2 ggplot2

We will be using the `ggplot2` package for making graphics in this class.

The first time on your machine you'll need to install the package:

```
install.packages("ggplot2")
```

Whenever you first want to plot during an `R` session, we need to load the library.

```
library(ggplot2)
```

2.1 Why visualize?

The sole purpose of visualization is communication. Visualization offers an alternative way of communicating numbers than simply using tables. Often, we can get more information out of our numbers graphically than with numerical summaries alone. Through the use of exploratory data analysis, we can see what the data can tell us beyond the formal modeling or hypothesis testing task.

For example, let's look at the following dataset.

```
anscombe
```

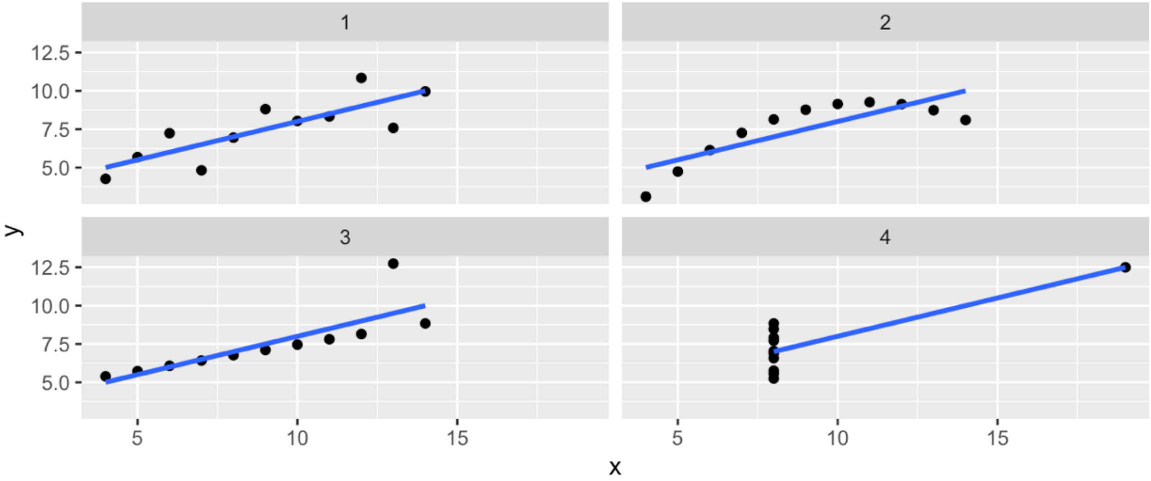
```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

Anscombe's Quartet is comprised of 4 datasets that have nearly identical simple statistical properties. Each dataset contains 11 (x, y) points with the same mean, median, standard deviation, and correlation coefficient between x and y.

dataset	mean_x	sd_x	mean_y	sd_y	cor
1	9	3.316625	7.500909	2.031568	0.8164205
2	9	3.316625	7.500909	2.031657	0.8162365
3	9	3.316625	7.500000	2.030424	0.8162867
4	9	3.316625	7.500909	2.030578	0.8165214

But this doesn't tell the whole story. Let's look closer at these datasets.

```
## `geom_smooth()` using formula 'y ~ x'
```



Visualizations can aid communication and make the data easier to perceive. It can also show us things about our data that numerical summaries won't necessarily capture.

2.2 A Grammar of Graphics

The grammar of graphics was developed by Leland Wilkinson (https://www.springer.com/gp/book/9780387245447). It is a set of grammatical rules for creating perceivable graphs. Rather than thinking about a limited set of graphs, we can think about graphical forms. This abstraction makes thinking, creating, and communicating graphics easier.

Statistical graphic specifications are expressed using the following components.

1. **data**: a set of data operations that create variables from datasets
2. **trans**: variable transformations

3. **scale**: scale transformations
4. **coord**: a coordinate system
5. **element**: graphs (points) and their aesthetic attributes (color)
6. **guide**: one or more guides (axes, legends, etc.)

ggplot2 is a package written by Hadley Wickham (https://vita.had.co.nz/papers/layered-grammar.html) that implements the ideas in the grammar of graphics to create layered plots.

ggplot2 uses the idea that you can build every graph with graphical components from three sources

1. the data, represented by geoms
2. the scales and coordinate system
3. the plot annotations

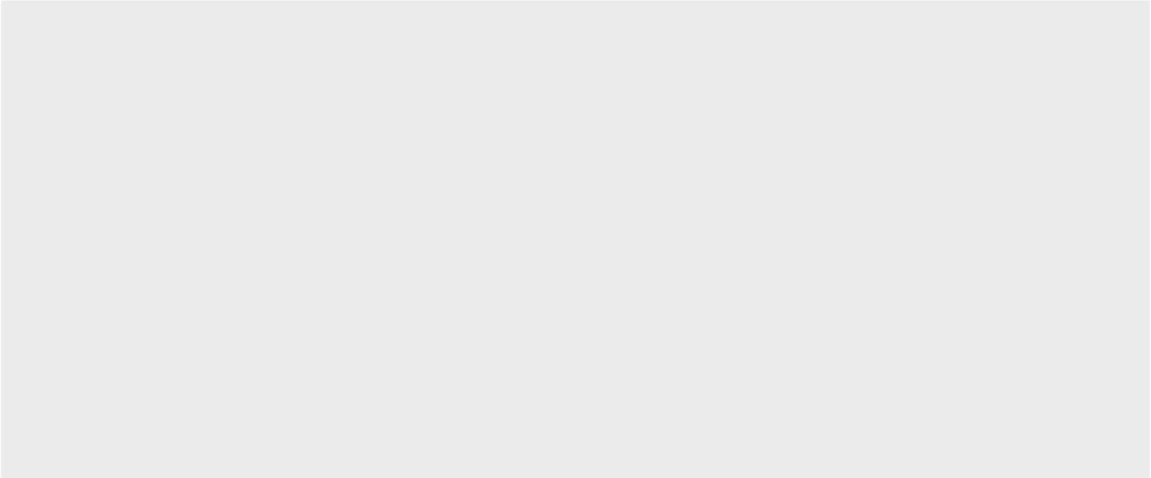
This works by mapping values in the data to visual properties of the geom (aesthetics) like size, color, and locations.

Let's build a graphic. We start with the data. We will use the diamonds dataset, and we want to explore the relationship between carat and price.

```
head(diamonds)
```

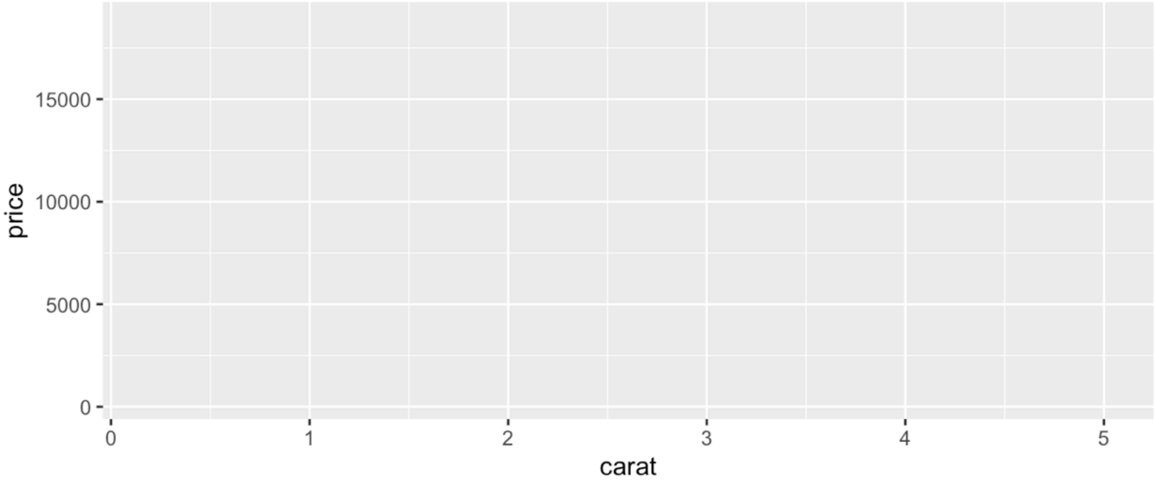
```
## # A tibble: 6 × 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
ggplot(data = diamonds)
```



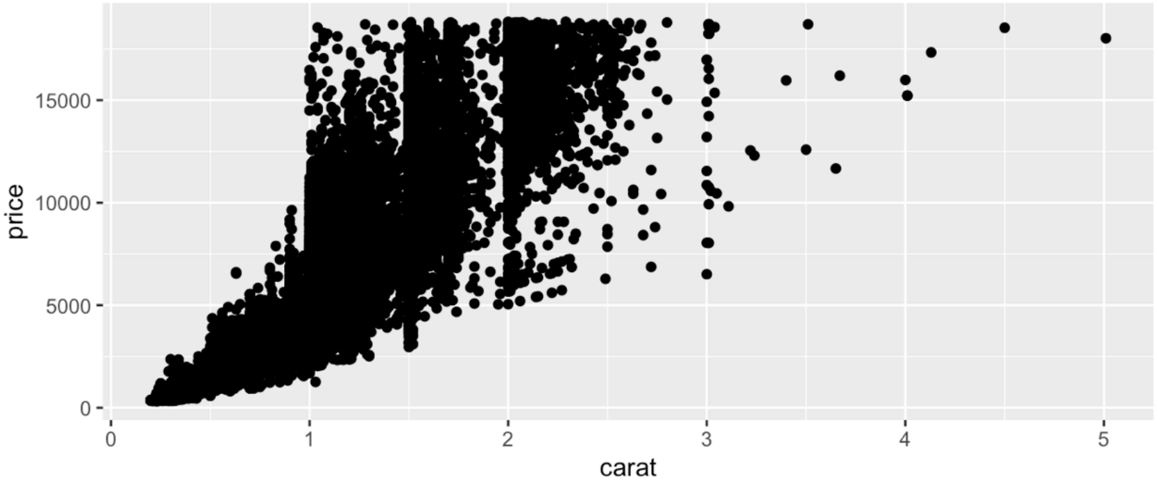
Next we need to specify the aesthetic (variable) mappings.

```
ggplot(data = diamonds, mapping = aes(carat, price))
```



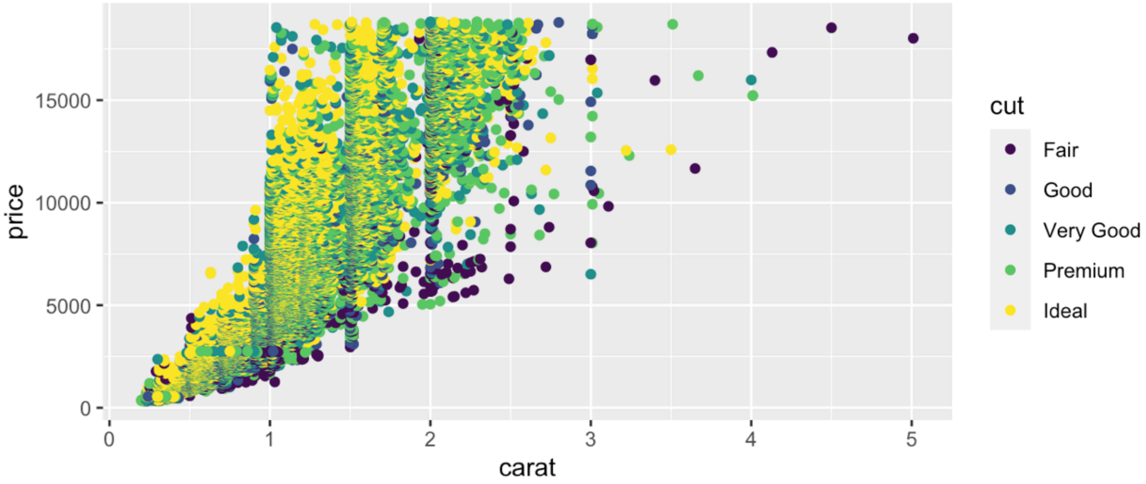
Now we choose a geom to display our data.

```
ggplot(data = diamonds, mapping = aes(carat, price)) +
  geom_point()
```



And add an aesthetic to our plot.

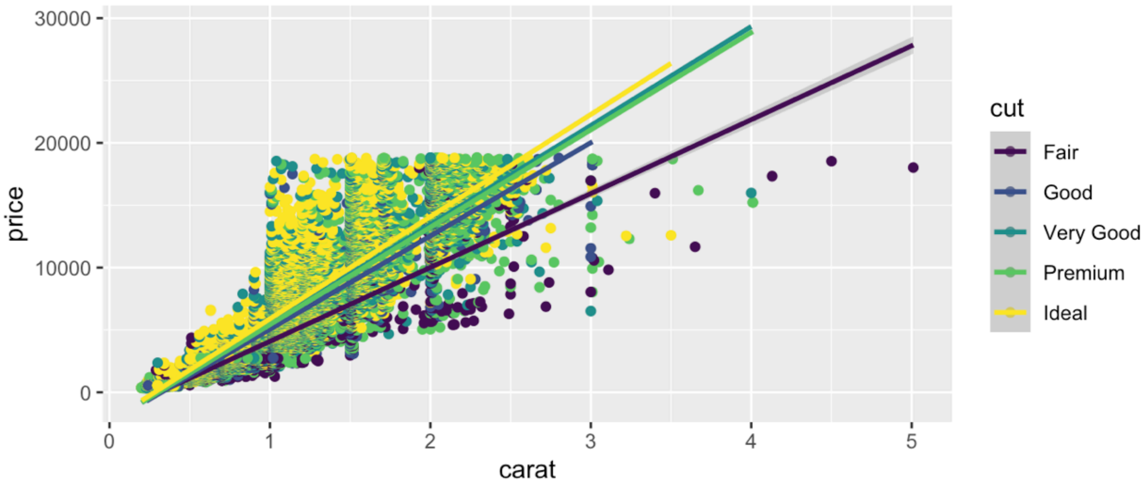
```
ggplot(data = diamonds, mapping = aes(carat, price)) +
  geom_point(aes(color = cut))
```



We could add another layer.

```
ggplot(data = diamonds, mapping = aes(carat, price)) +
  geom_point(aes(color = cut)) +
  geom_smooth(aes(color = cut), method = "lm")
```

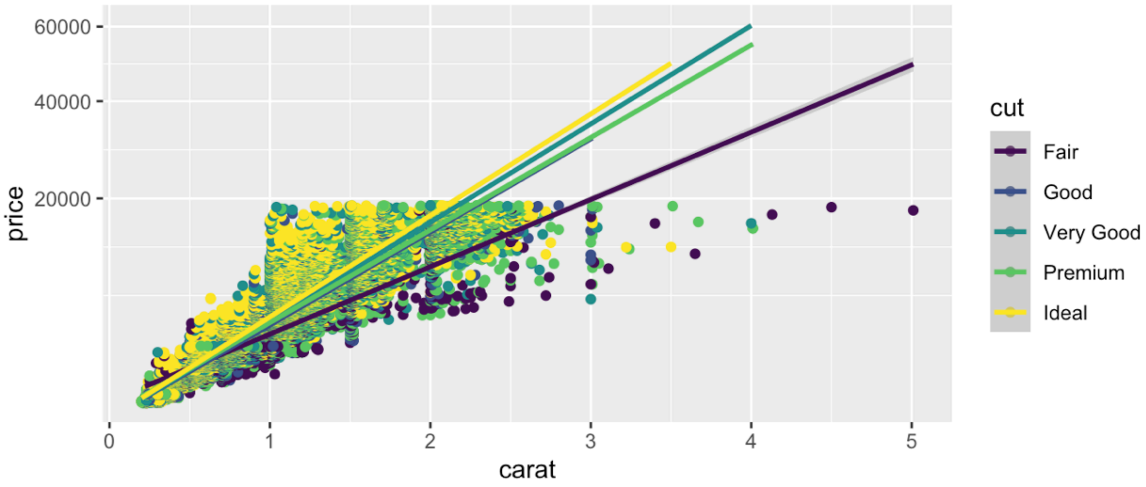
```
## `geom_smooth()` using formula 'y ~ x'
```



And finally, we can specify coordinate transformations.

```
ggplot(data = diamonds, mapping = aes(carat, price)) +
  geom_point(aes(color = cut)) +
  geom_smooth(aes(color = cut), method = "lm") +
  scale_y_sqrt()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Notice we can add on to our plot in a layered fashion.

2.3 Graphical Summaries

There are some basic charts we will use in this class that cover a wide range of cases. For univariate data, we can use dotplots, histograms, and barcharts. For two dimensional data, we can look at scatterplots and boxplots.

2.3.1 Scatterplots

Scatterplots are used for investigating relationships between two numeric variables. To demonstrate some of the flexibility of scatterplots in `ggplot2`, let's answer the following question.

> Do cars with big engines use more fuel than cars with small engines?

We will use the `mpg` dataset in the `ggplot2` package to answer the question. This dataset contains observations collected by the US Environmental Protection Agency on 38 models of car.

```
dim(mpg)
```

```
## [1] 234  11
```

```
summary(mpg)
```

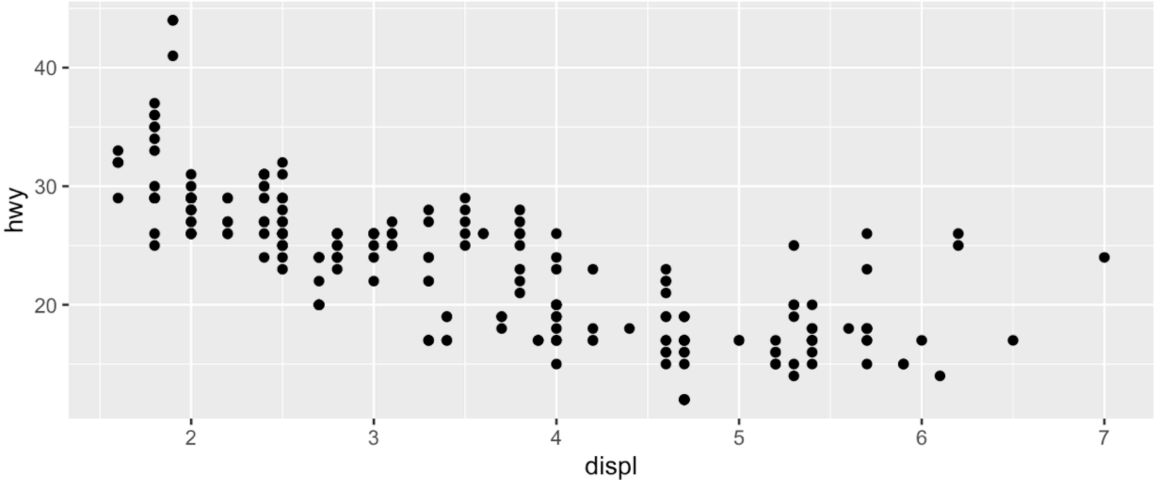
```
##  manufacturer          model               displ            year     
##  Length:234         Length:234         Min.   :1.600   Min.   :1999  
##  Class :character   Class :character   1st Qu.:2.400   1st Qu.:1999  
##  Mode  :character   Mode  :character   Median :3.300   Median :2004  
##                                        Mean   :3.472   Mean   :2004  
##                                        3rd Qu.:4.600   3rd Qu.:2008  
##                                        Max.   :7.000   Max.   :2008  
##       cyl           trans               drv                 cty      
##  Min.   :4.000   Length:234         Length:234         Min.   : 9.00  
##  1st Qu.:4.000   Class :character   Class :character   1st Qu.:14.00  
##  Median :6.000   Mode  :character   Mode  :character   Median :17.00  
##  Mean   :5.889                                         Mean   :16.86  
##  3rd Qu.:8.000                                         3rd Qu.:19.00  
##  Max.   :8.000                                         Max.   :35.00  
##       hwy             fl                 class          
##  Min.   :12.00   Length:234         Length:234         
##  1st Qu.:18.00   Class :character   Class :character   
##  Median :24.00   Mode  :character   Mode  :character   
##  Mean   :23.44                                         
##  3rd Qu.:27.00                                         
##  Max.   :44.00                                         
```

```
head(mpg)
```

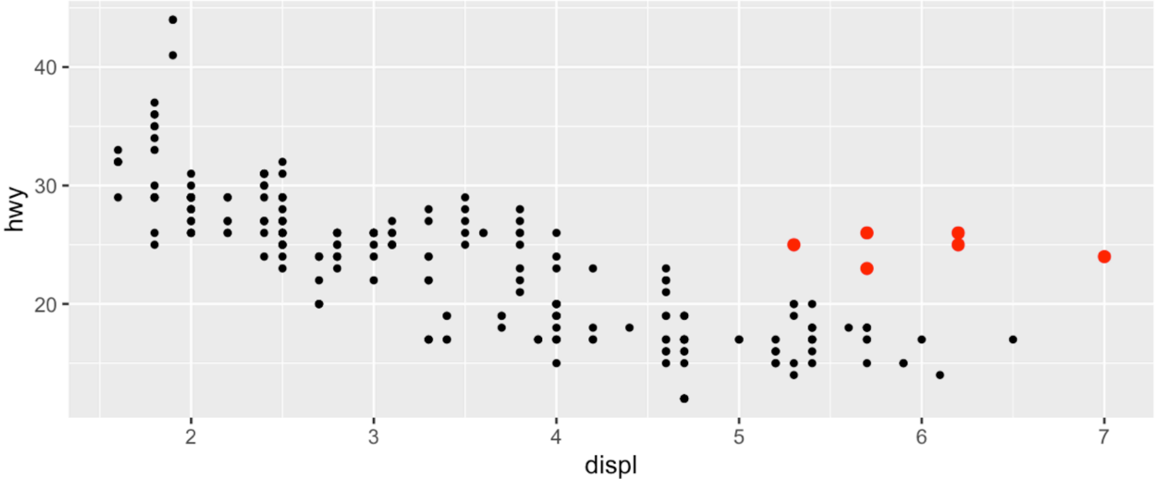
```
## # A tibble: 6 × 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy
fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int>
<chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29
p     compa…
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29
p     compa…
## 3 audi         a4      2    2008     4 manual(m6) f        20    31
p     compa…
## 4 audi         a4      2    2008     4 auto(av)   f        21    30
p     compa…
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26
p     compa…
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26
p     compa…
```

`mpg` contains the following variables: `displ`, a car's engine size, in liters, and `hwy`, a car's fuel efficiency on the highway, in miles per gallon (mpg).

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

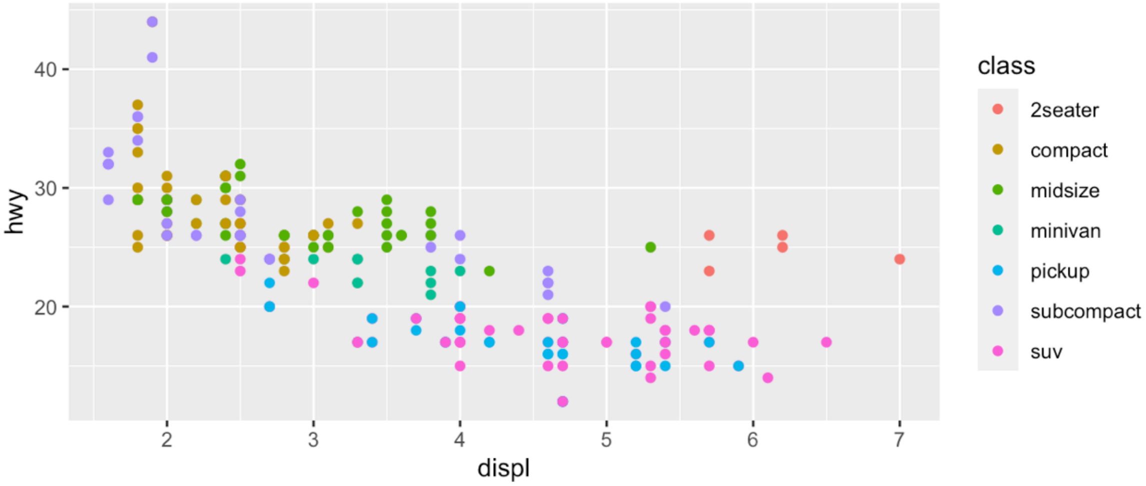


So we can say, yes, cars with larger engines have worse fuel efficiency. But there is more going on here.



The red points above seem to have higher `mpg` than they should based on engine size alone (outliers). Maybe there is a confounding variable we've missed. The `class` variable of the mpg dataset classifies cars into groups such as compact, midsize, and SUV.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(displ, hwy, colour = class))
```

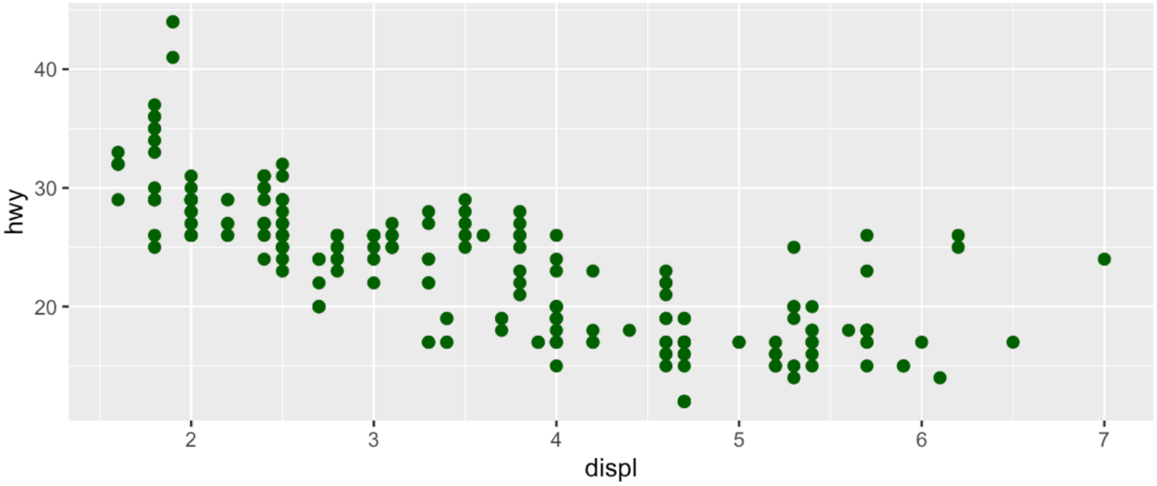


The colors show that many of the unusual points are two-seater cars, probably sports cars! Sports cars have large engines like SUVs and pickup trucks, but small bodies like midsize and compact cars, which improves their gas mileage.

Instead of color, we could also map a categorical variable (like `class`) to shape, size, and transparency (`alpha`).

So far we have mapped aesthetics to variables in our dataset. What happens if we just want to generally change the aesthetics of our plots, without tying that to data? We can specify general aesthetics as parameters of the `geom`, instead of specifying them as aesthetics (`aes`).

```r
ggplot(data = mpg) +
  geom_point(mapping = aes(displ, hwy), colour = "darkgreen", size = 2)
```



When interpreting a scatterplot we can look for big patterns in our data, as well as form, direction, and strength of relationships. Additionally, we can see small patterns and deviations from those patterns (outliers).

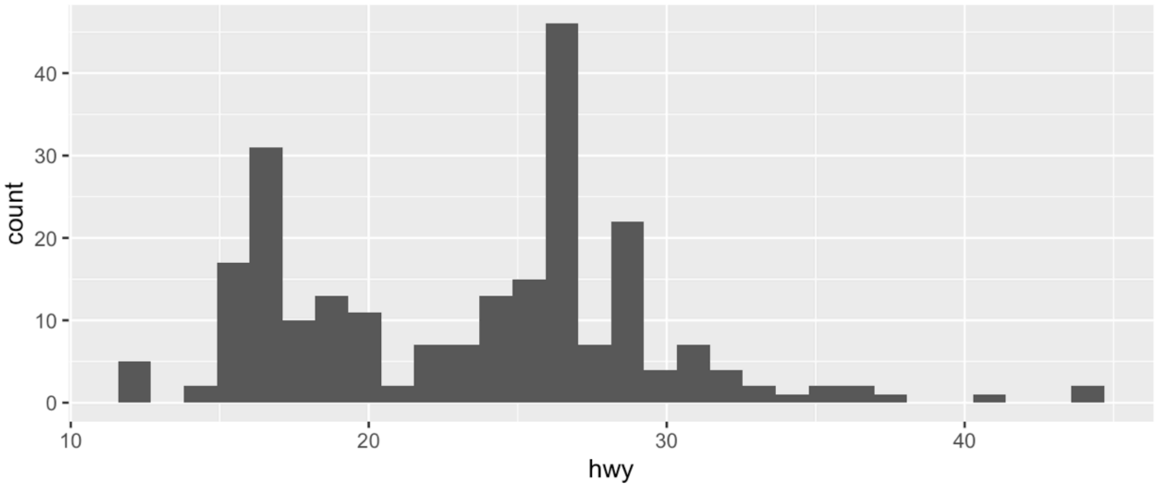
Your Turn

1. Make a scatterplot of `cty` vs. `hwy` mpg using the `mpg` dataset.
2. Describe the relationship that you see.
3. Map color and shape to type of drive the car is (see `?mpg` for details on the variables.). Do you see any patterns?
4. Alter your plot from part 3. to make all the points be larger.

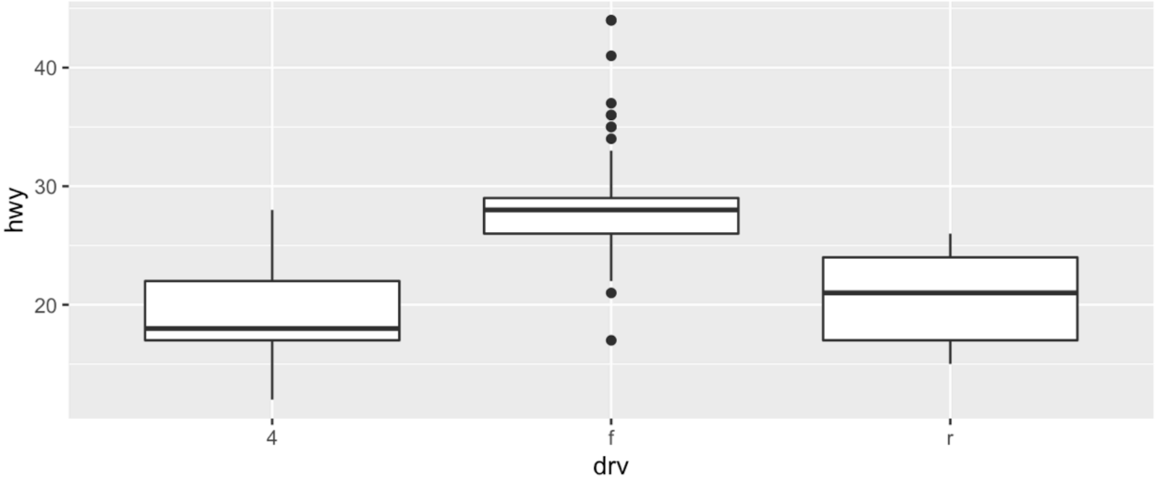
2.3.2 Histograms, Barcharts, and Boxplots

We can look at the distribution of continuous variables using **histograms** and **boxplots** and the distribution of discrete variables using **barcharts**.

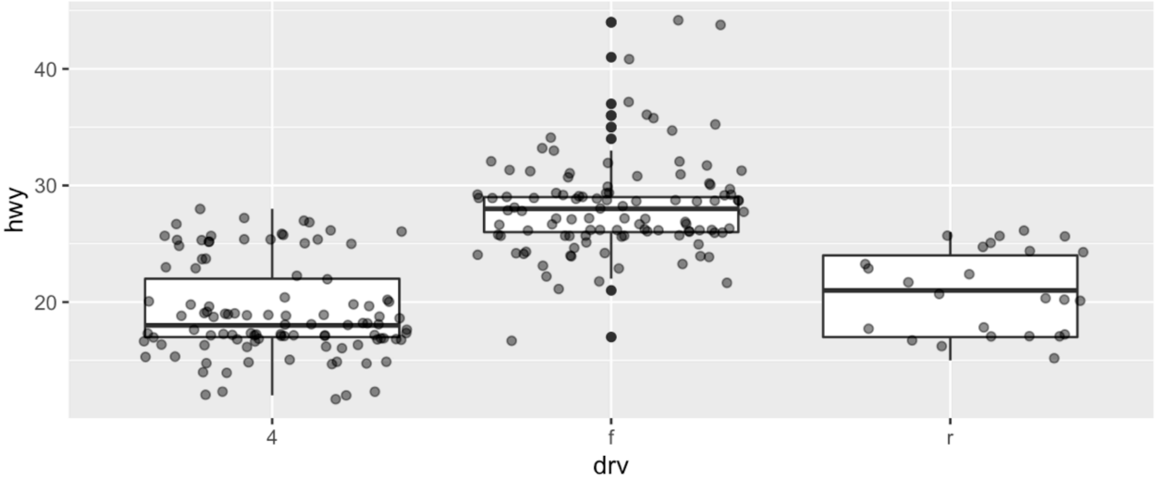
```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(hwy), bins = 30)
```



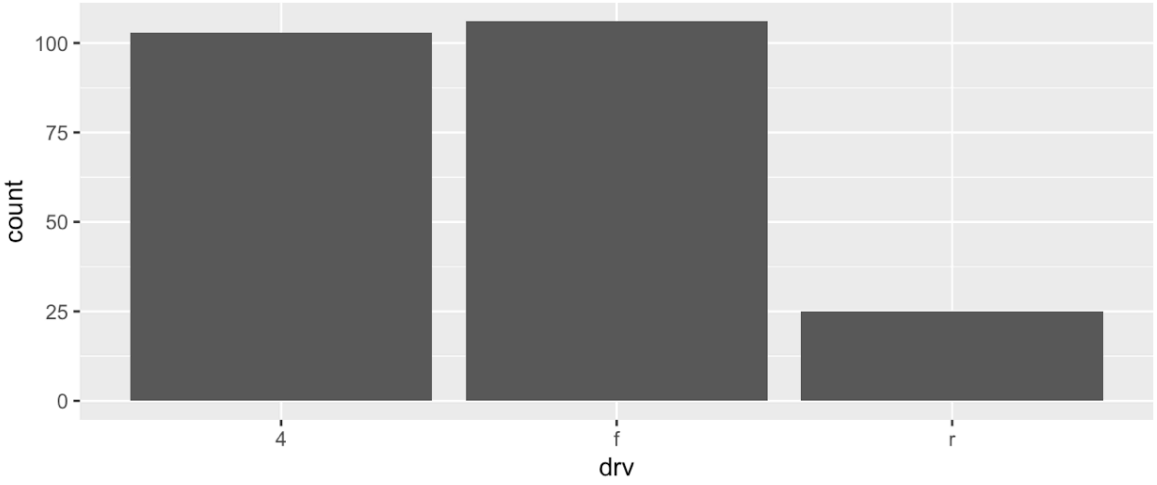
```
## histograms will look very different sometimes with different
        binwidths

ggplot(data = mpg) +
  geom_boxplot(mapping = aes(drv, hwy))
```



```
## boxplots allow us to see the distribution of a cts rv conditional on
        a discrete one
## we can also show the actual data at the same time
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(drv, hwy)) +
  geom_jitter(mapping = aes(drv, hwy), alpha = .5)
```



```
ggplot(data = mpg) +
  geom_bar(mapping = aes(drv))
```



```
## shows us the distribution of a categorical variable
```

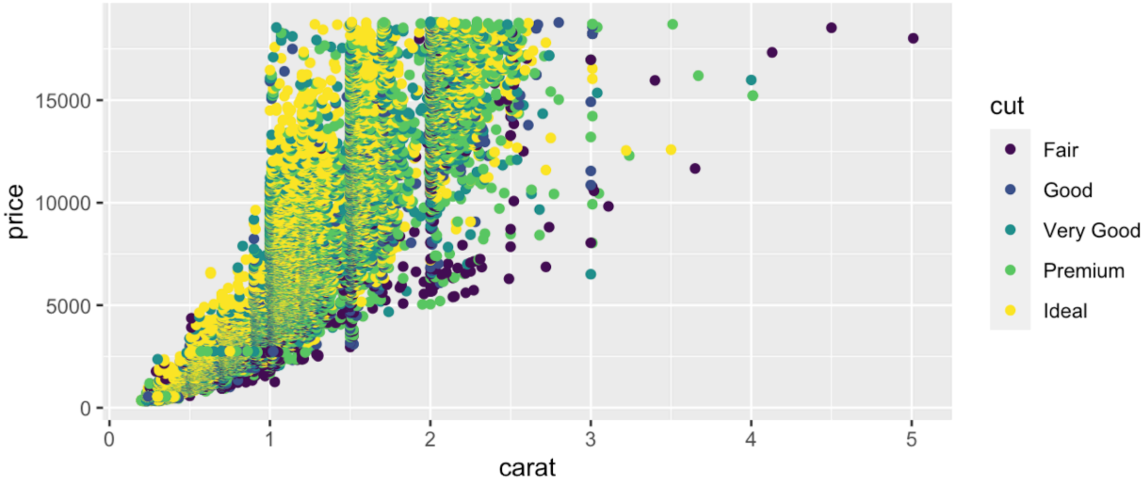
2.3.3 Facets

So far we've looked at

1. how one (or more) variables are distributed - barchart or histogram
2. how two variables are related - scatterplot, boxplot
3. how two variables are related, conditioned on other variables - color

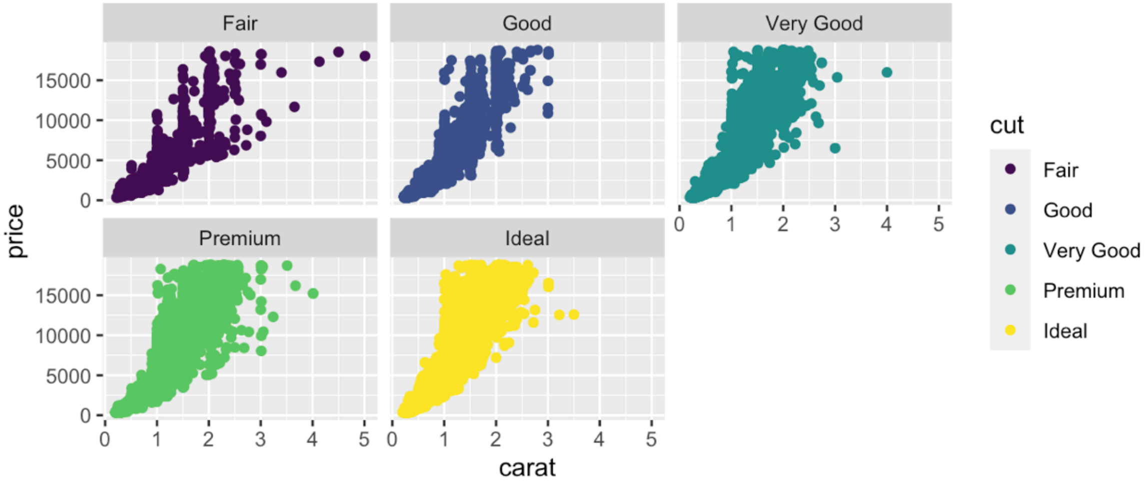
Sometimes color isn't enough to show conditioning because of crowded plots.

```
ggplot(data = diamonds, mapping = aes(carat, price)) +
  geom_point(aes(color = cut))
```



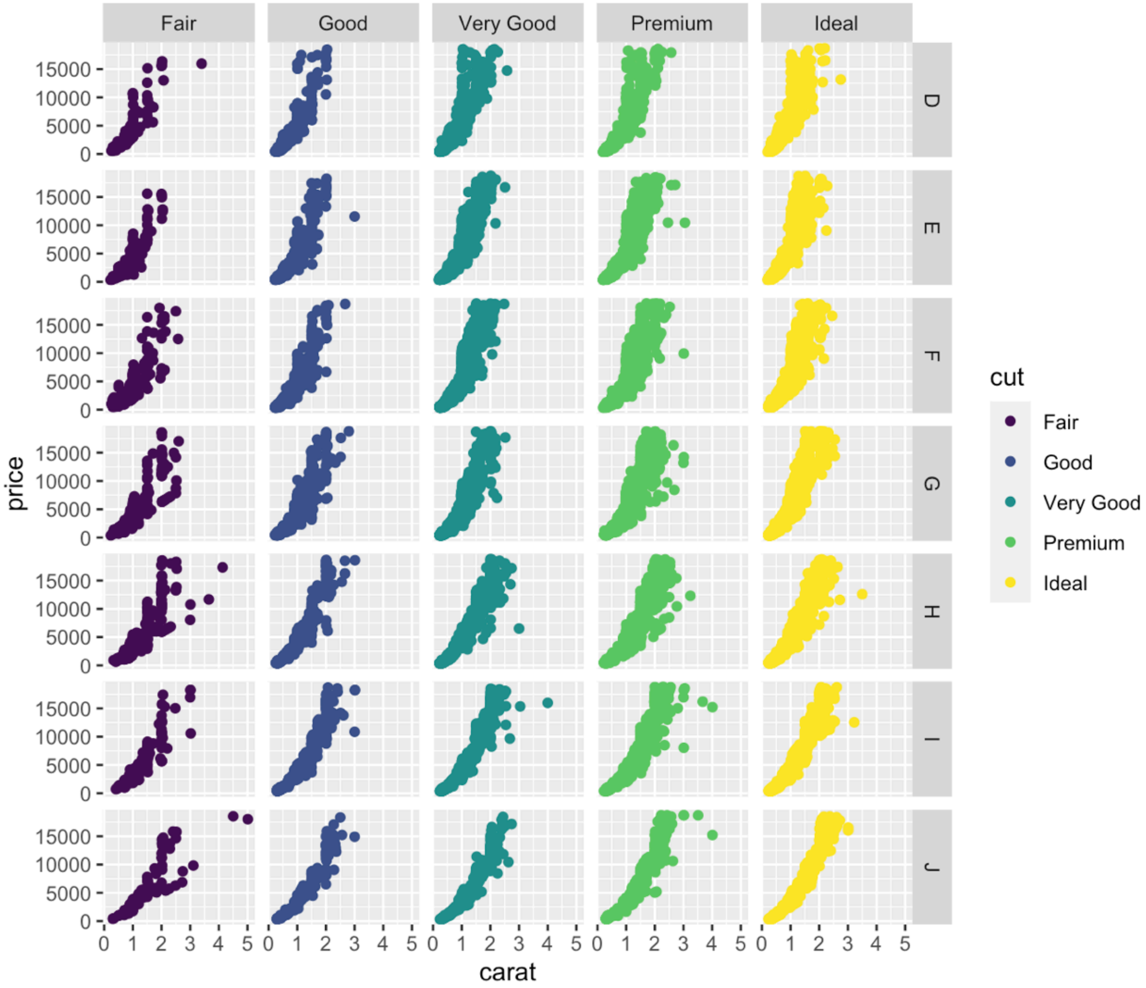
When this is the case, we can *facet* to display plots for different subsets. To do this, we specify row variables ~ column variables (or . for none).

```
ggplot(data = diamonds, mapping = aes(carat, price)) +
  geom_point(aes(color = cut)) +
  facet_wrap(. ~ cut)
```



If instead we have two variables we want to facet by, we can use `facet_grid()`.

```
ggplot(data = diamonds, mapping = aes(carat, price)) +
  geom_point(aes(color = cut)) +
  facet_grid(color ~ cut)
```


Fair
Good
Very Good
Premium
Ideal
D
E
F
G
H
I
J
price
15000
10000
5000
0
carat
0 1 2 3 4 5
cut
● Fair
● Good
● Very Good
● Premium
● Ideal

Your Turn

Using the `mpg` dataset,

1. Make a histogram of `hwy`, faceted by `drv`.
2. Make a scatterplot that incorporates color, shape, size, and facets.
3. BONUS - Color your histograms from `1.` by `cyl`. Did this do what you thought it would? (Look at `fill` and `group` as options instead).

2.4 Additional resources

Documentation and cheat sheets (https://ggplot2.tidyverse.org)

Book website (http://had.co.nz/ggplot2/)

Ch. 3 of R4DS (https://r4ds.had.co.nz/data-visualisation.html)