Class 05: Data Visualization with GGPLOT

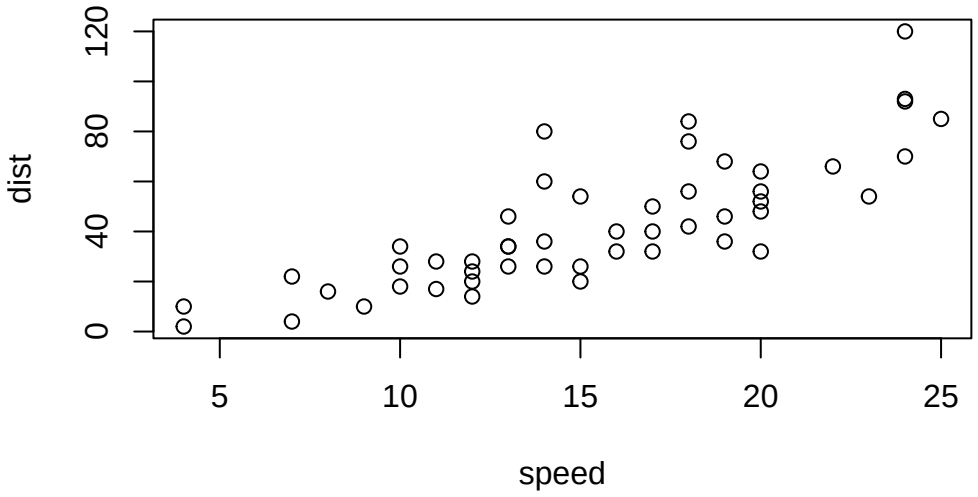
Max Gruber

4/19/23

Base R plotting

We are going to start by generating the plot of class 04. This code is plotting the **cars** dataset.

```
plot(cars)
```



Ggplot2

First, we need to install the package. We do this by using the `install.packages` command.

```
#install.packages('ggplot2')
```

After that, we need to load the package.

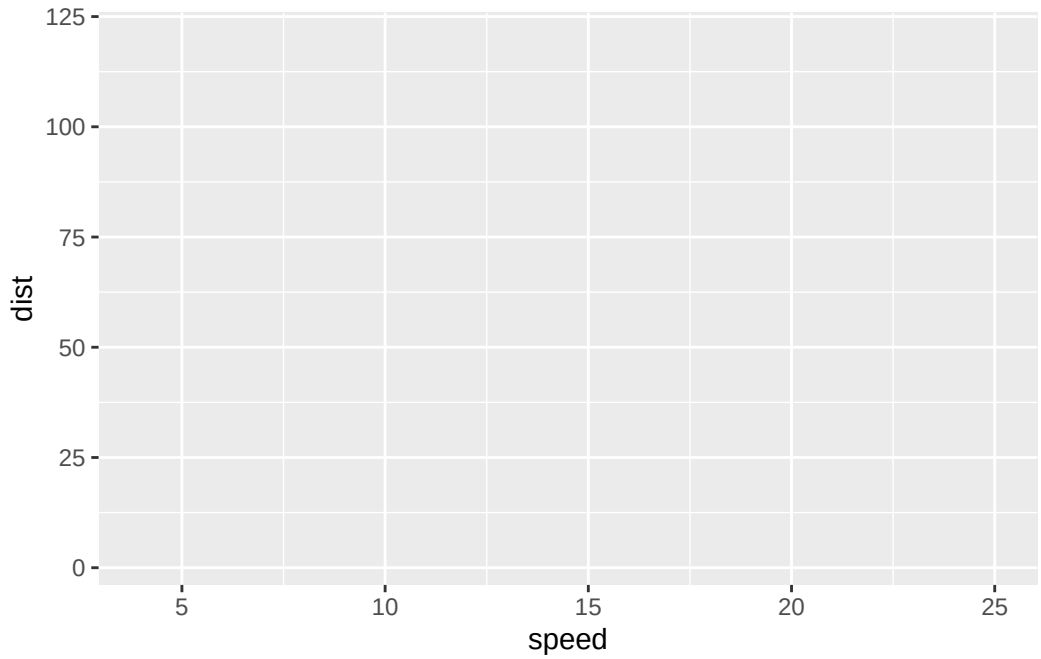
```
library(ggplot2)
```

We are going to build the plot of the cars dataframe by using ggplot2.

```
ggplot(data=cars)
```

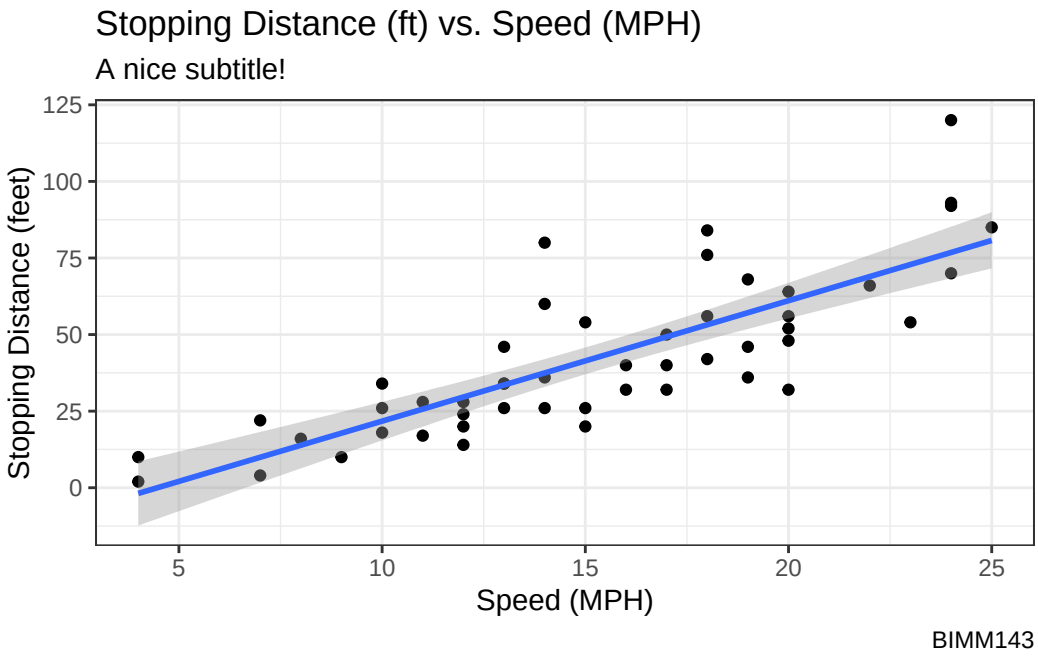


```
ggplot(data=cars) +
  aes(x=speed, y=dist)
```




```
ggplot(data=cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(title = 'Stopping Distance (ft) vs. Speed (MPH)',
       subtitle = 'A nice subtitle!',
       caption = 'BIMM143',
       x = 'Speed (MPH)',
       y = 'Stopping Distance (feet)') +
  theme_bw()
```

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



Q1. For which phases is data visualization important in our scientific workflows?

For Exploratory data analysis, detection of outliers, etc.

Q2. True or False? The ggplot2 package comes already installed with R?

False

Plotting gene expression data

Loading the data from the URL.

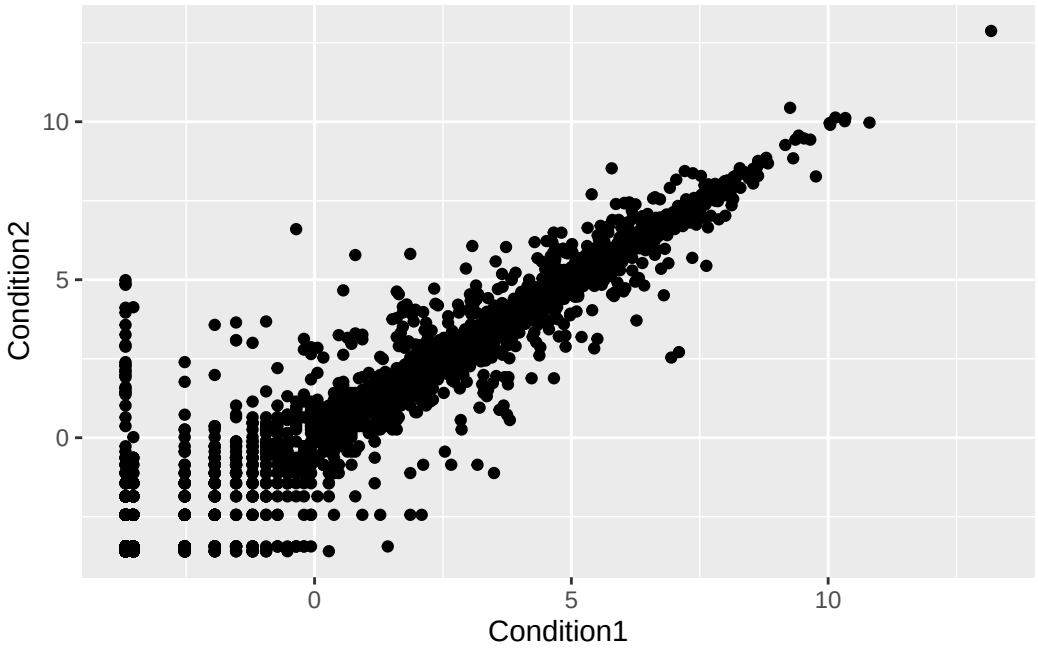
```
#Load data
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes)
```

```
        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
```

Initial ggplot

```
ggplot(data = genes) +
  aes(x = Condition1, y = Condition2) +
  geom_point()
```



Identifying the parts of the dataframe

```
nrow(genes)
```

```
[1] 5196
```

```
ncol(genes)
```

```
[1] 4
```

```
colnames(genes)
```

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

```
table(genes[,'State'])
```

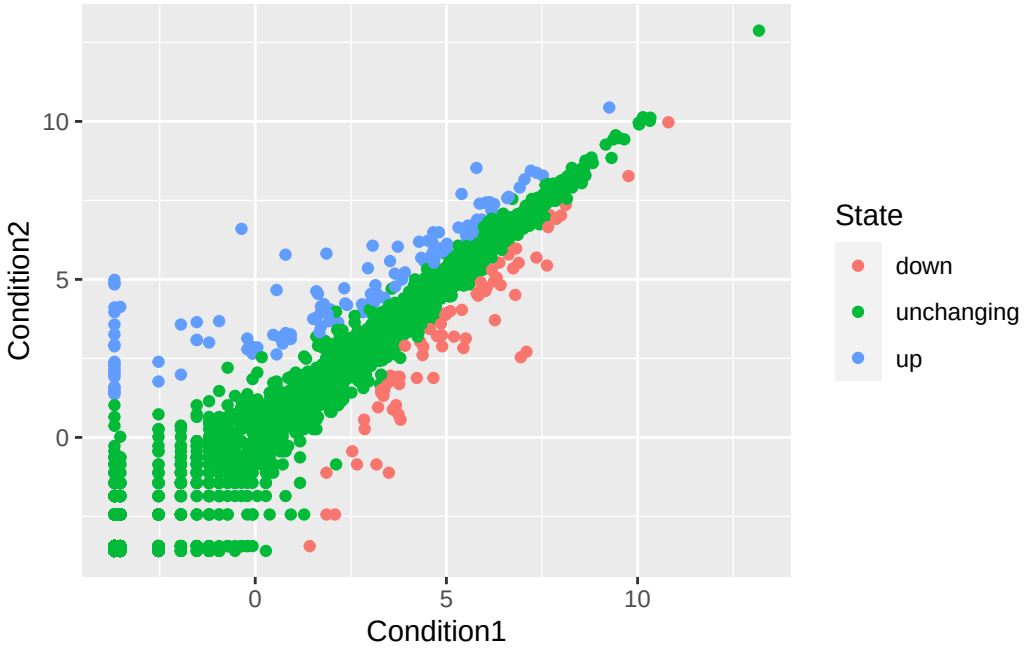
```
    down unchanging          up
      72       4997         127
```

```
round( table(genes$State)/nrow(genes) * 100, 2 )
```

```
    down unchanging          up
    1.39      96.17       2.44
```

Adding color to the plot

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



Q3. Use the `nrow()` function to find out how many genes are in this dataset. What is your answer?

5196

Q4. Use the `colnames()` function and the `ncol()` function on the `genes` data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find?

There are 4 columns, with the names: Gene, Condition1, Condition2, and State.

Q5. Use the `table()` function on the `State` column of this data.frame to find out how many 'up' regulated genes there are. What is your answer?

There are 72 genes that are downregulated, 4997 genes that are unchanging, and 127 genes that are upregulated.

Q6. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

2.44%

Let's change the color scheme and add some labels.

Q7. Nice, now add some plot annotations to the `p` object with the `labs()` function so your plot looks like the following:

```
p1 <- ggplot(data = genes) +
  aes(x = Condition1, y = Condition2, col= State) +
  geom_point()

p1 +
  scale_colour_manual(values=c("blue","gray","red")) +
    labs(title="Gene Expresion Changes Upon Drug Treatment",
         x="Control (no drug) ",
         y="Drug Treatment")
```