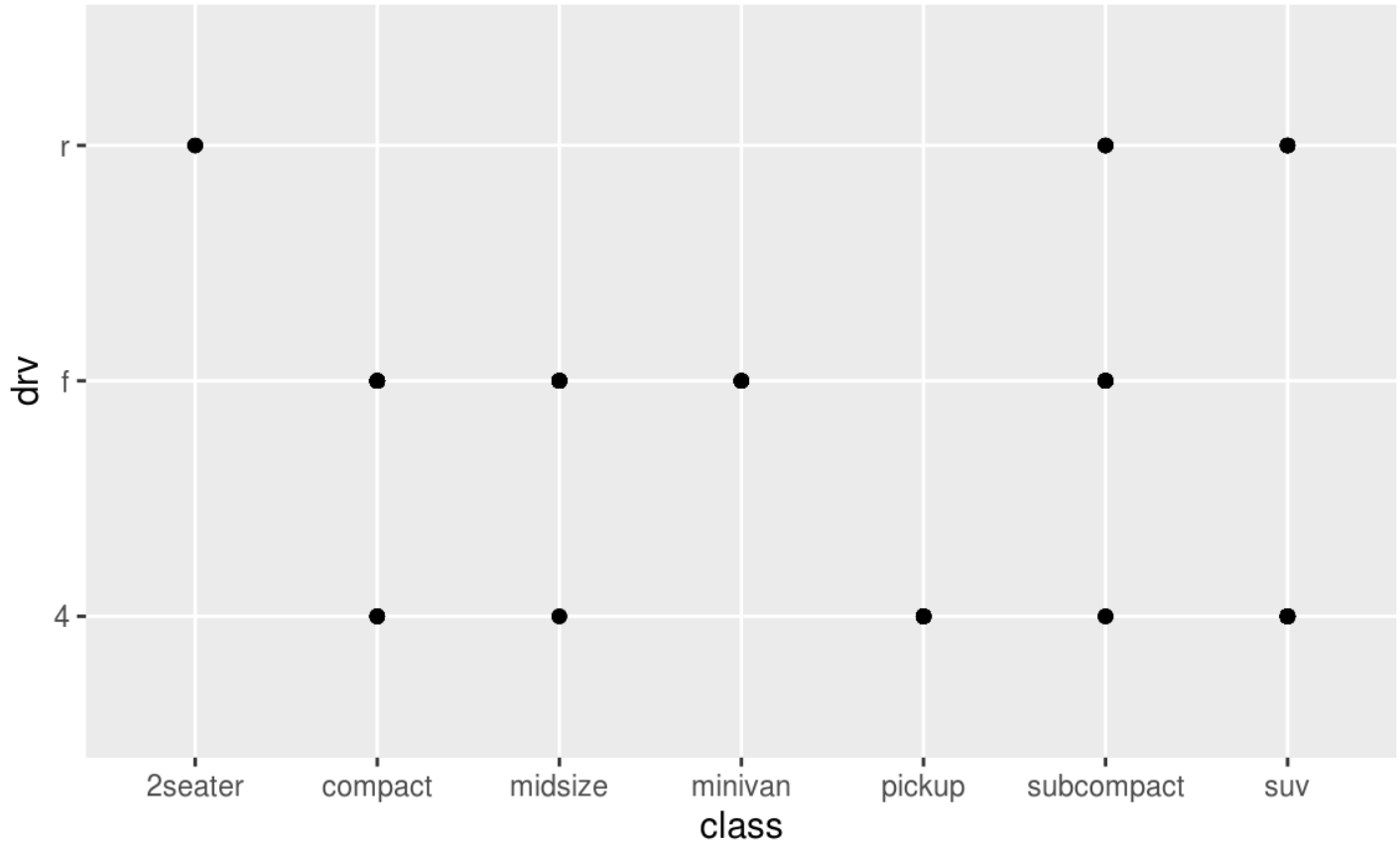
Professor McDonald
FNCE 5352 – Financial Programming and Modeling
March 10, 2020
Solutions to March 3 Assignment

3.2.4 Exercise 5

What happens if you make a scatter plot of `class` vs `drv`? Why is the plot not useful? The resulting scatterplot has only a few points.

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



A scatter plot is not a useful display of these variables since both `drv` and `class` are categorical variables. Since categorical variables typically take a small number of values, there are a limited number of unique combinations of (`x`, `y`) values that can be displayed. In this data, `mpg` takes 3 values and `class` takes 7 values, meaning that there are only 21 values that could be plotted on a scatterplot of `mpg` vs. `class`. In this data, there 12 values of (`mpg`, `class`) are observed.

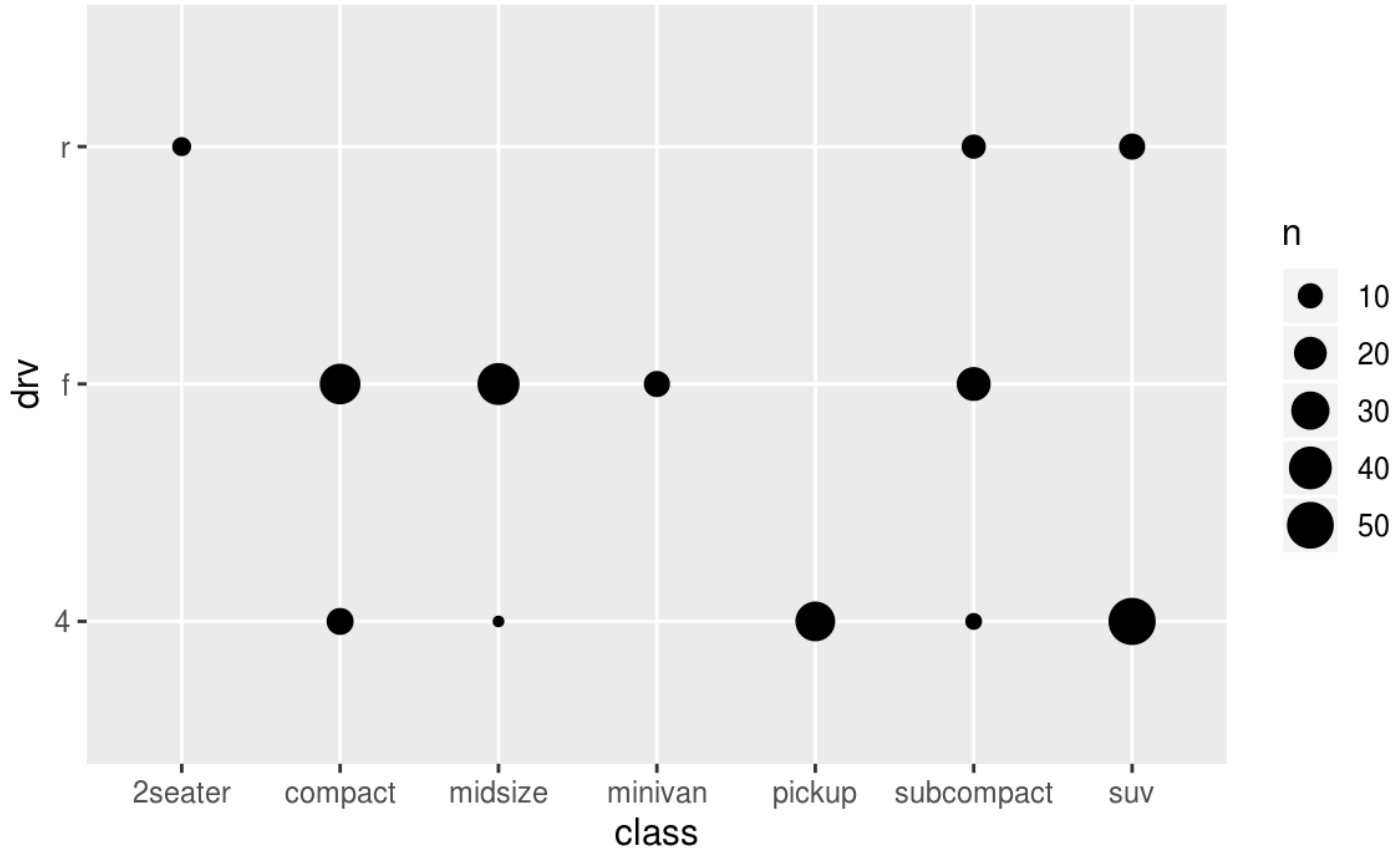
```
count(mpg, drv, class)
#> # A tibble: 12 x 3
#>   drv   class          n
```

```
#>   <chr> <chr>      <int>
#> 1 4     compact       12
#> 2 4     midsize        3
#> 3 4     pickup        33
#> 4 4     subcompact     4
#> 5 4     suv           51
#> 6 f     compact       35
#> # … with 6 more rows
```

A simple scatter plot does not show how many observations there are for each (`x`, `y`) value. As such, scatterplots work best for plotting a continuous x and a continuous y variable, and when all (`x`, `y`) values are unique.

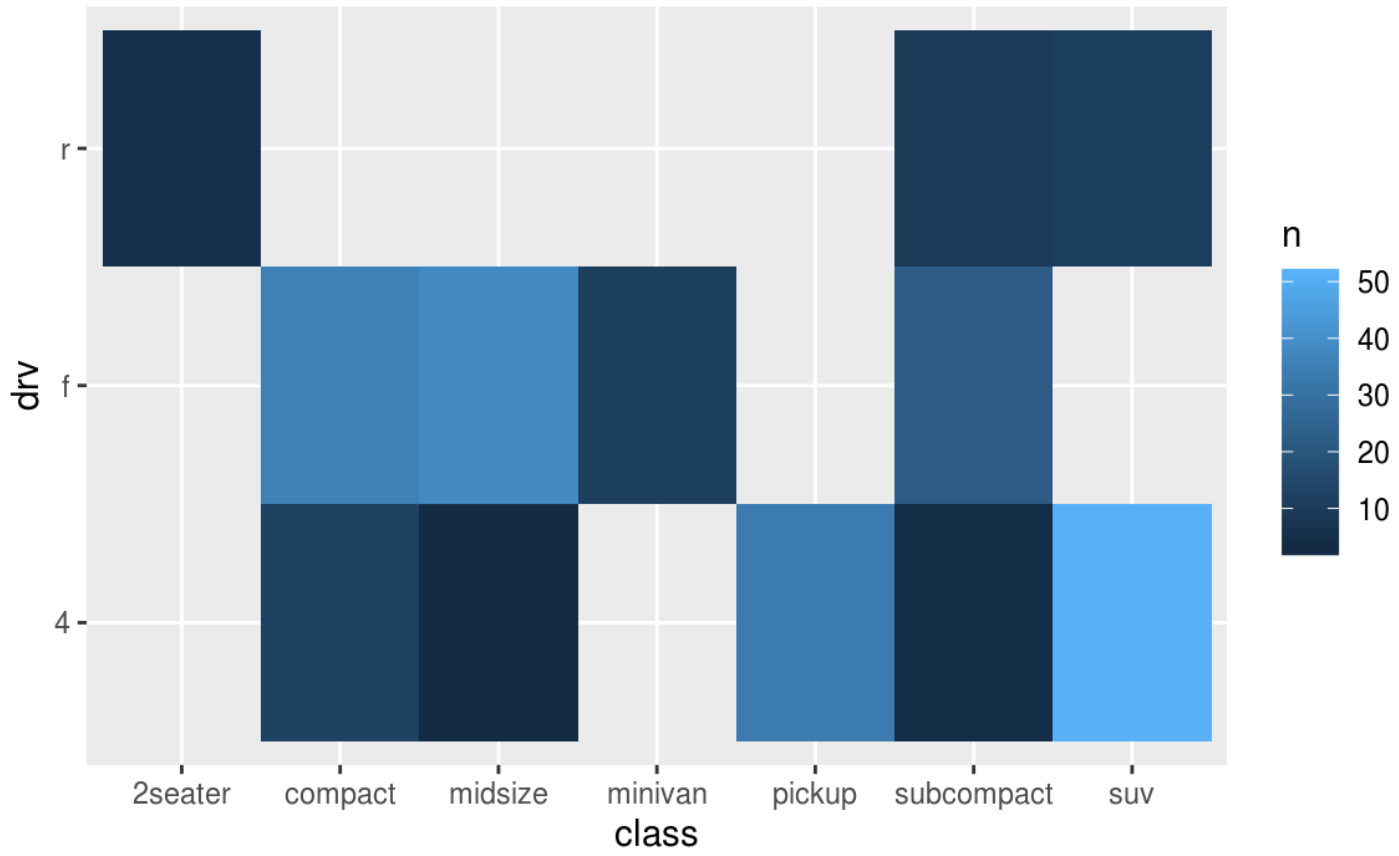
Warning: The following code uses functions introduced in a later section. Come back to this after reading section 7.5.2, which introduces methods for plotting two categorical variables. The first is `geom_count()` which is similar to a scatterplot but uses the size of the points to show the number of observations at an (`x`, `y`) point.

```
ggplot(mpg, aes(x = class, y = drv)) +
  geom_count()
```



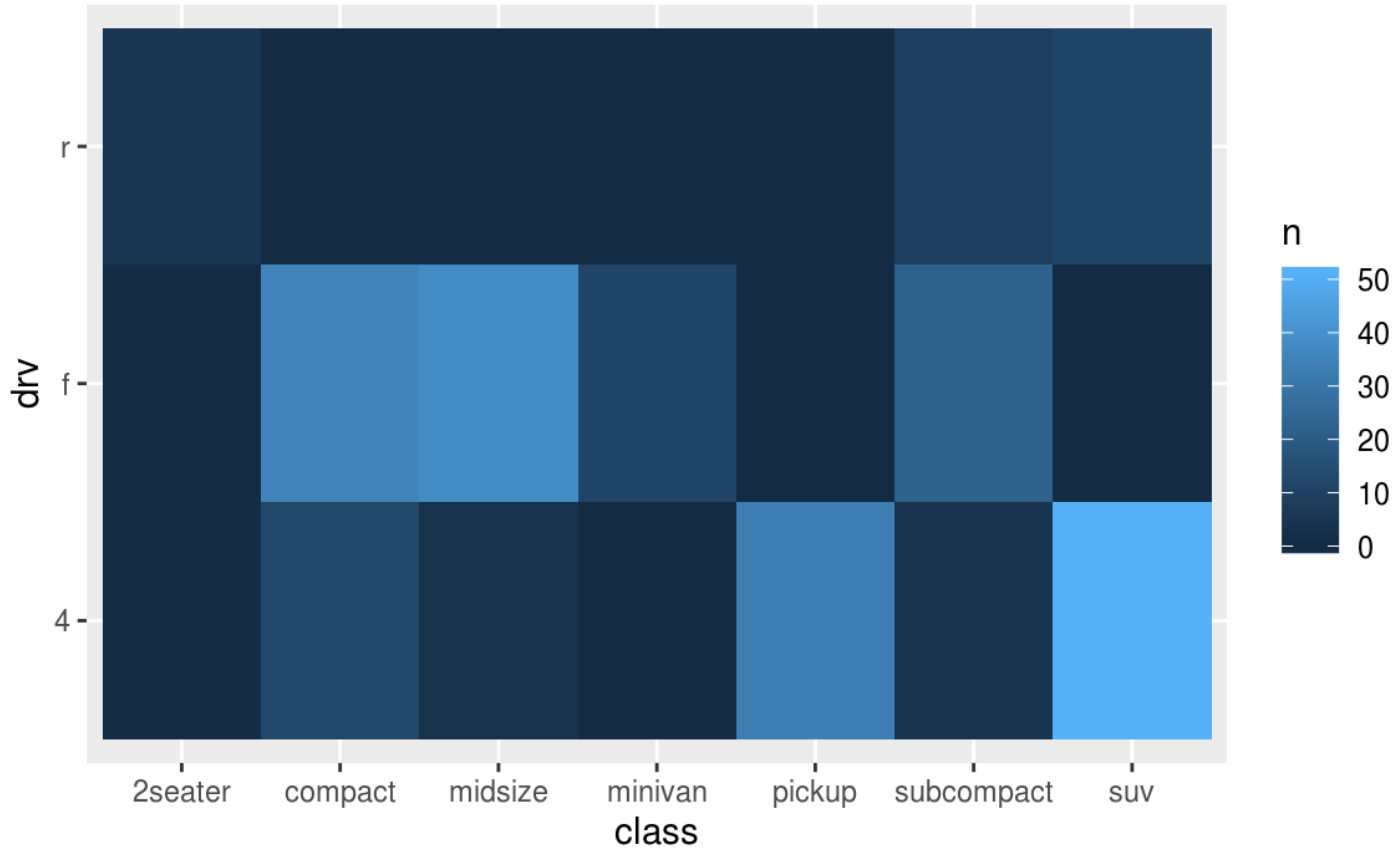
The second is `geom_tile()` which uses a color scale to show the number of observations with each (`x`, `y`) value.

```
mpg %>%
  count(class, drv) %>%
  ggplot(aes(x = class, y = drv)) +
  geom_tile(mapping = aes(fill = n))
```



In the previous plot, there are many missing tiles. These missing tiles represent unobserved combinations of `class` and `drv` values. These missing values are not unknown, but represent values of (`class`, `drv`) where `n = 0`. The `complete()` function in the tidyr package adds new rows to a data frame for missing combinations of columns. The following code adds rows for missing combinations of `class` and `drv` and uses the `fill` argument to set `n = 0` for those new rows.

```
mpg %>%
  count(class, drv) %>%
  complete(class, drv, fill = list(n = 0L)) %>%
  ggplot(aes(x = class, y = drv)) +
  geom_tile(mapping = aes(fill = n))
```



3.3.1 Exercise 2

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`?
The following list contains the categorical variables in `mpg`.

- `model`
- `trans`
- `drv`
- `fl`
- `class`

The following list contains the continuous variables in `mpg`.

- `displ`
- `year`
- `cyl`
- `cty`
- `hwy`

In the printed data frame, angled brackets at the top of each column provide type of each variable.

```
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(… f        18    29 p     comp…
#> 2 audi         a4      1.8  1999     4 manua… f        21    29 p     comp…
#> 3 audi         a4      2    2008     4 manua… 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 manua… f        18    26 p     comp…
#> # … with 228 more rows
```

Those with `<chr>` above their columns are categorical, while those with `<dbl>` or `<int>` are continuous. The exact meaning of these types will be discussed in the Vectors chapter.

Alternatively, `glimpse()` displays the type of each column.

```
glimpse(mpg)
#> Observations: 234
#> Variables: 11
#> $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "au…
#> $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quatt…
#> $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2…
#> $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 199…
#> $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, …
#> $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)",…
#> $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "…
#> $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17,…
#> $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25,…
#> $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "…
#> $ class        <chr> "compact", "compact", "compact", "compact", "compac…
```

3.6.1 Exercise 1

What geom would you use to draw a line chart? A boxplot? A histogram? An area chart?

- line chart: `geom_line()`
- boxplot: `geom_boxplot()`
- histogram: `geom_hist()`
- area chart: `geom_area()`

4.4 Practice 1

Why does this code not work?

```
my_variable <- 10
```

```
my_varıable
#> Error in eval(expr, envir, enclos): object 'my_varıable' not found
```

The variable being printed is `my_varıable`, not `my_variable`: the seventh character is "ı" ("LATIN SMALL LETTER DOTLESS I"), not "i".

While it wouldn't have helped much in this case, the importance of distinguishing characters in code is reasons why fonts which clearly distinguish similar characters are preferred in programming. It is especially important to distinguish between two sets of similar looking characters:

- the numeral zero (0), the Latin small letter O (o), and the Latin capital letter O (O),
- the numeral one (1), the Latin small letter I (i), the Latin capital letter I (I), and Latin small letter L (l).

In these fonts, zero and the Latin letter O are often distinguished by using a glyph for zero that uses either a dot in the interior or a slash through it. Some examples of fonts with dotted or slashed zero glyphs are Consolas, Deja Vu Sans Mono, Monaco, Menlo, Source Sans Pro, and FiraCode.

Error messages of the form `"object '...' not found"` mean exactly what they say. R cannot find an object with that name. Unfortunately, the error does not tell you why that object cannot be found, because R does not know the reason that the object does not exist. The most common scenarios in which I encounter this error message are

1. I forgot to create the object, or an error prevented the object from being created.
2. I made a typo in the object's name, either when using it or when I created it (as in the example above), or I forgot what I had originally named it. If you find yourself often writing the wrong name for an object, it is a good indication that the original name was not a good one.
3. I forgot to load the package that contains the object using `library()`.

Exercise 4.2

Tweak each of the following R commands so that they run correctly:

```
library(tidyverse)

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

fliter(mpg, cyl = 8)
filter(diamond, carat > 3)
```

The current version of *R4DS* includes the following code for this exercise:

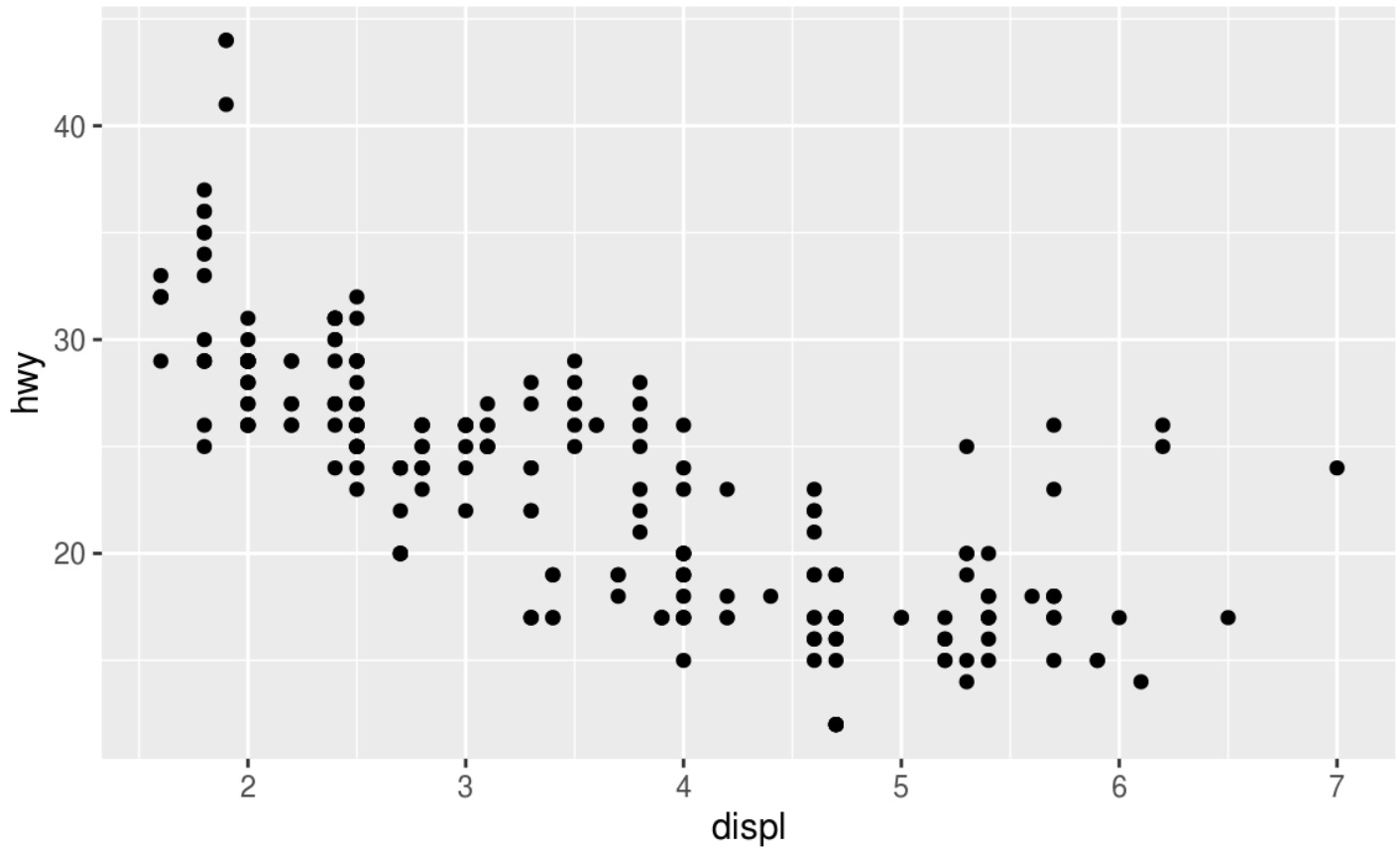
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

This is not the intended code for the exercise. See hadley/r4ds#760 and jrnold/r4ds-exercise-solutions#192.

```
ggplot(dota = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
#> Error in FUN(X[[i]], ...): object 'displ' not found
```

The error message is `argument "data" is missing, with no default`. This error is a result of a typo, `dota` instead of `data`.

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




```
fliter(mpg, cyl = 8)
#> Error in fliter(mpg, cyl = 8): could not find function "fliter"
```

R could not find the function `fliter()` because we made a typo: `fliter` instead of `filter`.

```
filter(mpg, cyl = 8)
#> Error: `cyl` (`cyl = 8`) must not be named, do you need `==`?
```

We aren't done yet. But the error message gives a suggestion. Let's follow it.

```
filter(mpg, cyl == 8)
#> # A tibble: 70 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         a6 qu…   4.2  2008     8 auto… 4        16    23 p     mids…
#> 2 chevrolet    c1500…   5.3  2008     8 auto… r        14    20 r     suv
#> 3 chevrolet    c1500…   5.3  2008     8 auto… r        11    15 e     suv
#> 4 chevrolet    c1500…   5.3  2008     8 auto… r        14    20 r     suv
#> 5 chevrolet    c1500…   5.7  1999     8 auto… r        13    17 r     suv
#> 6 chevrolet    c1500…   6    2008     8 auto… r        12    17 r     suv
#> # … with 64 more rows
filter(diamond, carat > 3)
#> Error in filter(diamond, carat > 3): object 'diamond' not found
```

R says it can't find the object `diamond`. This is a typo; the data frame is named `diamonds`.

```
filter(diamonds, carat > 3)
#> # A tibble: 32 x 10
```

```
#>   carat cut     color clarity depth table price     x     y     z
#>   <dbl> <ord>   <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
#> 1  3.01 Premium I     I1       62.7    58  8040  9.1   8.97  5.67
#> 2  3.11 Fair    J     I1       65.9    57  9823  9.15  9.02  5.98
#> 3  3.01 Premium F     I1       62.2    56  9925  9.24  9.13  5.73
#> 4  3.05 Premium E     I1       60.9    58 10453  9.26  9.25  5.66
#> 5  3.02 Fair    I     I1       65.2    56 10577  9.11  9.02  5.91
#> 6  3.01 Fair    H     I1       56.1    62 10761  9.54  9.38  5.31
#> # … with 26 more rows
```

How did I know? I started typing in `diamond` and RStudio completed it to `diamonds`. Since `diamonds`includes the variable `carat` and the code works, that appears to have been the problem.

Exercise 4.3

Press *Alt* + *Shift* + *K*. What happens? How can you get to the same place using the menus?

This gives a menu with keyboard shortcuts. This can be found in the menu under `Tools -> Keyboard Shortcuts Help`.