

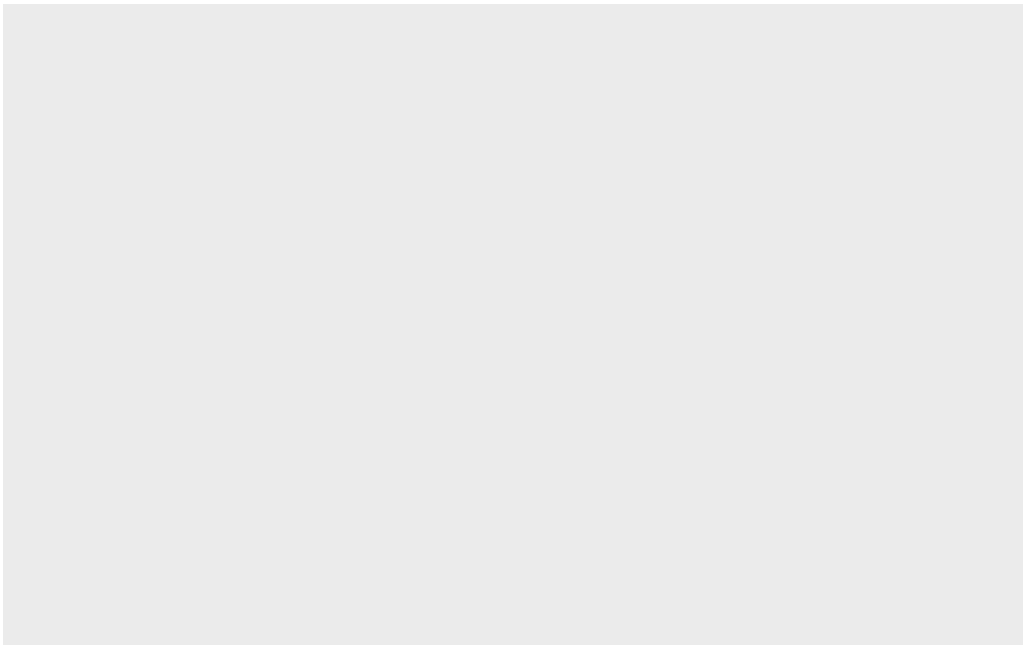
Class 05: Data Visualization with GGPLOT

Morgan Black (PID A14904860)

```
#One time install of package  
#install.packages("ggplot2")
```

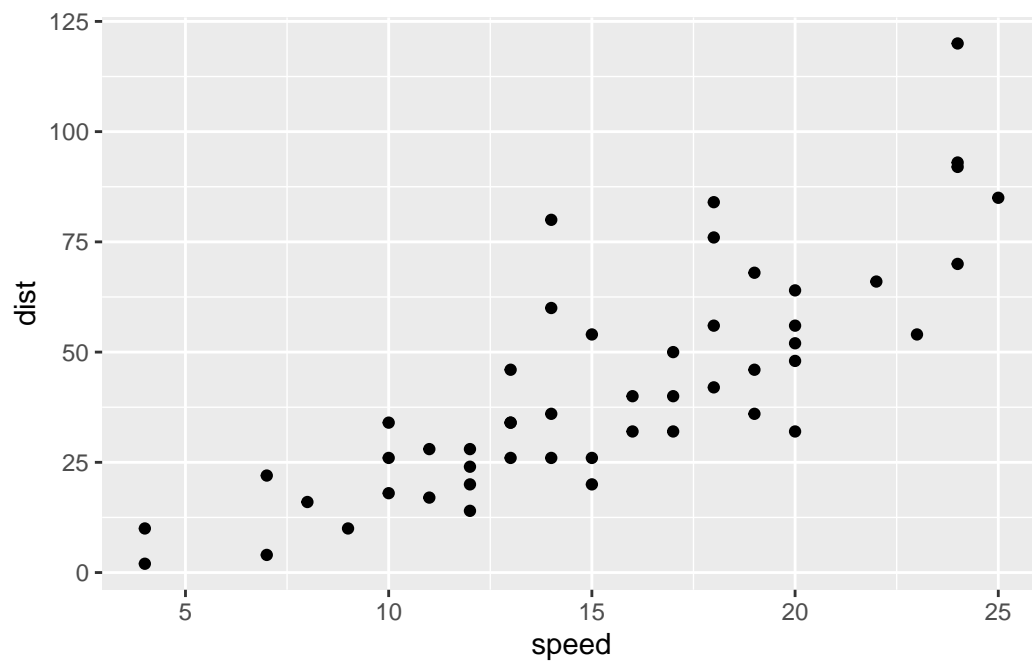
```
#Load the package each session  
library(ggplot2)
```

```
#Creates just a blank plot canvas without any points  
ggplot(cars)
```



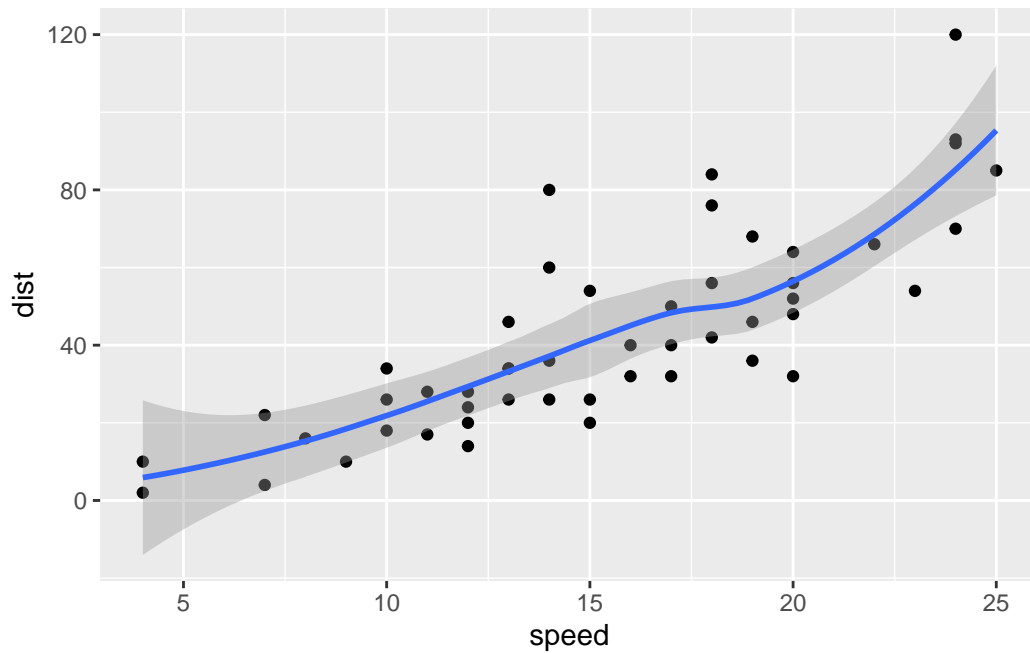
```
#Creates a plot specifying the x and y data from the cars dataset, and  
#plotting them as points on the plot  
ggplot(cars) +
```

```
aes(x=speed, y=dist) +  
geom_point()
```



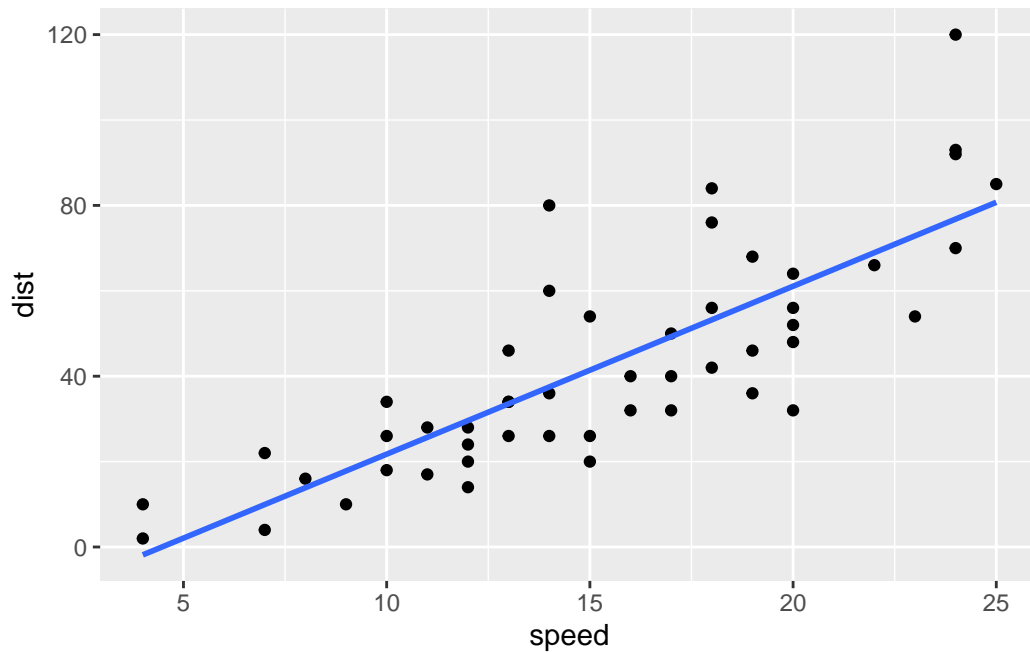
```
#Add a fitted trendline to the point plot to help visualize the relationship  
#between the x and y variables  
ggplot(cars) +  
  aes(x= speed, y=dist) +  
  geom_point() +  
  geom_smooth()
```

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



```
#Add a trendline fitting a linear model without a shaded error region,
#instead of the trendline with a shaded error region from the previous plot
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)
```

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

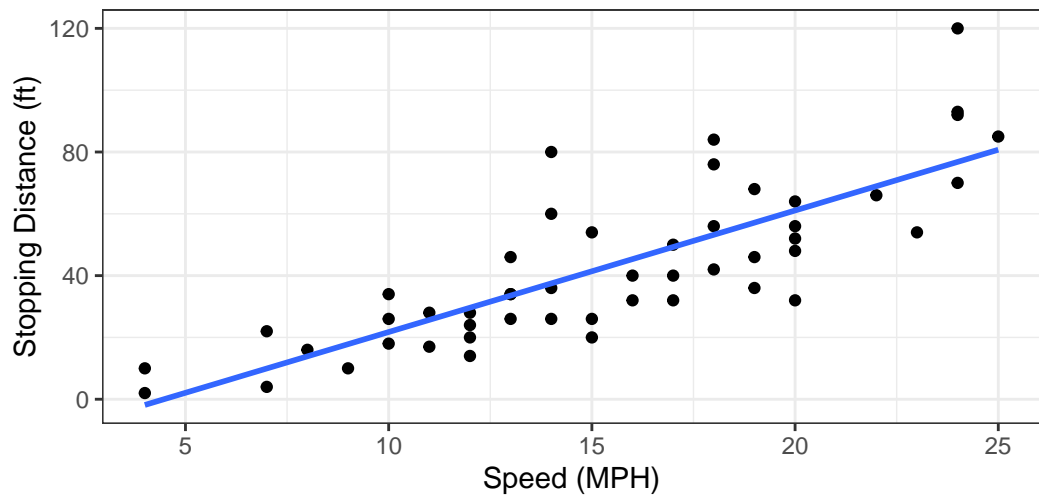


```
#Add labels to the overall plot, x and y axes, a subtitle to explain the plot
#in more detail, and a caption below to note the dataset used to make the plot
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  labs(title="Speed and Stopping Distances of Cars",
       x="Speed (MPH)",
       y="Stopping Distance (ft)",
       subtitle = "Stopping distance of different cars based on their speed
prior to breaking",
       caption="Dataset: 'cars'") +
  geom_smooth(method="lm", se=FALSE) +
  theme_bw()
```

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

Speed and Stopping Distances of Cars

Stopping distance of different cars based on their speed prior to breaking



Dataset: 'cars'

```
#Call and read input dataset of RNAseq data for drug vs no drug treatment. Look
#at just the first 6 lines of 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

```
#How many genes are in this dataset?
nrow(genes)
```

```
[1] 5196
```

```
#What are the column names, and how many columns are there?
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

```
#How many upregulated genes are there?  
table(genes$State)
```

down	unchanging	up
72	4997	127

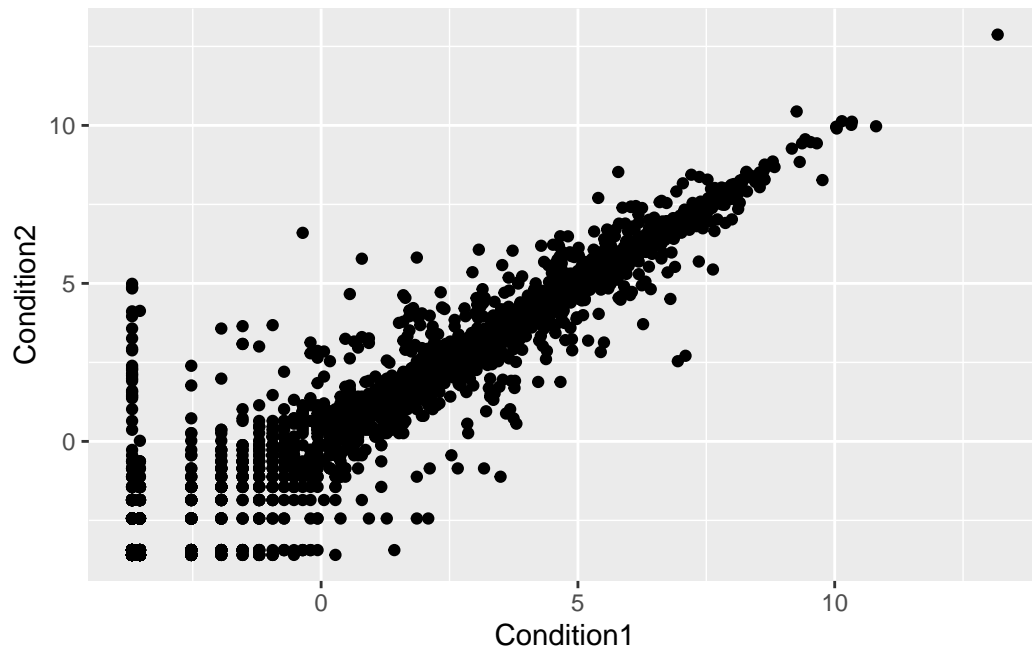
```
#There are 127 upregulated genes here
```

```
#What fraction of total genes is upregulated in this dataset, to 2 sig figs?  
round((table(genes$State) / nrow(genes))*100, 2)
```

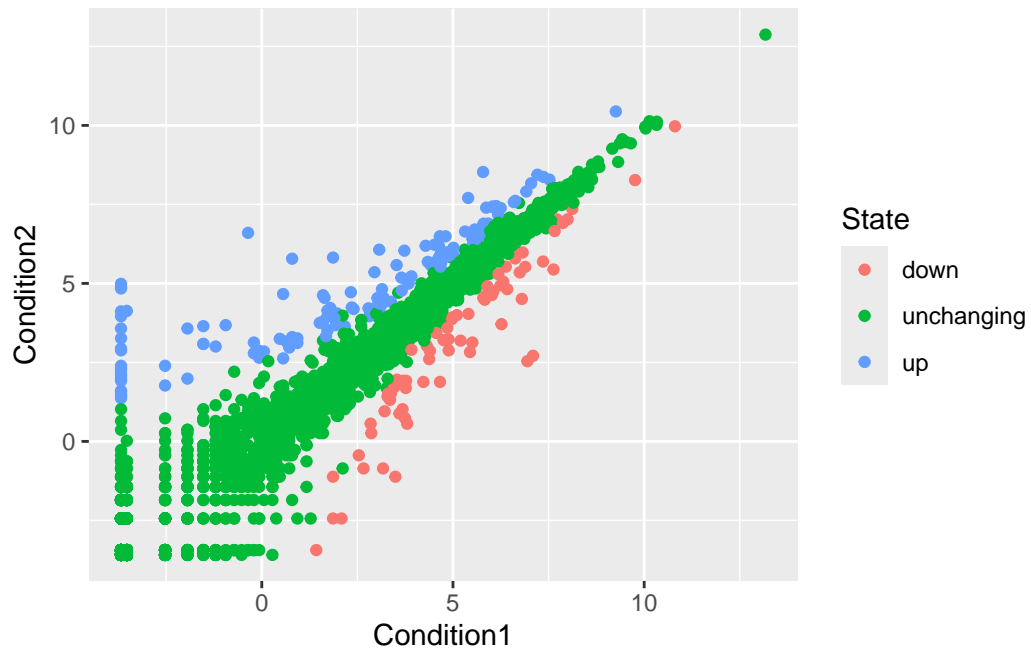
down	unchanging	up
1.39	96.17	2.44

```
#2.44
```

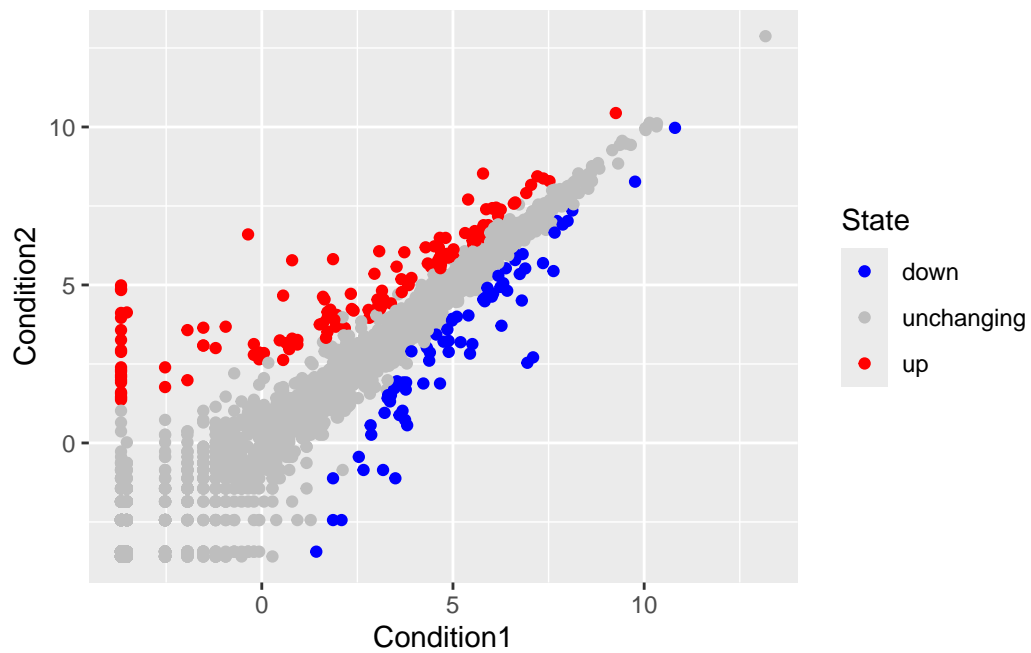
```
#Make a basic scatter plot of this dataset  
ggplot(genes) +  
  aes(x=Condition1, y=Condition2) +  
  geom_point()
```



```
#Add a layer of color to the point plot by mapping the "State" column to  
#point color  
p <- ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) +  
  geom_point()  
p
```



```
#Change the colors of the points
p + scale_colour_manual( values=c("blue","gray","red") )
```




```
#Add labels and annotations
p + scale_colour_manual(values=c("blue","gray","red")) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
       x="Control (no drug) ",
       y="Drug Treatment")
```

