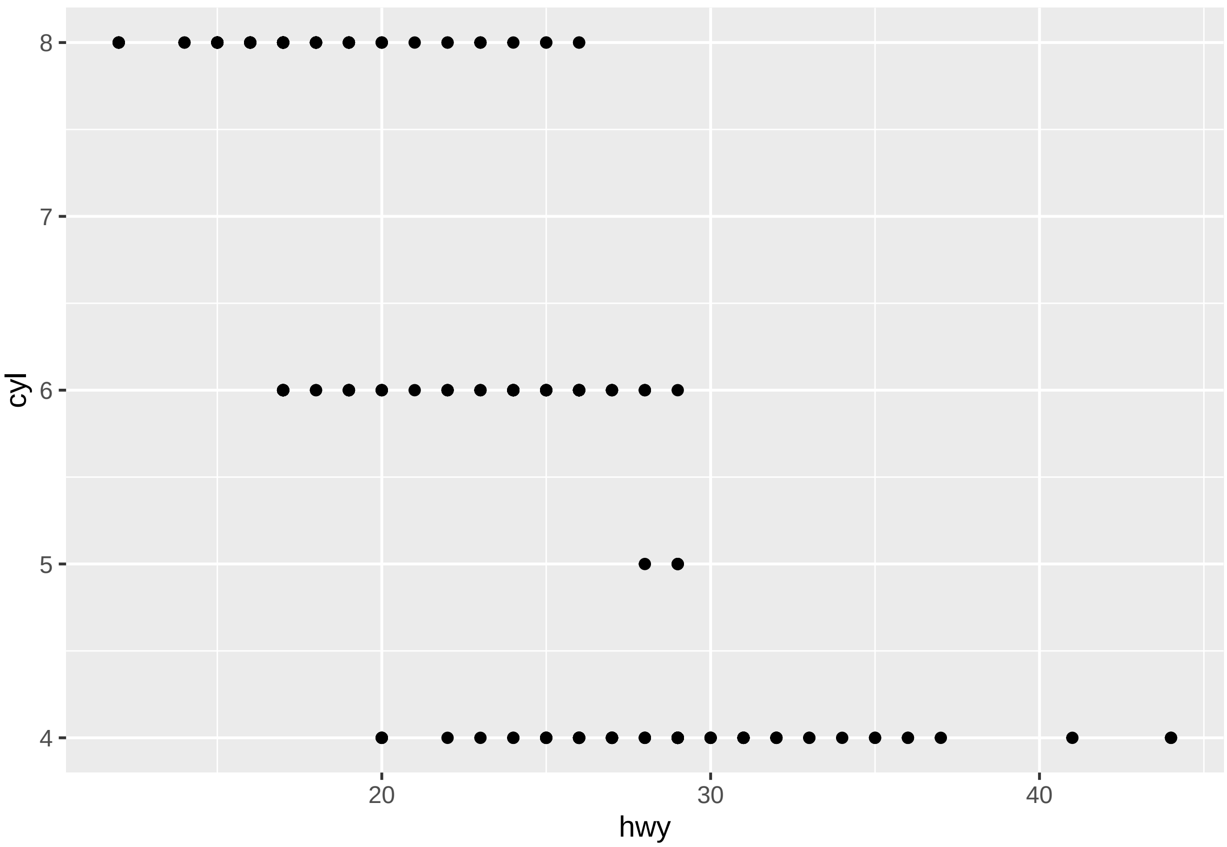
```
## [1] 234
```

```
ncol(mpg)
```

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

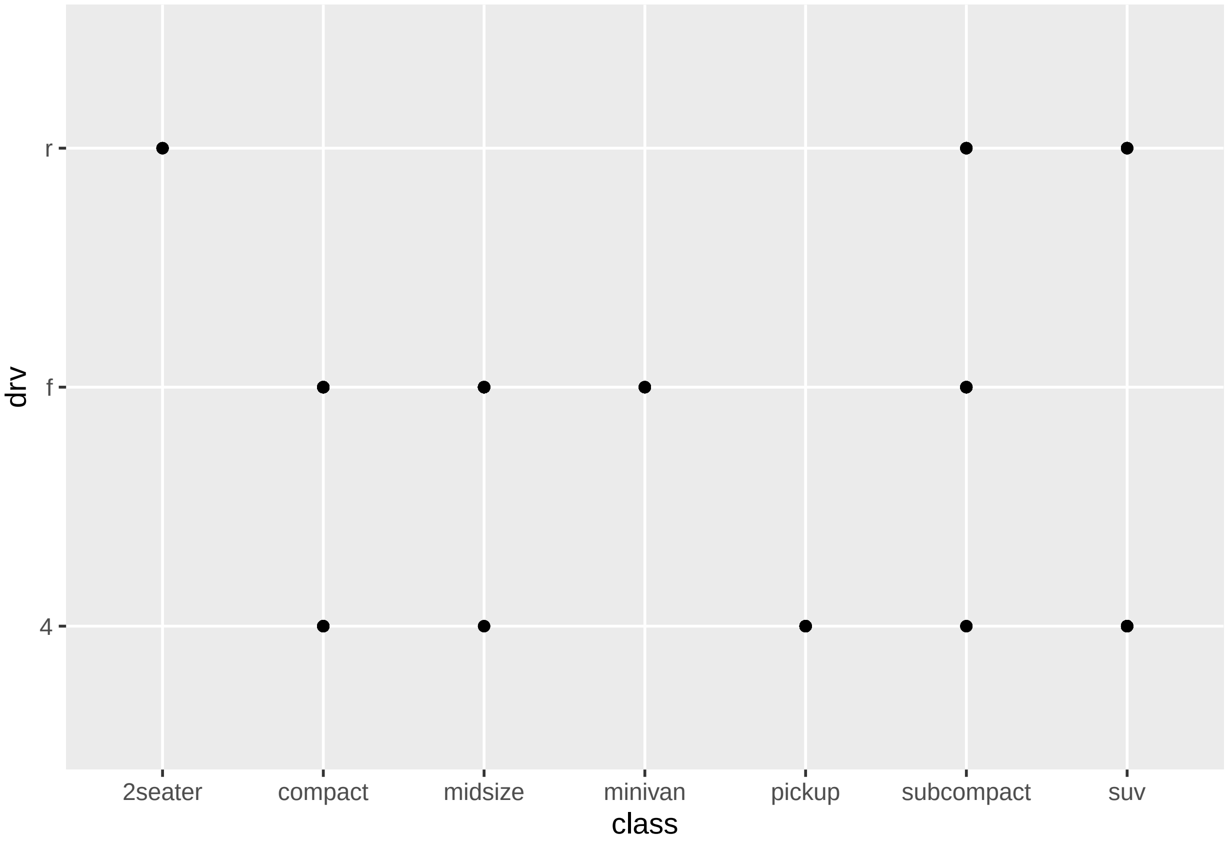
- What does the drv variable describe? Read the help for ?mpg to find out. Type of drive train, where f = front-wheel drive, r = rear wheel drive, 4 = 4wd
- Make a scatterplot of hwy vs cyl.

```
ggplot(data = mpg) +
        geom_point(mapping = aes(x = hwy, y = cyl))
```

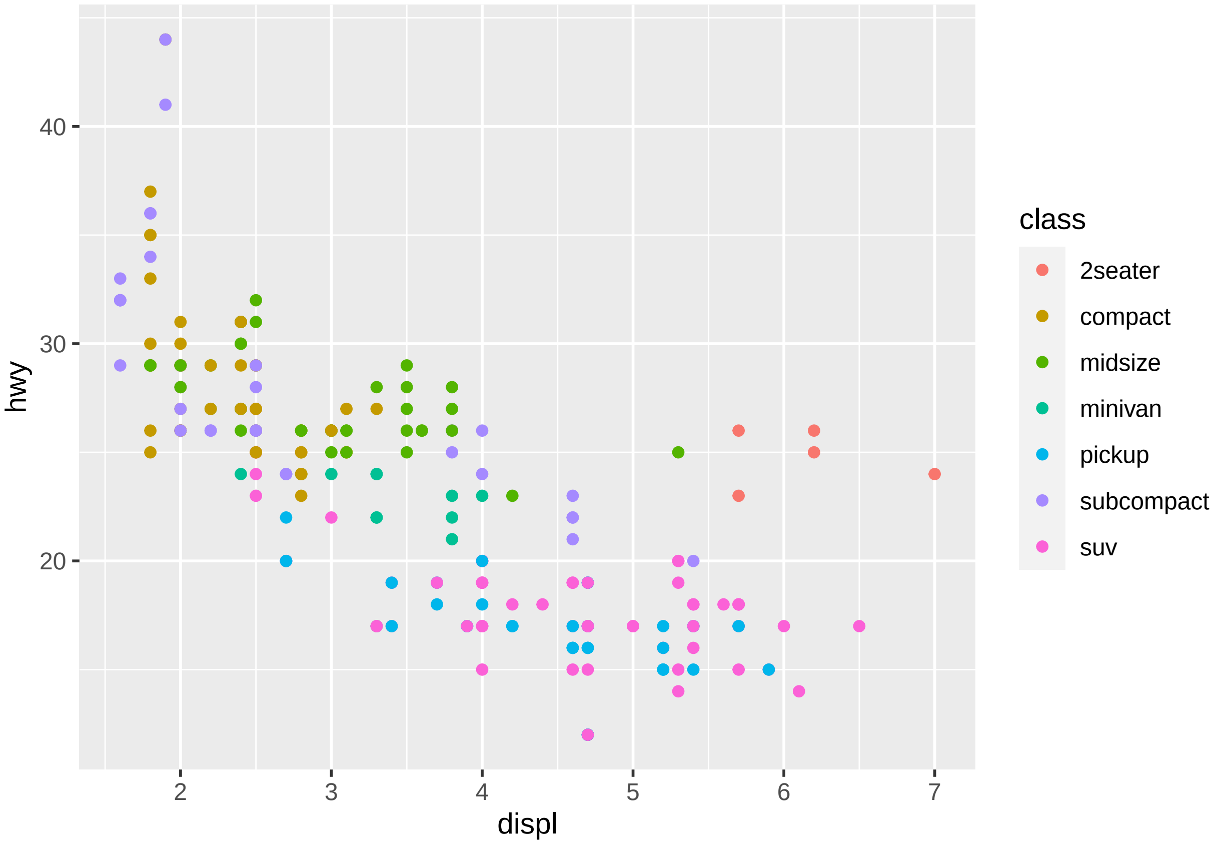


- What happens if you make a scatterplot of class vs drv? Why is the plot not useful?

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

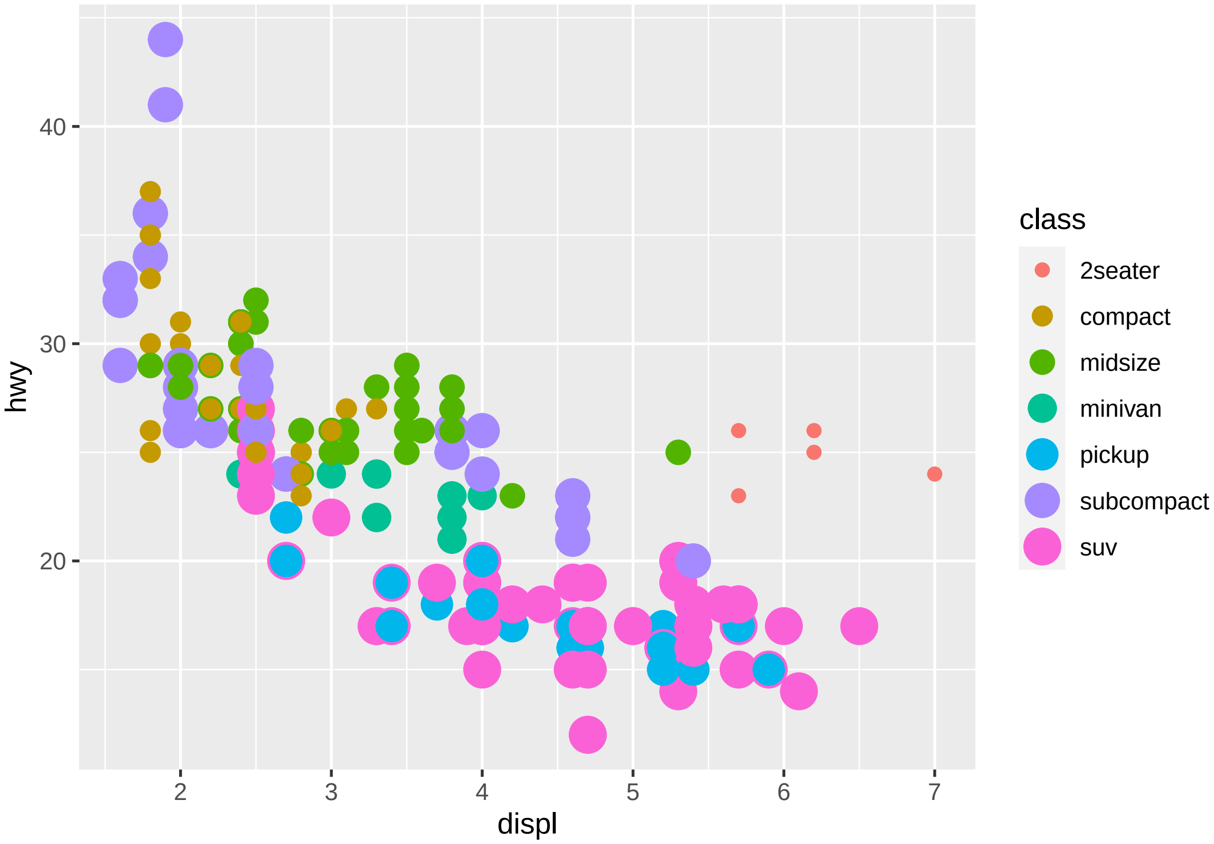


```
ggplot(data = mpg ) +
          geom_point (mapping = aes(x = displ, y = hwy,
                                    color = class))
```



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

```
## Warning: Using size for a discrete variable is not advised.
```

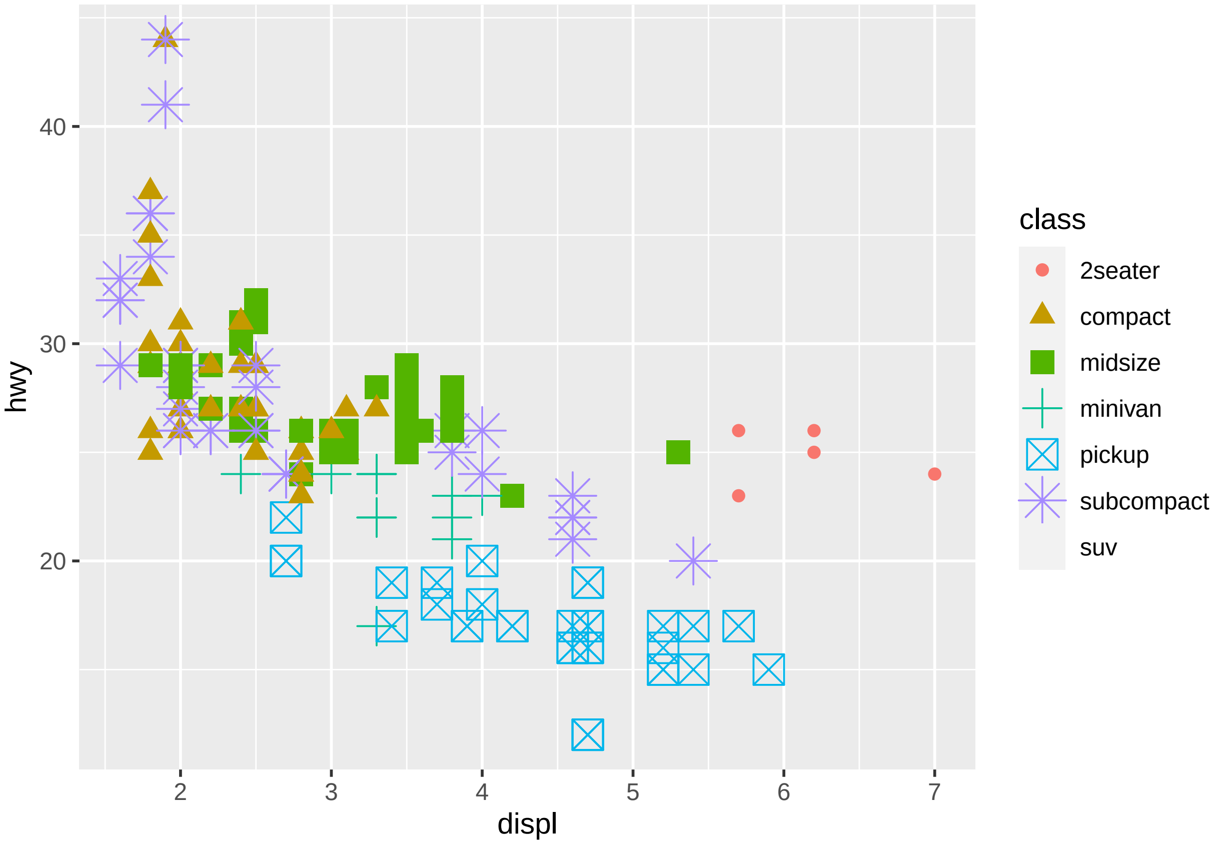


```
ggplot(data =  mpg) +
          geom_point (mapping = aes(x = displ , y =hwy,
                                    color = class,
                                    size = class,
                                    shape = class))
```

```
## Warning: Using size for a discrete variable is not advised.
```

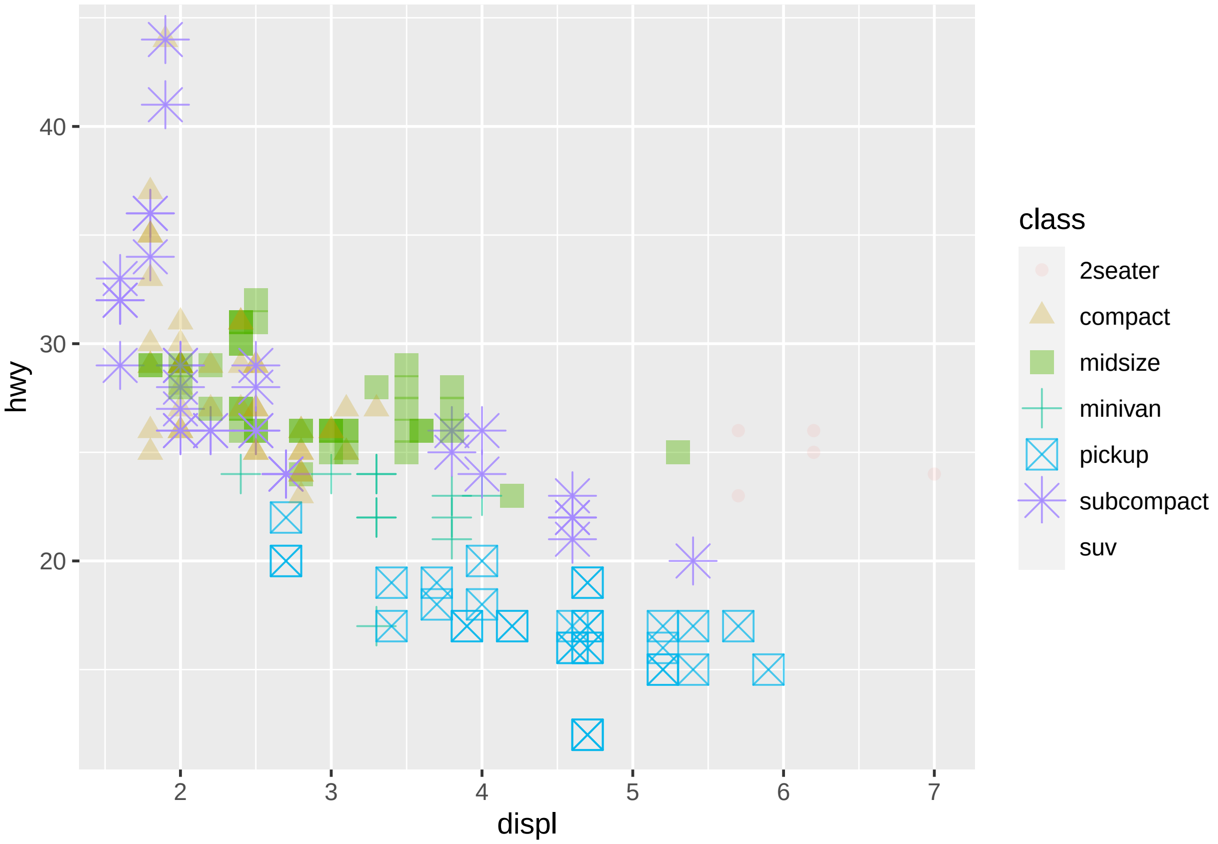
```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 62 rows containing missing values (geom_point).
```



```r
ggplot(data =  mpg) +
          geom_point (mapping = aes(x = displ , y =hwy,
                                    color = class,
                                    size = class,
                                    shape = class,
                                    alpha = class))
```

```
## Warning: Using size for a discrete variable is not advised.

## Warning: Using alpha for a discrete variable is not advised.

## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.

## Warning: Removed 62 rows containing missing values (geom_point).
```



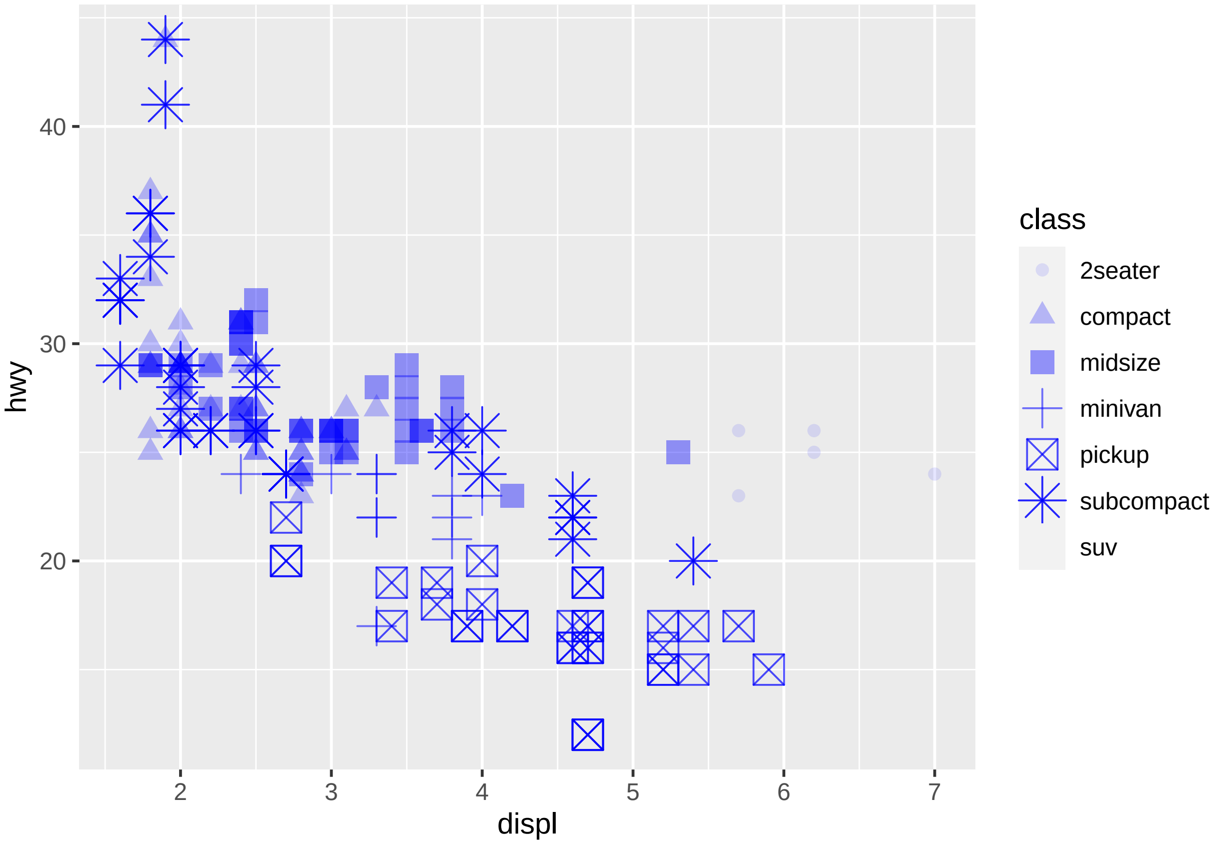
```
ggplot(data =  mpg) +
          geom_point (mapping =
                        aes(x = displ , y =hwy,
                        size = class,
                        shape = class,
                        alpha = class), color = "blue")
```

```
## Warning: Using size for a discrete variable is not advised.
```

```
## Warning: Using alpha for a discrete variable is not advised.
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

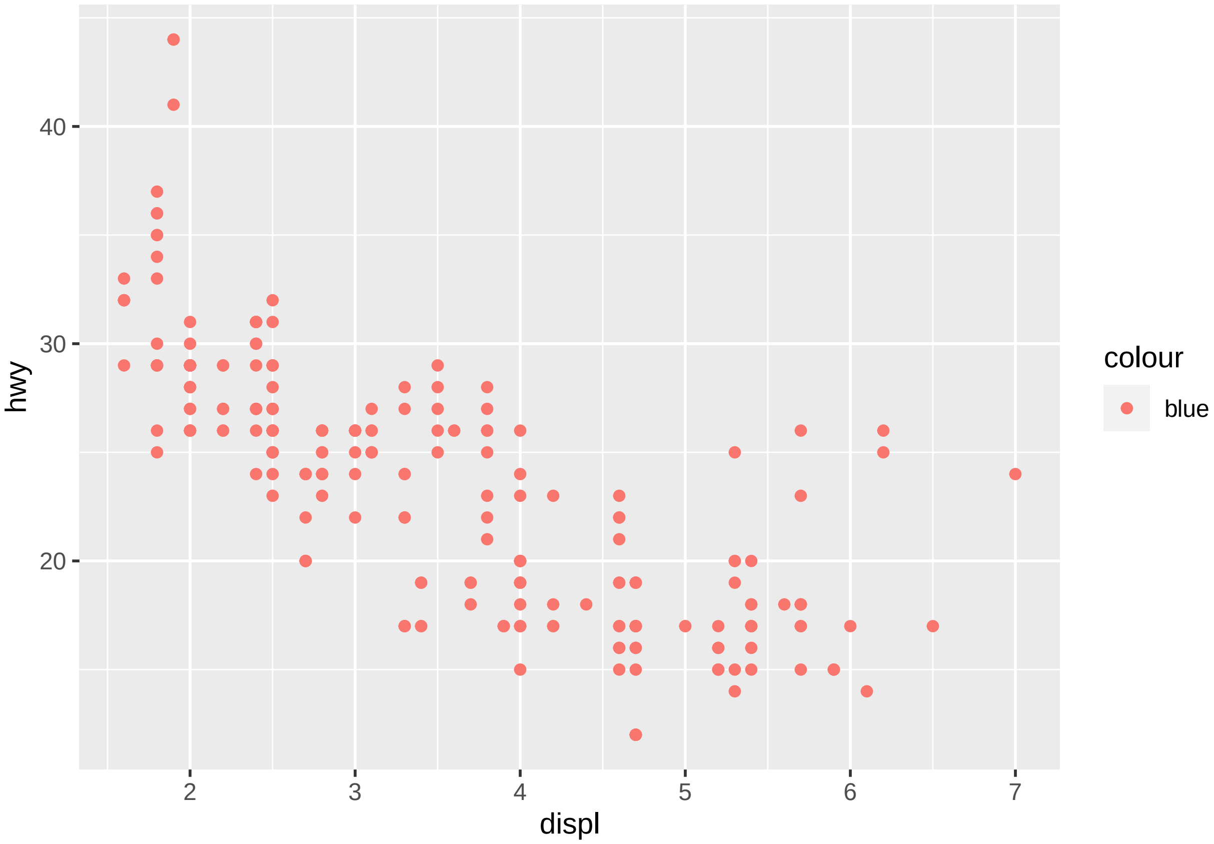
```
## Warning: Removed 62 rows containing missing values (geom_point).
```



3.3.1 Exercises

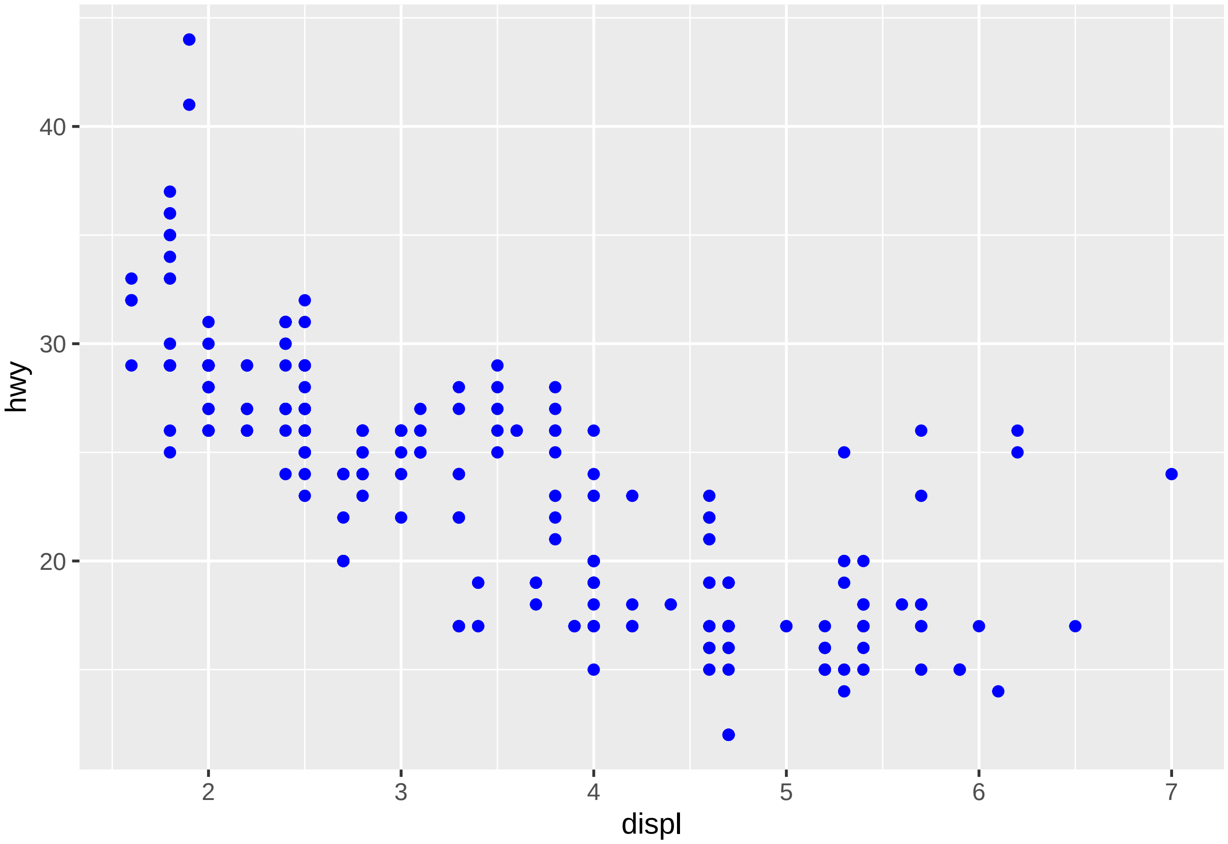
- What's gone wrong with this code? Why are the points not blue?

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



### Manual mapping of aesthetics must be specified outside the ***aes*** argument.

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



- Which variables in mpg are categorical? Which variables are continuous? (Hint: type ?mpg to read the documentation for the dataset). How can you see this information when you run mpg?

```
?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" ...
```

```
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(l~ f        18    29 p     comp~
##  2 audi         a4         1.8  1999     4 manual~ f        21    29 p     comp~
##  3 audi         a4         2    2008     4 manual~ f        20    31 p     comp~
##  4 audi         a4         2    2008     4 auto(a~ f        21    30 p     comp~
##  5 audi         a4         2.8  1999     6 auto(l~ f        16    26 p     comp~
##  6 audi         a4         2.8  1999     6 manual~ f        18    26 p     comp~
##  7 audi         a4         3.1  2008     6 auto(a~ f        18    27 p     comp~
##  8 audi         a4 quat~   1.8  1999     4 manual~ 4        18    26 p     comp~
##  9 audi         a4 quat~   1.8  1999     4 auto(l~ 4        16    25 p     comp~
## 10 audi         a4 quat~   2    2008     4 manual~ 4        20    28 p     comp~
## # ... with 224 more rows
```

Since, mpg is a tibble, simply printing it would tell me about the variables.

We can also use str() to view how the variables are coded.

No categorical variable is present in this tibble. displ, year, cyl, cty, hwy are continuous.

- Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?

```
#ggplot(data =  mpg) +
          #geom_point (mapping = aes(x = displ, y = hwy,
                                    #size = cyl,
                                    #color = cyl,
                                    #shape = cyl))
```

Continuous variables can not be mapped to shape.

- What happens if you map the same variable to multiple aesthetics?

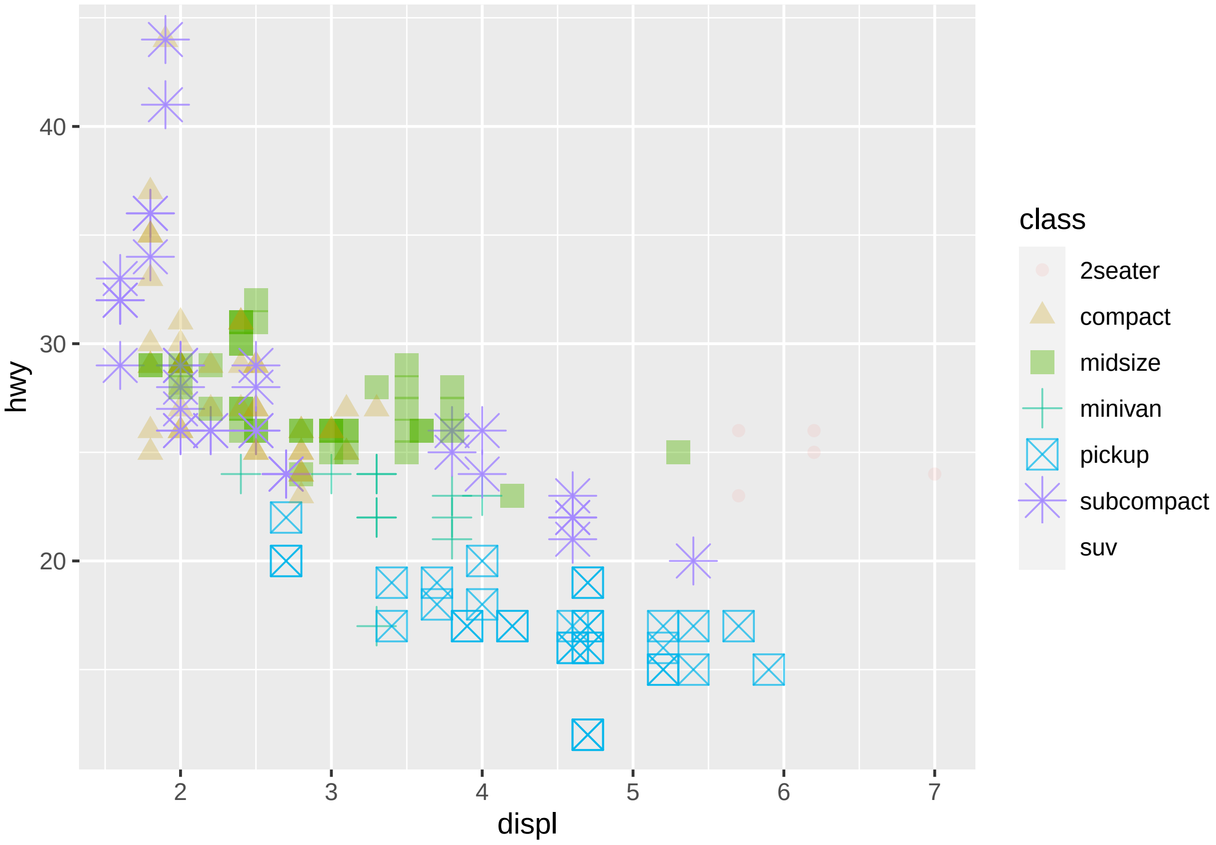
```
ggplot(data =  mpg) +
          geom_point (mapping = aes(x = displ , y =hwy,
                                    color = class,
                                    size = class,
                                    shape = class,
                                    alpha = class))
```

```
## Warning: Using size for a discrete variable is not advised.
```

```
## Warning: Using alpha for a discrete variable is not advised.
```

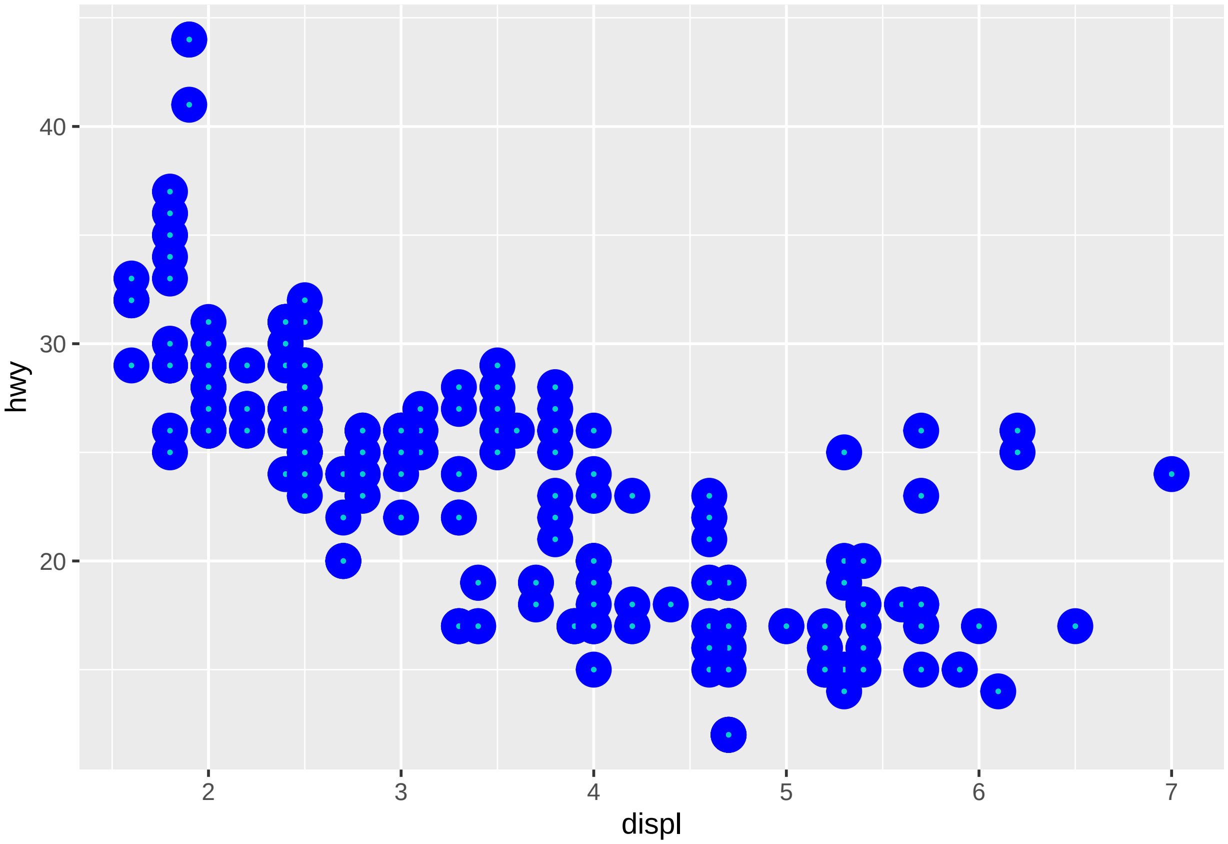
```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 62 rows containing missing values (geom_point).
```

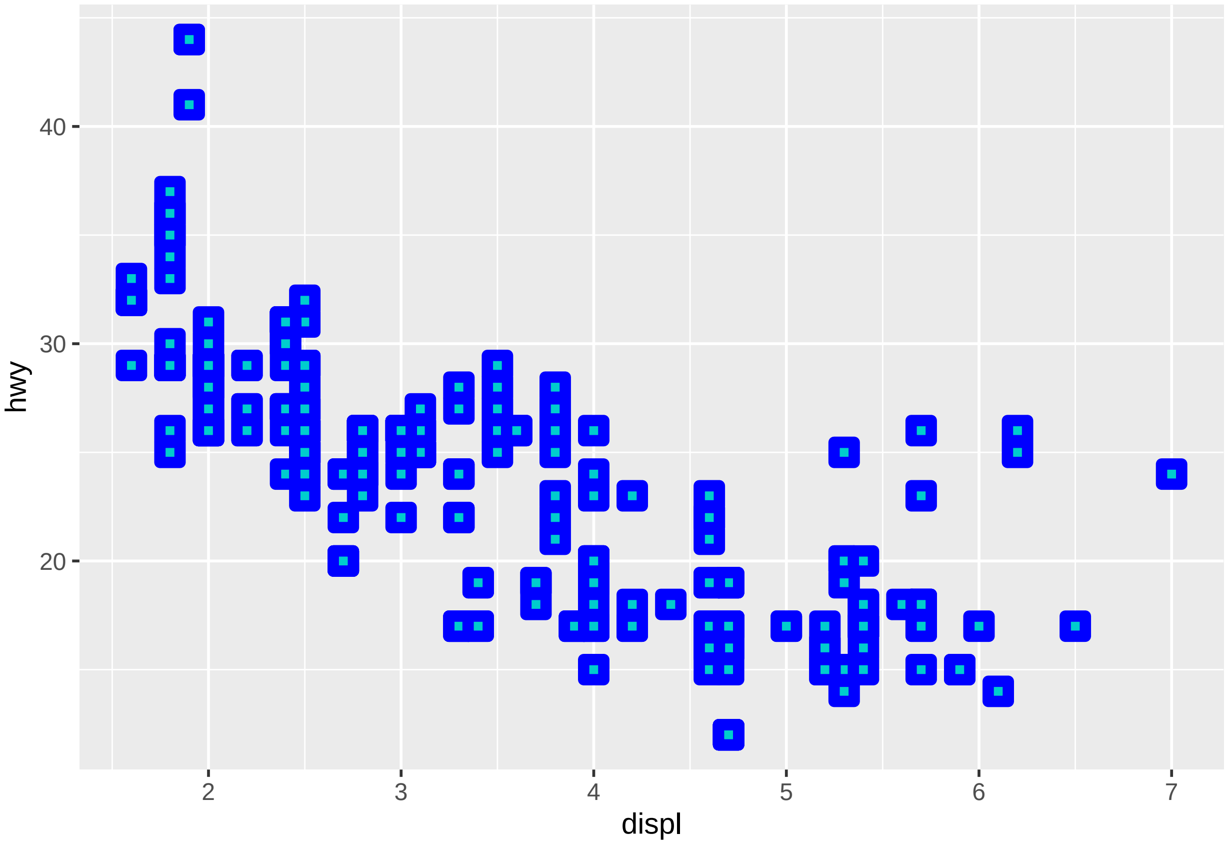


- What does the stroke aesthetic do? What shapes does it work with? (Hint: use ?geom_point)

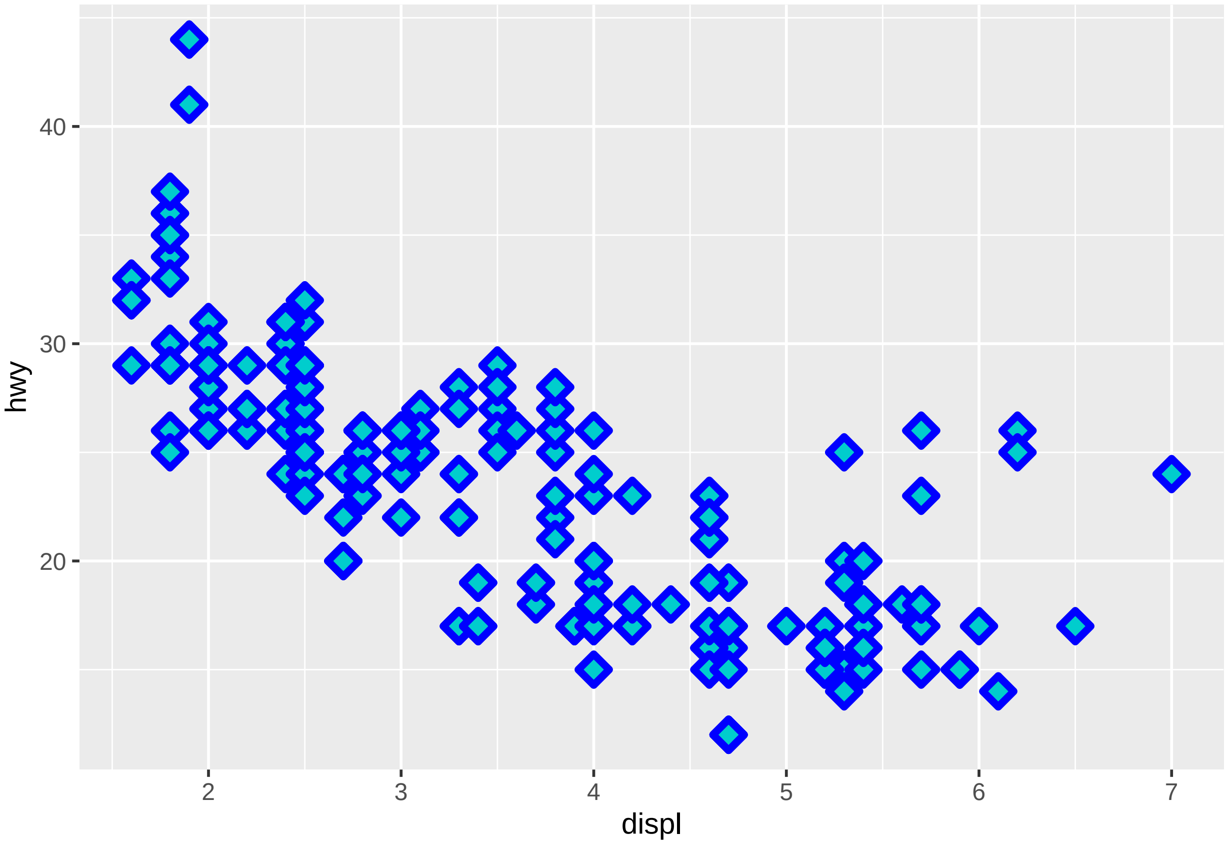
```
?geom_point

# For shapes that have a border (21-24), we can color the inside (fill()) and
# outside(border-color()) separately. The stroke aesthetic can be used to modify the width of the borde

# The hollow shapes (0-14) have a border determined by color;
# The solid shapes (15-18) are filled with colour;
# The filled shapes (21-24) have a border of colour and are filled with fill.

ggplot(mpg, aes(displ, hwy)) +
  geom_point(shape = 21, colour = "blue", fill = "cyan3", size = 1, stroke = 4)
```



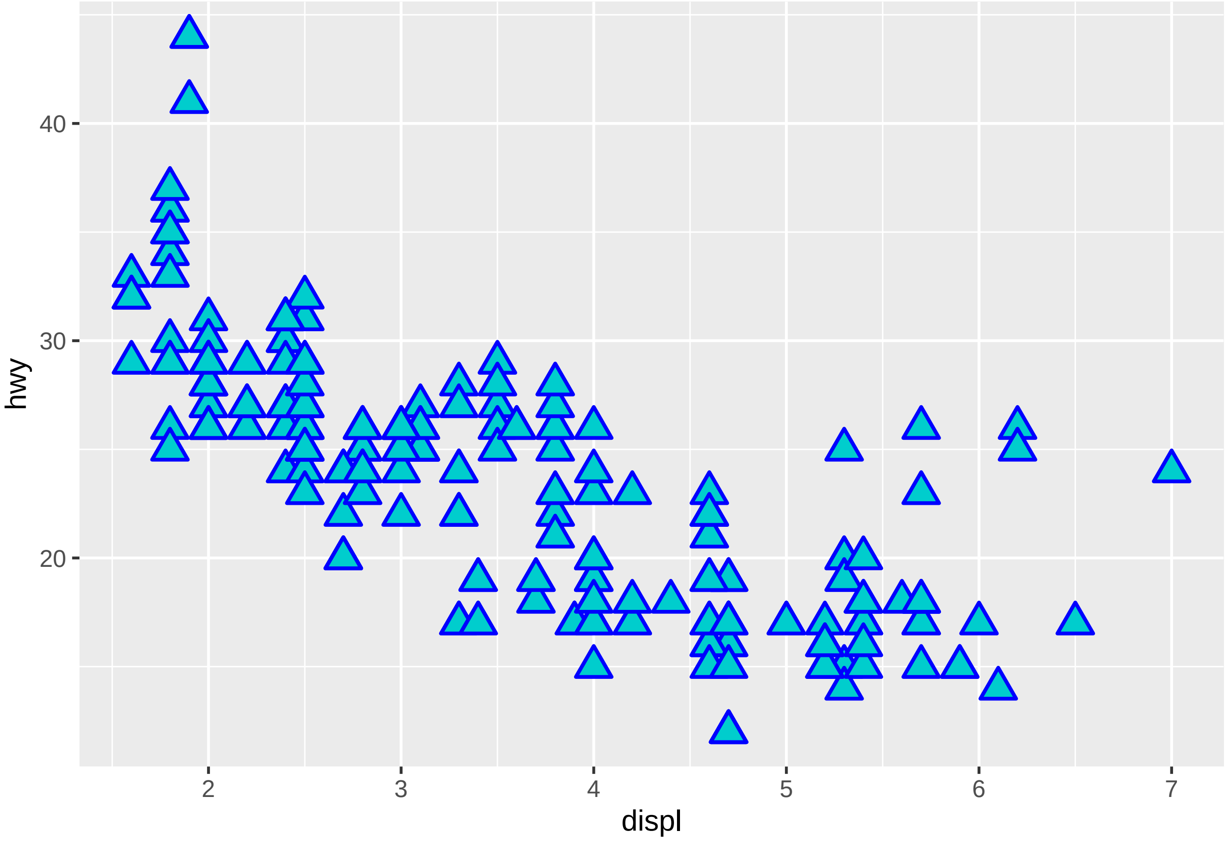
```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(shape = 22, colour = "blue", fill = "cyan3", size = 2, stroke = 3)
```



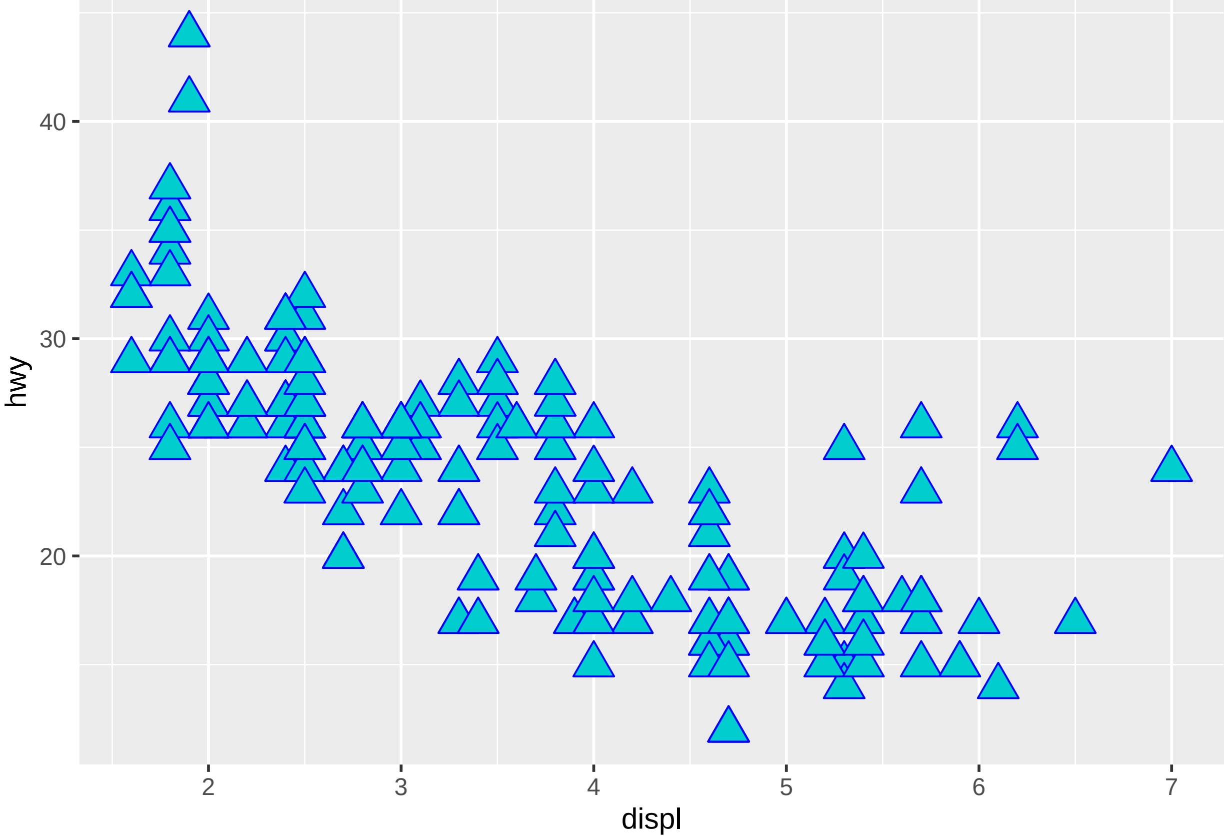
```r
ggplot(mpg, aes(displ, hwy)) +
  geom_point(shape = 23, colour = "blue", fill = "cyan3", size = 3, stroke = 2)
```



```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(shape = 24, colour = "blue", fill = "cyan3", size = 4, stroke = 1)
```



```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(shape = 24, colour = "blue", fill = "cyan3", size = 5, stroke = 0.5)
```



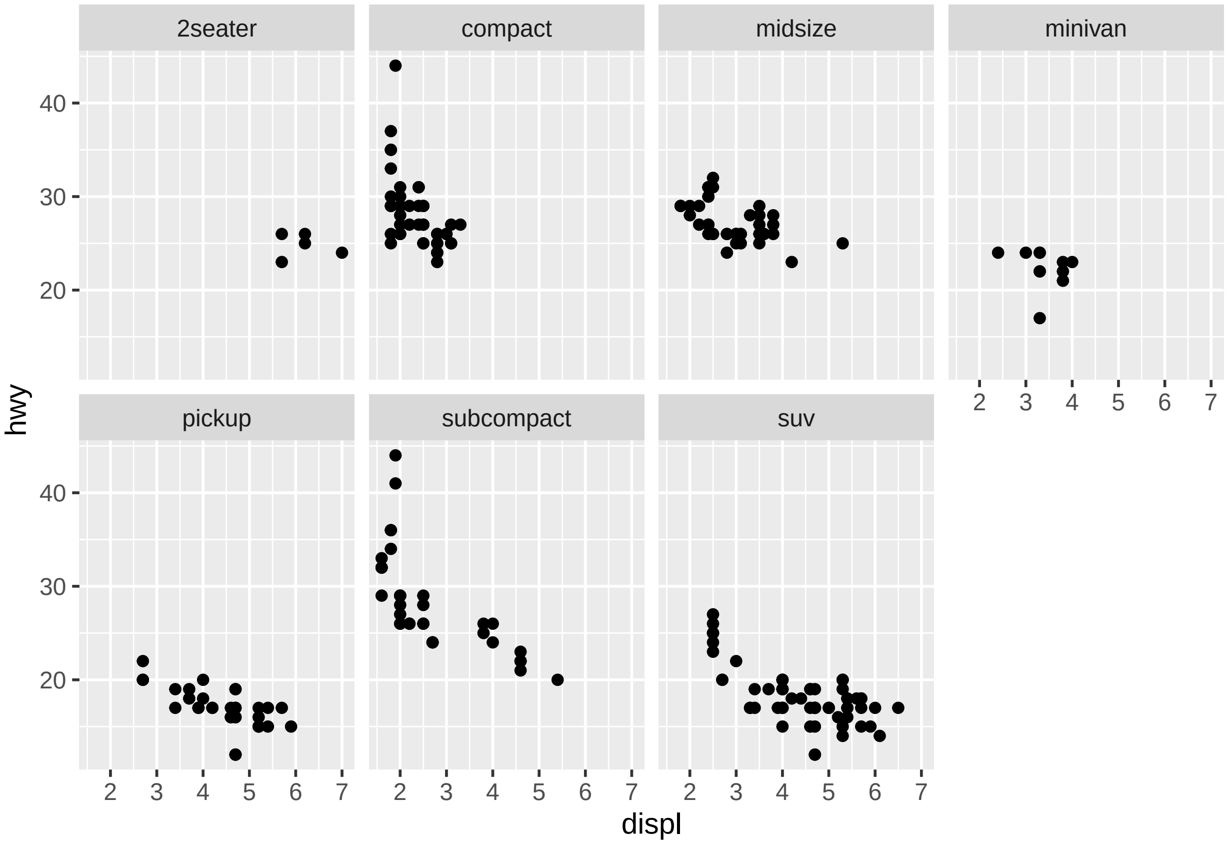
+

What happens if you map an aesthetic to something other than a variable name, like aes(colour = displ < 5)? Note, you'll also need to specify x and y.

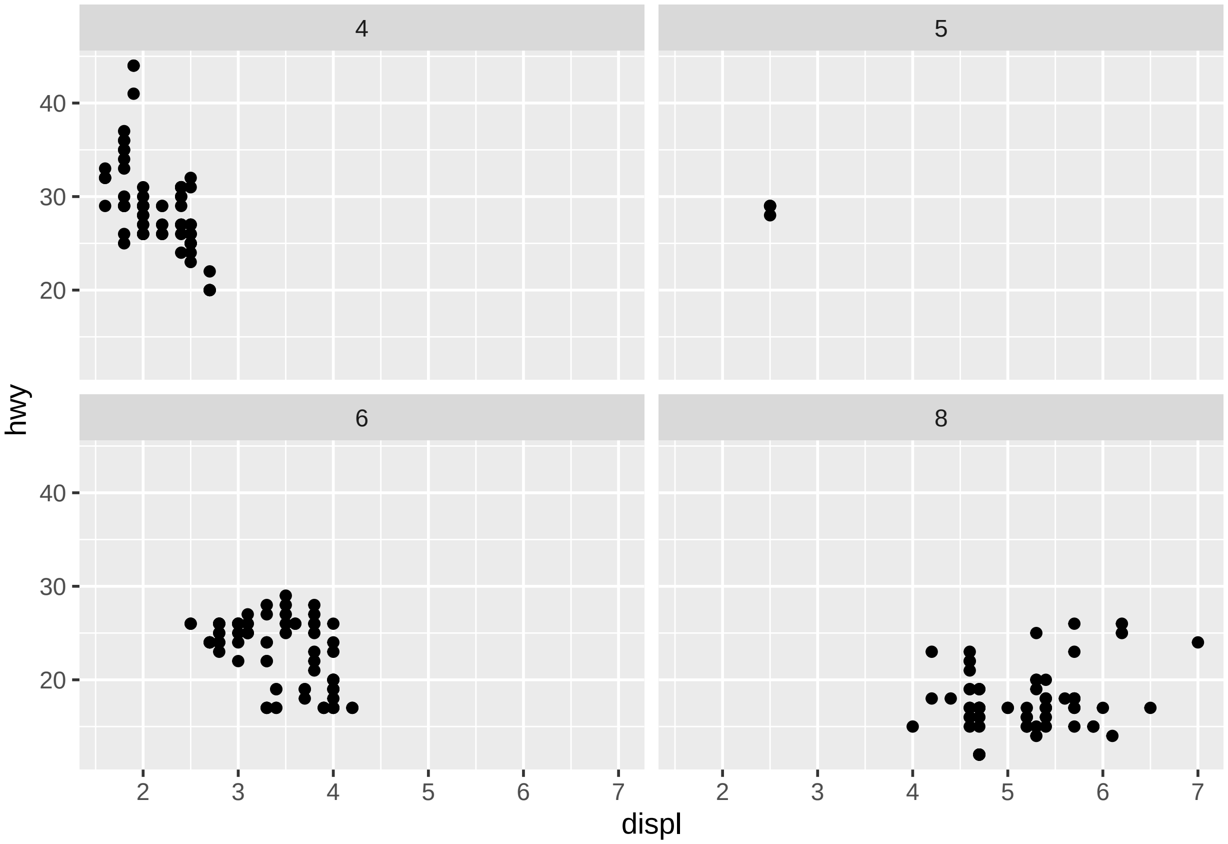
```
#ggplot(mpg) +
  #geom_point(mapping = aes(displ, hwy), shape = 24,
             #colour = displ < 5, fill = "cyan3",
             #size = 10, stroke = 0.5)
```

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

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```



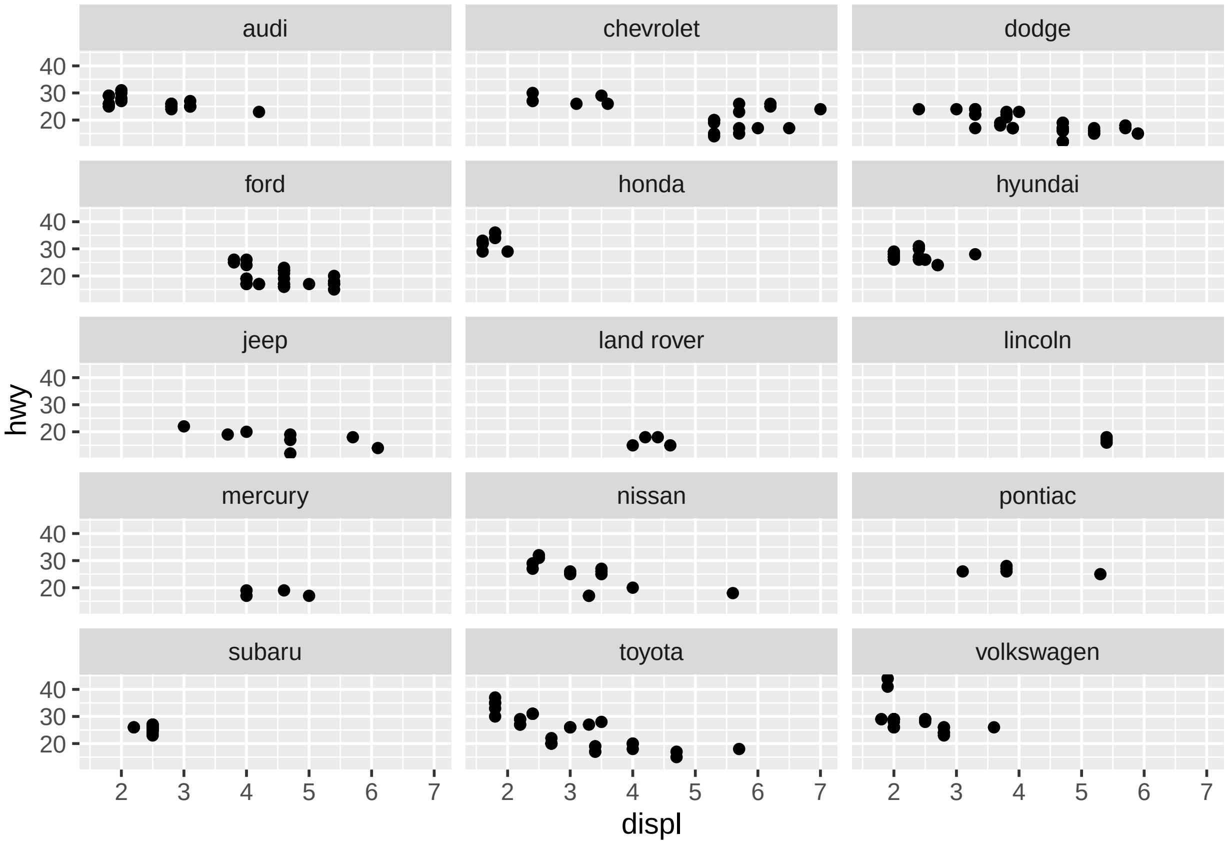
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ cyl, nrow = 2)
```



```r
table(mpg$manufacturer)
```

```
##
##       audi  chevrolet      dodge       ford      honda    hyundai       jeep
##         18         19         37         25          9         14          8
## land rover    lincoln    mercury     nissan    pontiac     subaru     toyota
##          4          3          4         13          5         14         34
## volkswagen
##         27
```

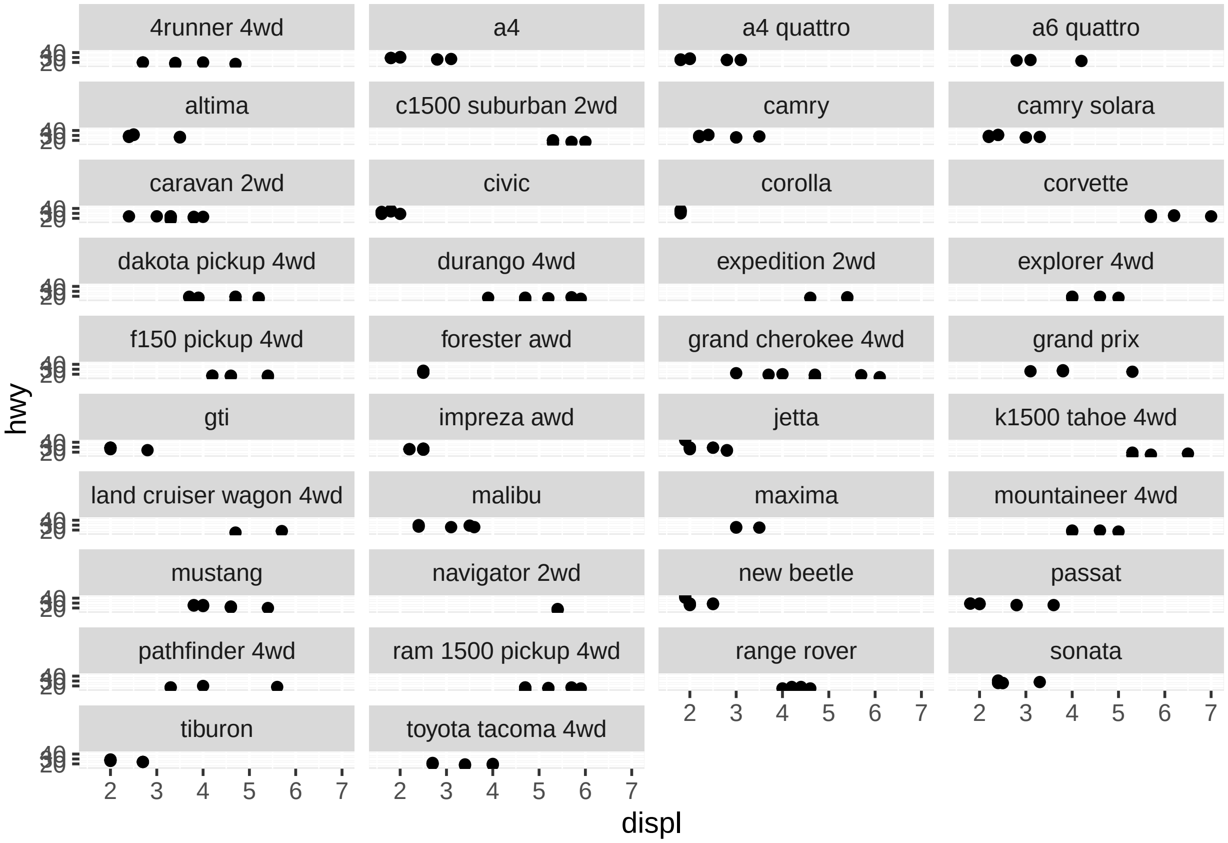
```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ manufacturer, nrow = 5)
```



```r
table(mpg$model)
```

```
##
##            4runner 4wd                     a4             a4 quattro
##                      6                      7                      8
##             a6 quattro                 altima     c1500 suburban 2wd
##                      3                      6                      5
##                  camry           camry solara            caravan 2wd
##                      7                      7                     11
##                  civic                corolla               corvette
##                      9                      5                      5
##      dakota pickup 4wd            durango 4wd         expedition 2wd
##                      9                      7                      3
##           explorer 4wd        f150 pickup 4wd           forester awd
##                      6                      7                      6
##     grand cherokee 4wd             grand prix                    gti
##                      8                      5                      5
##            impreza awd                  jetta        k1500 tahoe 4wd
##                      8                      9                      4
## land cruiser wagon 4wd                 malibu                 maxima
##                      2                      5                      3
##        mountaineer 4wd                mustang          navigator 2wd
##                      4                      9                      3
##             new beetle                 passat         pathfinder 4wd
##                      6                      7                      4
##    ram 1500 pickup 4wd            range rover                 sonata
##                    10                     4                      7
##               tiburon      toyota tacoma 4wd
##                     7                     7
```

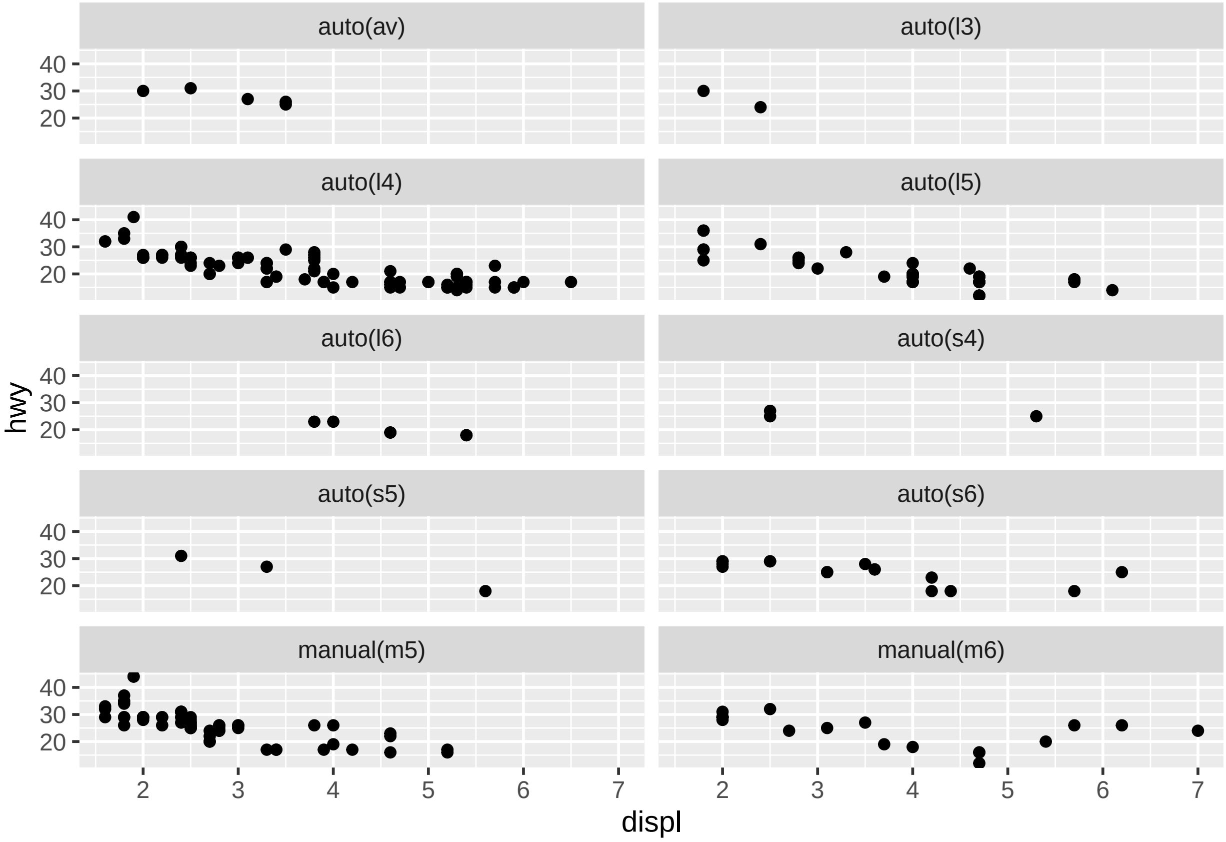
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ model, nrow = 10)
```



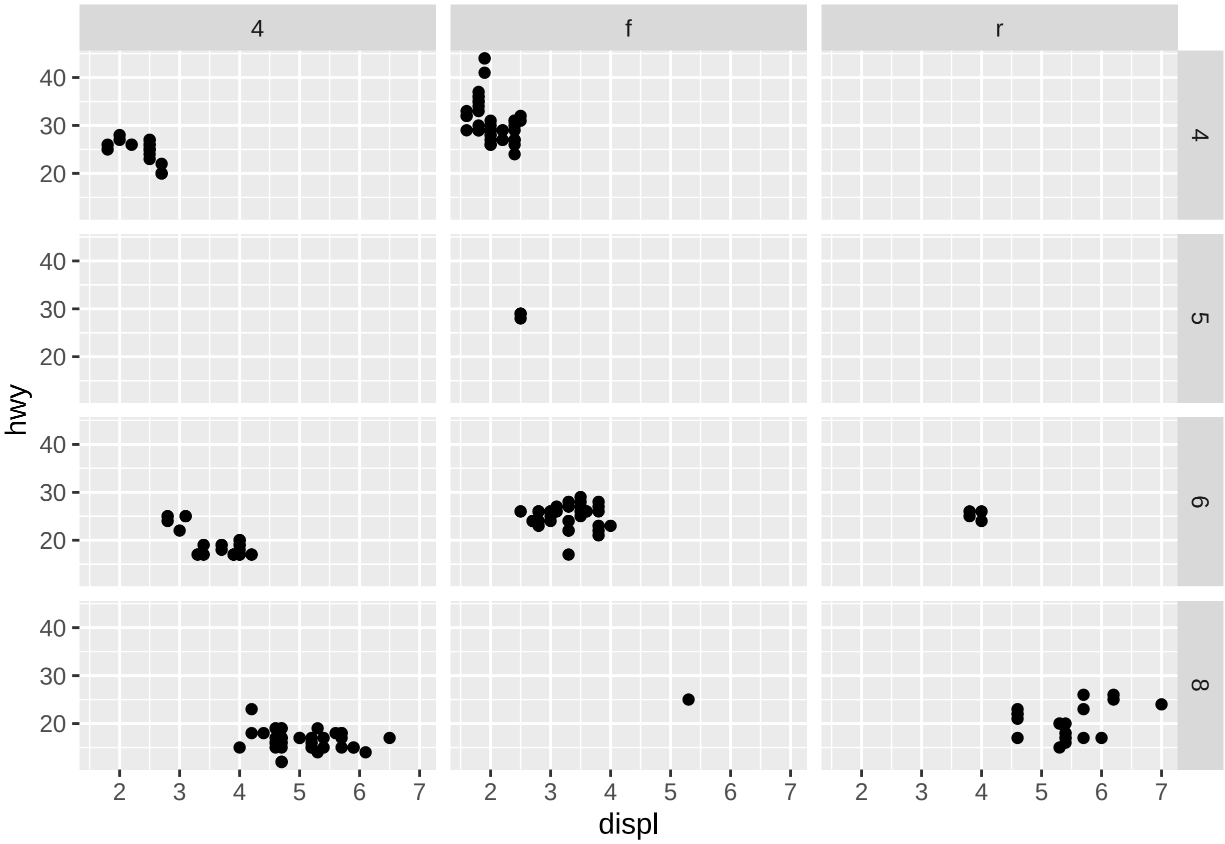
```
table(mpg$trans)
```

```
## 
##   auto(av)   auto(l3)   auto(l4)   auto(l5)   auto(l6)   auto(s4)   auto(s5) 
##          5          2         83         39          6          3          3 
##   auto(s6) manual(m5) manual(m6) 
##         16         58         19
```

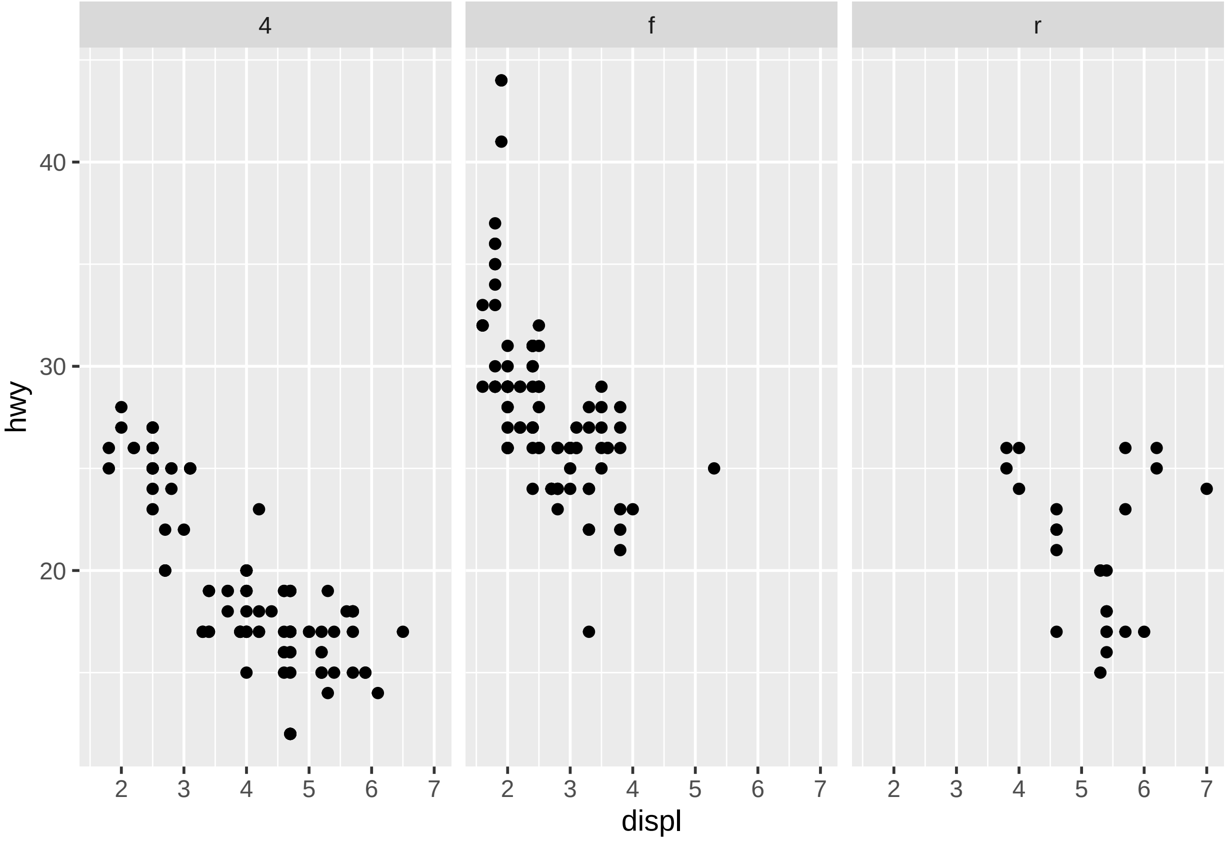
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ trans, nrow = 5)
```



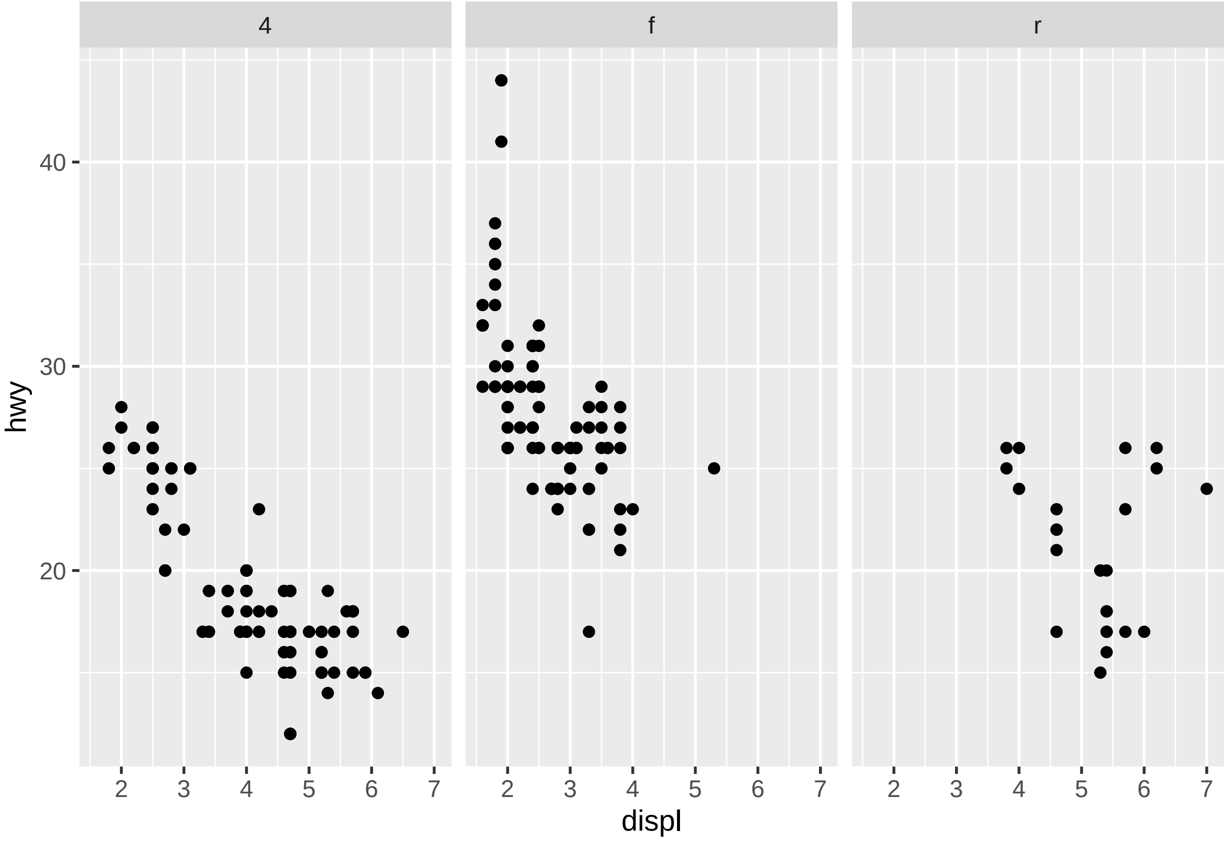
```
ggplot(data = mpg ) +
          geom_point(mapping = aes(x = displ, y =hwy)) +
          facet_grid(cyl~drv)
```



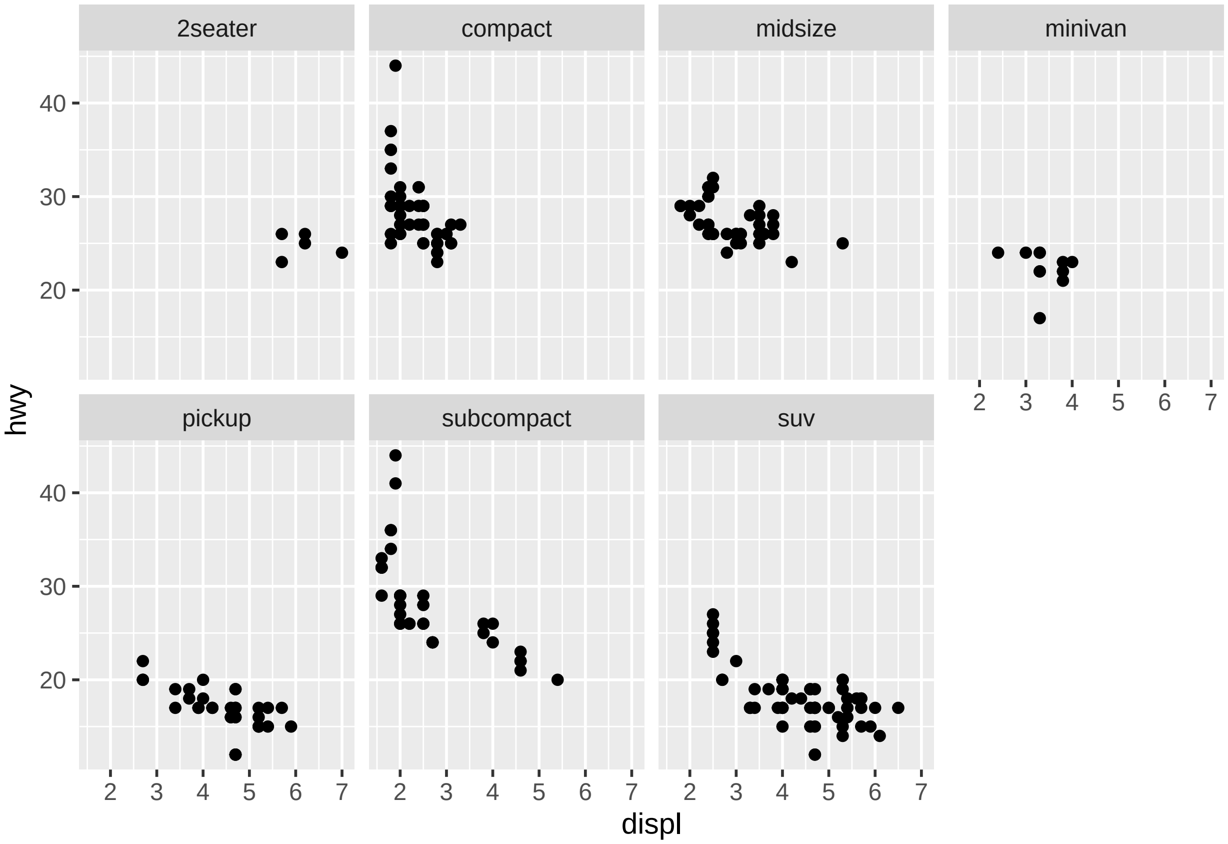
```
ggplot(data =  mpg) +
          geom_point(mapping = aes(x = displ, y = hwy)) +
          facet_grid(.~drv)
```



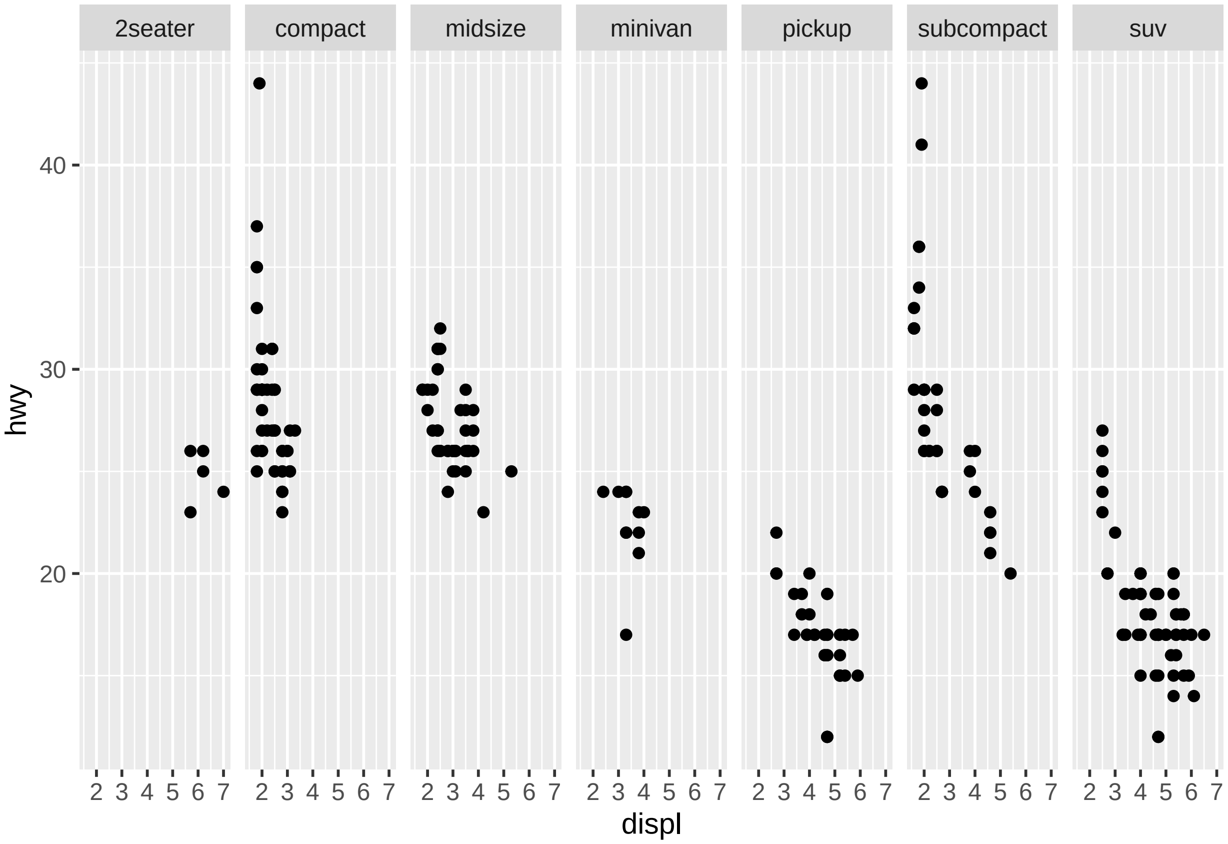
```
ggplot(data =  mpg) +
          geom_point(mapping = aes(x = displ, y = hwy)) +
          facet_grid(~drv)
```



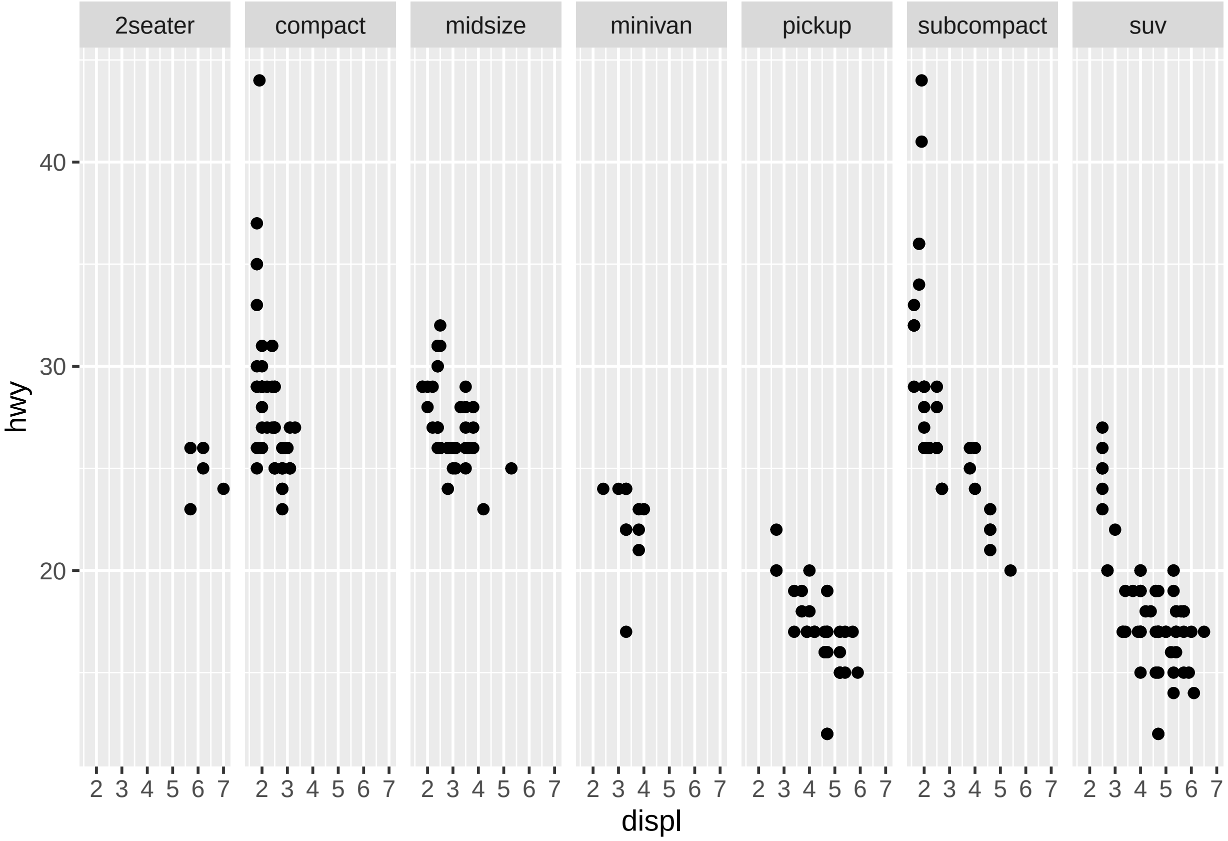
```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```



```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(.~ class)
```



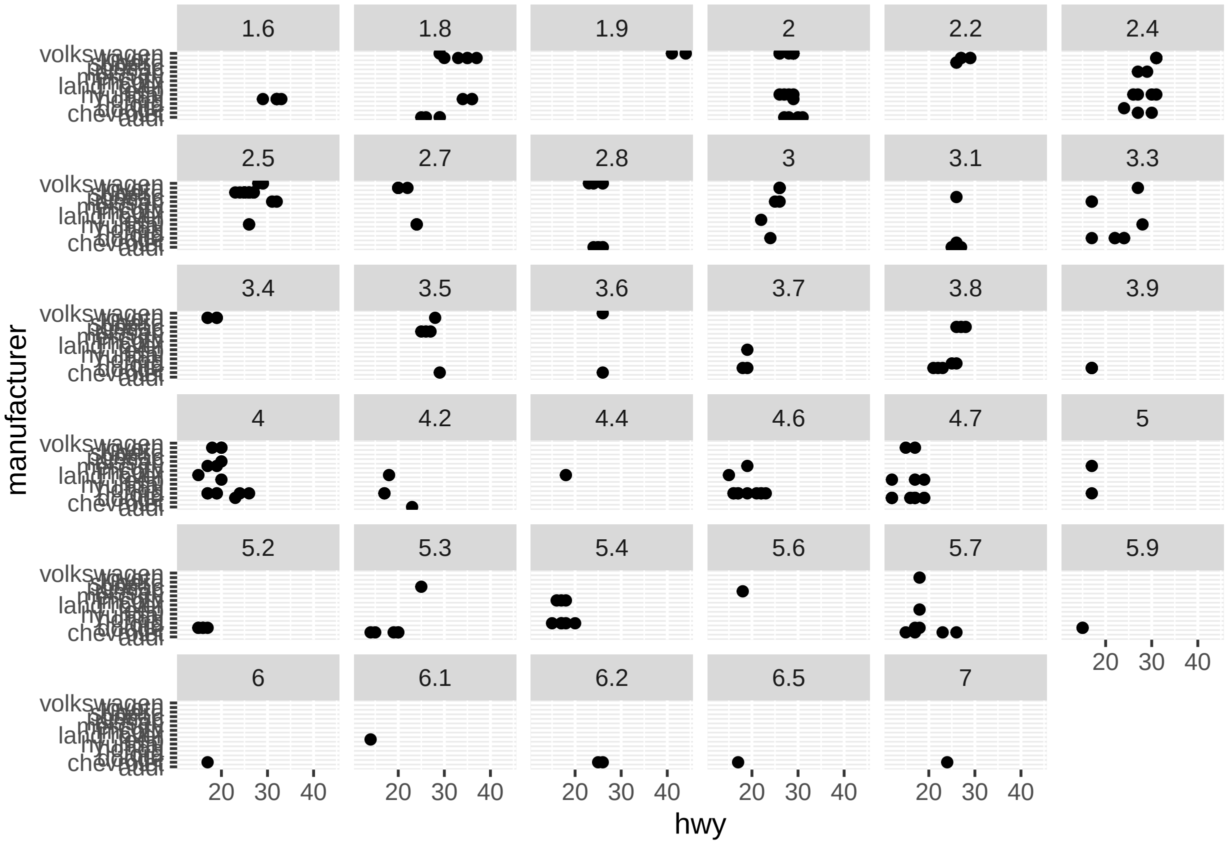
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(~ class)
```



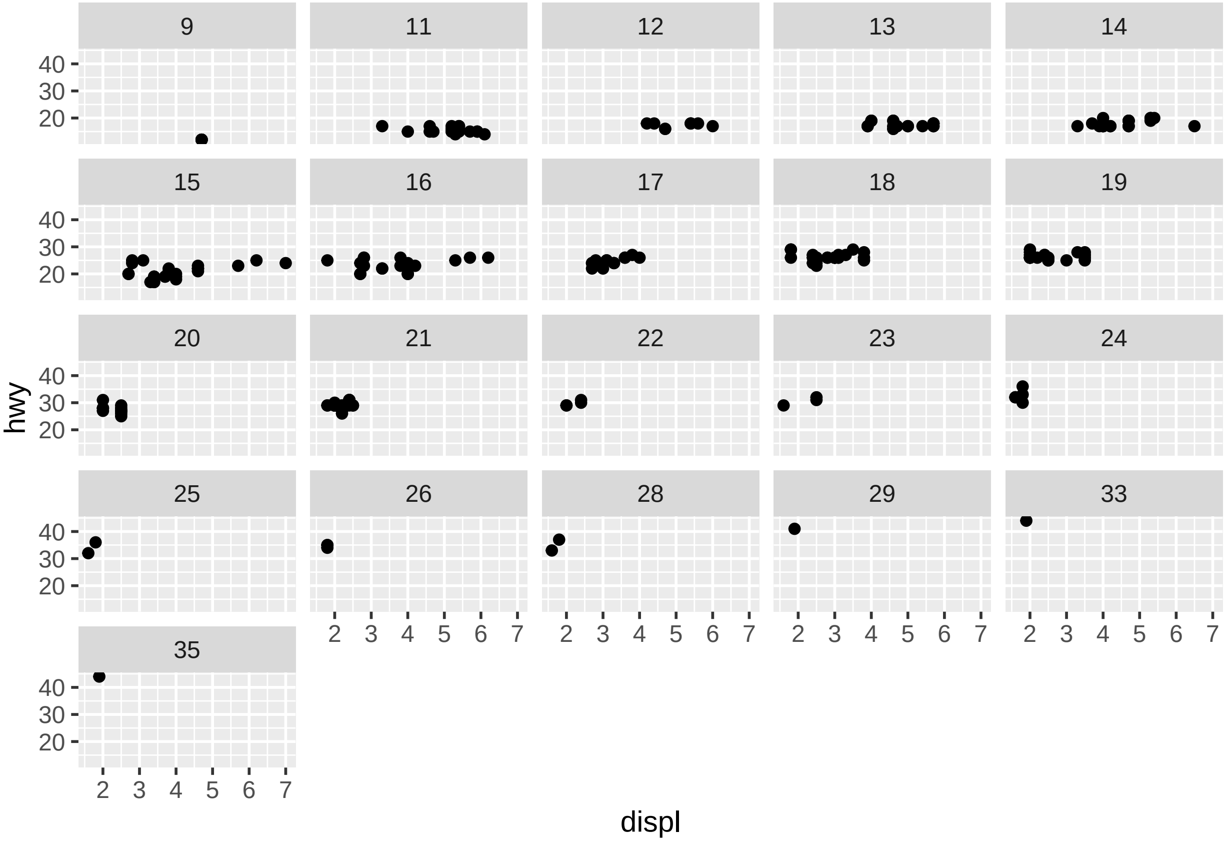
3.5.1 Exercises

- What happens if you facet on a continuous variable?

```
ggplot(data = mpg ) +
        geom_point(mapping = aes(x = hwy, y = manufacturer)) +
        facet_wrap(~ displ)
```



```
ggplot(data = mpg ) +
          geom_point(mapping = aes(x = displ, y = hwy)) +
          facet_wrap(~ cty)
```

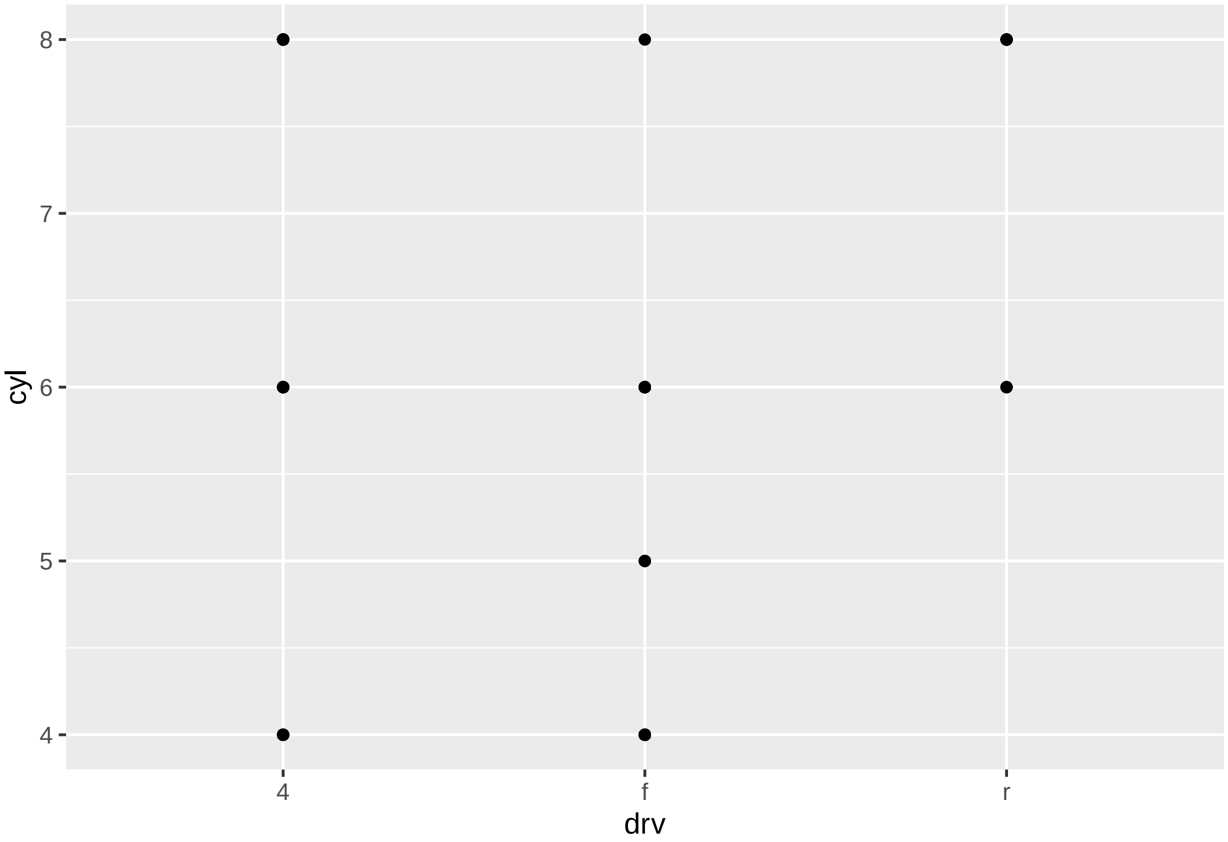


```r
table(mpg$cty)
```

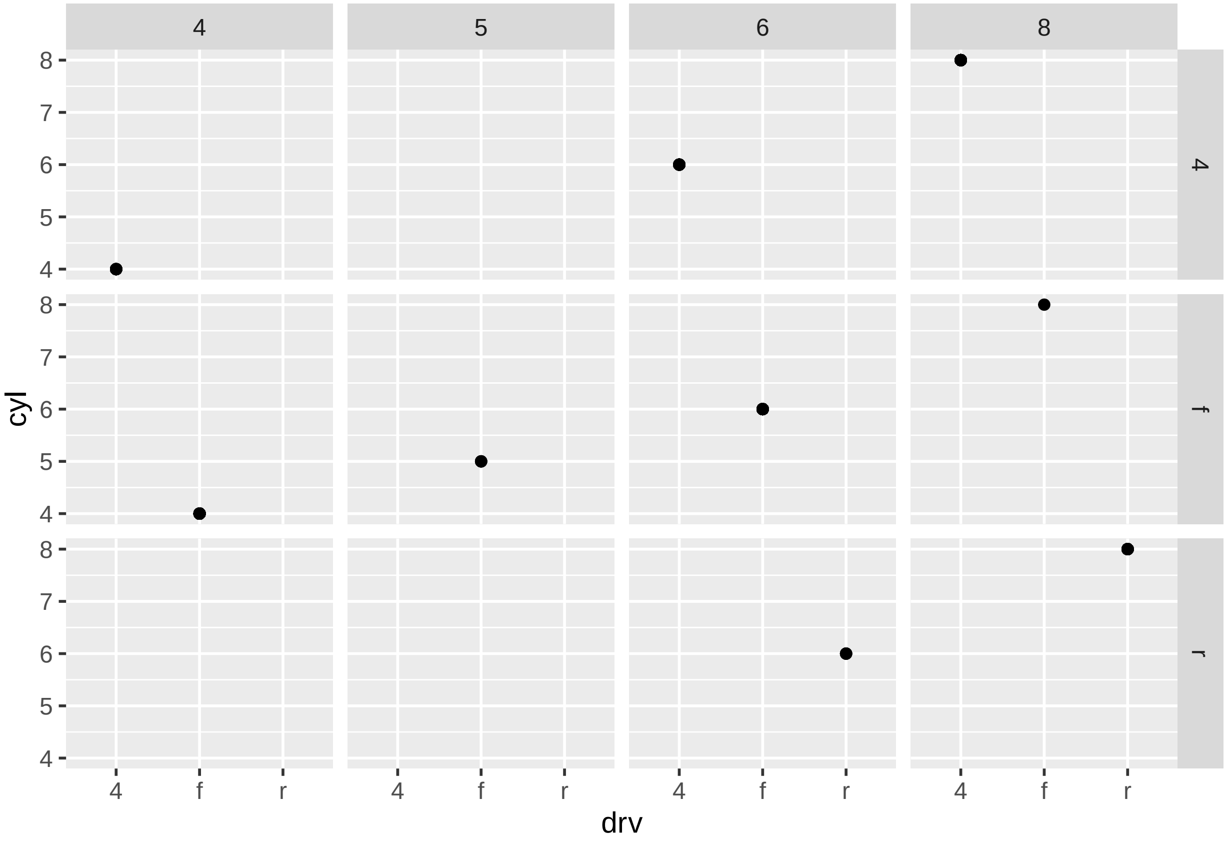
```
##
##   9  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  28  29  33  35
##   5  20   8  21  19  24  19  16  26  20  11  23   4   3   5   2   3   2   1   1   1
```

- What do the empty cells in plot with facet_grid(drv ~ cyl) mean? How do they relate to this plot?

```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = cyl))
```



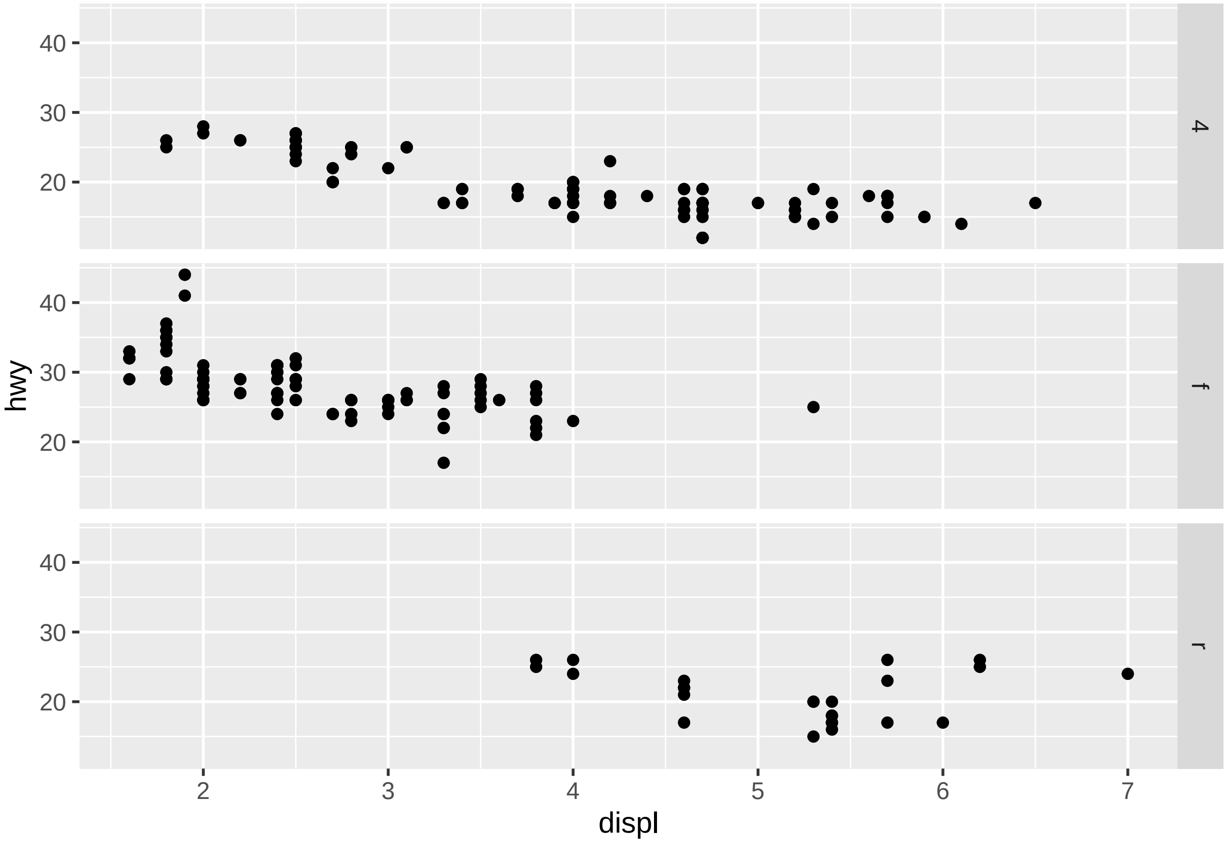
```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = cyl)) +
  facet_grid(drv~cyl)
```



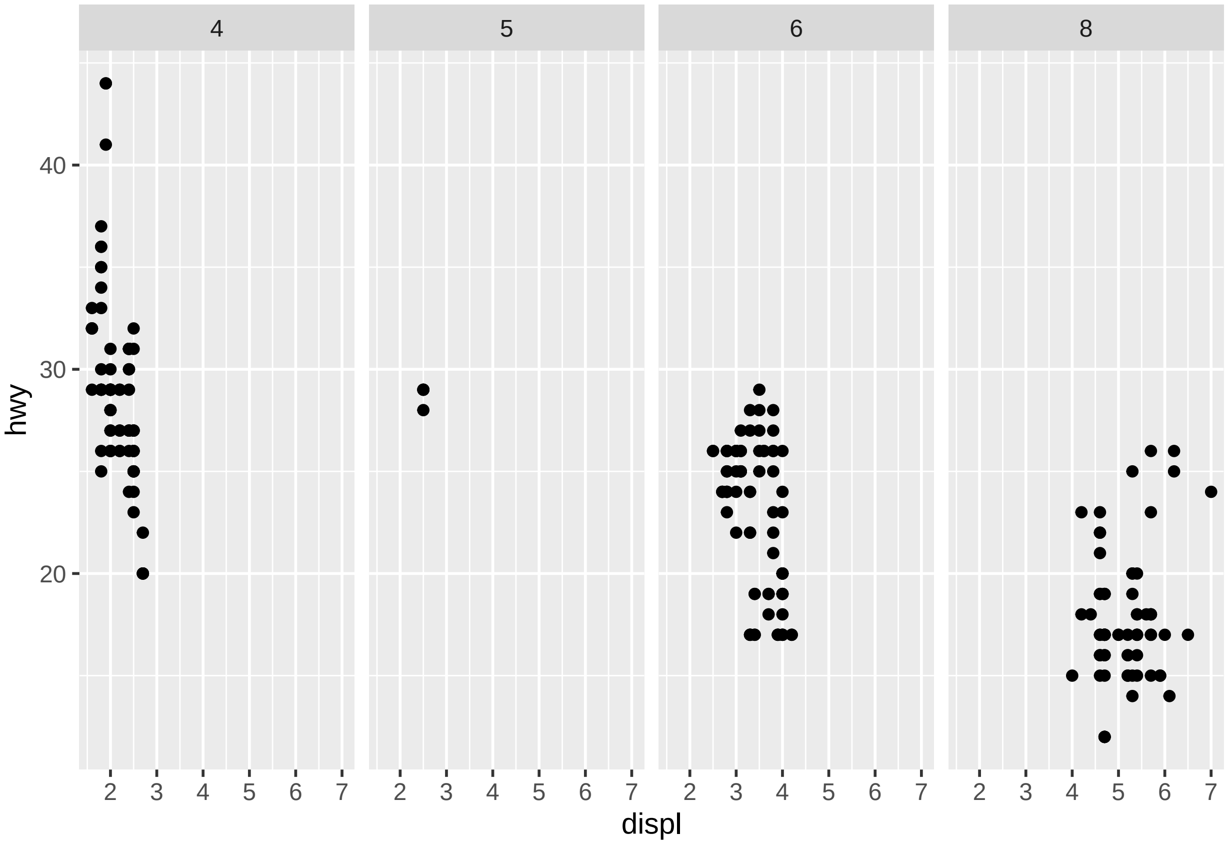
```
### Empty plots mean there is no car to match this configuration. For example, there is no 4WD car that
```

- What plots does the following code make? What does . do?

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ .)
```



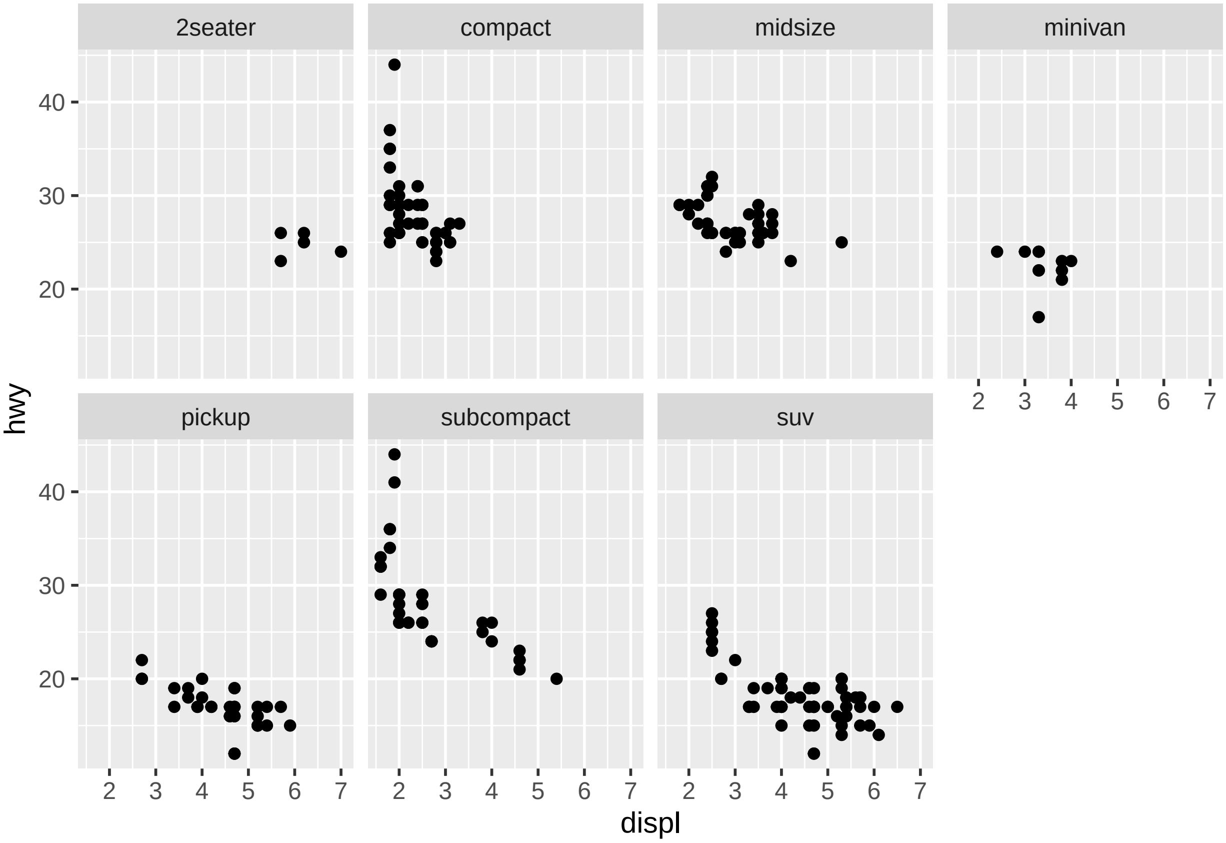
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(. ~ cyl)
```



```
###. (dot) represents how do i want to graph to orient. For example, ___facet_grid(drv ~ .)___ means i
```

- Take the first faceted plot in this section: What are the advantages to using faceting instead of the colour aesthetic? What are the disadvantages? How might the balance change if you had a larger dataset?

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```



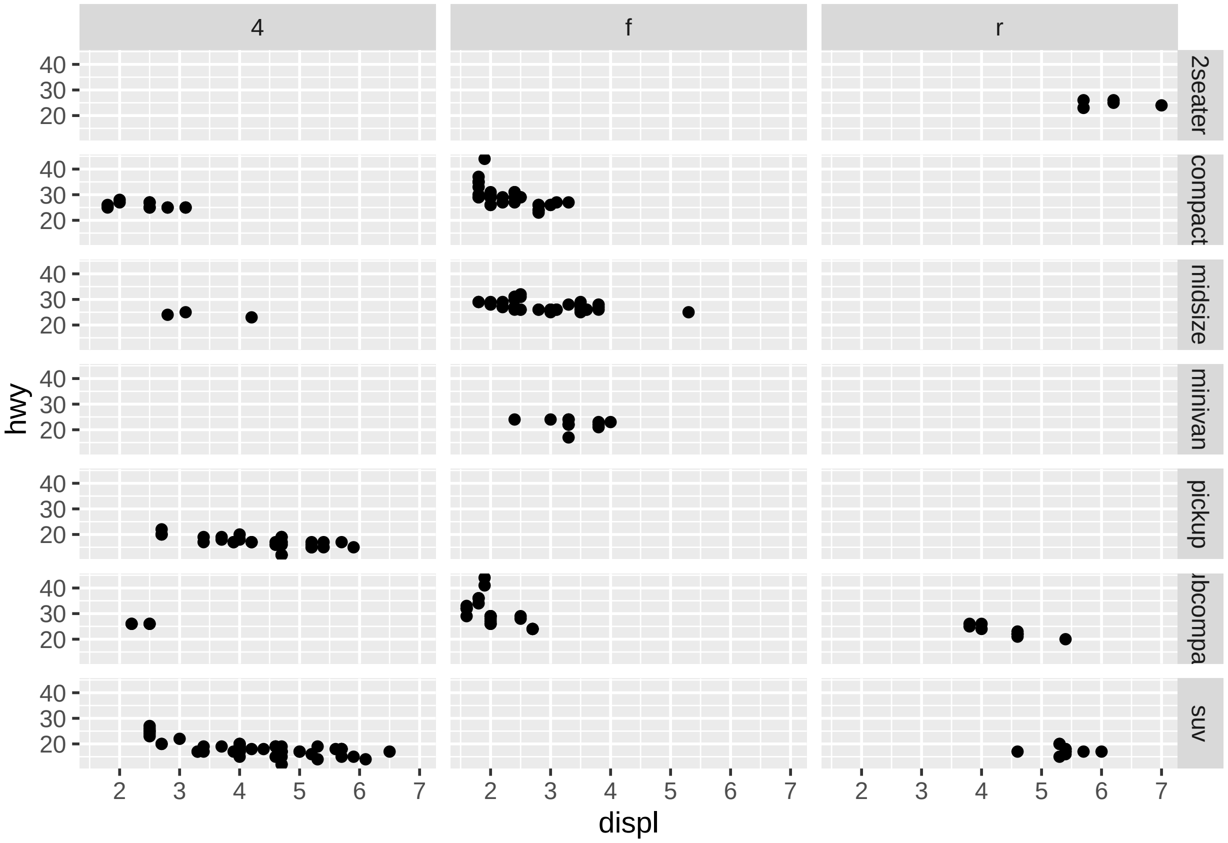
```
###Faceting makes it easy to understand because of descrete graphs compared to colors.
###For larger datasets, faceting may make the graph too cramped.
```

- Read ?facet_wrap. What does nrow do? What does ncol do? What other options control the layout of the individual panels? Why doesn't facet_grid() have nrow and ncol arguments?

```
?facet_wrap
?facet_grid

# facet_wrap wraps a 1d sequence of panels into 2d.
# This is generally a better use of screen space than facet_grid()
# because most displays are roughly rectangular.

# facet_grid() forms a matrix of panels defined by row and column faceting variables.
# It is most useful when you have two discrete variables, and all combinations of the
# variables exist in the data.

### nrow and ncol represents number of rows and columns respectively.
### scales, shrink, labeller, switch are other options to control the layout of individual panels.
### ___facet_grid()___ forms a matrix, that's why it doesn't have nrow and ncol arguments ??
```

- When using facet_grid() you should usually put the variable with more unique levels in the columns. Why?

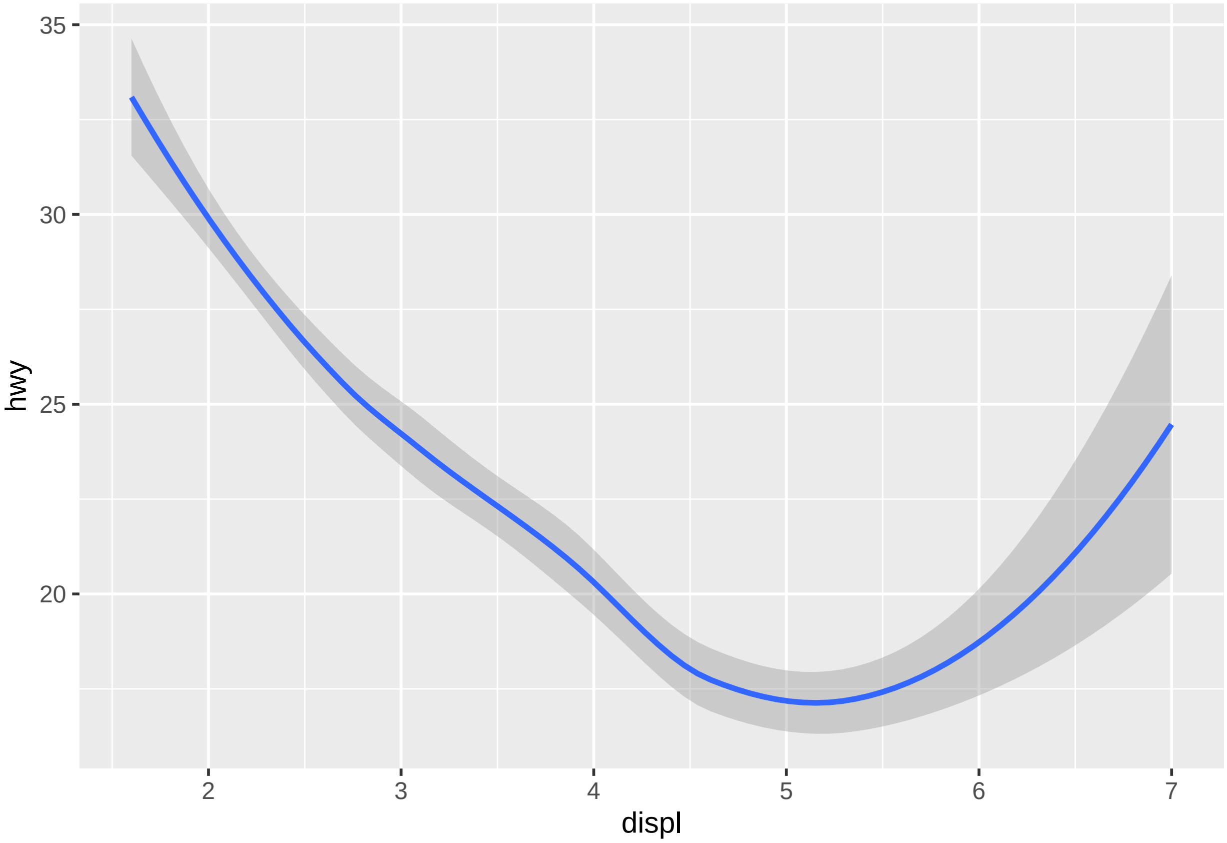
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(class ~ drv)
```



```
### Putting variables with many levels in the rows shrinks the space like the above graph.
### That's why it recommended to put the variable with more unique levels in the columns.
```

```
ggplot(data =  mpg) +
          geom_smooth(mapping = aes(x = displ , y = hwy))
```

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


35
30
25
20
hwy
2
3
4
5
6
7
displ