Solutions to G. Grolemund & H. Wickhams's R for Data Science

Krista DeStasio

7/11/2017

Contents

Note: *This is a work in progress.*

A Brief Introduction to This File

This R file walks through G. Grolemund & H. Wickhams's online text, "R for Data Science." Much of the code is sourced directly from the book and credit belongs to the authors. Here, some sections of code are heavily commented so that the beginning R programmer can read through and understand what each line of code does and compare it to their own as they work through the text. Throughout, the book provides the primary and most thorough explanation. **For the greatest learning benefit, I suggest you attempt each exercise on your own before looking at the code or write-ups provided here.** Of course, there is more than one way to write code and you may find a more elegant solution that you prefer.

For those new to R and RStudio, it may be of additional benefit to knit the document and examine how the code in the Rmd file is visually expressed in the resultant knitted document. For example, see how the `["R for Data Science."](http://r4ds.had.co.nz/index.html)` is expressed as a hyperlink in the preceeding paragraph where it was not surrounded by tick-marks and compare that to how the same text is expressed in this paragraph when surrounded by ticks. See also the difference in appearance when knitting to different document types (HTML, PDF, Word).

Tip: *If you are using RStudio, click the text next to the orange # box at the bottom of the editor window to easily navigate the code chunks.*

Tip: *Use the `?` before any command to view the documentation on that function. Do this often. For example, type `?setwd` to see a description, usage, arguments, and more for the function `setwd()`.*

Chapter 3, Data Visualisation

To really understand ggplot2, I highly recommend reading "The Layered Grammar of Graphics" as suggested at the beginning of Chapter 3.

The mpg data frame

```
str(mpg) # Look at the structure of the mpg data frame
```

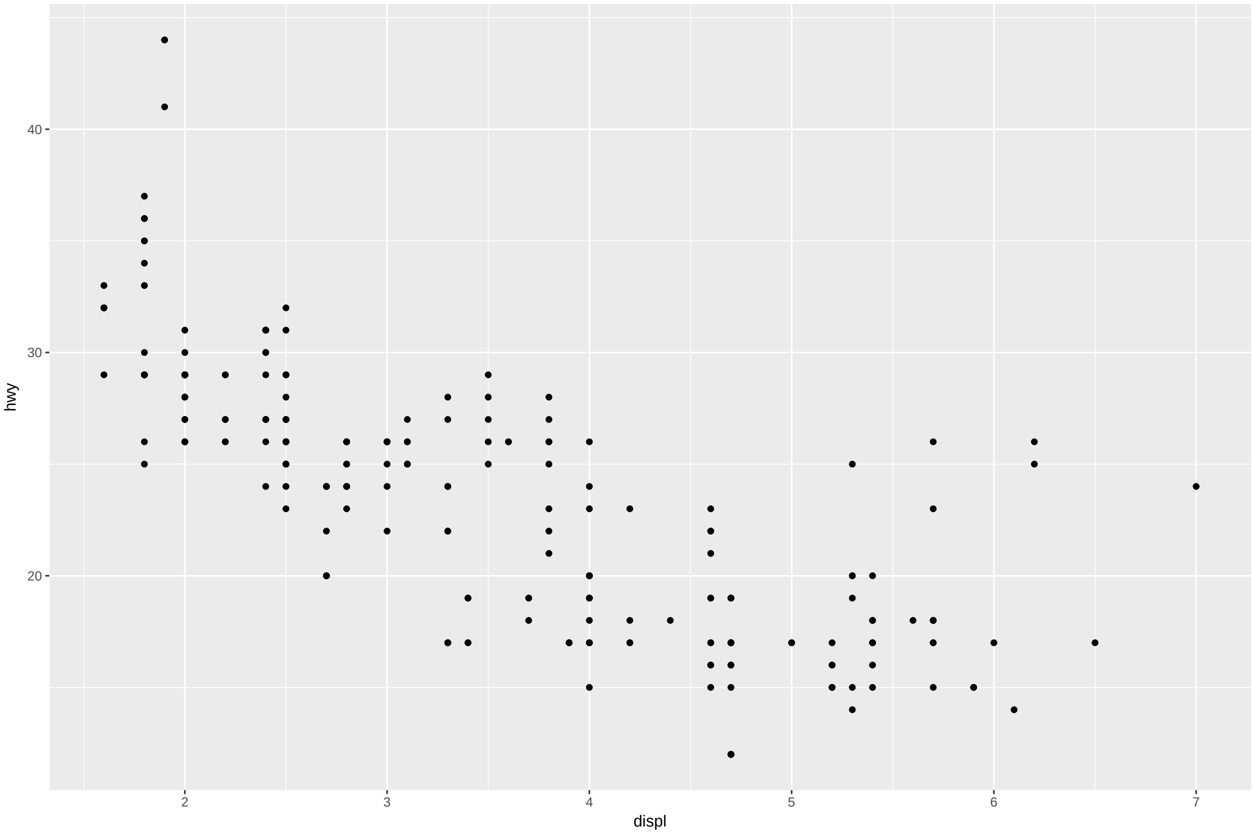
```
## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11 variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

```
mpg # Look at the first 10 rows of the mpg data frame
```

```
## # A tibble: 234 x 11
##    manufacturer      model displ  year   cyl      trans   drv   cty   hwy
##           <chr>      <chr> <dbl> <int> <int>      <chr> <chr> <int> <int>
##  1         audi         a4   1.8  1999     4   auto(l5)     f    18    29
##  2         audi         a4   1.8  1999     4 manual(m5)     f    21    29
##  3         audi         a4   2.0  2008     4 manual(m6)     f    20    31
##  4         audi         a4   2.0  2008     4   auto(av)     f    21    30
##  5         audi         a4   2.8  1999     6   auto(l5)     f    16    26
##  6         audi         a4   2.8  1999     6 manual(m5)     f    18    26
##  7         audi         a4   3.1  2008     6   auto(av)     f    18    27
##  8         audi a4 quattro   1.8  1999     4 manual(m5)     4    18    26
##  9         audi a4 quattro   1.8  1999     4   auto(l5)     4    16    25
## 10         audi a4 quattro   2.0  2008     4 manual(m6)     4    20    28
## # ... with 224 more rows, and 2 more variables: fl <chr>, class <chr>
```

Hypothesis: There is a negative linear relationship between engine size and fuel efficiency, such that as engine size increases fuel efficiency decreases.

```
ggplot(data=mpg) + # specify data frame
    geom_point(mapping = aes(x = displ, y = hwy)) # specify that plot is a scatterplot with displ on th
```



The plot confirms the hypothesis that there is a negative relationship between engine size and fuel efficiency.

Making graphs with ggplot2

Template:

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Exercises 3.2.4

1. There are no visible results from the code below.

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

2. Based on the output from `str(mpg)`, we see that there are 234 rows and 11 columns in the mpg data frame.

```
# Alternative means of finding number of rows and columns
nrow(mpg) # Pring the number of rows
```

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

```
ncol(mpg)
```

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

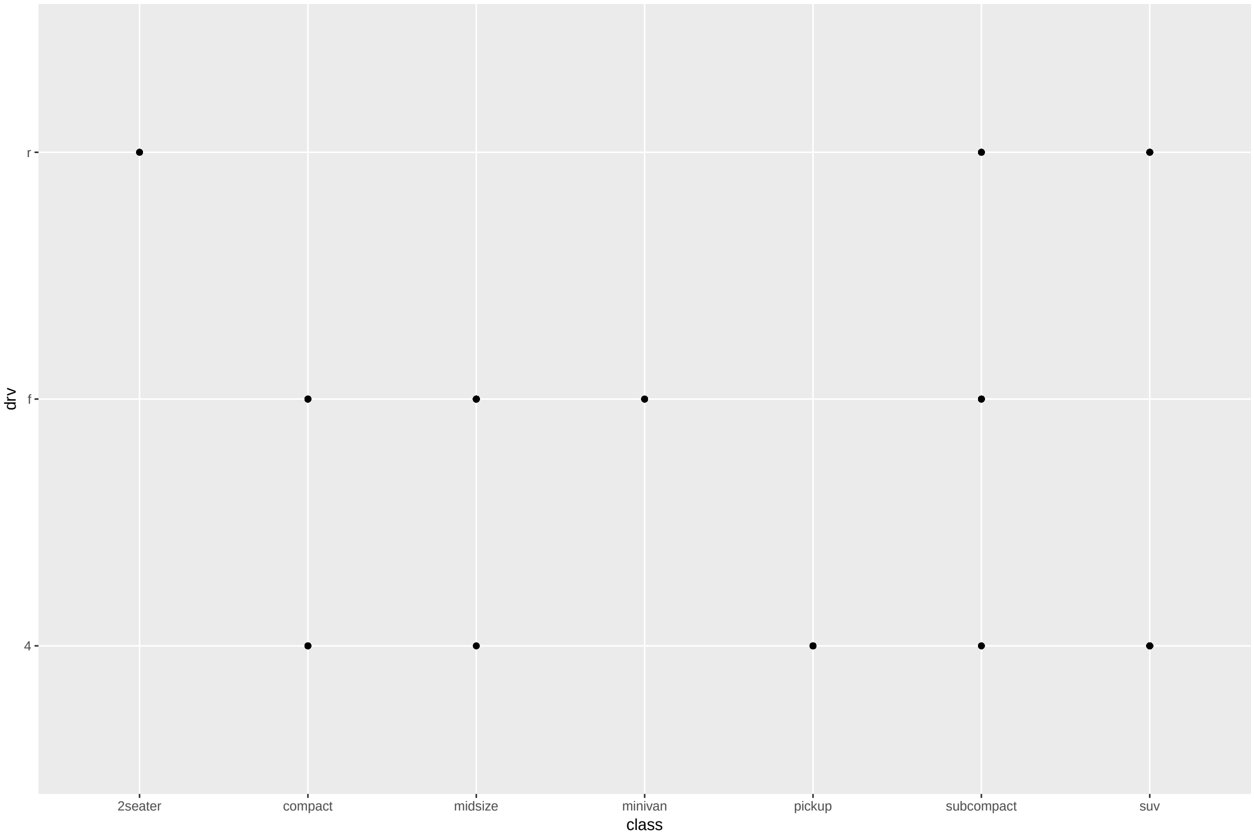
There are 234 rows and 11 columns in the mpg data frame.

3. The `drv` variable describes whether the vehicle is front, rear, or 4-wheel drive.

```
?mpg
```

4. The plot is not useful because the variables are categorical and multiple points are plotted atop one another. We are unable to determine from this plot how many observations there are of each class-drive combination.

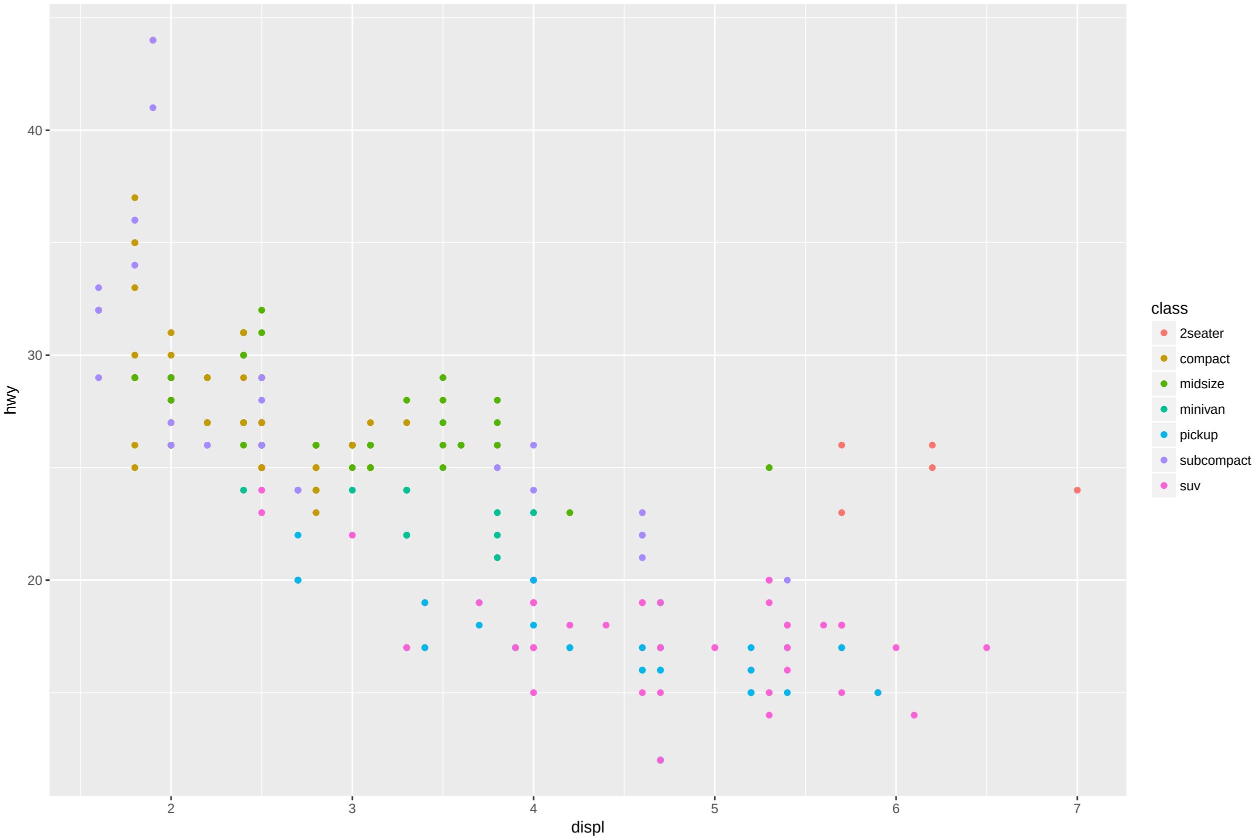
```
ggplot(data=mpg) +
    geom_point(mapping = aes(x=class, y=drv))
```



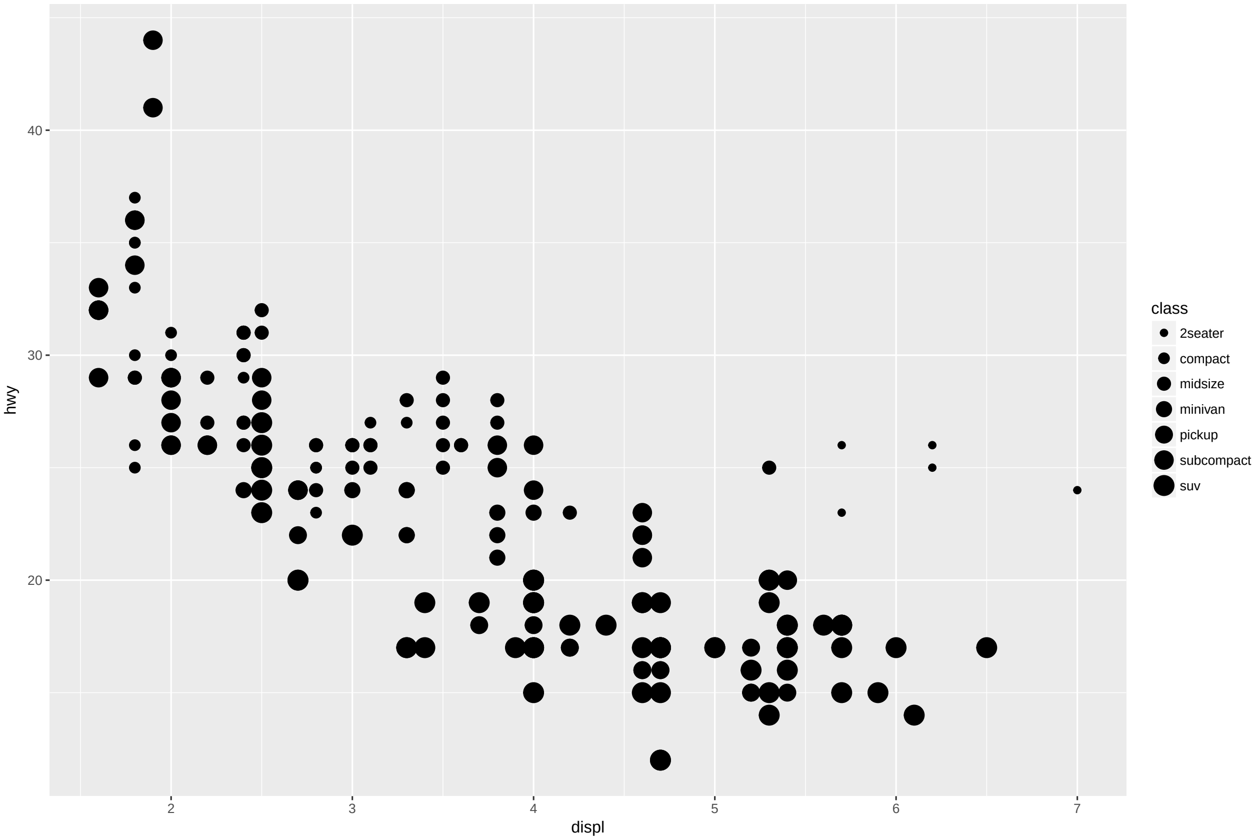
Aesthetic mappings

Test the hypothesis that the cars highlighted in red are hybrids by mapping car class to an aesthetic.

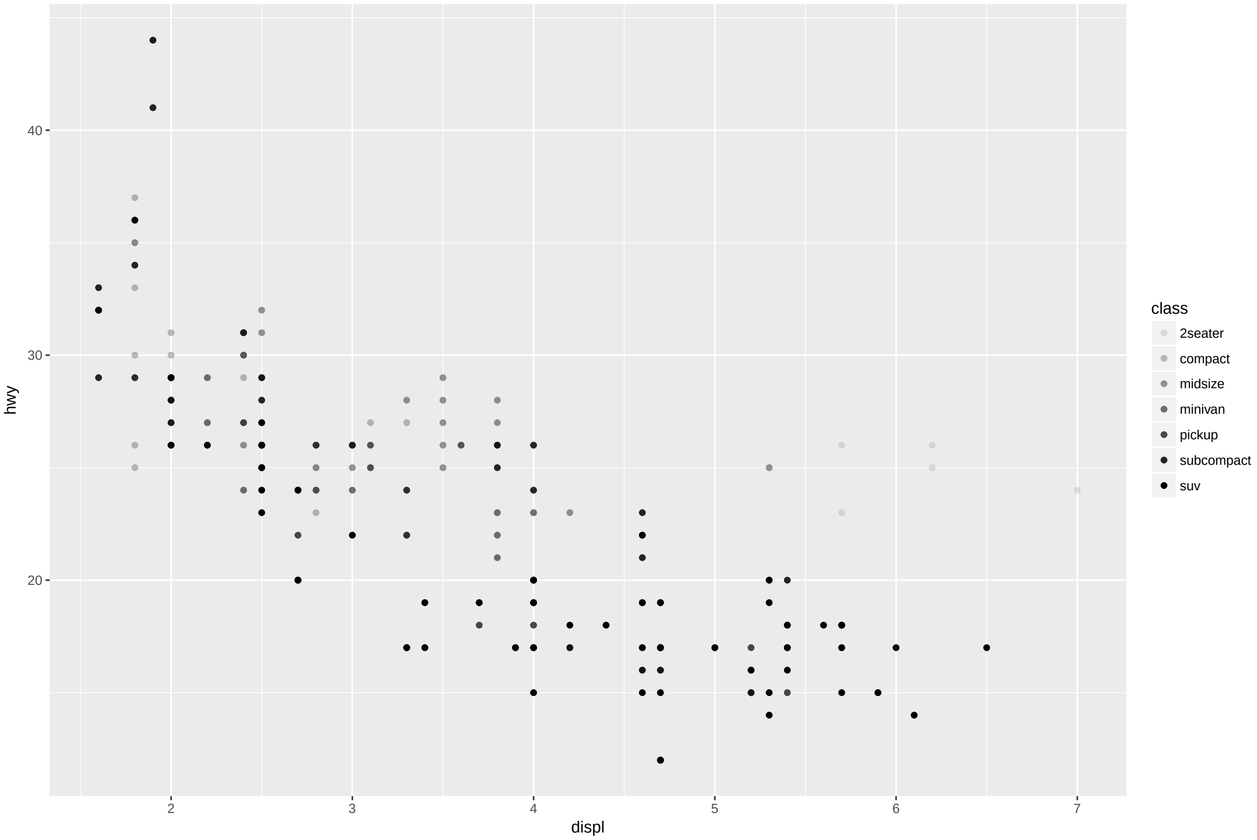
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class)) # map class to the color aesthetic so th
```



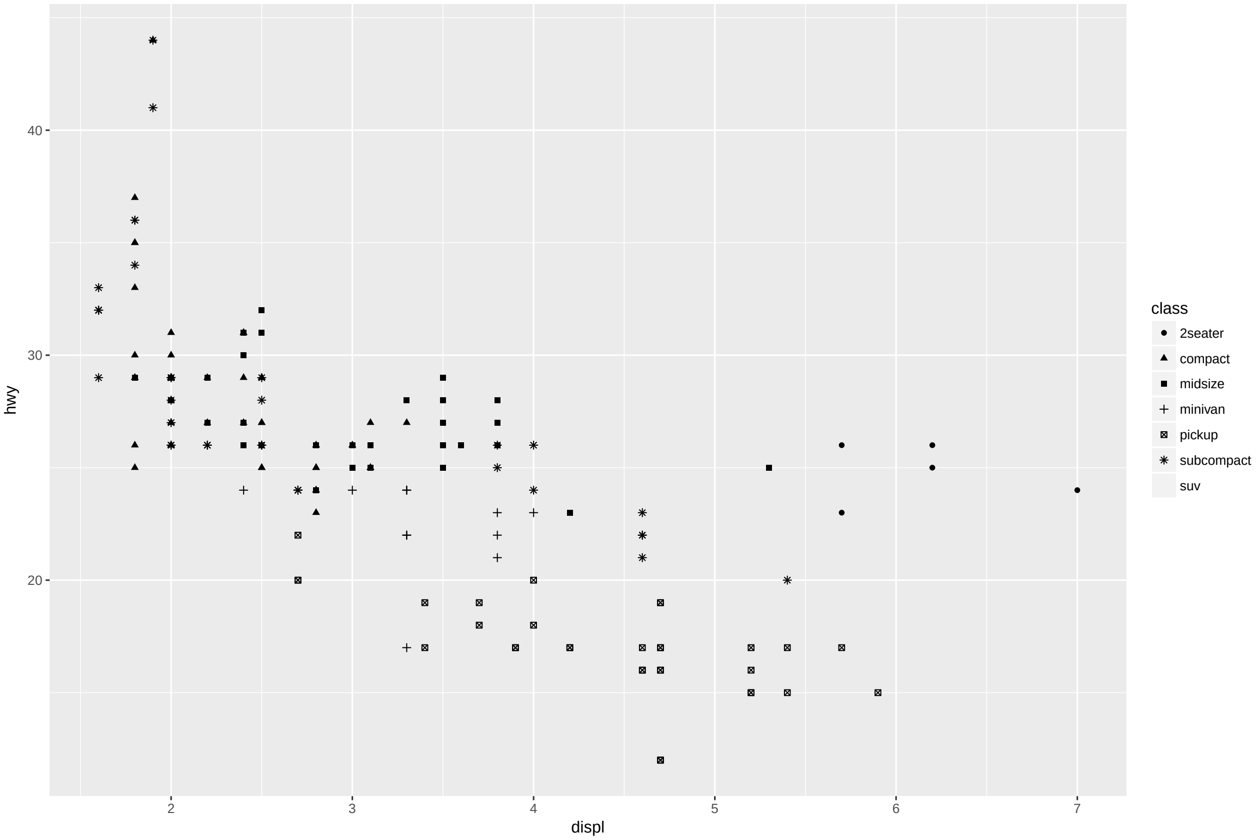
```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = class))
```



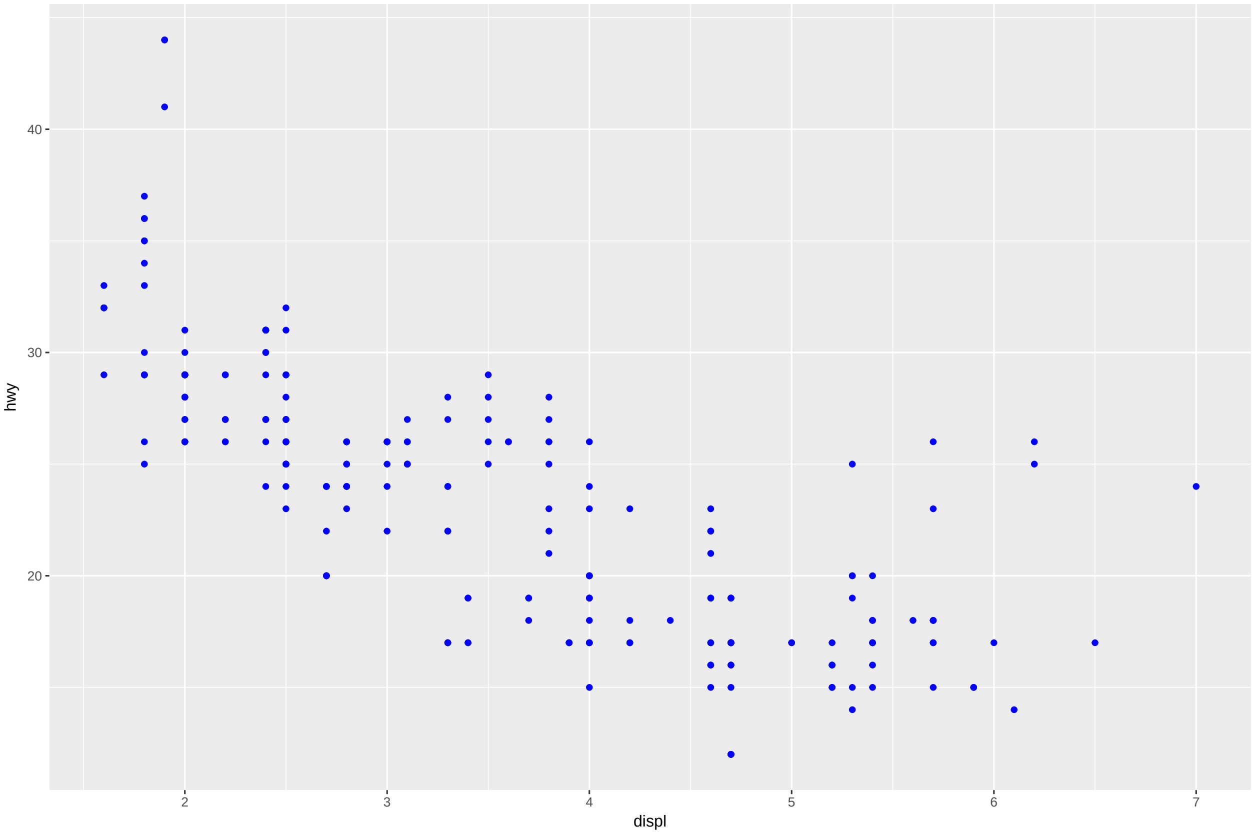
```
# Left
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
```



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



```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue") # Set the aesthetic outside of aes() to
```



Aesthetic shapes:

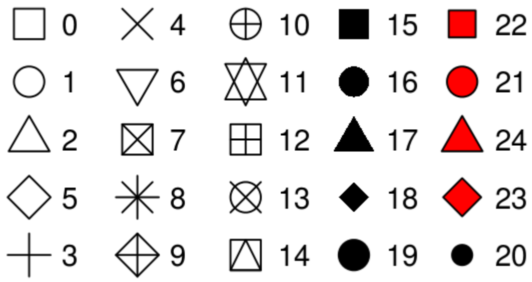
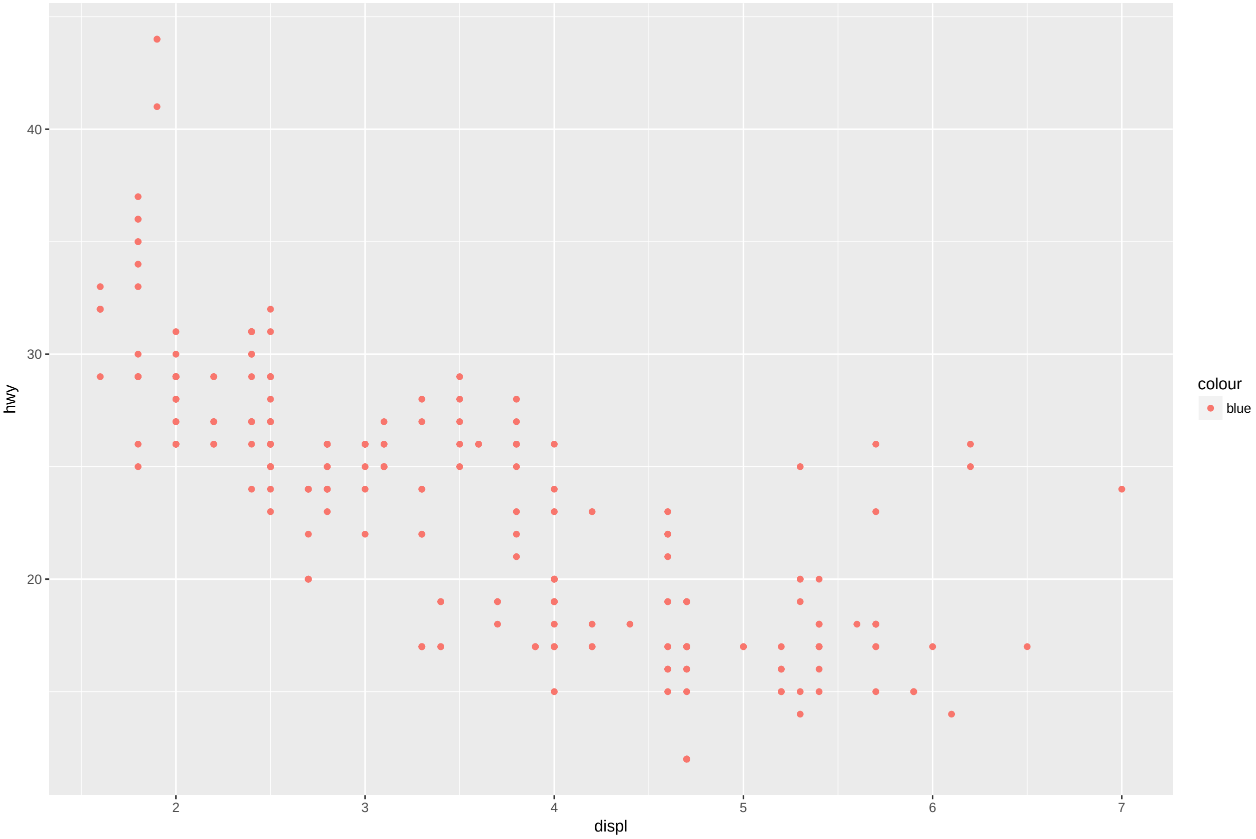


Figure 1:

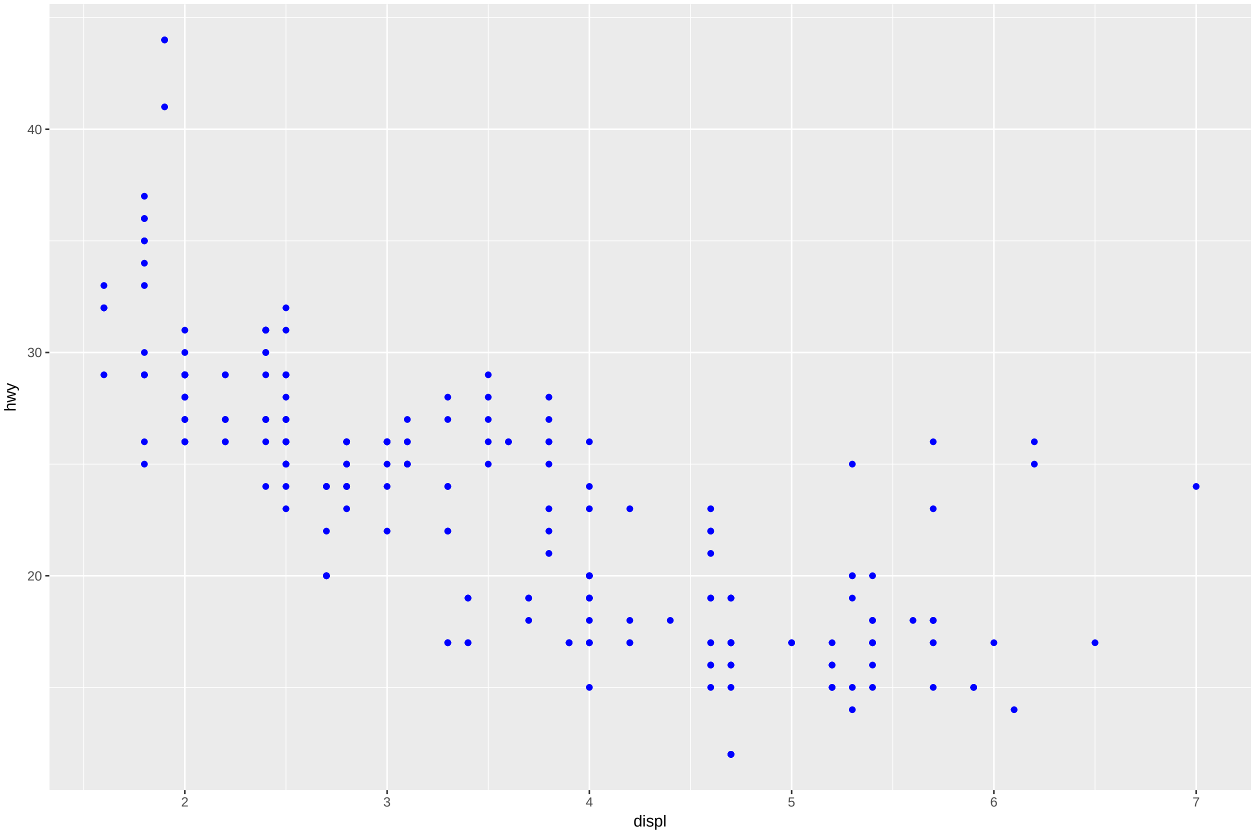
Exercises 3.3.1

1. The points are not blue because the color aesthetic is set inside `aes()`.

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



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



2. To determine what the categorical and continuous variables are, one can either view the tibble by typing `mpg` or by viewing the documentation `?mpg`. One may decide whether a variable is categorical or continuous by checking whether it is stored as a character, integer, or double (floating point integer) value. However, this can lead to miscategorization in some cases. For example, while year is an integer, it is typically considered a whole number, a discrete variable without a meaningful 0 value anchor, and therefore not continuous.

The categorical variables are:

- model
- year (discrete, rather than categorical)
- trans
- drv
- fl
- class

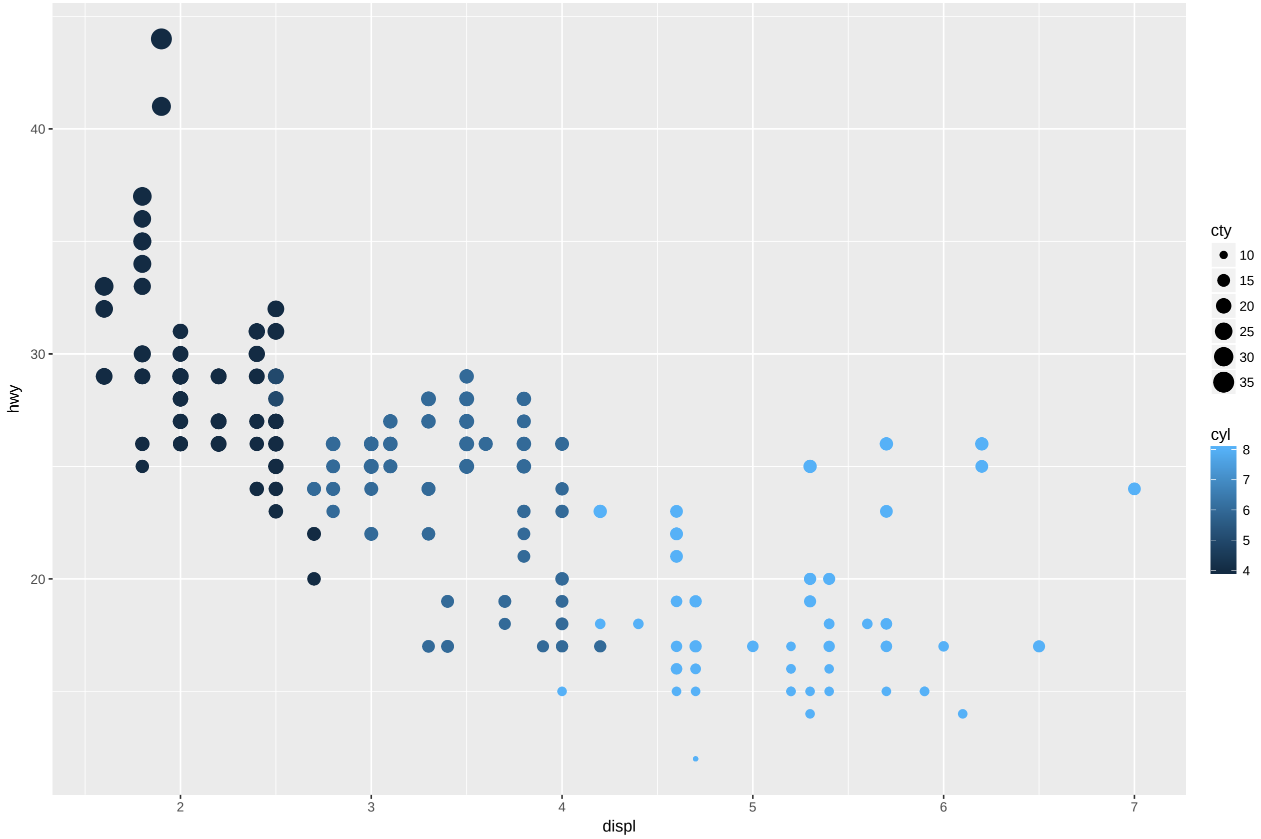
The continuous variables are:

- displ
- cyl
- cty
- hwy
- year (in this data set, year is treated as an integer variable. Better to consider this “quantitative”, rather than “continuous”)

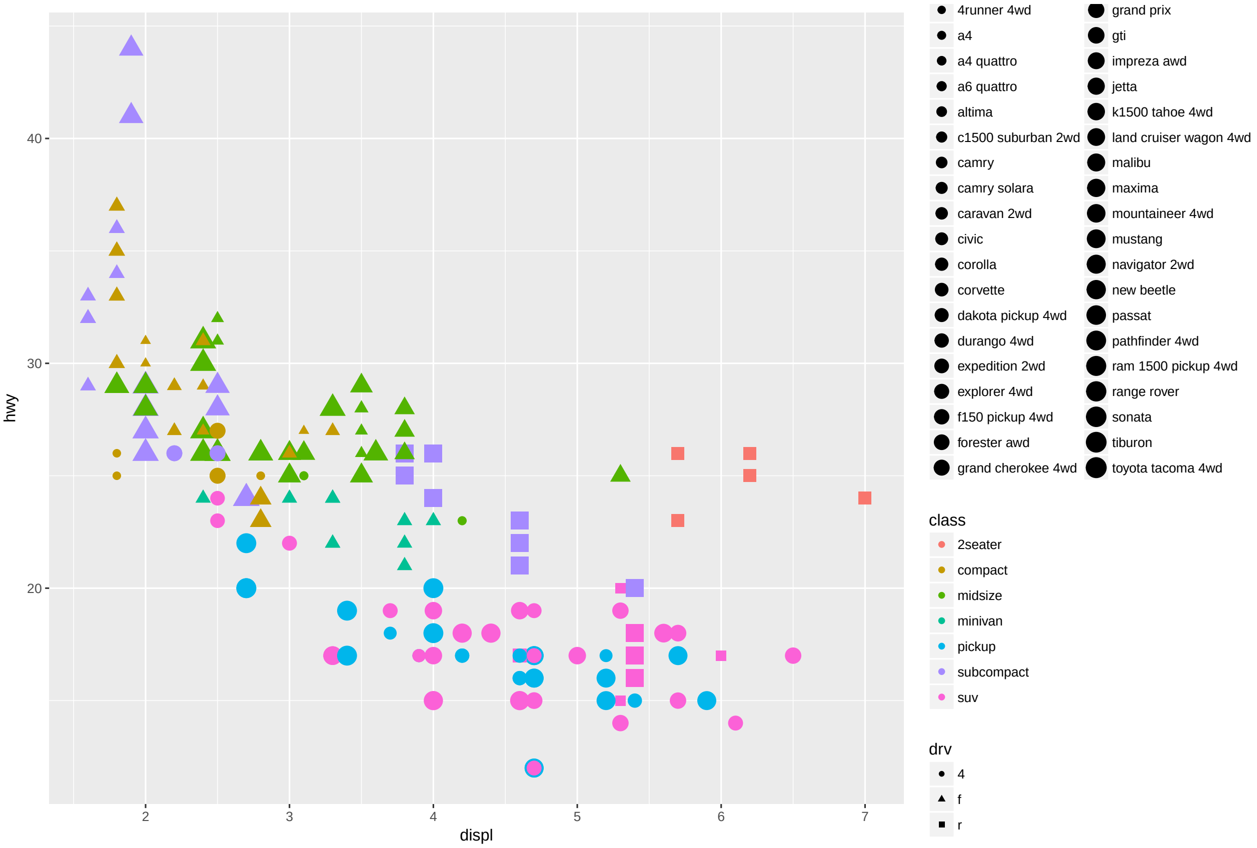
3. A continuous variable cannot be mapped to shape. When mapped to size or color, the continuous variable is binned by equal intervals (in this case, intervals of 5 mpg). When mapped to the size aesthetic, points scale by the intervals. Continuous variables when mapped to a color aesthetic are mapped along a gradient scale.

```
# Mapping a continuous variable to the shape aesthetic
ggplot(data=mpg) +
    geom_point(mapping = aes(x = displ, y = hwy, shape = cty))
```

```
# Mapping continuous variables to the color and size aesthetics
ggplot(data=mpg) +
    geom_point(mapping = aes(x = displ, y = hwy, color = cyl, size = cty))
```

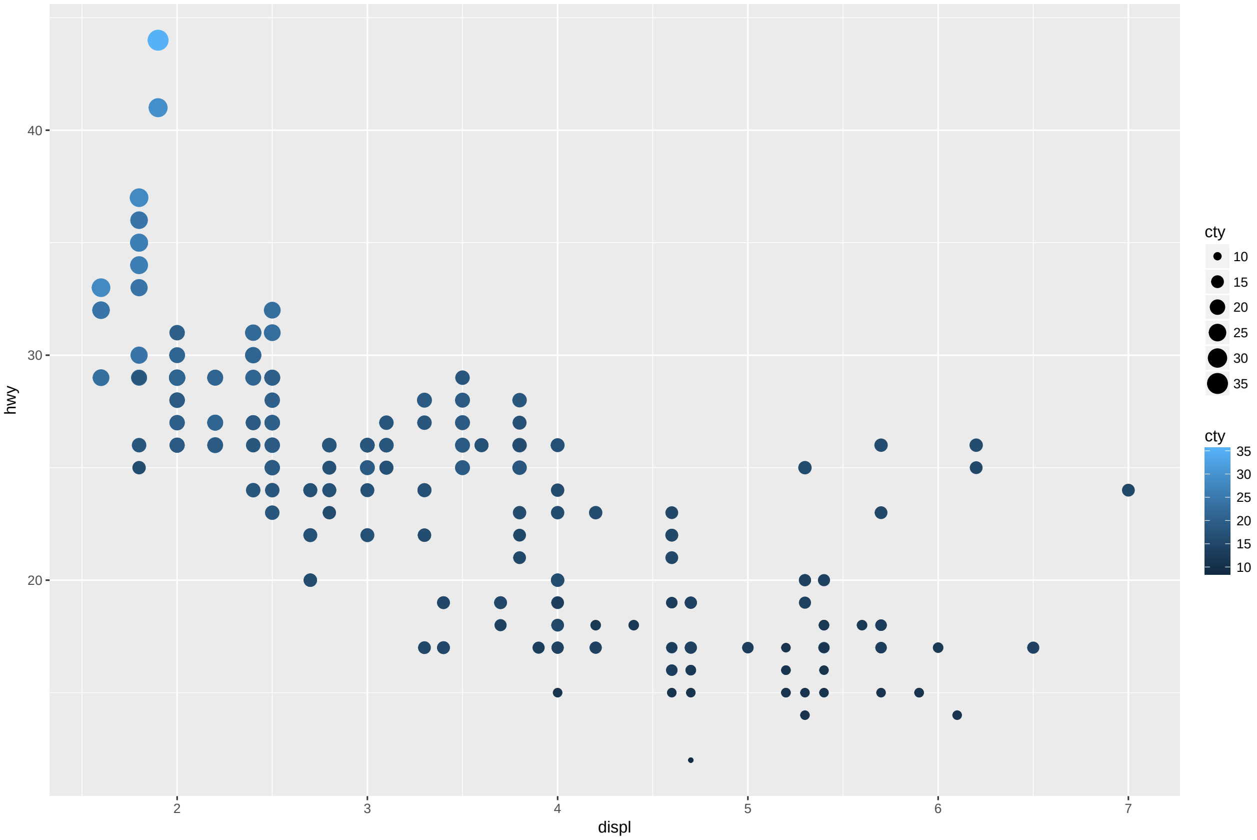


```
# Mapping categorical variables to size, color, and shape
ggplot(data=mpg) +
    geom_point(mapping = aes(x = displ, y = hwy, size = model, color = class, shape = drv))
```



4. When the same variable is mapped to multiple aesthetics, it is represented by those aesthetics.

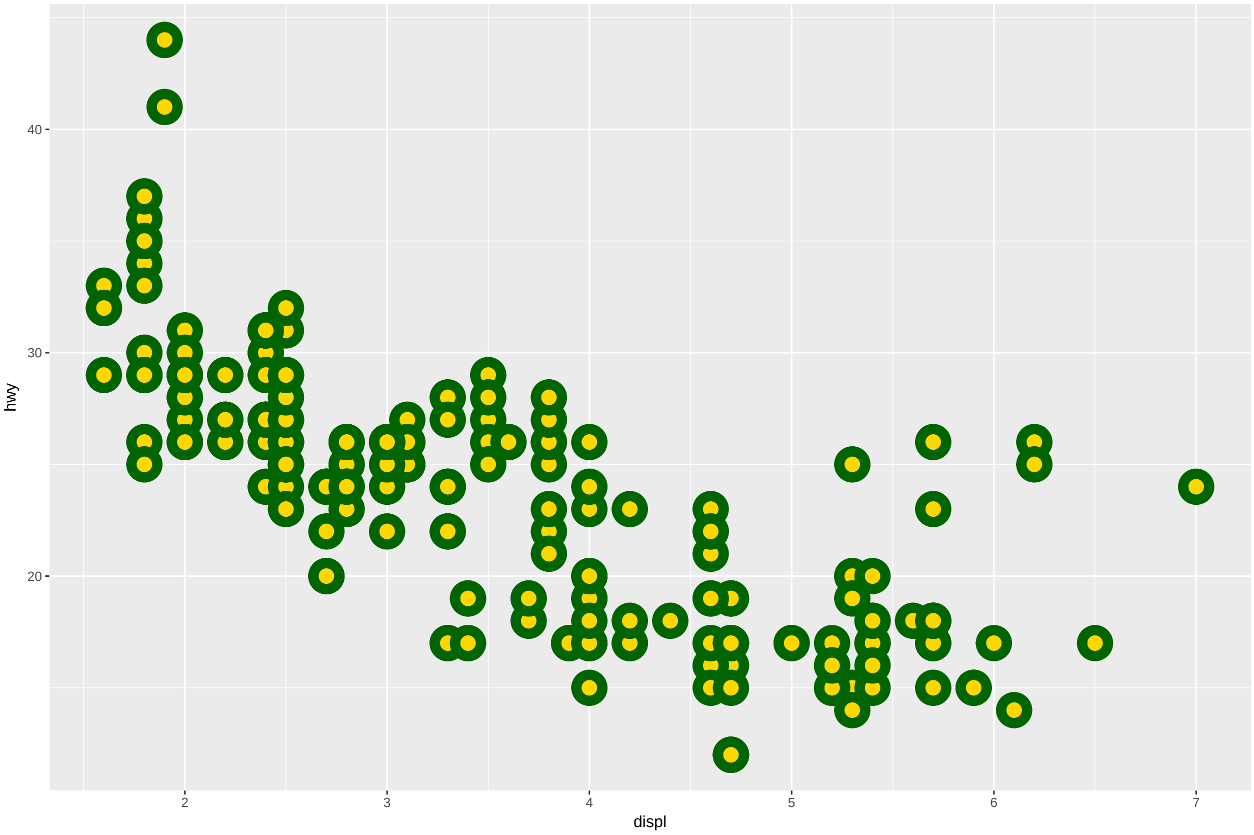
```
# Mapping the same variable to multiple aesthetics
ggplot(data=mpg) +
    geom_point(mapping = aes(x = displ, y = hwy, color = cty, size = cty)) # Here, city is mapped to th
```



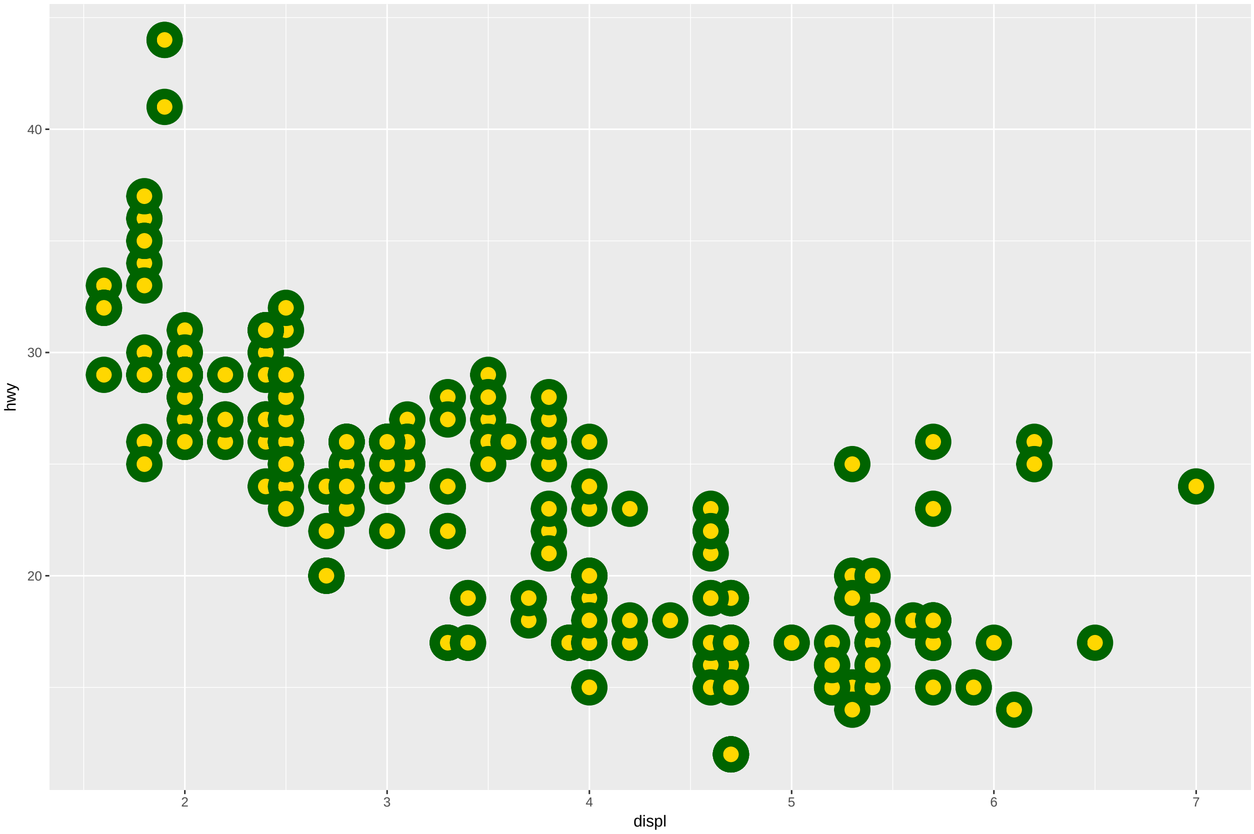
5. According to the R documentation:
   "For shapes that have a border (like 21), you can colour the inside and outside separately. Use the **stroke** aesthetic to modify the width of the border."

Tip: You can find documentation of available colors here.

```
?geom_point

# Example using `stroke`
ggplot(data=mpg)+
    geom_point(mapping = aes(x=displ, y=hwy), shape = 21, colour = "darkgreen", fill = "gold", size = 5
```

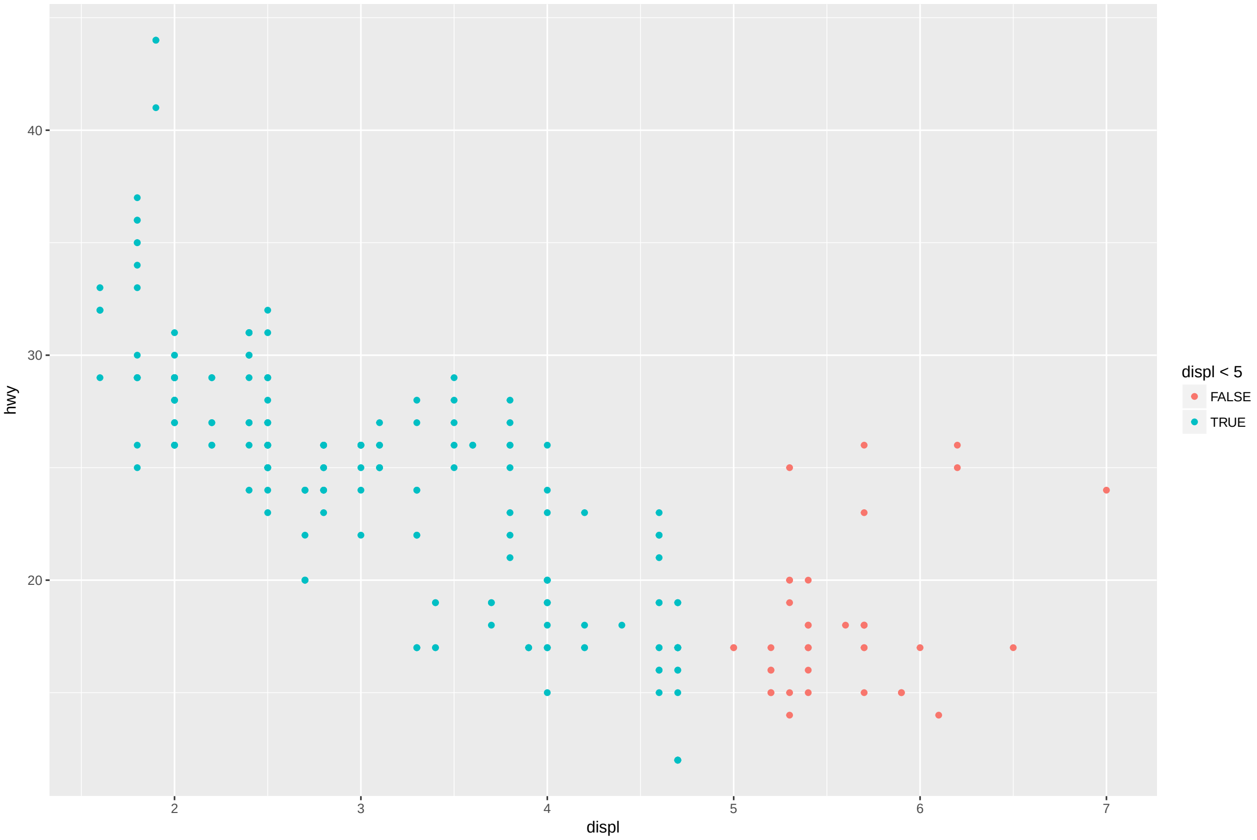


```
# Just for fun, let's write short-hand code make the same plot
ggplot(mpg, aes(displ, hwy)) +
  geom_point(shape = 21, colour = "darkgreen", fill = "gold", size = 5, stroke = 5)
```



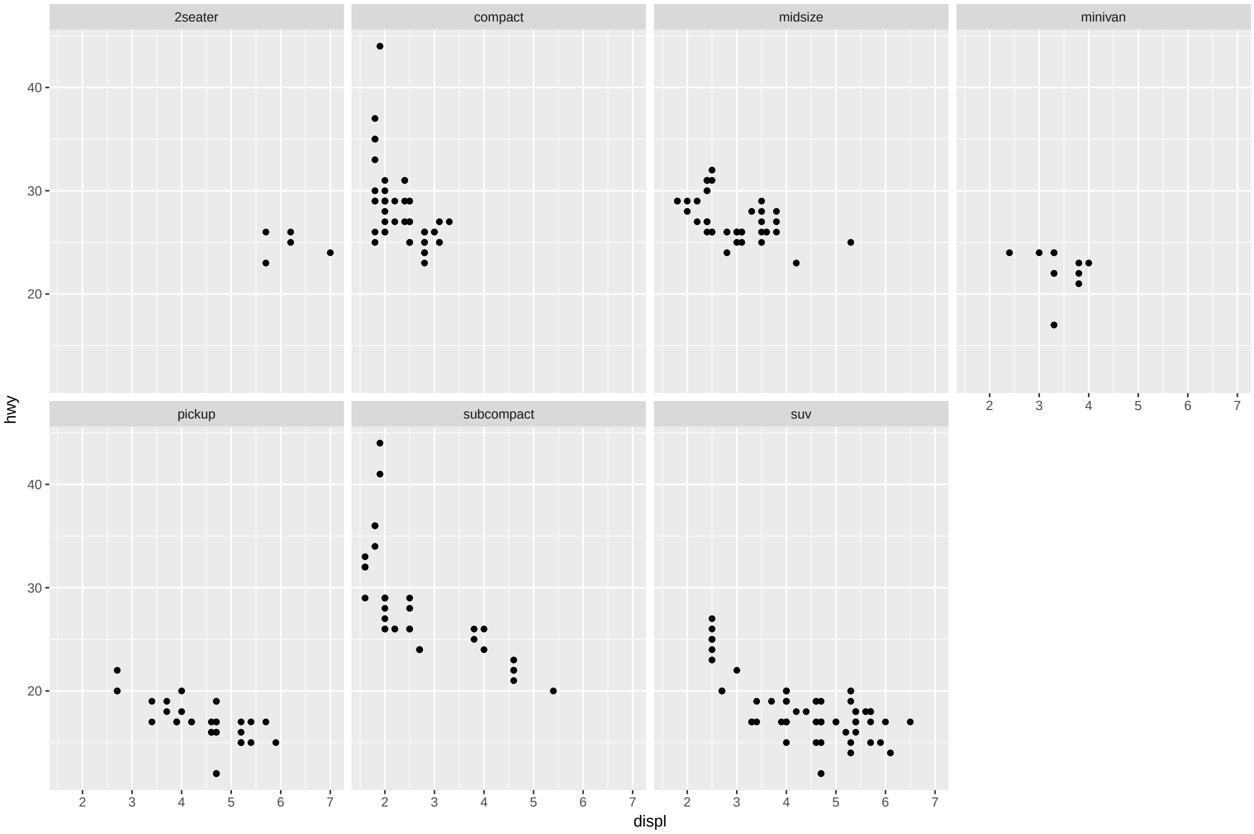
6. Setting the color aesthetic to `displ < 5` will assign one color to all x-axis (hwy) values < 5 and a different color to x-axis values ≥ 5. Since the color palette is not specified, default colors are used.

```
ggplot(data=mpg) +
    geom_point(mapping = aes(x = displ, y = hwy, color = displ < 5))
```

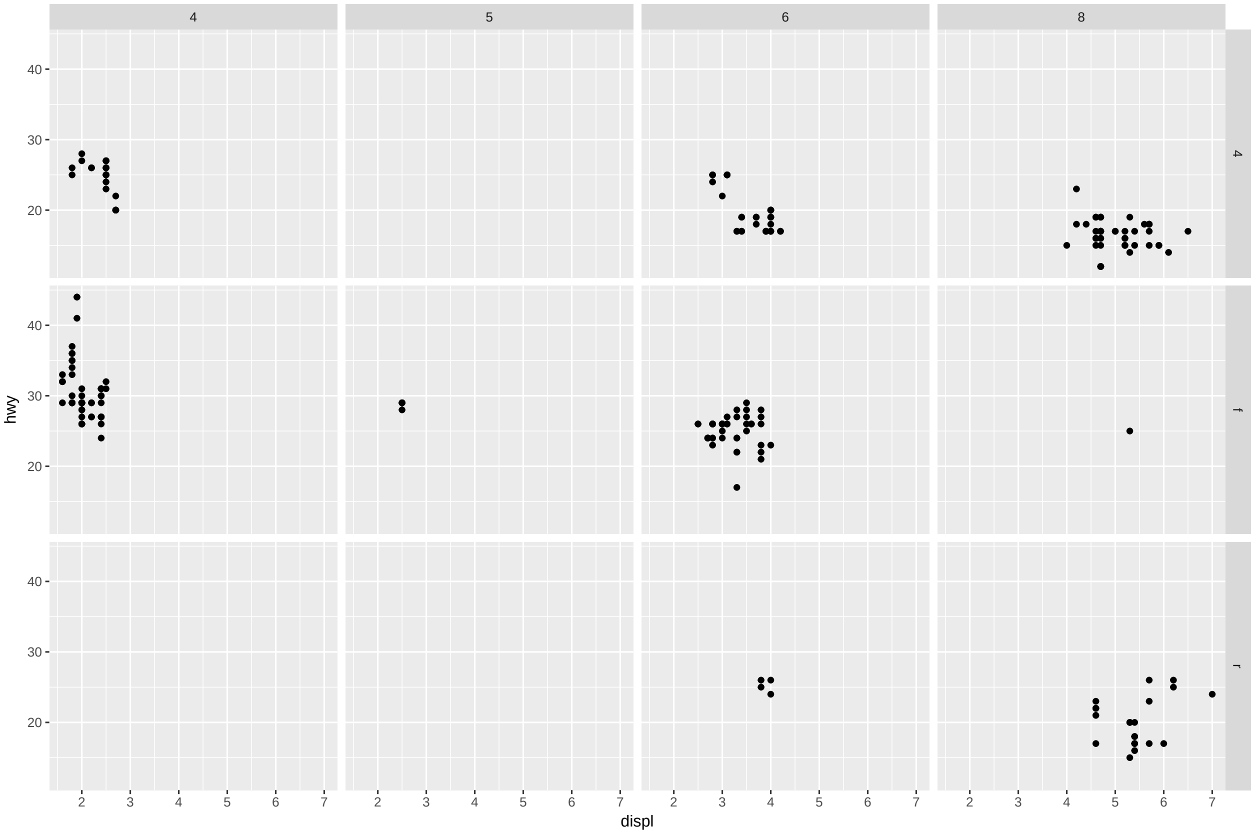


Facets

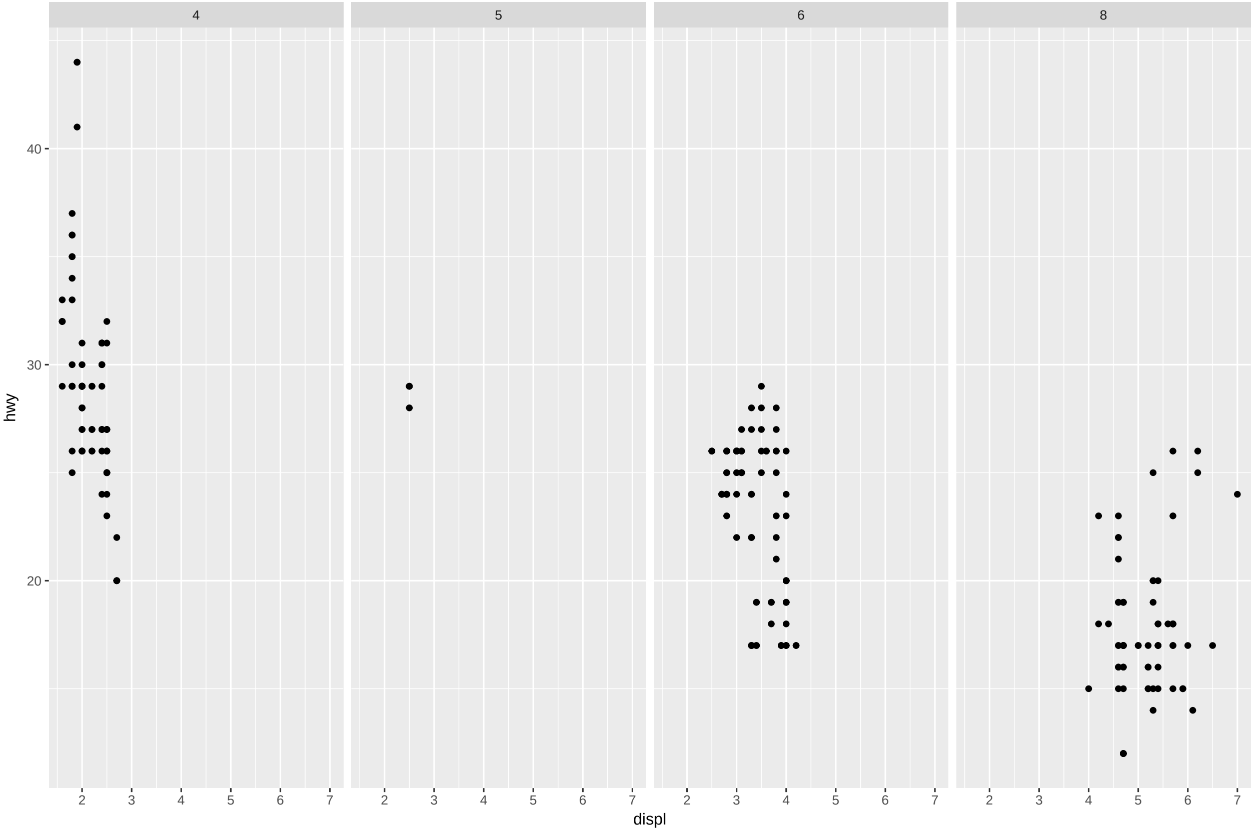
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2) # This will create a separate plot for each class of vehicle and will f
```



```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ cyl) # This will create a grid of plots with one plot for each combination of drv an
```



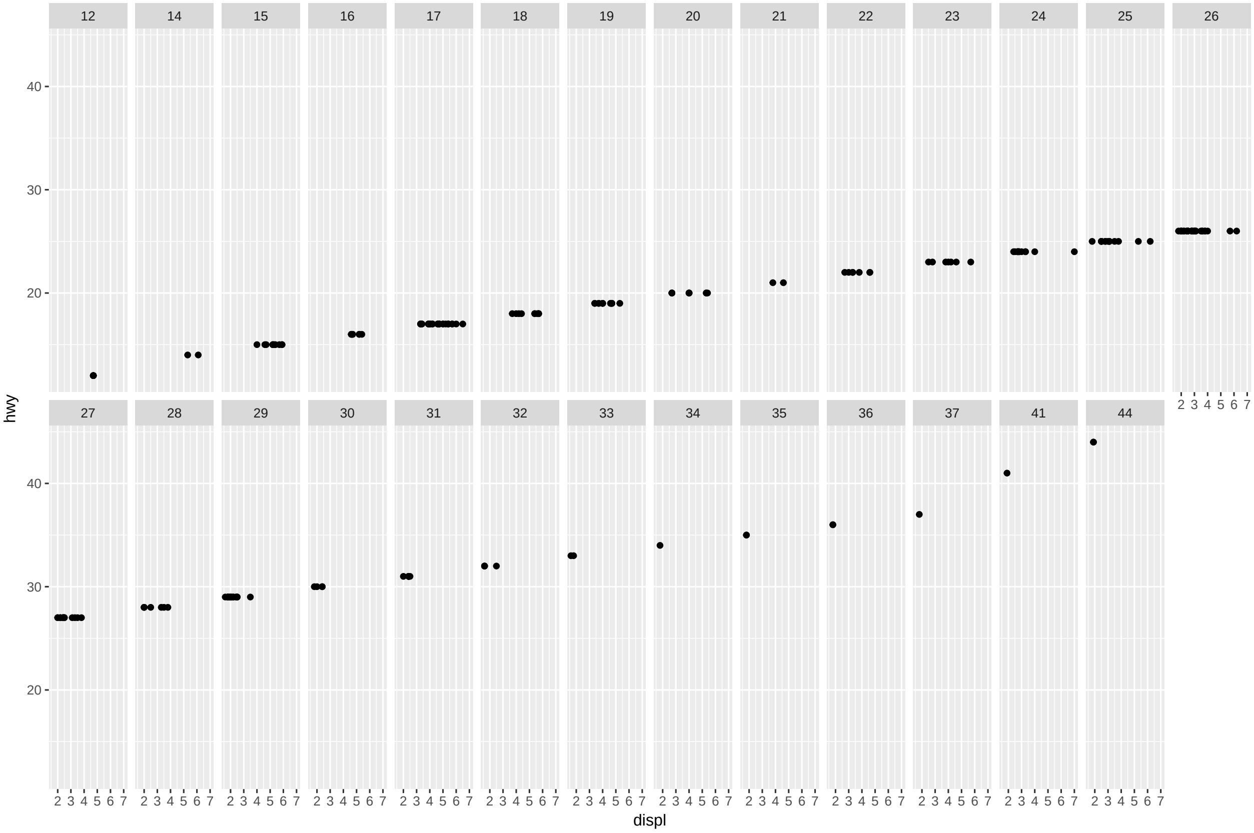
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
    facet_grid(. ~ cyl) # Use the . to create plots for each level of cylinder (cyl) in the columns dim
```



Exercises 3.5.1

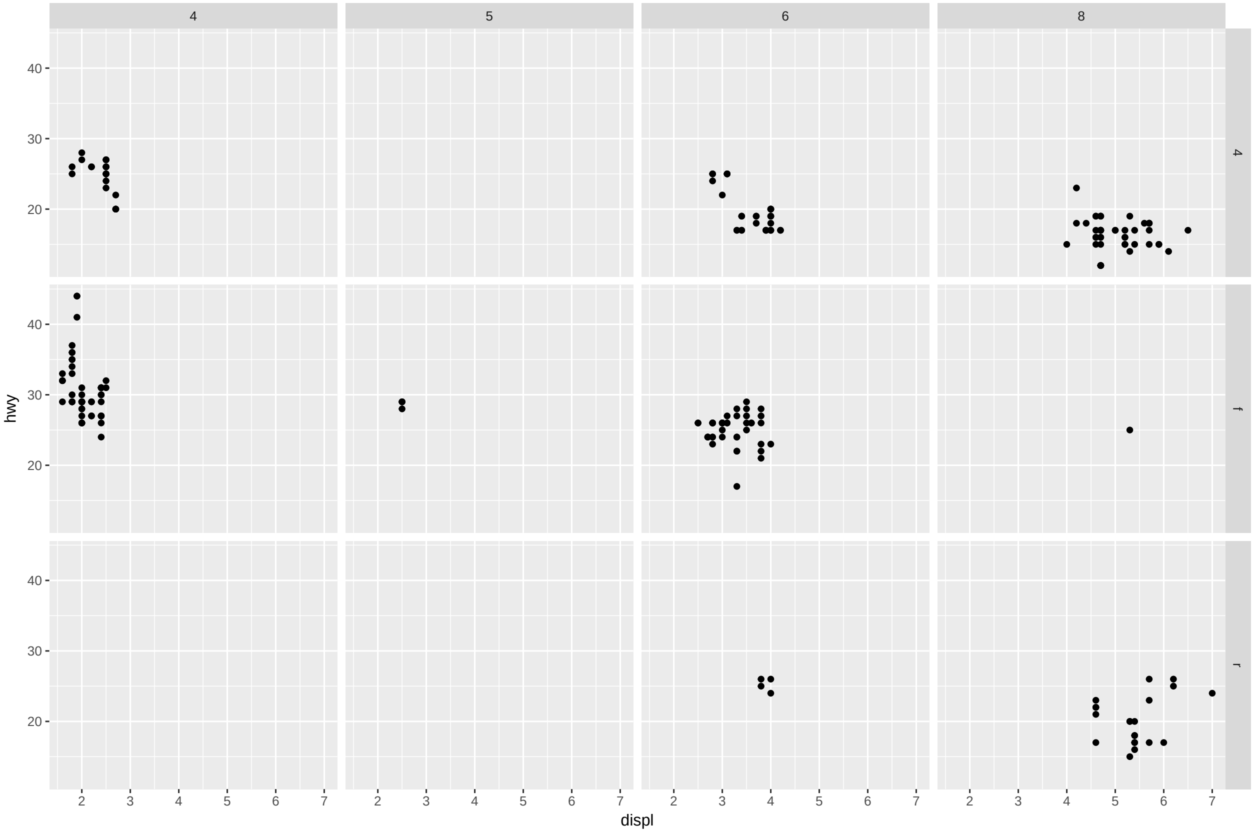
1. If faceting is done with a continuous variable, a plot is created for each value for which there is at least one observation.

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

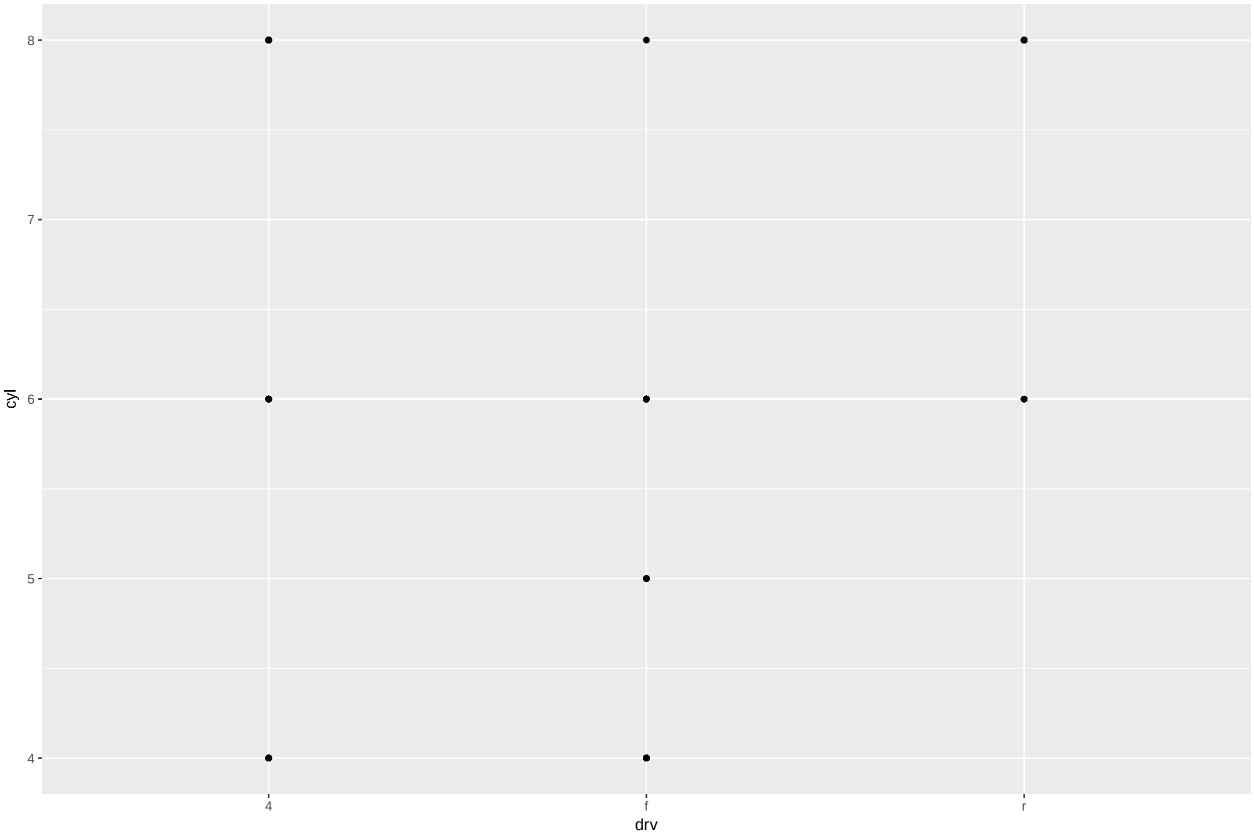


2. The empty cells in the plot with `facet_grid(drv ~ cyl)` indicate that there are no cars with at the intersection of that number of cylinders and that type of drive (e.g. no cars with 5 cylinders and 4-wheel drive). The absence of vehicles corresponding to specific cylinde r-drive combinations is also evident in the second plot. Those intersections in the second plot without a point correspond to the empty cells in the first plot (see again cars with 5 cylinders on the y-axis and 4-wheel drive on the x-axis).

```
# First plot, with drive and cylinder are faceted
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ cyl)
```

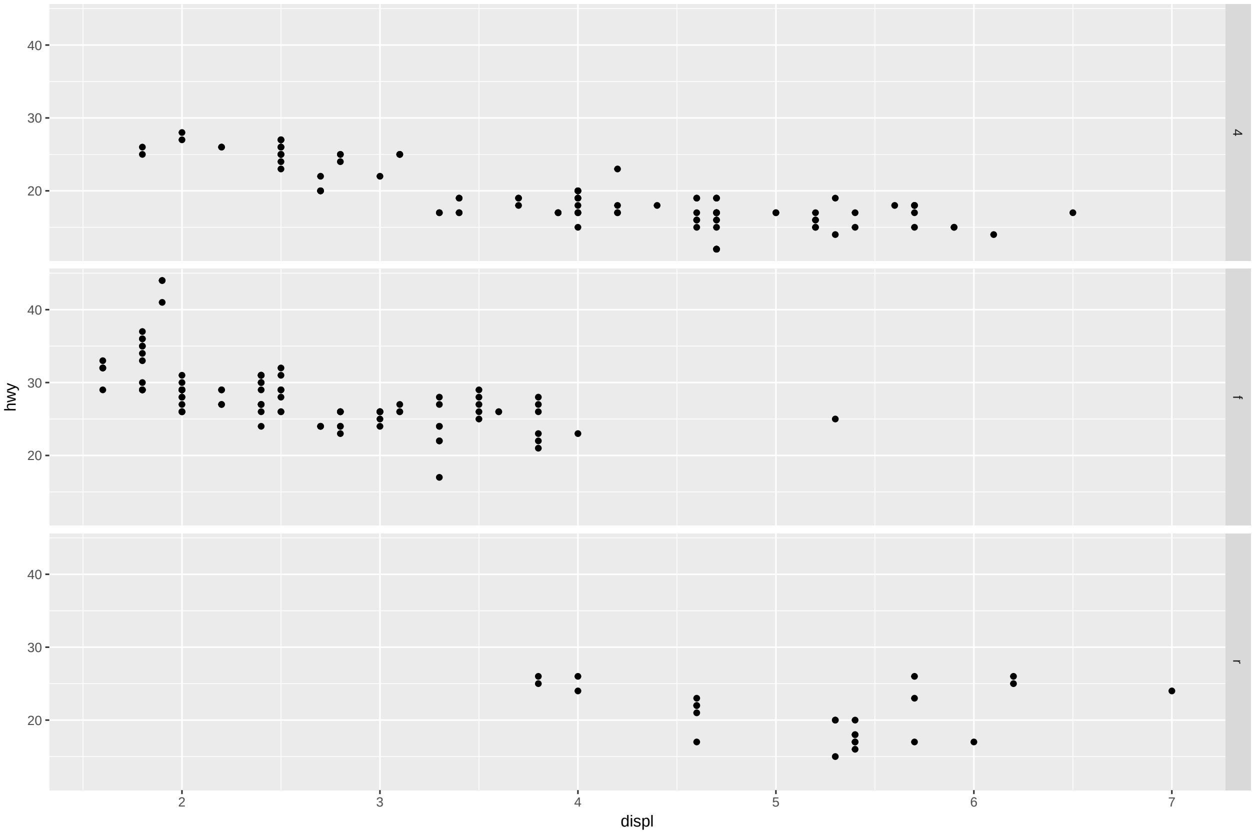


```
# Second plot, with drive and cylinder represented on the axes of a single plot
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = cyl))
```

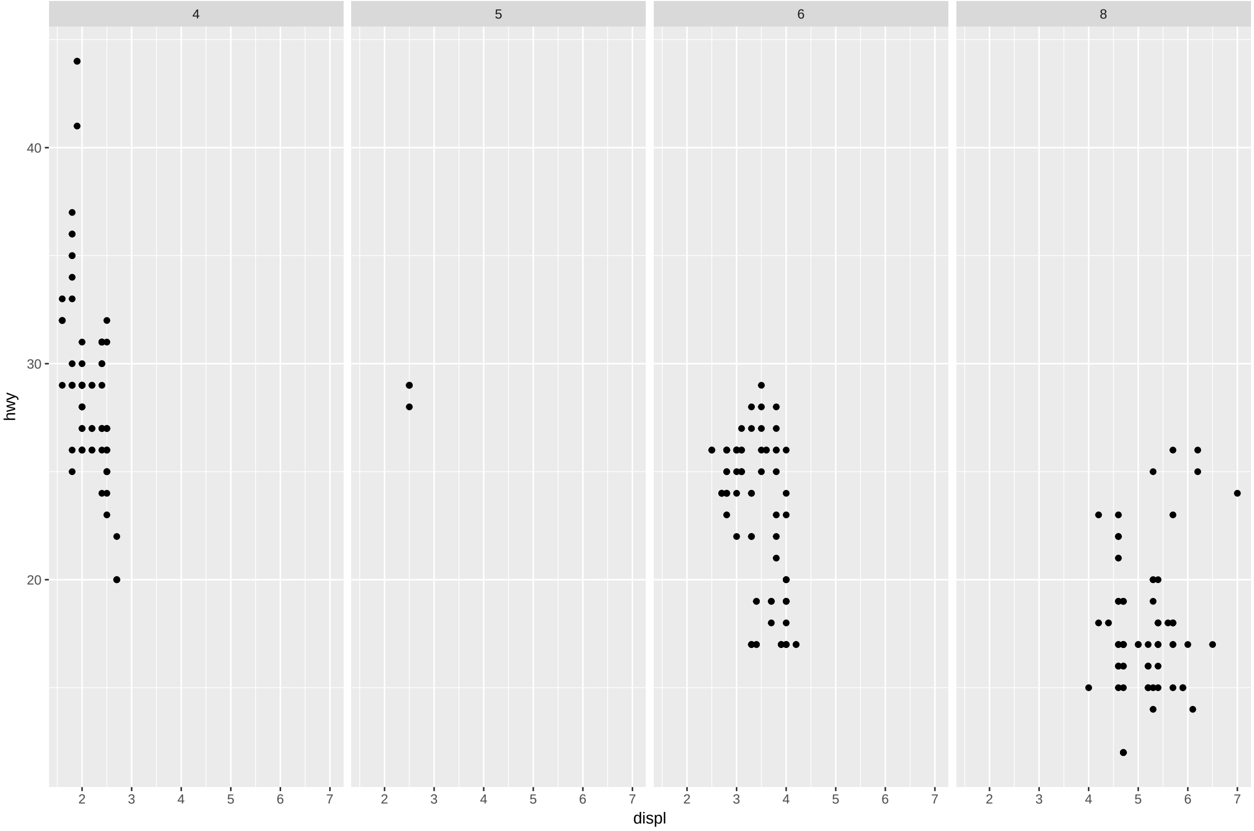


3. The first plot shows highway miles per gallon and engine displacement faceted by drive type. The . in the second position specifies that drive type should be displayed in rows. The second plot shows highway miles per gallon and engine displacement faceted by number of cylinders. The . in the first position specifies that number of cylinders should be displayed in columns.

```r
# Plot of highway mpg and engine displacement faceted by drive type
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ .)
```



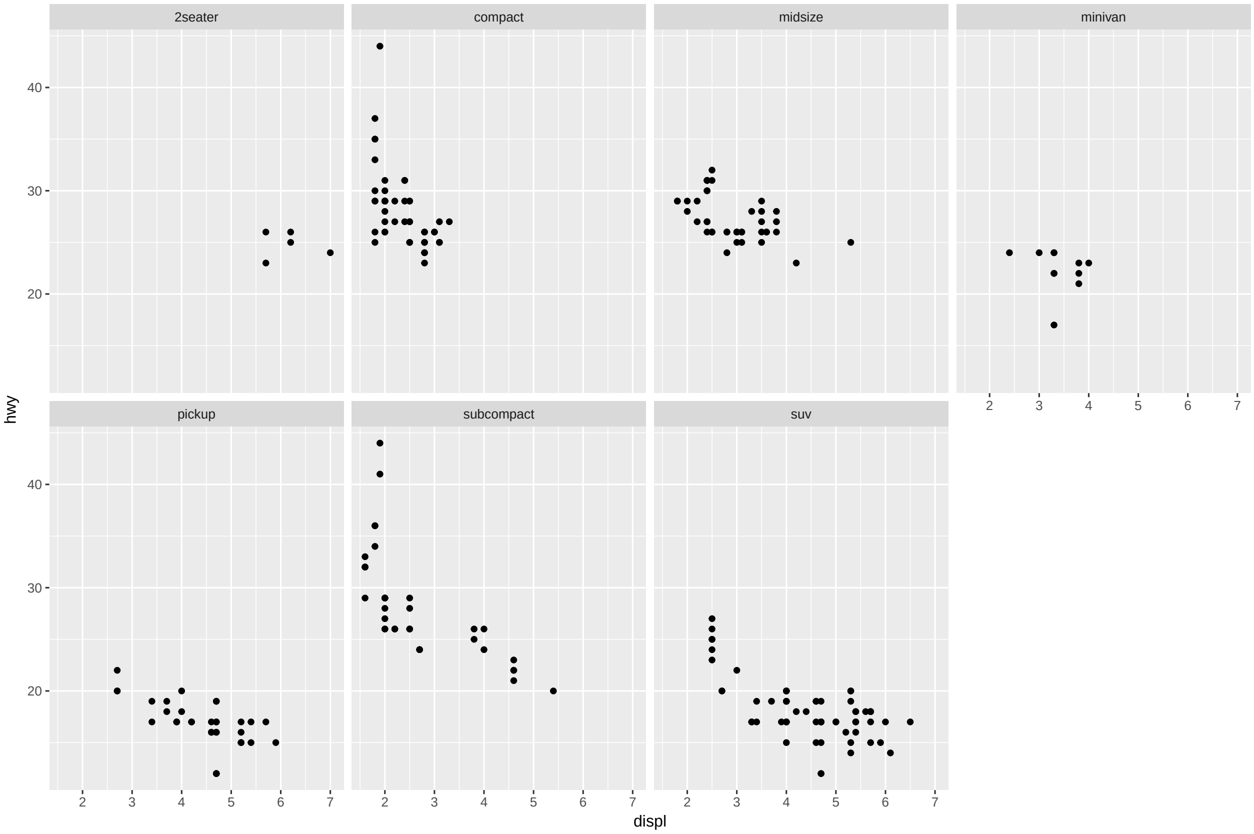
```
# The above is the same as the following except that the drive labels shift from right to top aligned.
#ggplot(data = mpg) +
#  geom_point(mapping = aes(x = displ, y = hwy)) +
#  facet_wrap(~ drv, nrow = 3)

# Plot of highway mpg and engine displacement faceted by number of cylinders
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(. ~ cyl)
```



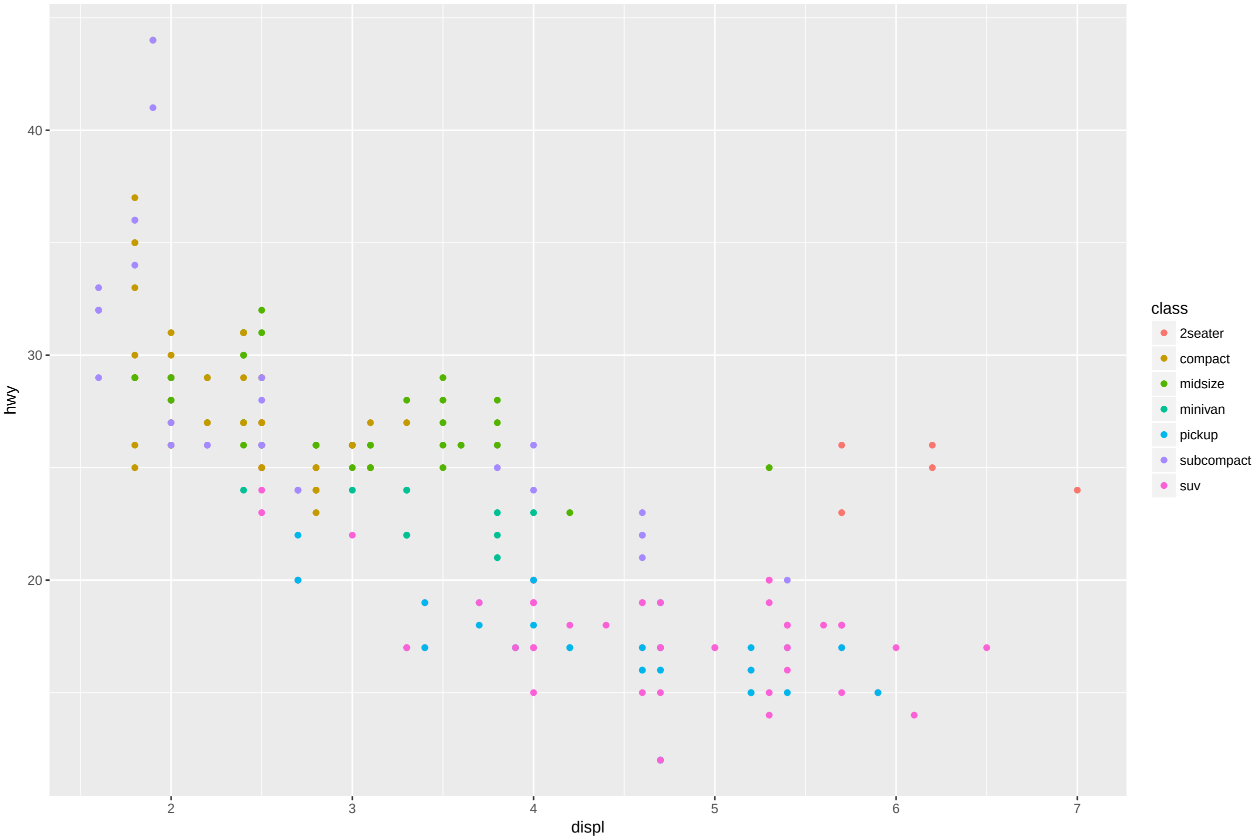
```
# The above is the same as the following. Uncomment and run the code to see.
#ggplot(data = mpg) +
#    geom_point(mapping = aes(x = displ, y = hwy)) +
#    facet_wrap(~ cyl, nrow = 1)
```

4. The advantage of using faceting rather than the color aesthetic is that with separate plots it is easier to see the shape and spread of the data points for each level of the variable. A disadvantage is that it's difficult to see the overall shape and spread of the observations across levels of the faceted variable. While using the color aesthetic works well with the mpg dataset, with a larger dataset, the likelihood of overlapping data points increases and with enough overlapping observations jittering may be insufficient. It may therefore be preferable to use faceting with large datasets.

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



```r
# Plot with color aesthetic
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```



5.

`nrow` - specifies the number of rows into which the faceted plots are fitted.
`ncol` - specifies the number of columns into which the faceted plots are fitted.
`facet_grid()` does not have `nrow` or `ncol` arguments because the number of rows and columns is determined by the number of levels of the row and column facetting variables.

```
?facet_wrap
```

6. One should put the variable with more unique levels in the columns so the plots can extend vertically where there is more space. The horizontal space is limited by the page width and adding more plots compresses them, making them difficult to read.