# ggplot2 demo

## Class 8

Maxwell Austensen

2018-10-28 (updated: 2018-10-28)

# Plotting with `ggplot2` in **R**

Here is a sample dataset we'll be using for some examples.

```
mtcars
```

```
## # A tibble: 32 x 11
##      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##  * <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  21       6  160    110  3.9   2.62  16.5     0     1     4     4
##  2  21       6  160    110  3.9   2.88  17.0     0     1     4     4
##  3  22.8     4  108     93  3.85  2.32  18.6     1     1     4     1
##  4  21.4     6  258    110  3.08  3.22  19.4     1     0     3     1
##  5  18.7     8  360    175  3.15  3.44  17.0     0     0     3     2
##  6  18.1     6  225    105  2.76  3.46  20.2     1     0     3     1
##  7  14.3     8  360    245  3.21  3.57  15.8     0     0     3     4
##  8  24.4     4  147.    62  3.69  3.19  20       1     0     4     2
##  9  22.8     4  141.    95  3.92  3.15  22.9     1     0     4     2
## 10  19.2     6  168.   123  3.92  3.44  18.3     1     0     4     4
## # ... with 22 more rows
```

```
ggplot(data = mtcars)
```

Every plot begins with the function `ggplot()` and providing the dataframe you want to use.

```
ggplot(data = mtcars) +
  aes(x = mpg, y = disp)
```

The second part of every plot is to specify the mapping of columns form the dataframe to aesthetic elements of the plot. This is done with the function `aes()`. Every function call for a ggplot2 plot is added together with `+`.

In the `aes()` function some of the most common aesthetic elements are:

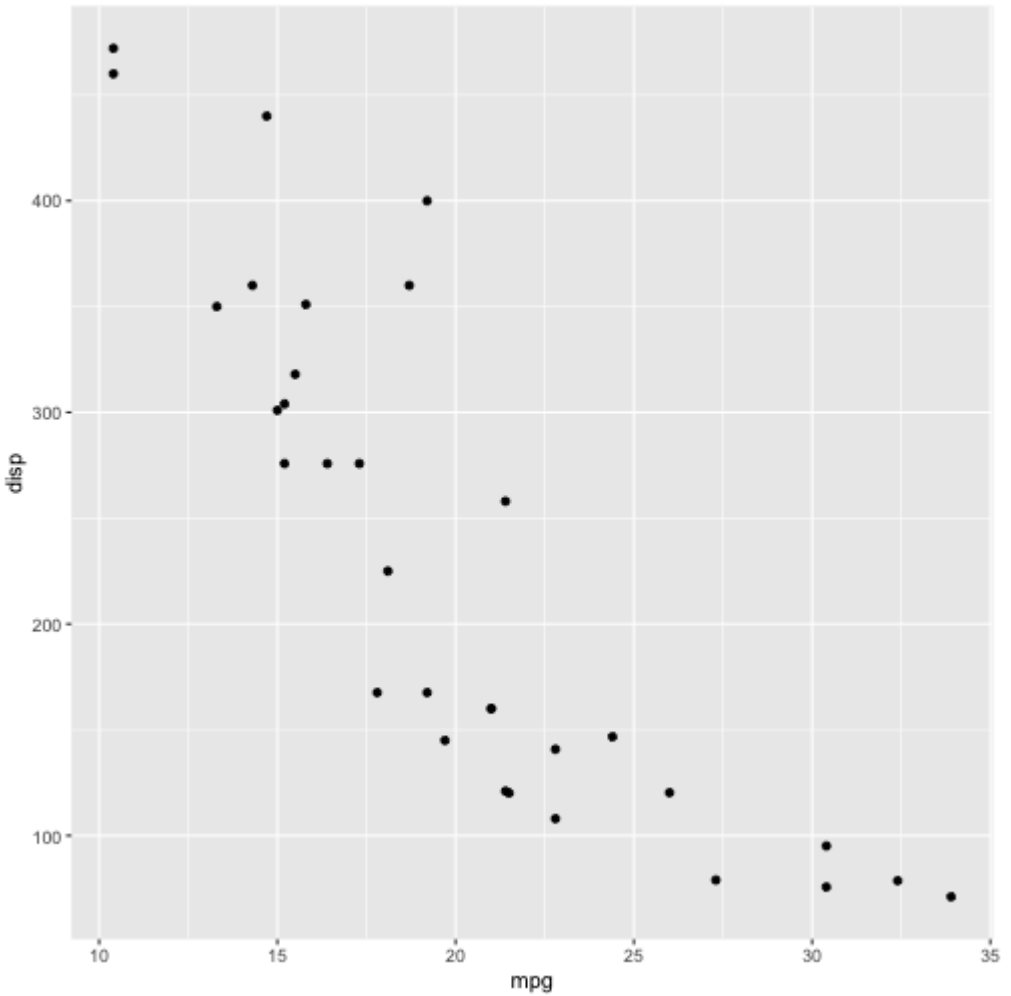
- `x` for the x-axis
- `y` for the y-axis
- `color` for the color of data elements (lines, points, and outline of shapes)
- `fill` for the fill color of shapes
- `label` for data labels
- `group` for separate data elements by some grouping variable
- `shape` for the shape of data points
- `linetype` for the style

```
ggplot(data = mtcars) +
  aes(x = mpg, y = disp) +
  geom_point()
```

The third main element of every plot is the geometries, which determine how the data is displayed on the plot. The geometries are determined by the `geom_*()` functions. Some examples are:

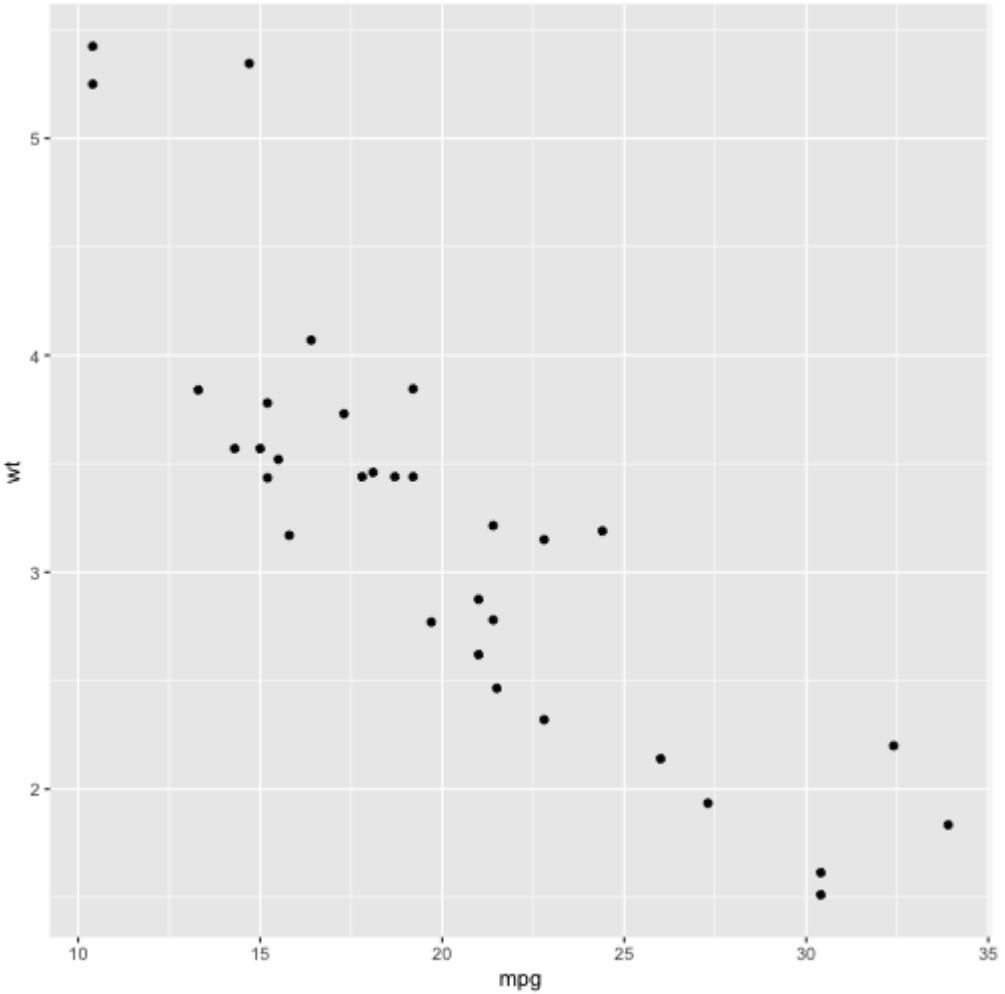
- `geom_point()` for scatter plot
- `geom_line()` for line graphs
- `geom_bar()` for bar graphs, usually representing the number of rows by some grouping column
- `geom_col()` for bar graphs where the height of the bars is specified with a column of the dataframe
- `geom_histogram()` for histograms
- `geom_sf()` for a type of spatial dataframe called "simple features" for creating maps, which we'll cover later
- and many more!

```
ggplot(data = mtcars) +
  aes(x = mpg, y = disp) +
  geom_point()
```

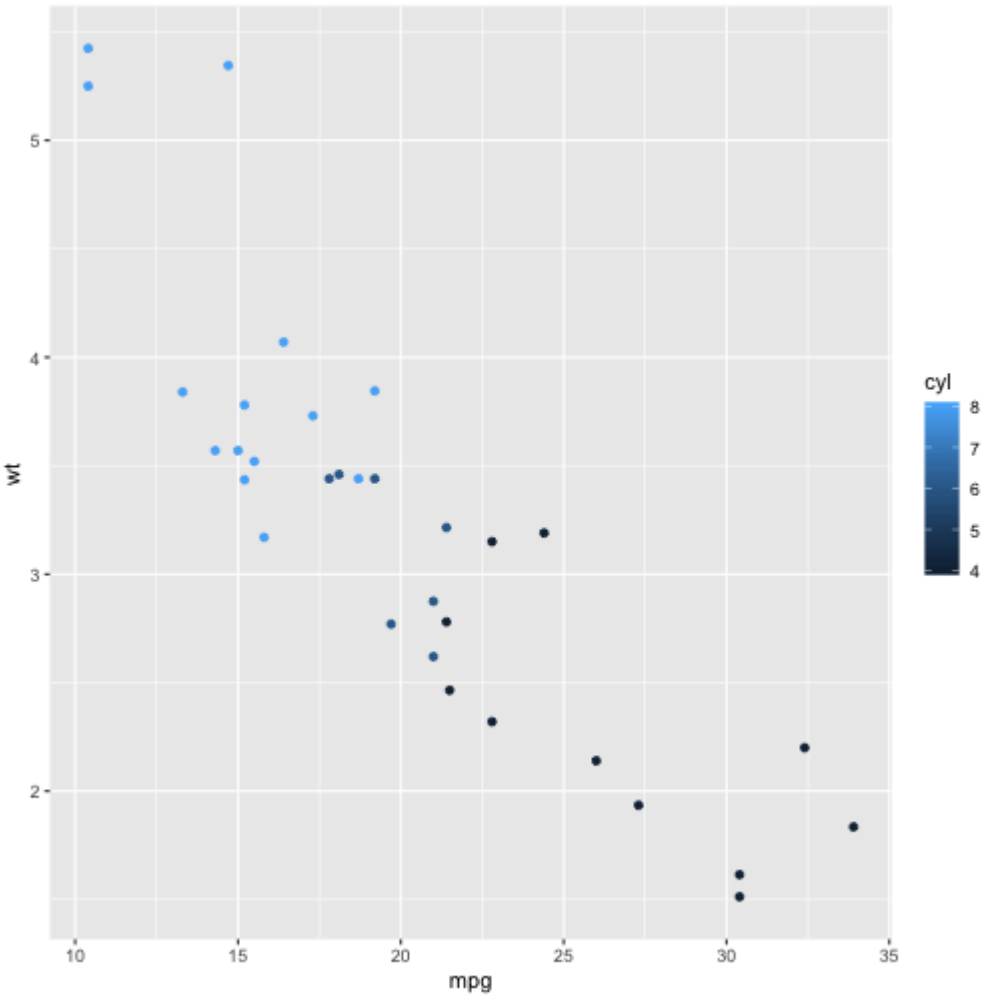


# Some examples

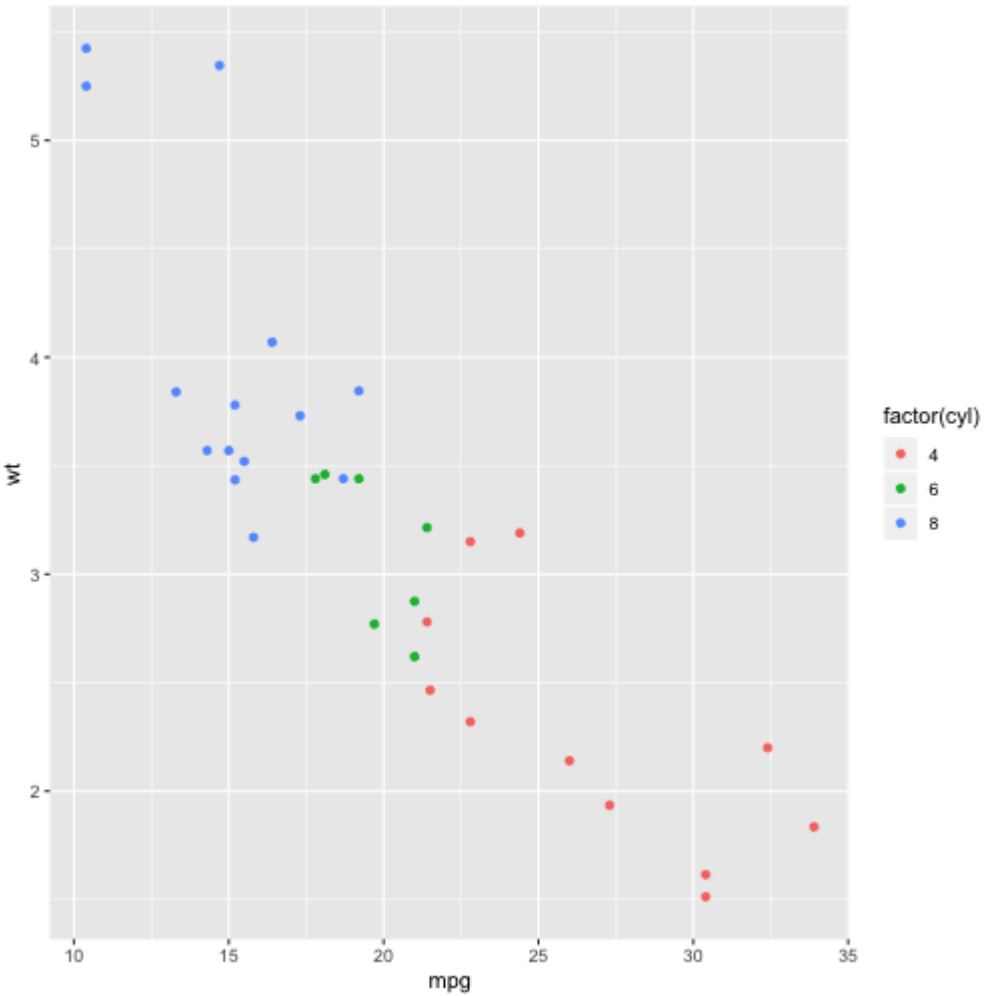
```
ggplot(mtcars) +
  aes(x = mpg, y = wt) +
  geom_point()
```



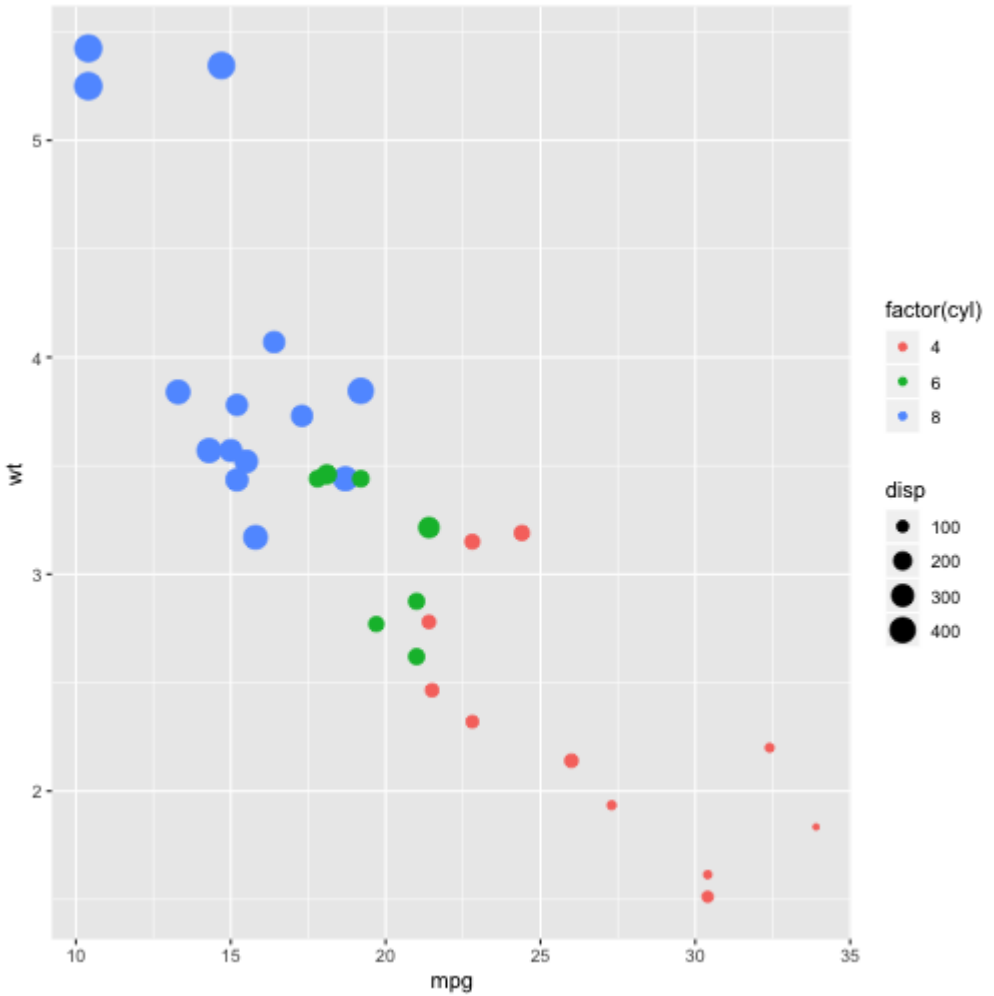
```
ggplot(mtcars) +
  aes(x = mpg, y = wt, color = cyl) +
  geom_point()
```



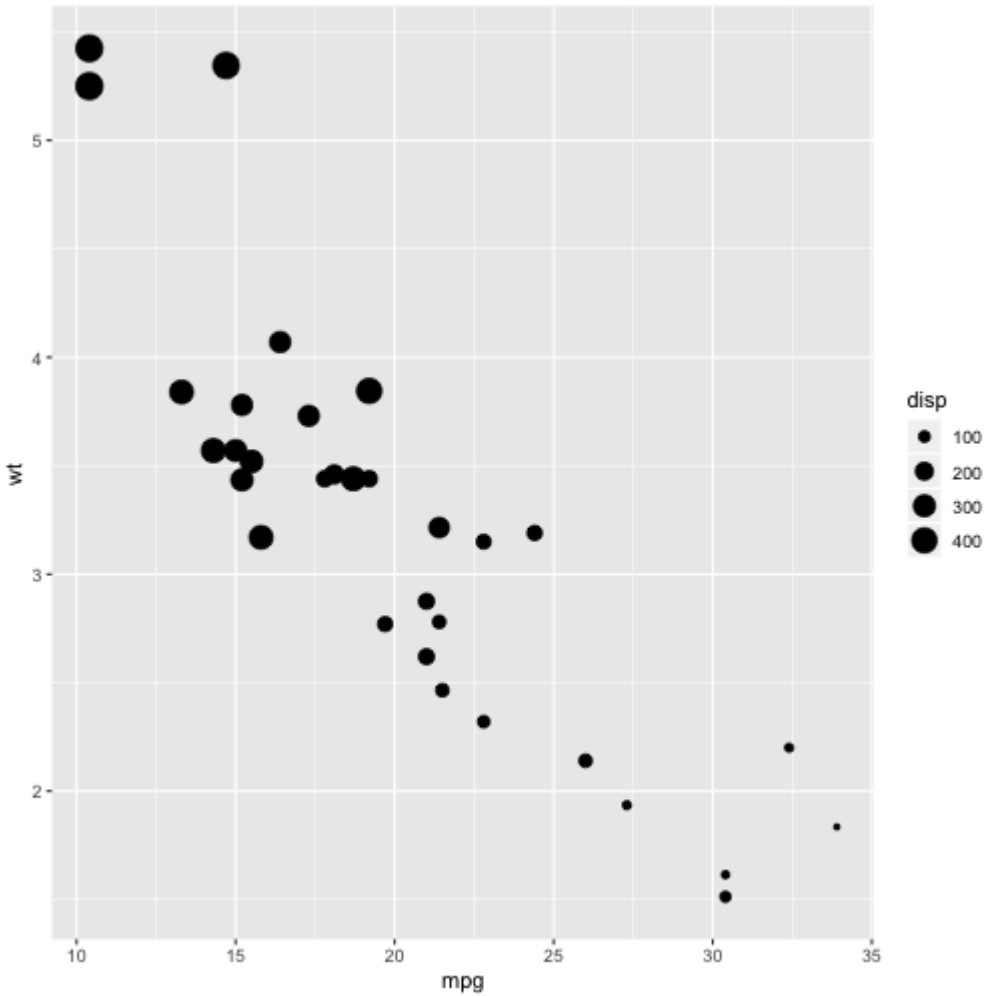
```
ggplot(mtcars) +
  aes(x = mpg, y = wt, color = factor(cyl)) +
  geom_point()
```



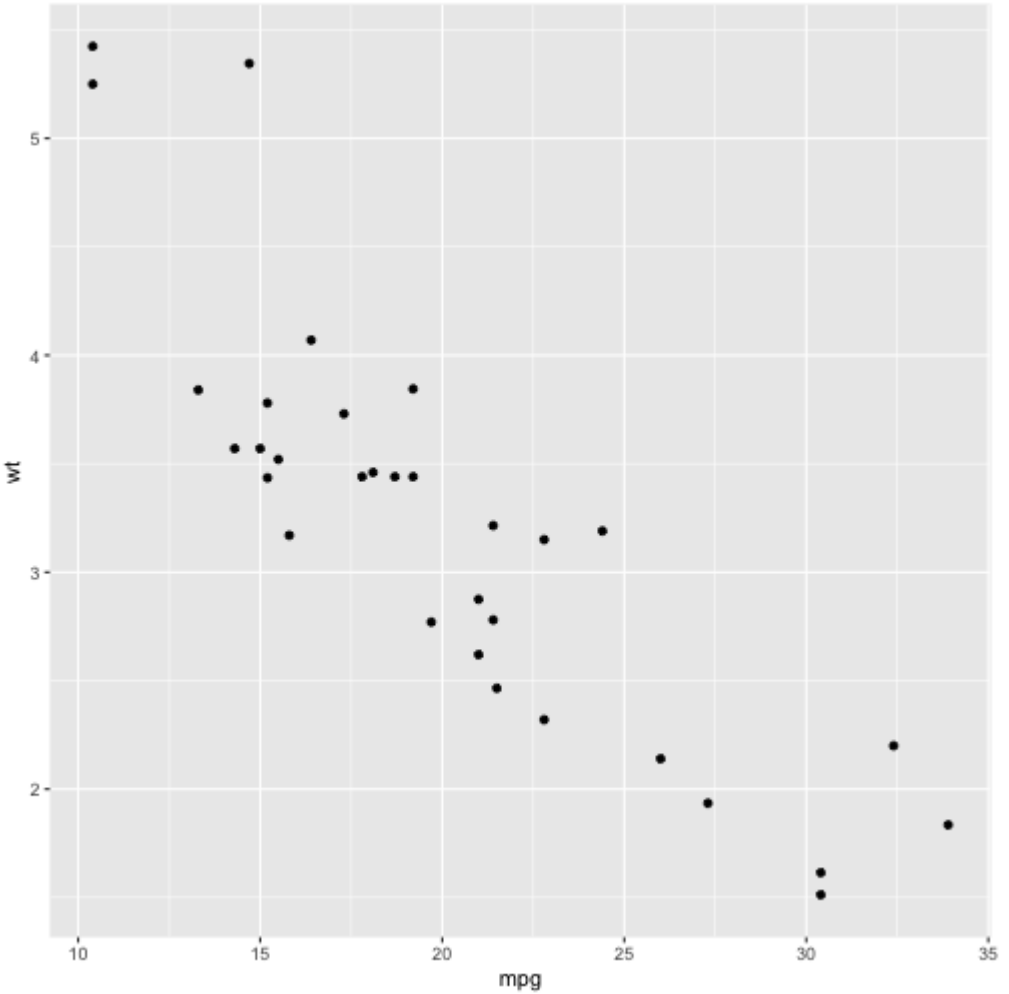
```
ggplot(mtcars) +
  aes(x = mpg, y = wt, color = factor(cyl), size = disp) +
  geom_point()
```



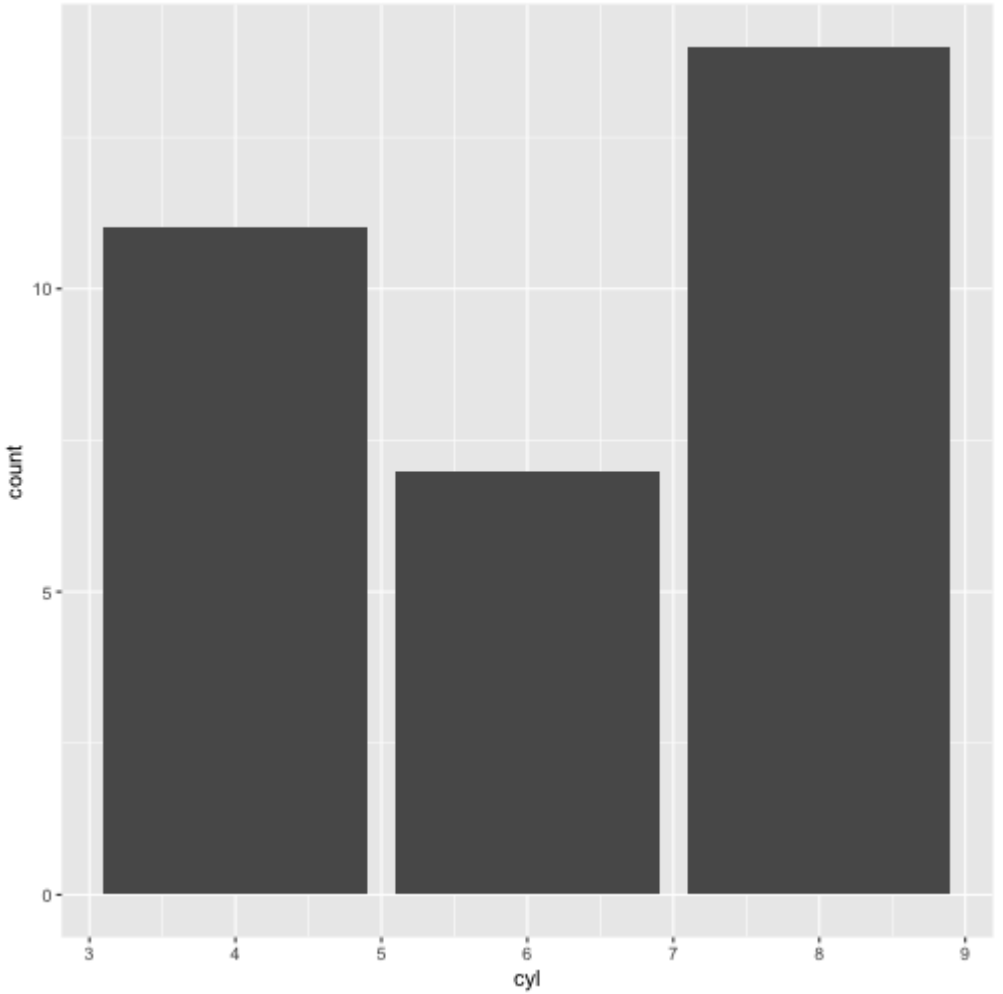
```
ggplot(mtcars) +
  aes(x = mpg, y = wt, size = disp) +
  geom_point()
```



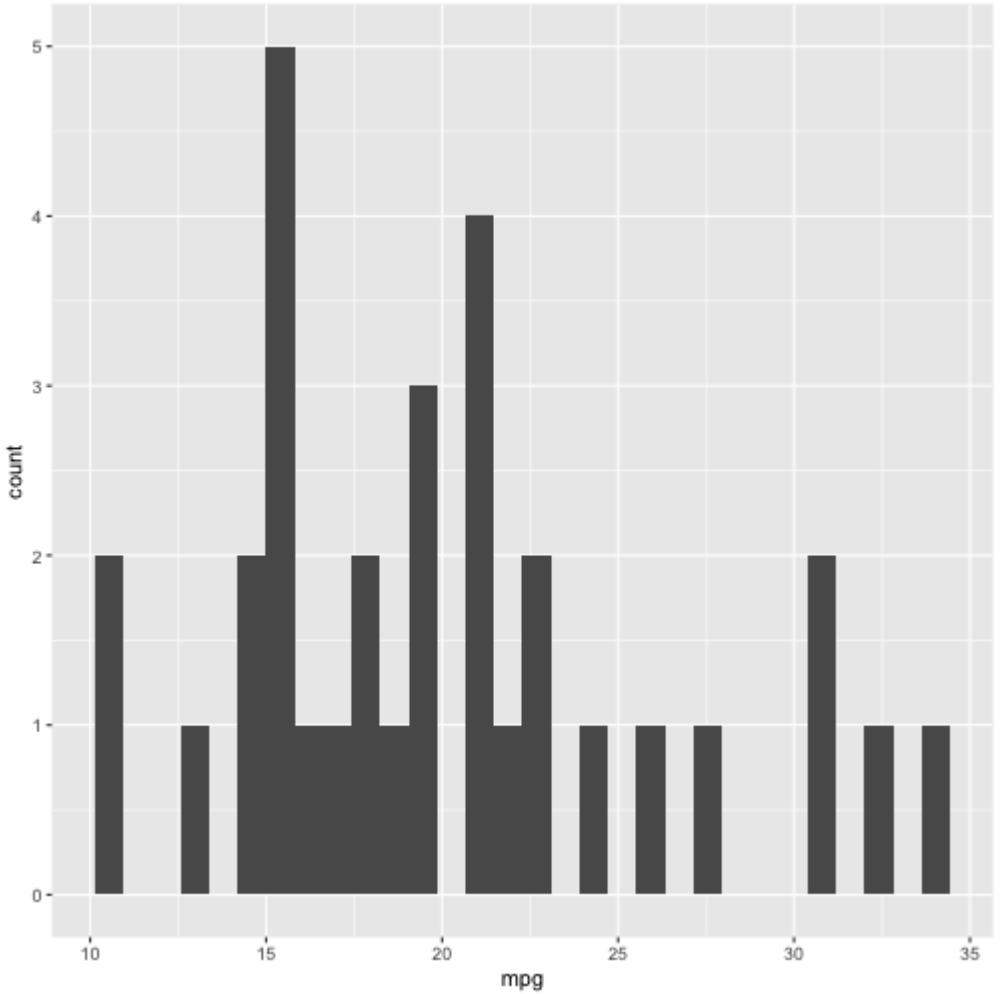
```
ggplot(mtcars) +
  aes(x = mpg, y = wt) +
  geom_point()
```



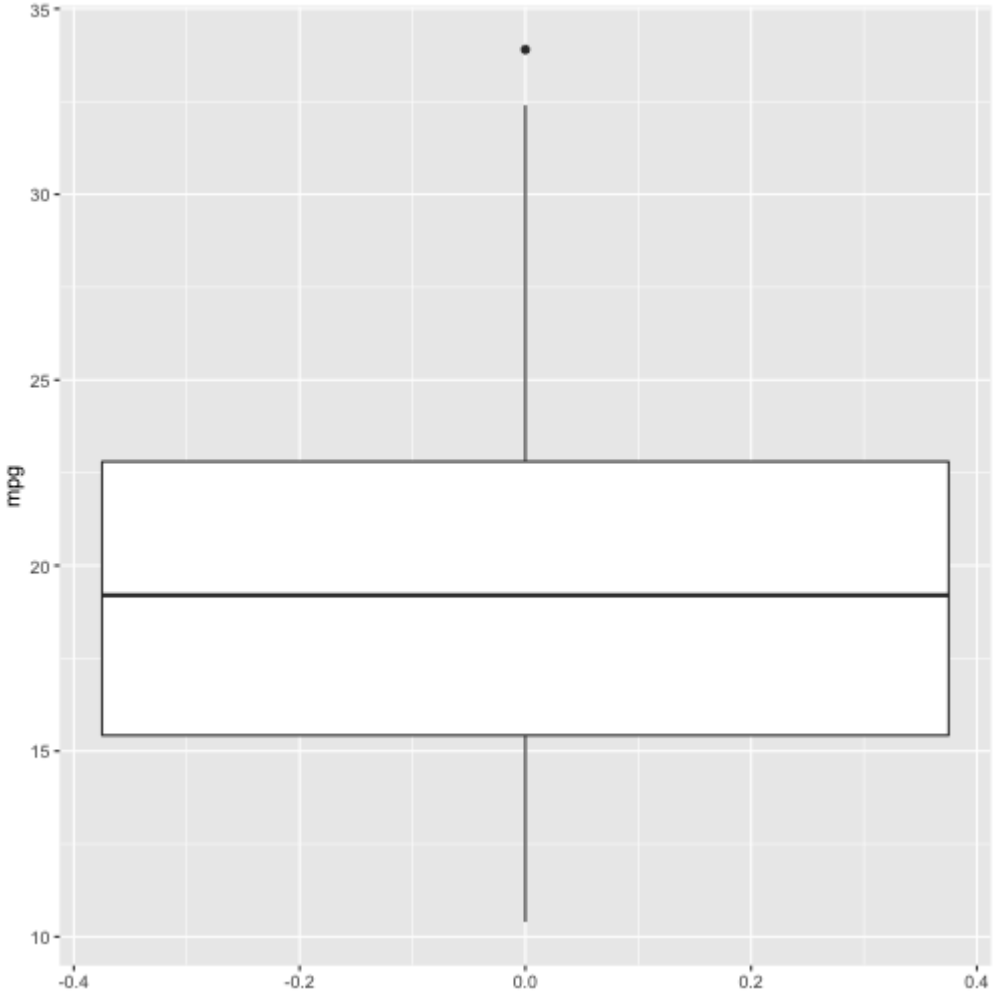
```
ggplot(mtcars) +
  aes(x = cyl) +
  geom_bar()
```



```
ggplot(mtcars) +
  aes(x = mpg) +
  geom_histogram()
```

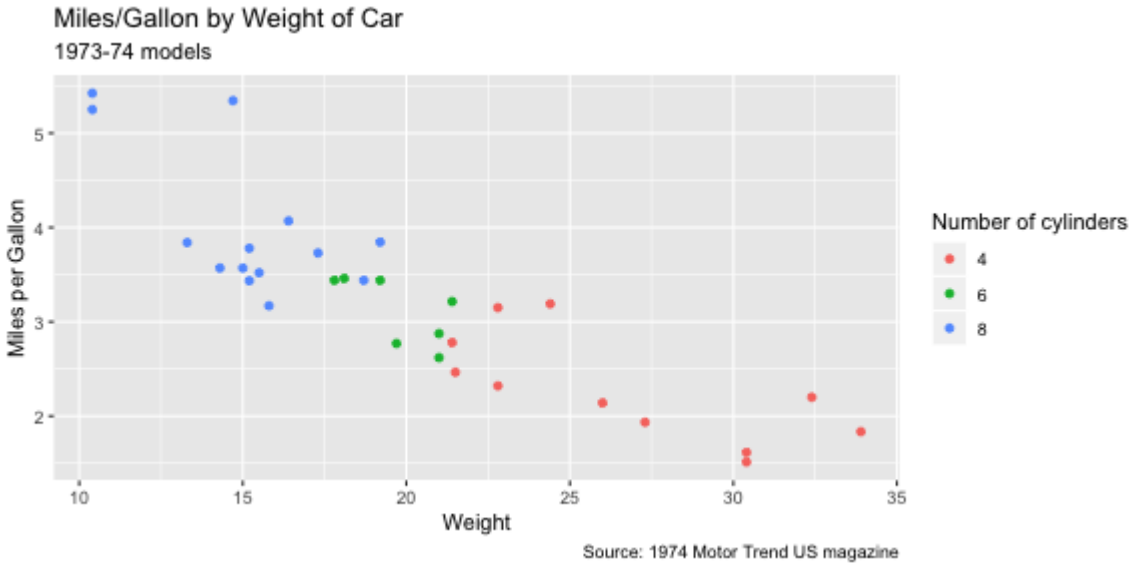


```
ggplot(mtcars) +
  aes(y = mpg) +
  geom_boxplot()
```



You can apply titles and labels to you plot using `labs()`:

```
ggplot(mtcars) +
  aes(x = mpg, y = wt, color = factor(cyl)) +
  geom_point() +
  labs(
    title = "Miles/Gallon by Weight of Car",
    subtitle = "1973-74 models",
    x = "Weight",
    y = "Miles per Gallon",
    color = "Number of cylinders",
    caption = "Source: 1974 Motor Trend US magazine"
  )
```



# Live Demo!