Class 5: ggplot

Dhruv

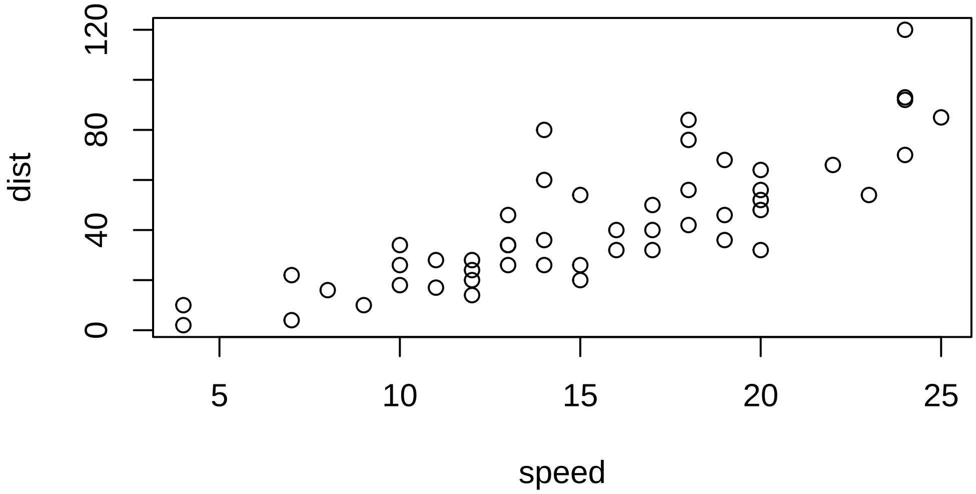
Making figures and graphs with R **"base" R** is the default program to make plots. This can be accessed using the `plot()` function.

```
cars
```

```
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
11    11   28
12    12   14
13    12   20
14    12   24
15    12   28
16    13   26
17    13   34
18    13   34
19    13   46
20    14   26
21    14   36
22    14   60
23    14   80
24    15   20
25    15   26
26    15   54
```

```
27     16   32
28     16   40
29     17   32
30     17   40
31     17   50
32     18   42
33     18   56
34     18   76
35     18   84
36     19   36
37     19   46
38     19   68
39     20   32
40     20   48
41     20   52
42     20   56
43     20   64
44     22   66
45     23   54
46     24   70
47     24   92
48     24   93
49     24  120
50     25   85
```

```
plot(cars)
```

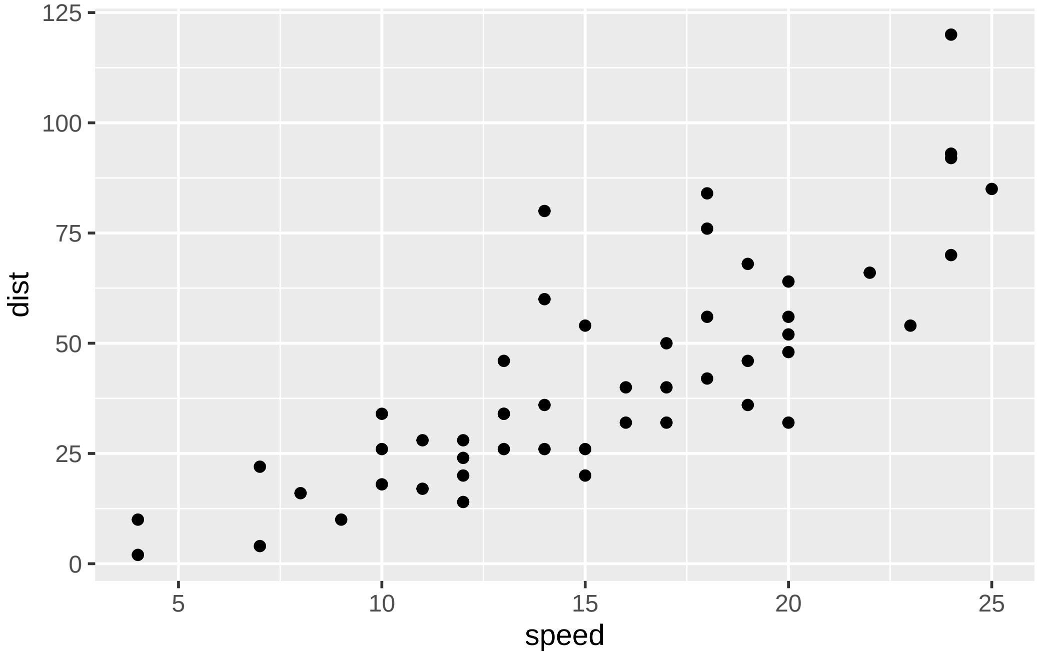


Popular package to do data visualization is **ggplot2**

ggplot(cars) - this wont work

Before using an add-on package we must first install it: `install.packages("ggplot2")`. Next, this must be loaded.

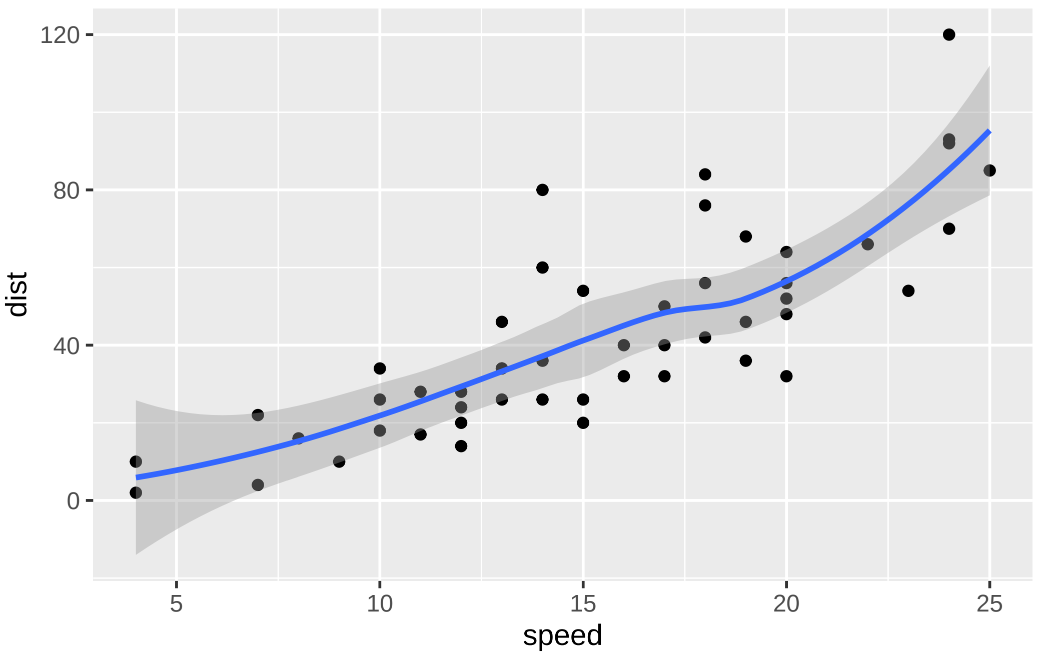
```
library(ggplot2)
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



"base" r is shorter for simpler graphs, and **ggplot** works better for complex graphs. Let's try to make the above plot more complex now

```
library(ggplot2)
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth()
```

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



Every ggplot has at minimum 3 layers

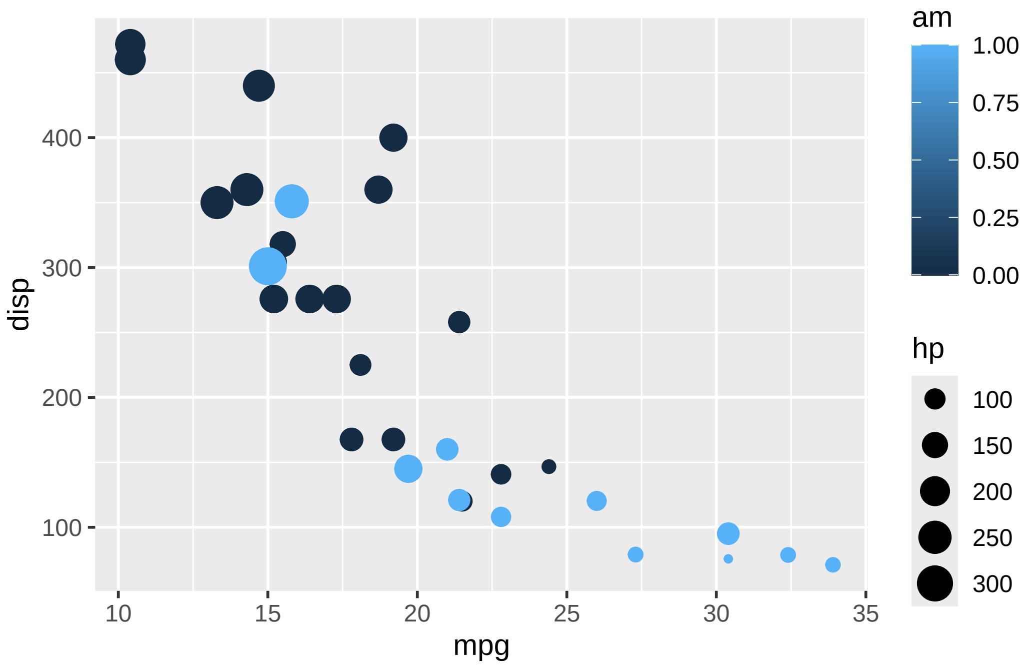
- **data** (data.frame with things you want to plot)
- **aes**thetics**
- **geoms**s - quite a few including `geom_line()`, `geom_col()`, `geom_point()`

```
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

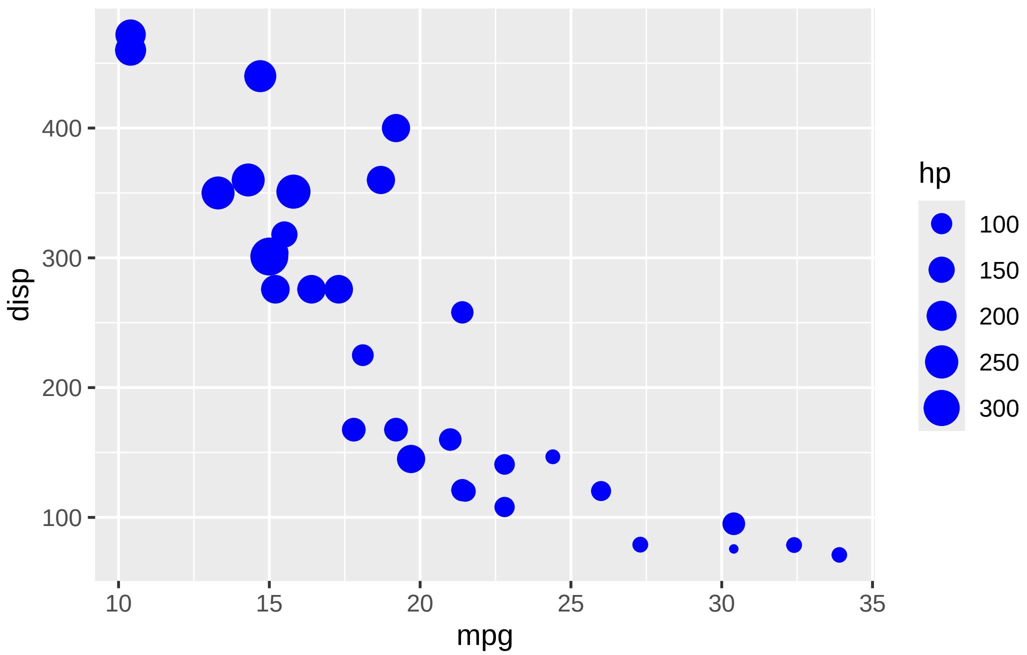
ggplot

```
ggplot(mtcars) +
  aes(x = mpg, y = disp, size = hp, col = am) +
  geom_point()
```



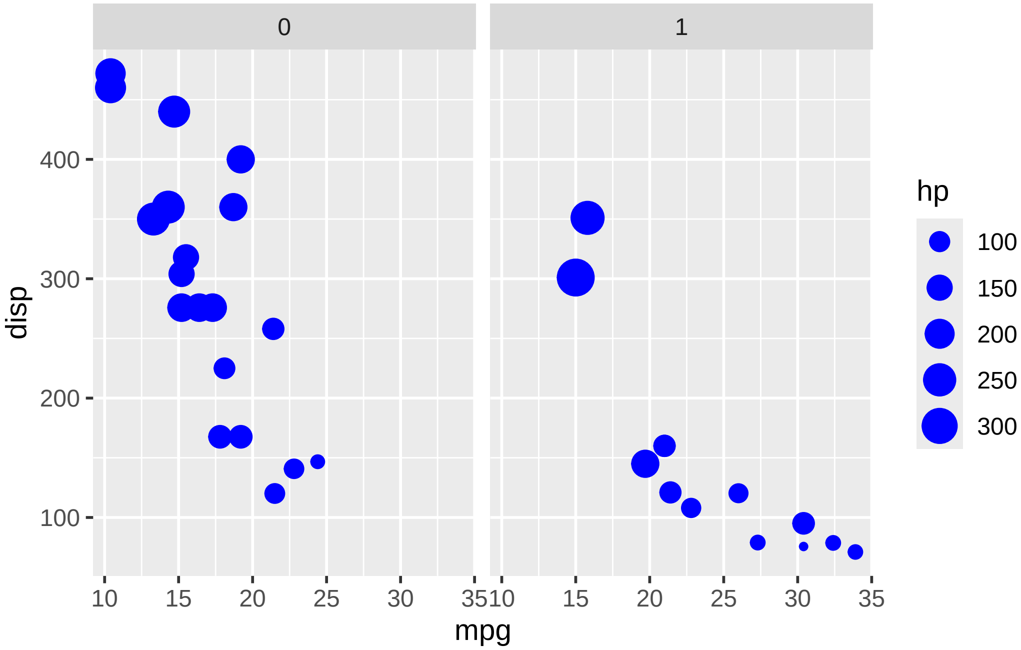
Now color all points blue

```
ggplot(mtcars) +
  aes(x = mpg, y = disp, size = hp) +
  geom_point(col = "blue")
```



can make the old plot faceted

```r
ggplot(mtcars) +
  aes(x = mpg, y = disp, size = hp) +
  geom_point(col = "blue") +
  facet_wrap("am")
```



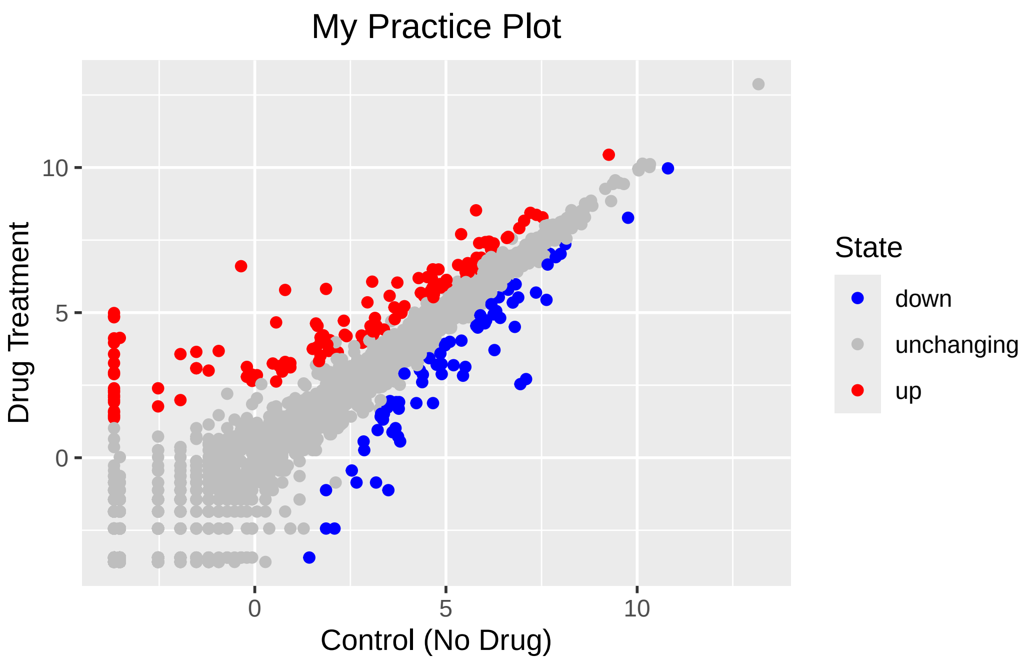
Now we will work through the lab sheet

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes)
```

```
        Gene Condition1 Condition2      State
1      A4GNT -3.6808610 -3.4401355 unchanging
2       AAAS  4.5479580  4.3864126 unchanging
3      AASDH  3.7190695  3.4787276 unchanging
4       AATF  5.0784720  5.0151916 unchanging
5       AATK  0.4711421  0.5598642 unchanging
6 AB015752.4 -3.6808610 -3.5921390 unchanging
```

```
p <- ggplot(genes) + aes(x = Condition1, y = Condition2, col = State) +
  geom_point()
```

```
p + scale_color_manual(values = c("blue", "grey","red")) + labs(title = "My Practice Plot",
```



```
nrow(genes)
```

```
[1] 5196
```

There are 5196 genes in this dataset

The `table()` function is useful to look at how many of each entries are there. this is compared to `unique()` which tells you the actual unique variables but not how many of each.

```
table(genes$State)
```

```

      down unchanging         up 
        72       4997        127 
```

What fraction are up, down, or unchanging. Total genes = `nrow(genes)`. Can divide table by this.

```
colnames(genes)
```

```
[1] "Gene"       "Condition1" "Condition2" "State"
```

```
ncol(genes)
```

```
[1] 4
```

```
nrow(genes)
```

```
[1] 5196
```

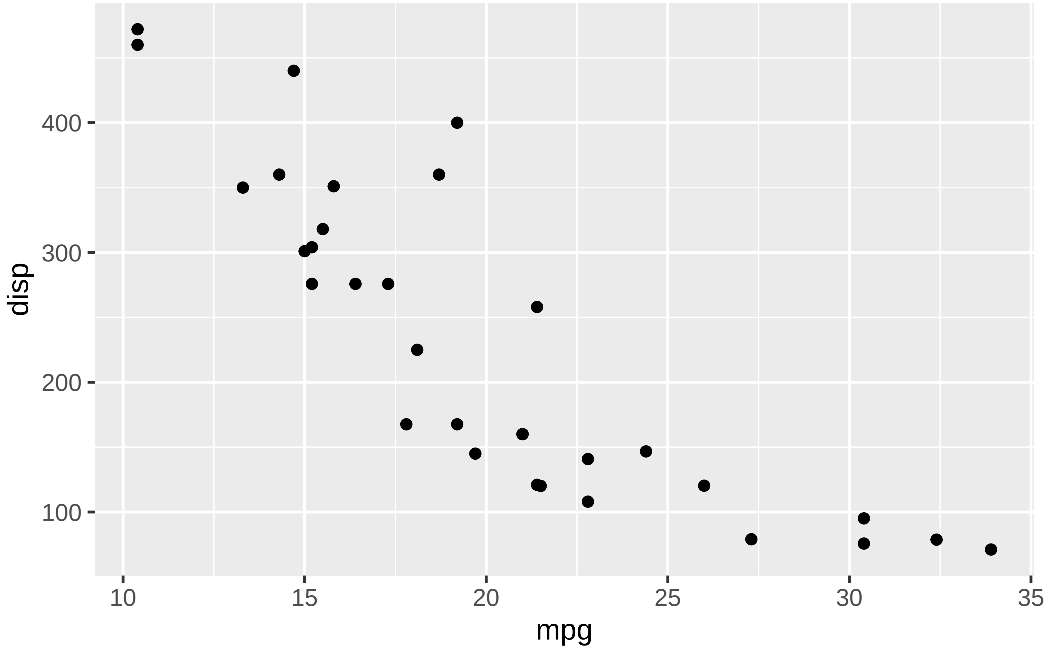
```
round(table(genes$State)/nrow(genes), 3)*100
```

```

       down unchanging         up 
        1.4       96.2        2.4 
```

Key points: Saving plots with **ggsave()** “types” of plots with ‘geoms_ Multi-plot layout with **patchwork** package

```
ggplot(mtcars) +
  aes(mpg, disp) +
  geom_point()
```



```r
ggsave("myplot.pdf")
```

```
Saving 5.5 x 3.5 in image
```