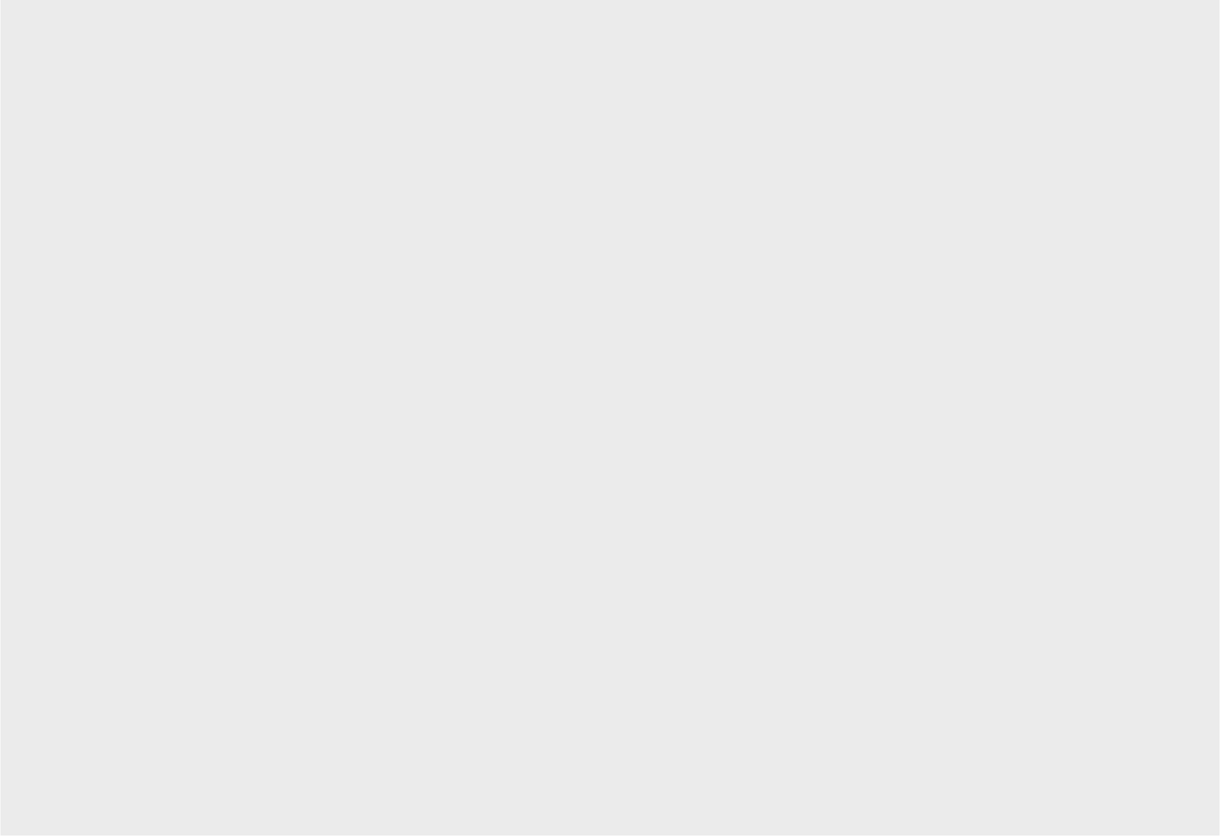
Class 05 Data Analysis and Visualization

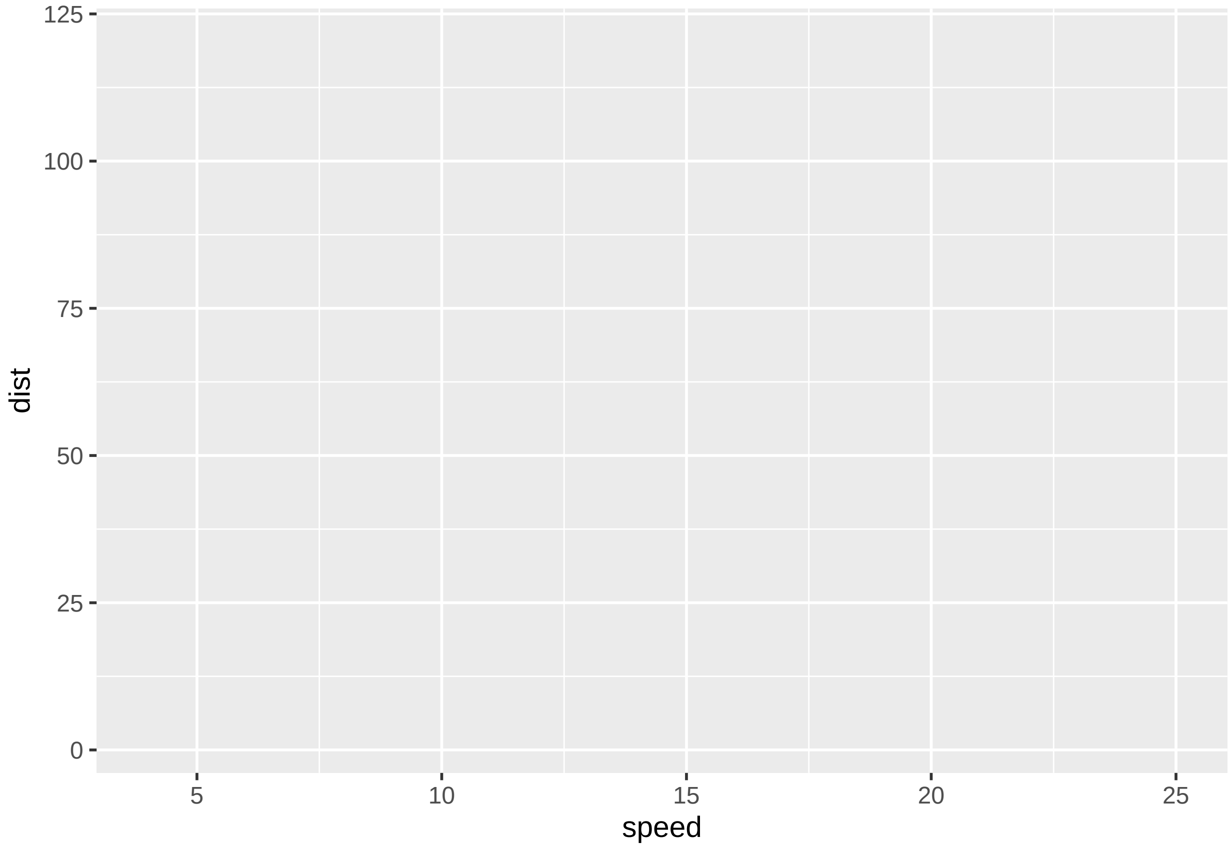
Angelita Rivera (PID: A15522236)

October 12th, 2021

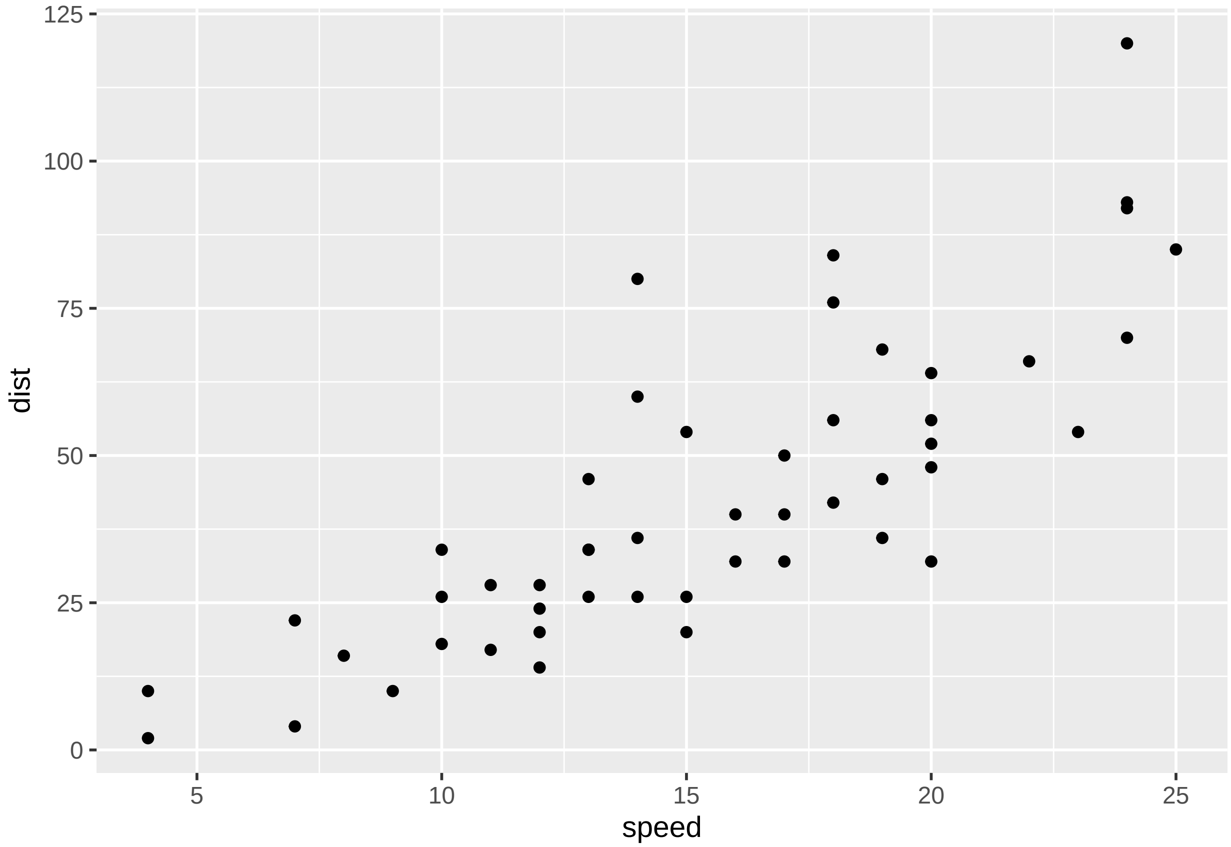
```
# Let's start with a scatterplot

# Need to load it up first
library(ggplot2)

# Every ggplot has a data + aes + geoms

# Select data set
ggplot(data = cars)
```



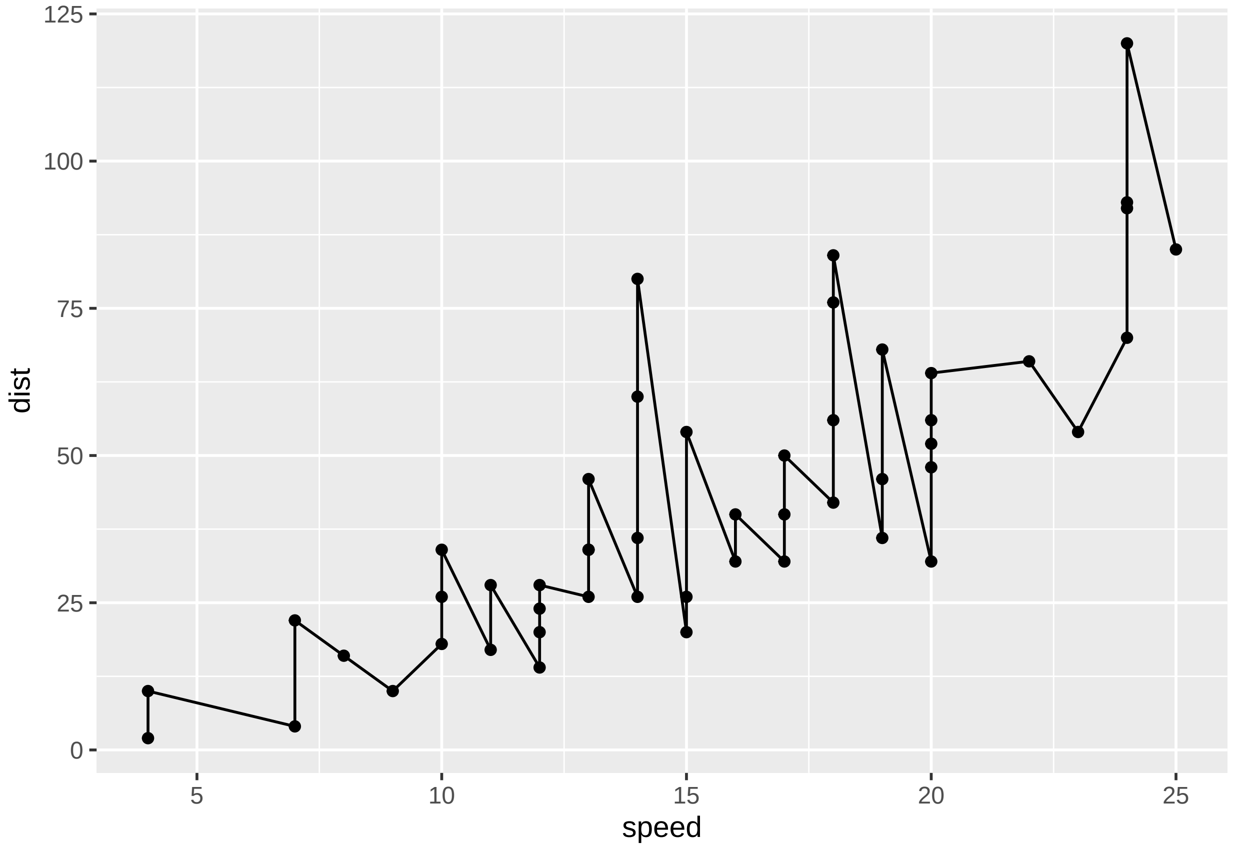
```
#Add aes (gives parameters)
ggplot(data = cars) + aes(x = speed, y=dist)
```



```
#Add geoms (adds points)
ggplot(data = cars) + aes(x = speed, y=dist) + geom_point()
```

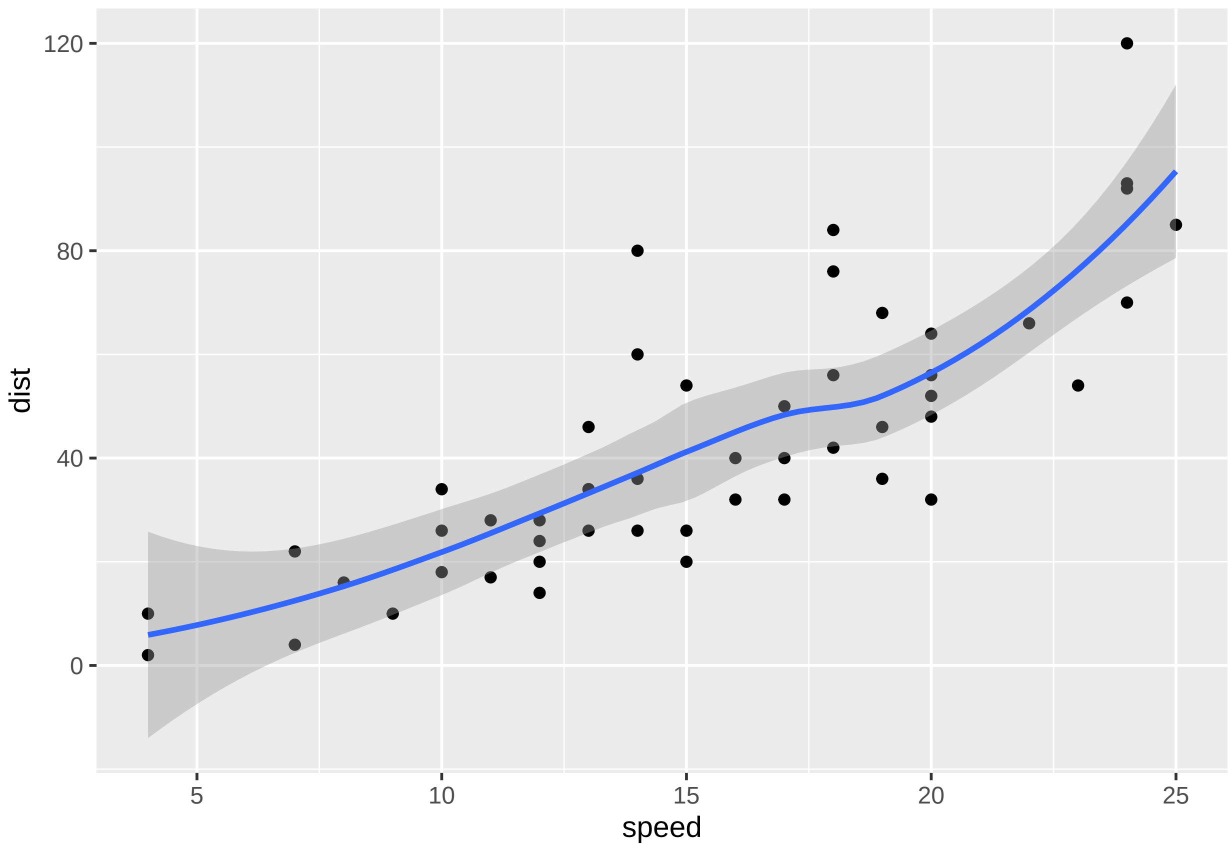


```
#Add another layer (ex. add a line on top)
ggplot(data = cars) + aes(x = speed, y=dist) + geom_point() + geom_line()
```



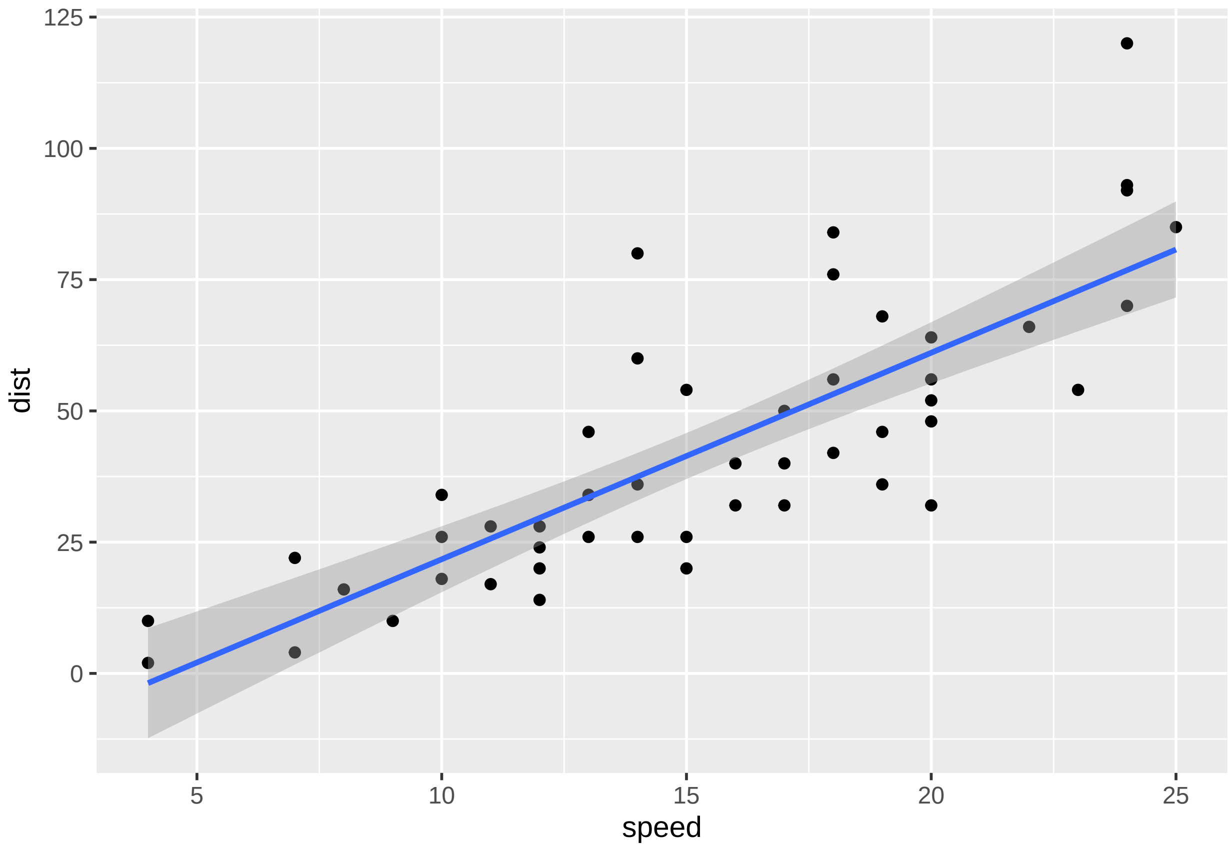
```
#Show a trend line to the scatter plot
ggplot(data = cars) + aes(x = speed, y=dist) + geom_point() + geom_smooth()
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



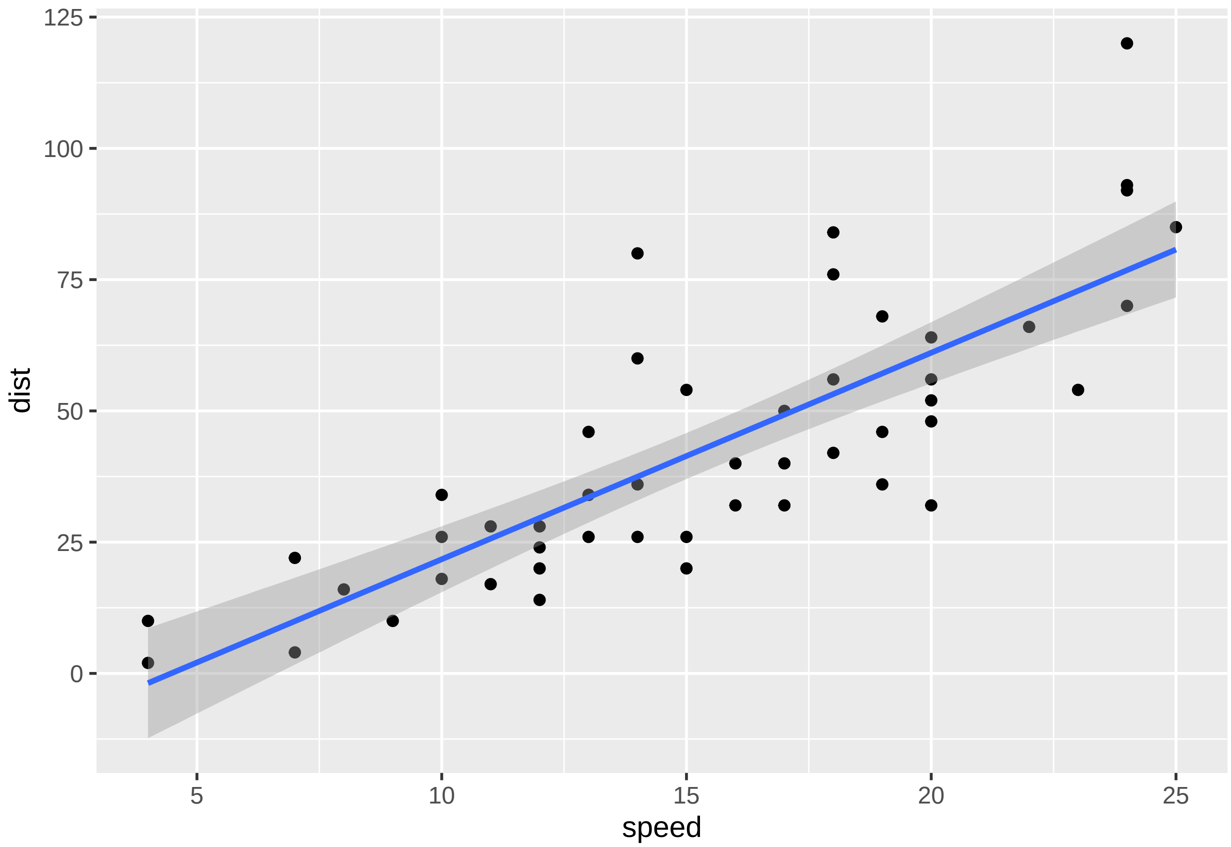
```
#Change to a linear model
ggplot(data = cars) + aes(x = speed, y=dist) + geom_point() + geom_smooth(method="lm")
```

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



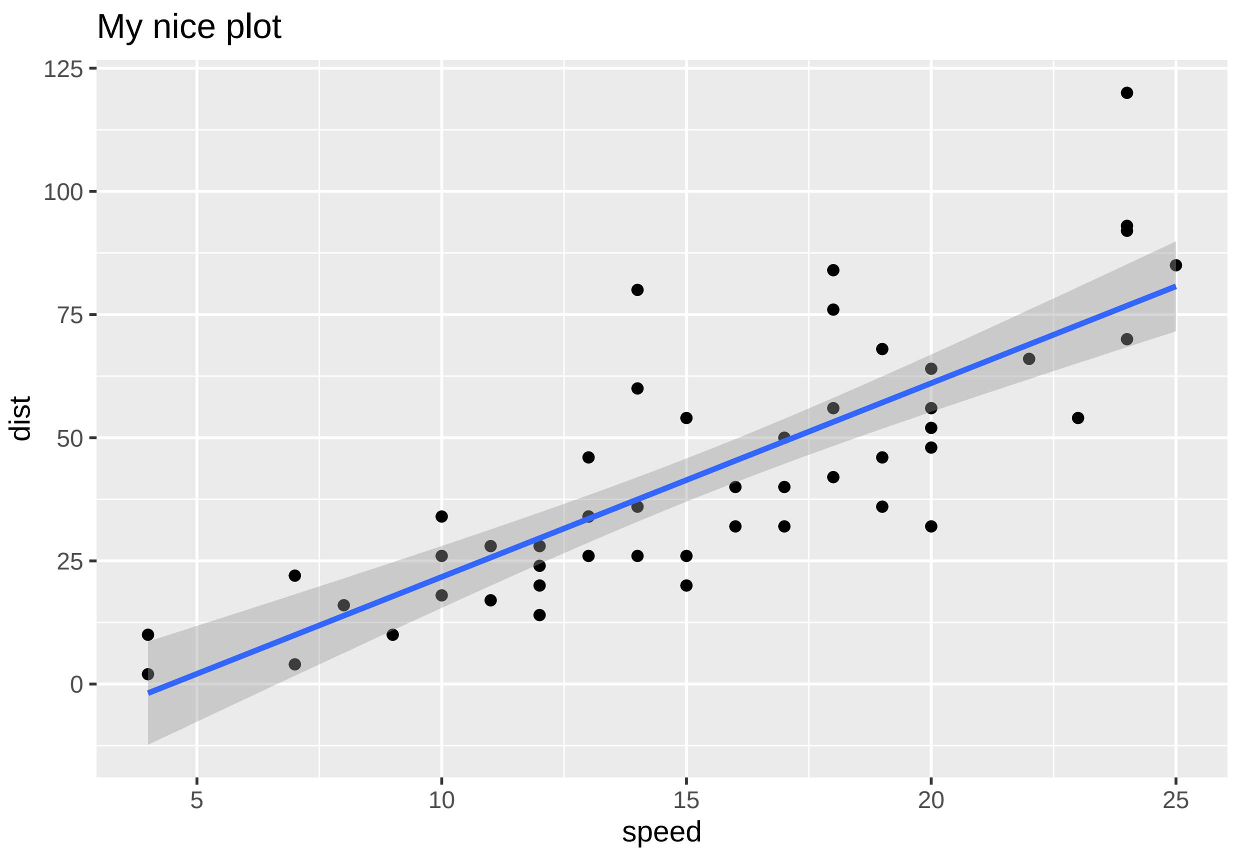
```
#Save as p
p <- ggplot(data = cars) + aes(x = speed, y=dist) + geom_point() + geom_smooth(method="lm")

#Call p
p
```

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



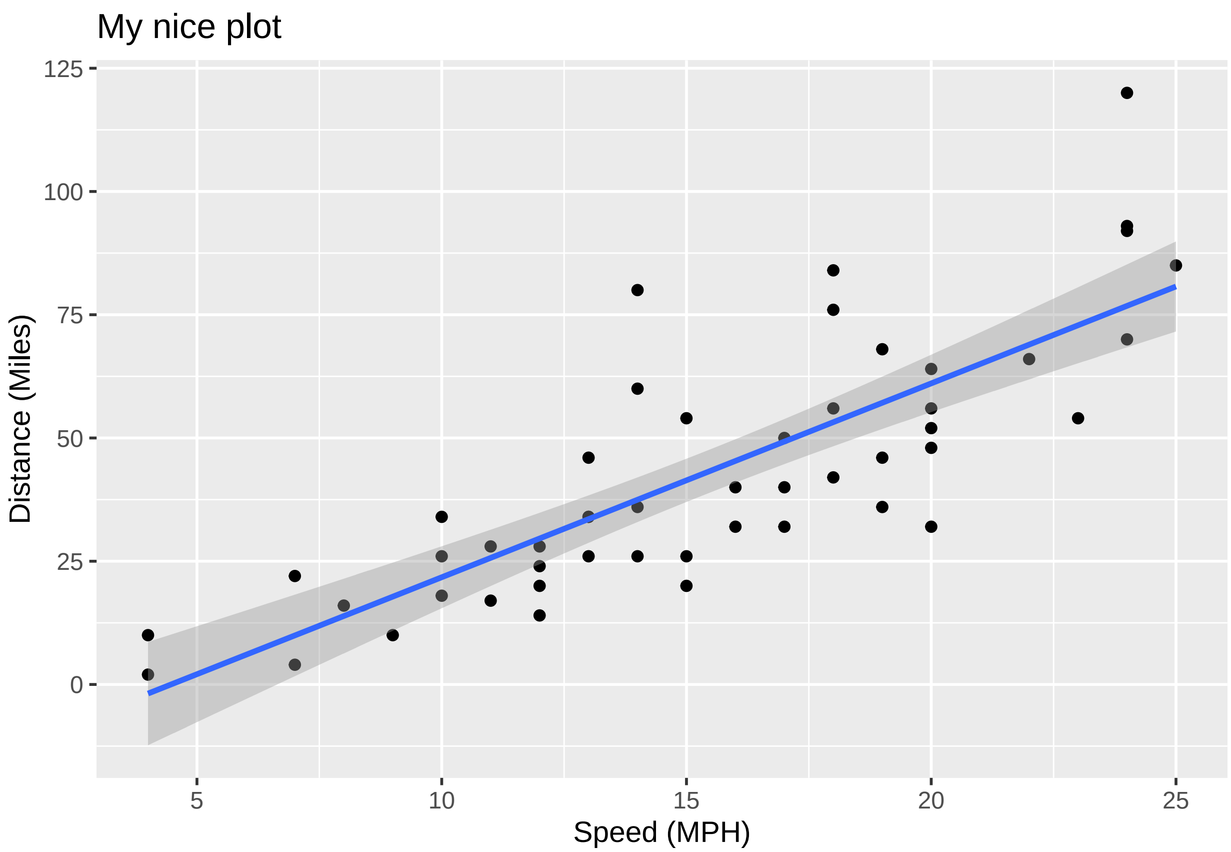
```
#Add title of plot
p + labs(title="My nice plot")
```

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

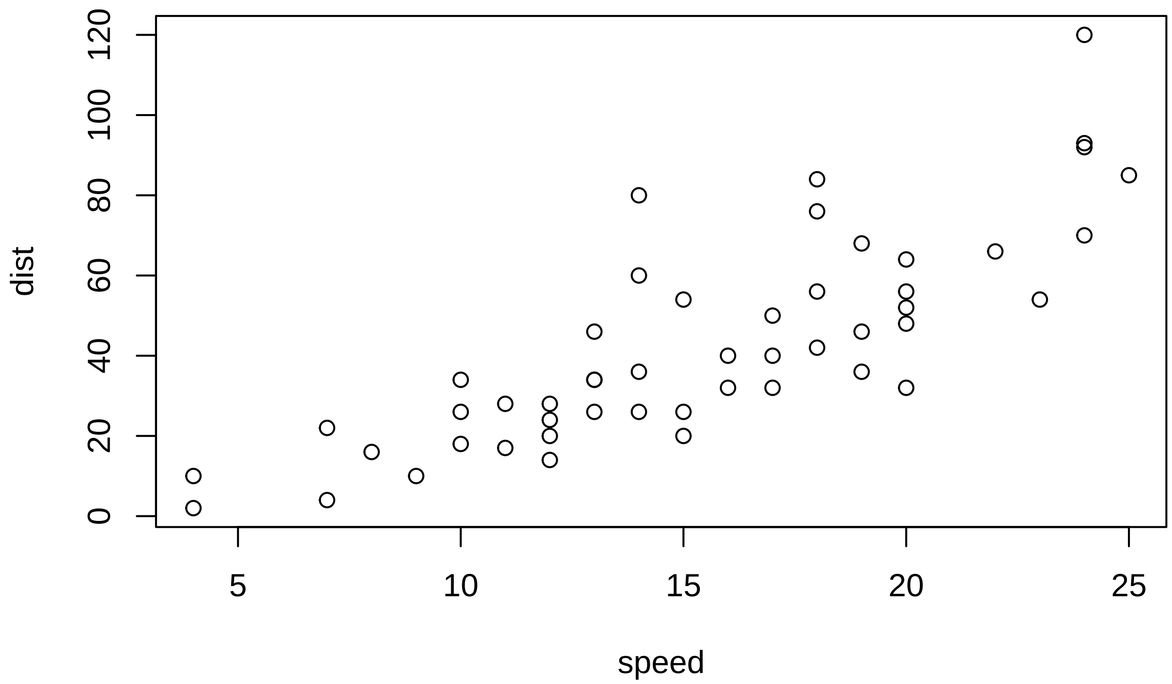


```
#Change x and y labels
p + labs(title="My nice plot", x="Speed (MPH)", y="Distance (Miles)")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

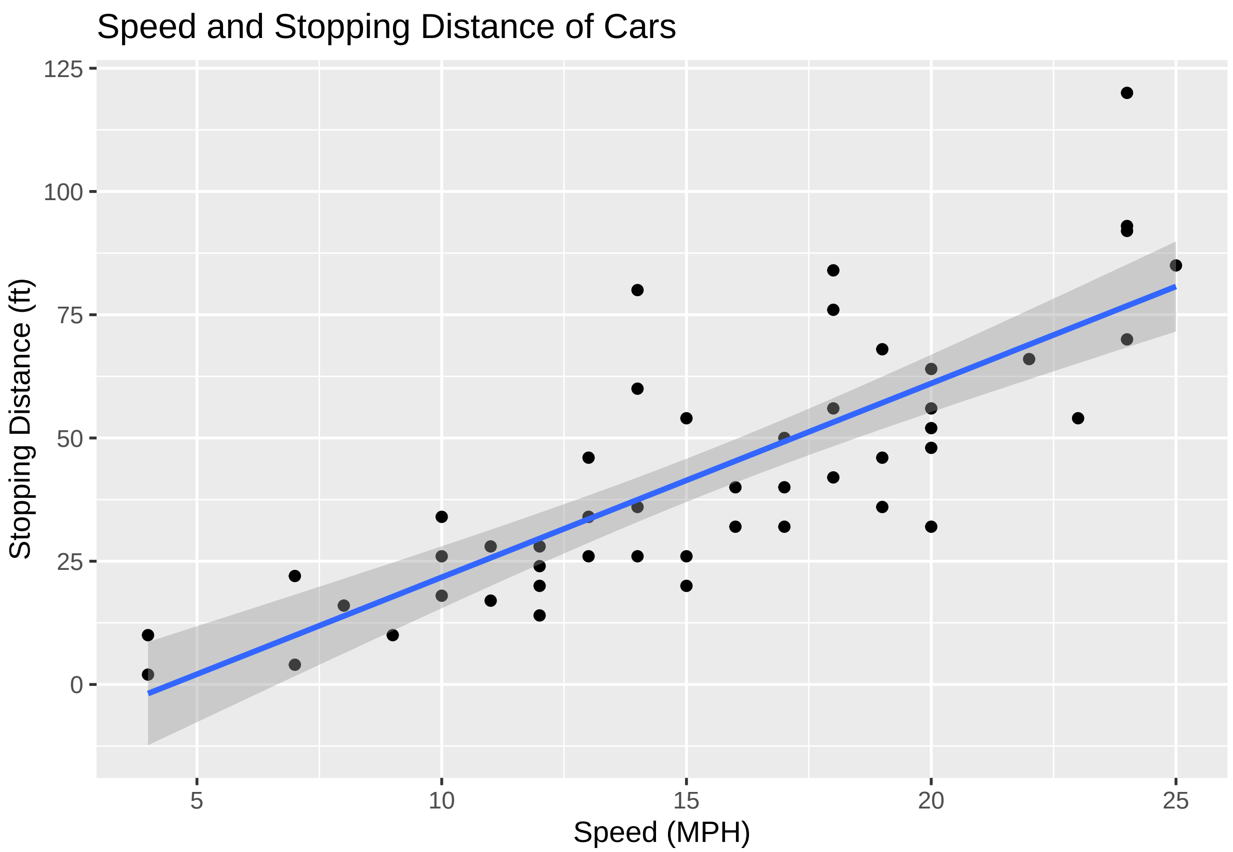


```
#Base graphics is shorter
plot(cars)
```



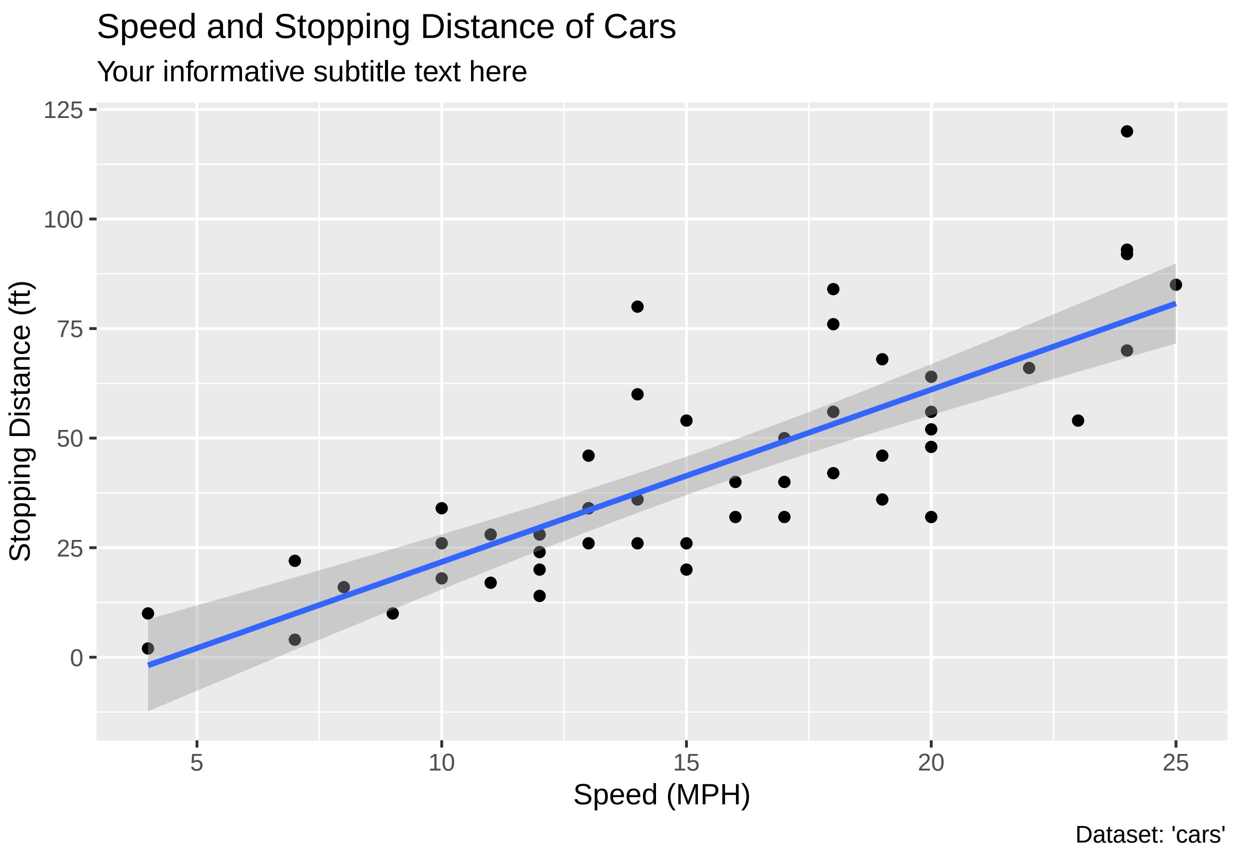
```
#But you can change every aspect of ggplots buy just adding another layer

#Different titles
p + labs(title="Speed and Stopping Distance of Cars", x="Speed (MPH)", y="Stopping Distance (ft)")
```

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



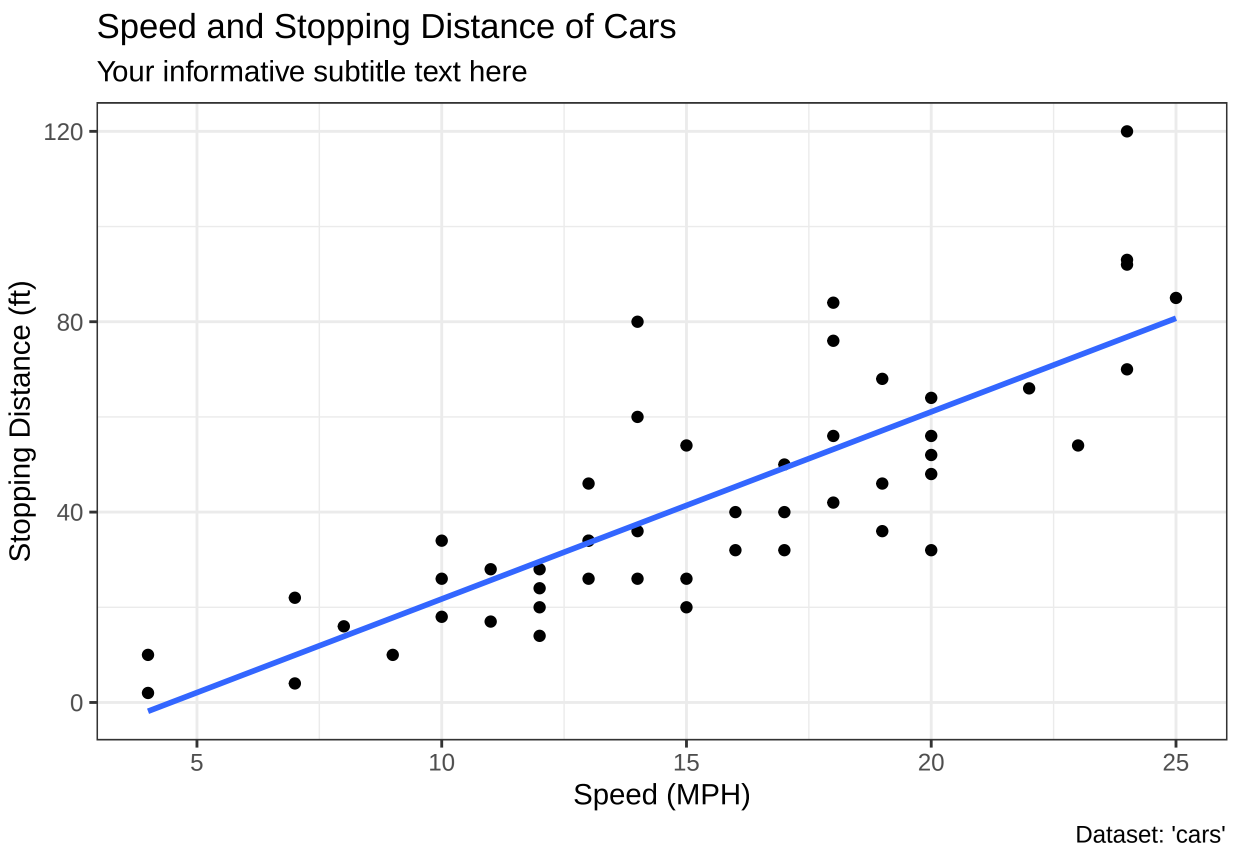
```
#Adding subtitle and caption
p + labs(title="Speed and Stopping Distance of Cars", x="Speed (MPH)", y="Stopping Distance (ft)", subti
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



```
#Adding se=FALSE) + theme_bw
ggplot(data = cars) + aes(x = speed, y=dist) + geom_point() + geom_smooth(method="lm", se=FALSE)+ labs(
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



```
#Save as p
p <- p <- ggplot(data = cars) + aes(x = speed, y=dist) + geom_point() + geom_smooth(method="lm", se=FAL

#New GENE dataset
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
```

```
#Count how many rows
nrow(genes)
```

```
## [1] 5196
```

```
#What are the column names?
colnames(genes)
```

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

```r
#How many columns are there?
ncol(genes)
```

```
## [1] 4
```

```r
#Summary table of one column --> in this case "State"
table(genes$State)
```

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

```r
#Answering the question: what fraction of total genes is up-regulated in this date set?
round(table(genes$State)/nrow(genes)*100,2)
```

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

```r
#Step 1: Recognize "State" column tells you whether it's down, up or unchanging
#Create a SUMMARY column of "State"
table(genes$State)
```

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

```r
#^Summarizes how many up, down, unchanging
#Step 2: Know you want to divide SUMMARY of up-regulated genes by TOTAL number of genes
table(genes$State) / nrow(genes)
```

```
## 
##       down unchanging         up 
## 0.01385681 0.96170131 0.02444188
```

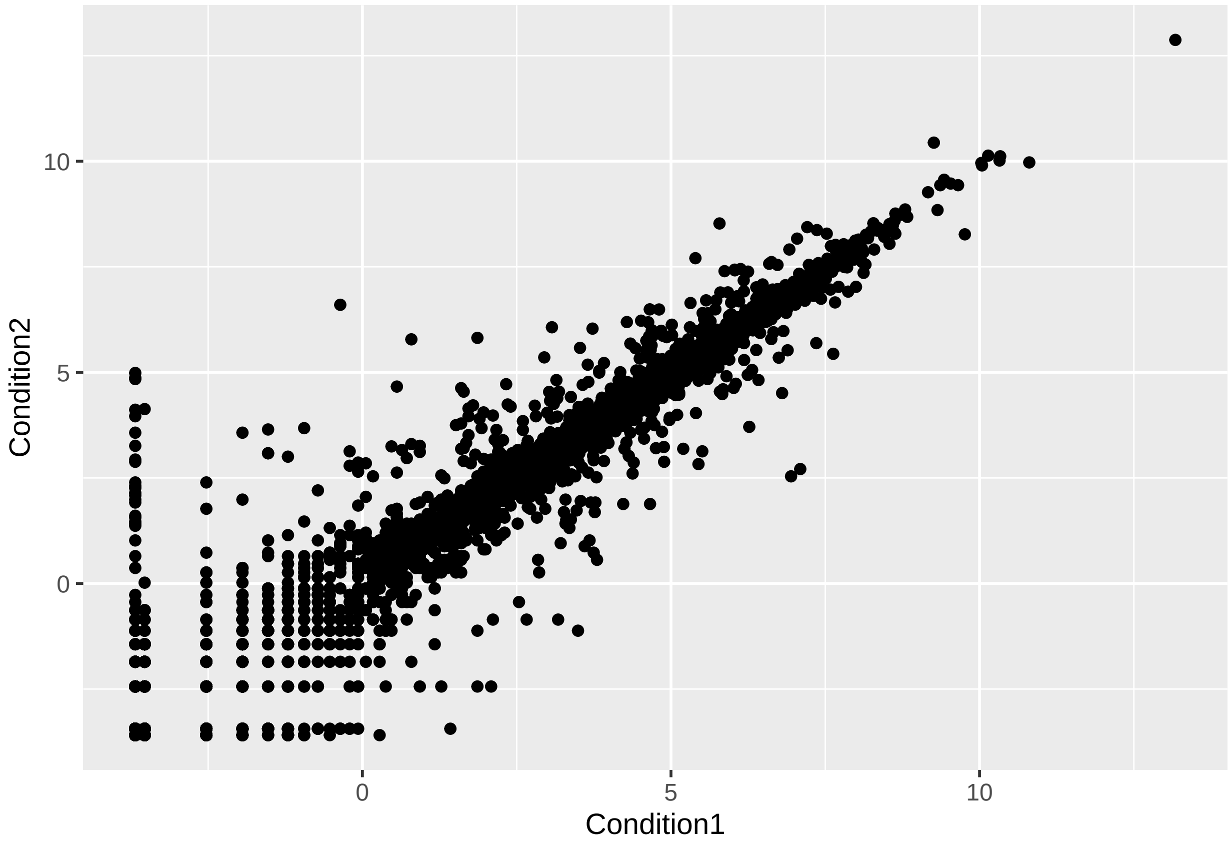
```r
#Gives you 0.0 numbers, we want percentage!
#AND WE WANT TWO SIGNIFICANT FIGURES
#To get two significant figures:
round(table(genes$State) / nrow(genes), 2)
```

```
## 
##       down unchanging         up 
##       0.01       0.96       0.02
```

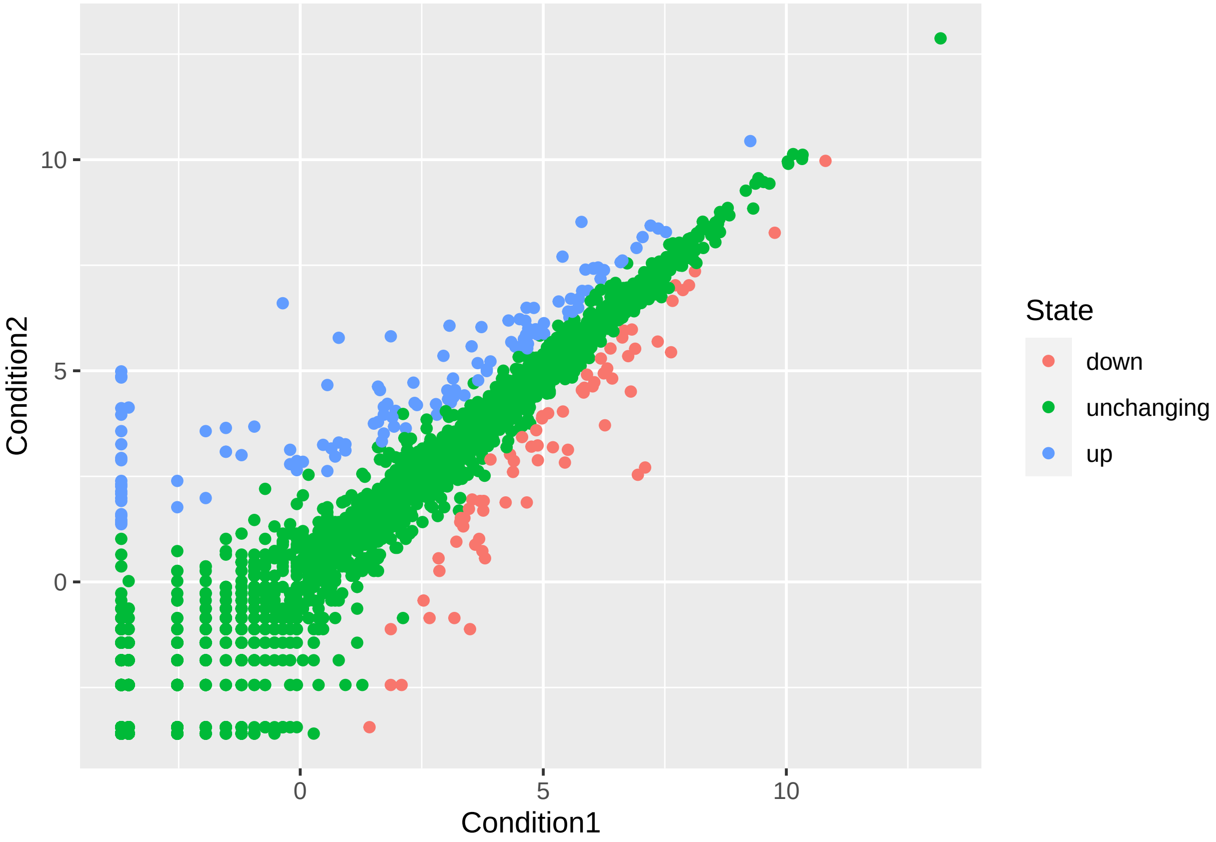
```r
#Still need percentage, so have to multiply by 100
round(table(genes$State) / nrow(genes) *100,2)
```

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

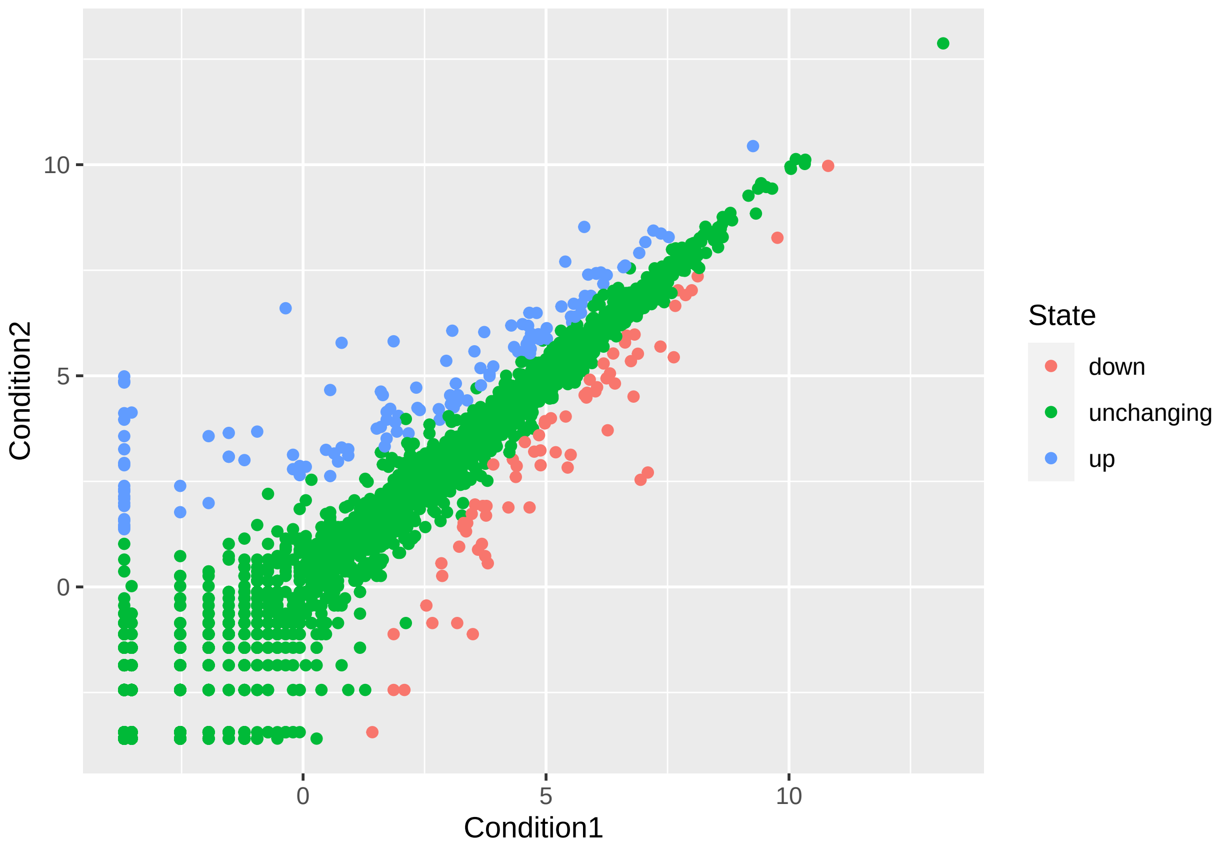
```
#Now we get percentage, rounded to 2 significant figures! (:

#Launch ggplot
library(ggplot2)

#Make a scatterplot of Gene
ggplot(genes) + aes(x=Condition1, y=Condition2) + geom_point()
```



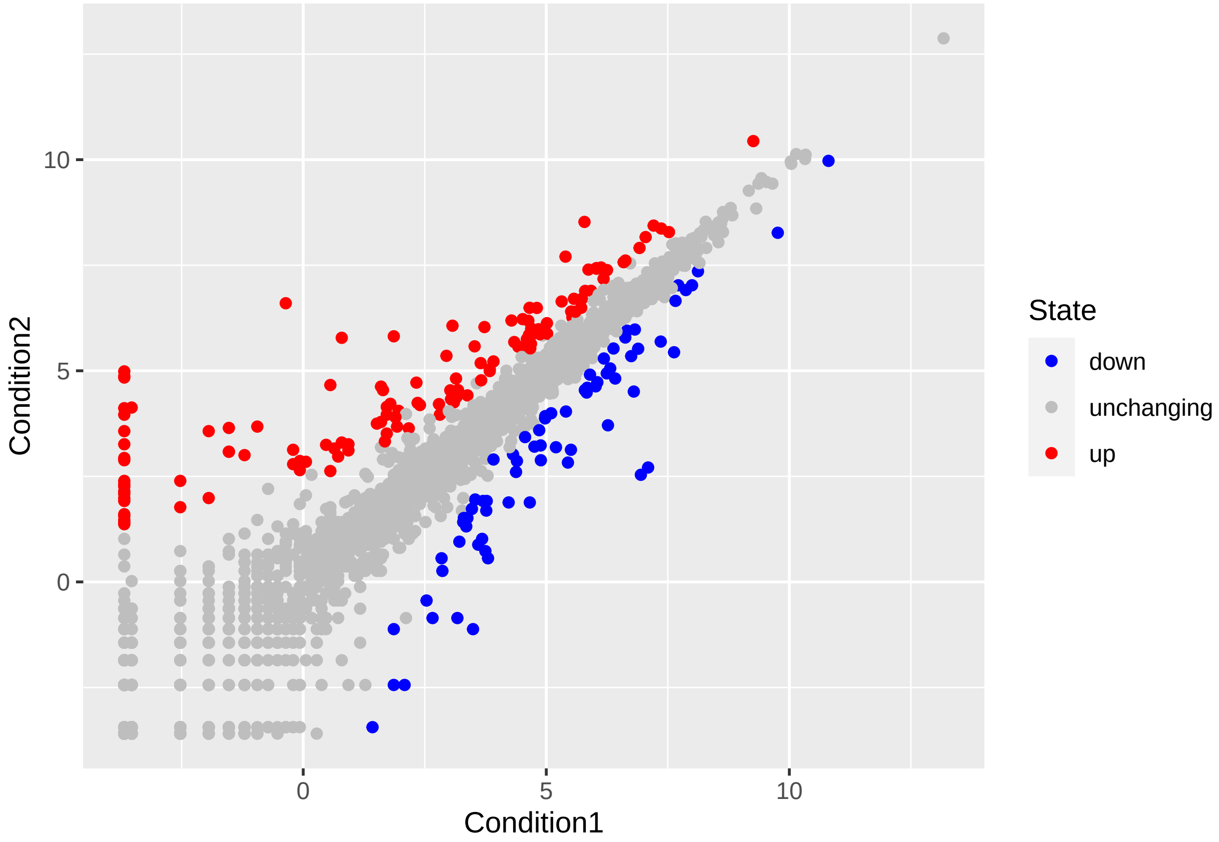
```
#Add column=State, so the plot can show which genes are down, up, and unchanging
ggplot(genes) + aes(x=Condition1, y=Condition2, col=State) + geom_point()
```



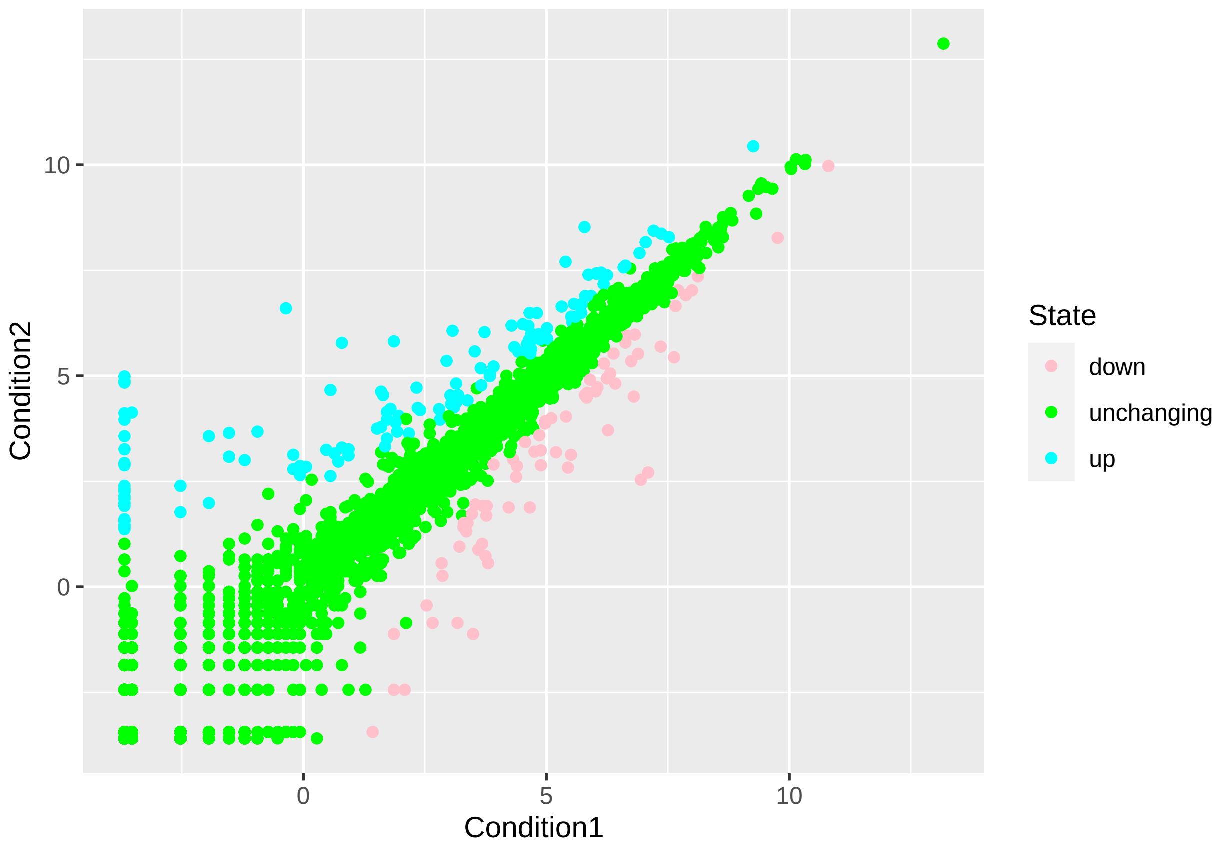
```
#Save as p
p <- ggplot(genes) + aes(x=Condition1, y=Condition2, col=State) + geom_point()
#DON'T FORGET TO LAUNCH p so R knows
p
```



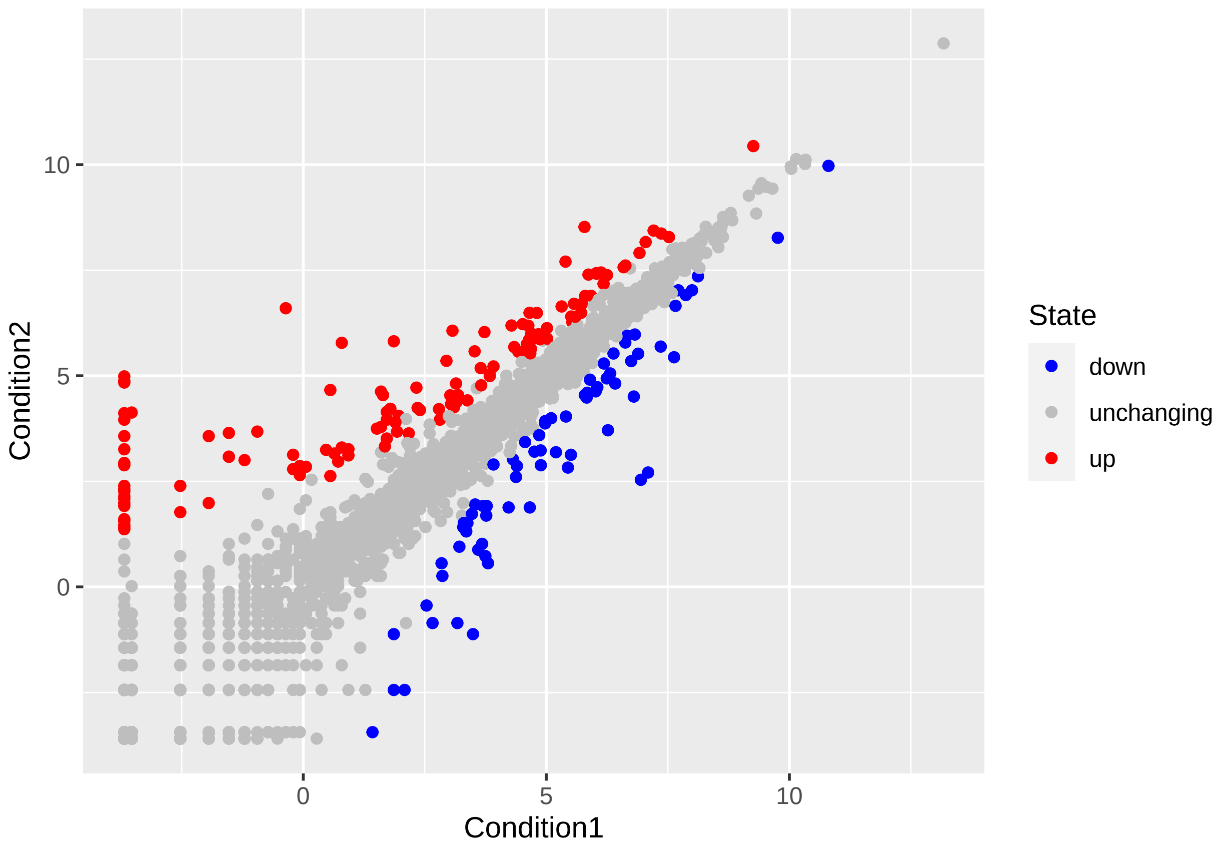
```
#Add more layers
#Change color scheme
p + scale_color_manual (values=c("blue","gray","red"))
```



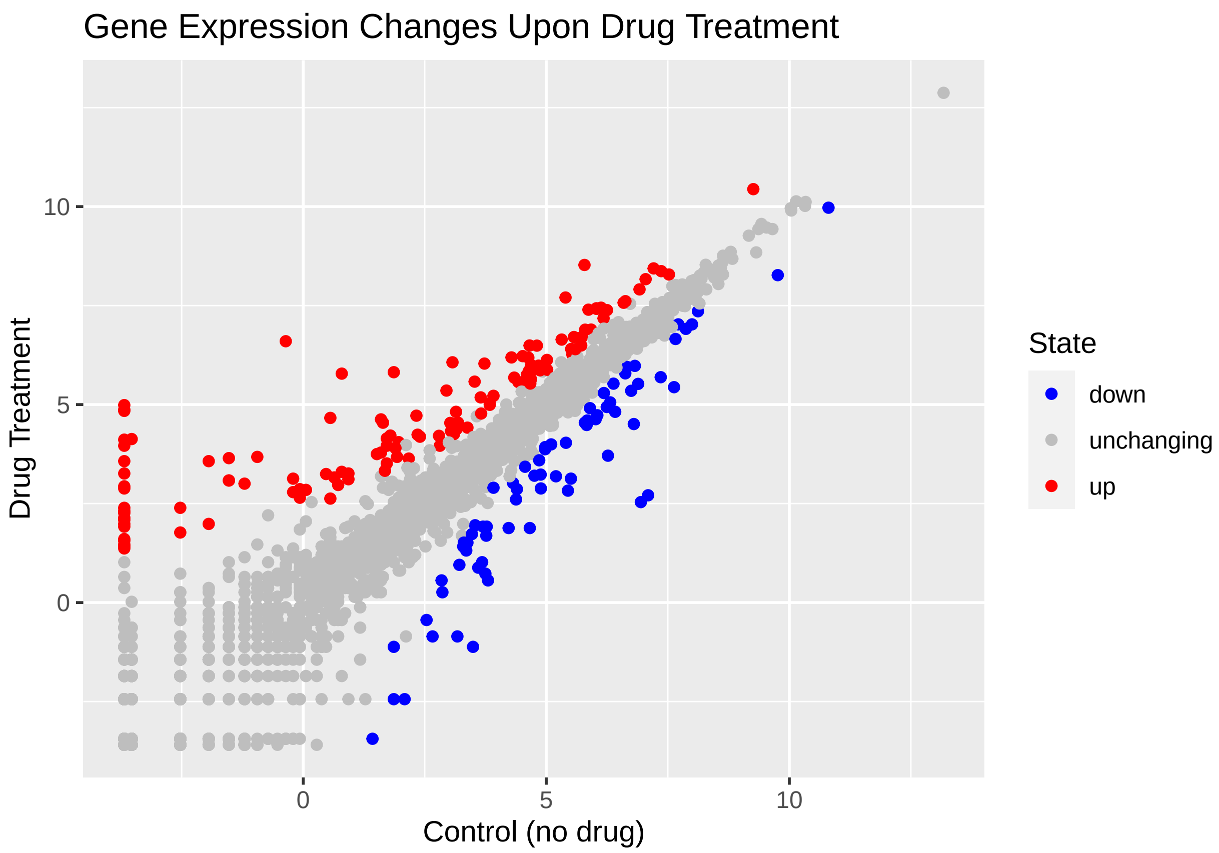
```
#You can change to any colors!
p + scale_color_manual (values=c("pink","green","cyan"))
```



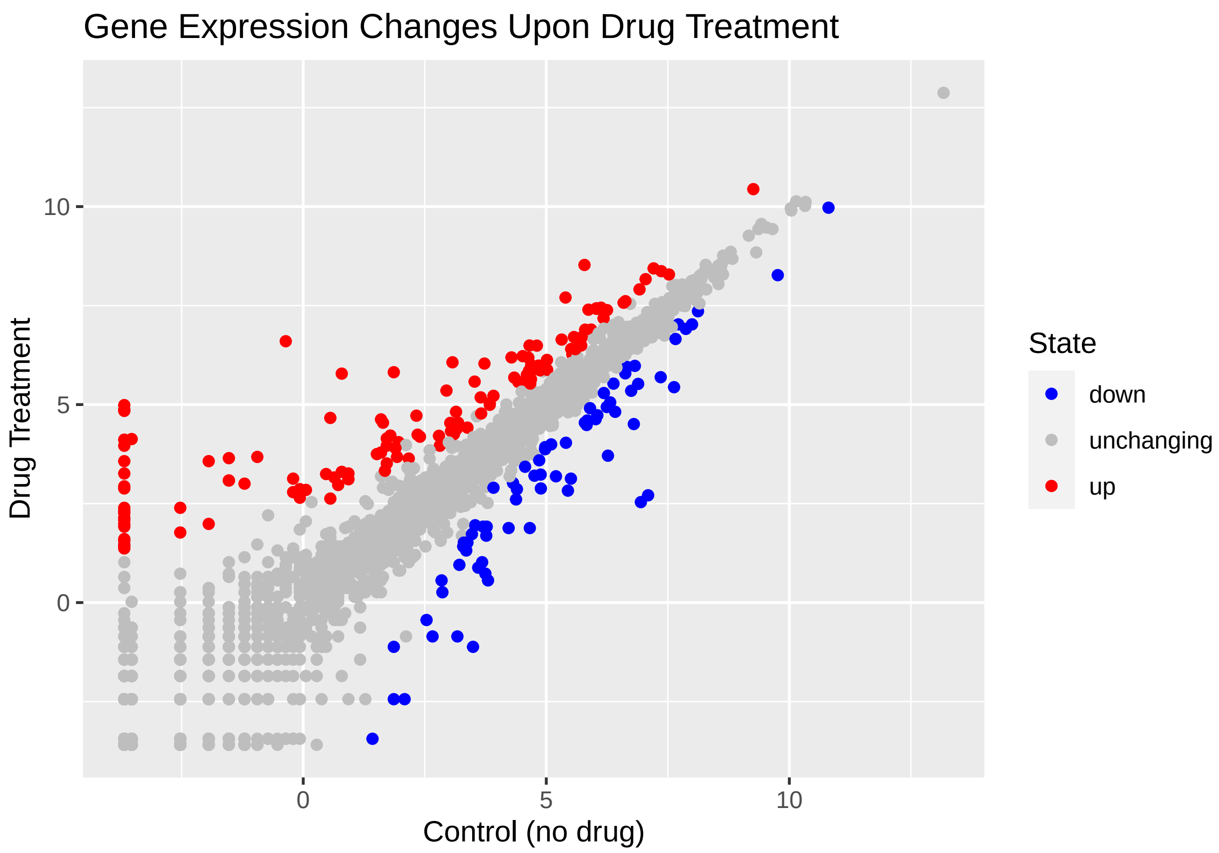
```
#To update p with new changes:
p <- p + scale_color_manual (values=c("blue","gray","red"))
#Launch p
p
```



```
#Add title, x labels, y labels
p + labs(title="Gene Expression Changes Upon Drug Treatment", x="Control (no drug)", y= "Drug Treatment
```



```
#Save new changes
p <- p + labs(title="Gene Expression Changes Upon Drug Treatment", x="Control (no drug)", y= "Drug Trea
#Launch p
p
```


Gene Expression Changes Upon Drug Treatment
Drug Treatment
Control (no drug)
10
5
0
0
5
10
State
down
unchanging
up