

# class05.R

Anu

2021-10-12

```
# 'title: "Class 05 Data Visualization"
# 'author: "Anu Chaparala (PID: A15484707)"
# today's objective: learn how to use ggplot(), aes(), and geom_point()

#start with scatter plot
#first, load ggplot2 using library(ggplot2)
#then, call ggplot2(cars)
#note, ggplot() function alone just defines the data set for the plot and creates an empty base on top
#so, aes() aesthetics, maps variables (i.e. columns) from your data set to the visual features of the plot

#every ggplot has a data + aes + geoms

#load data set
cars
```

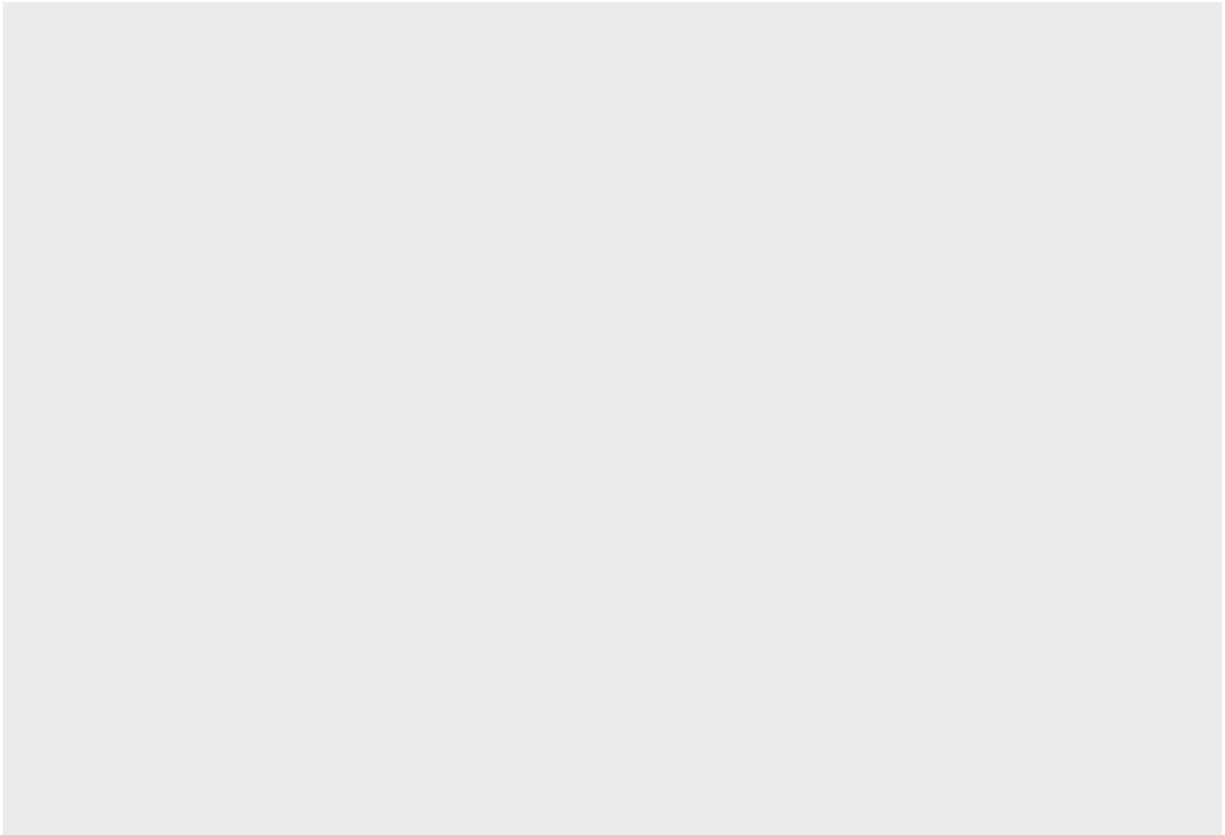
```
##      speed dist
## 1         4    2
## 2         4   10
## 3         7    4
## 4         7   22
## 5         8   16
## 6         9   10
## 7        10   18
## 8        10   26
## 9        10   34
## 10       11   17
## 11       11   28
## 12       12   14
## 13       12   20
## 14       12   24
## 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
```

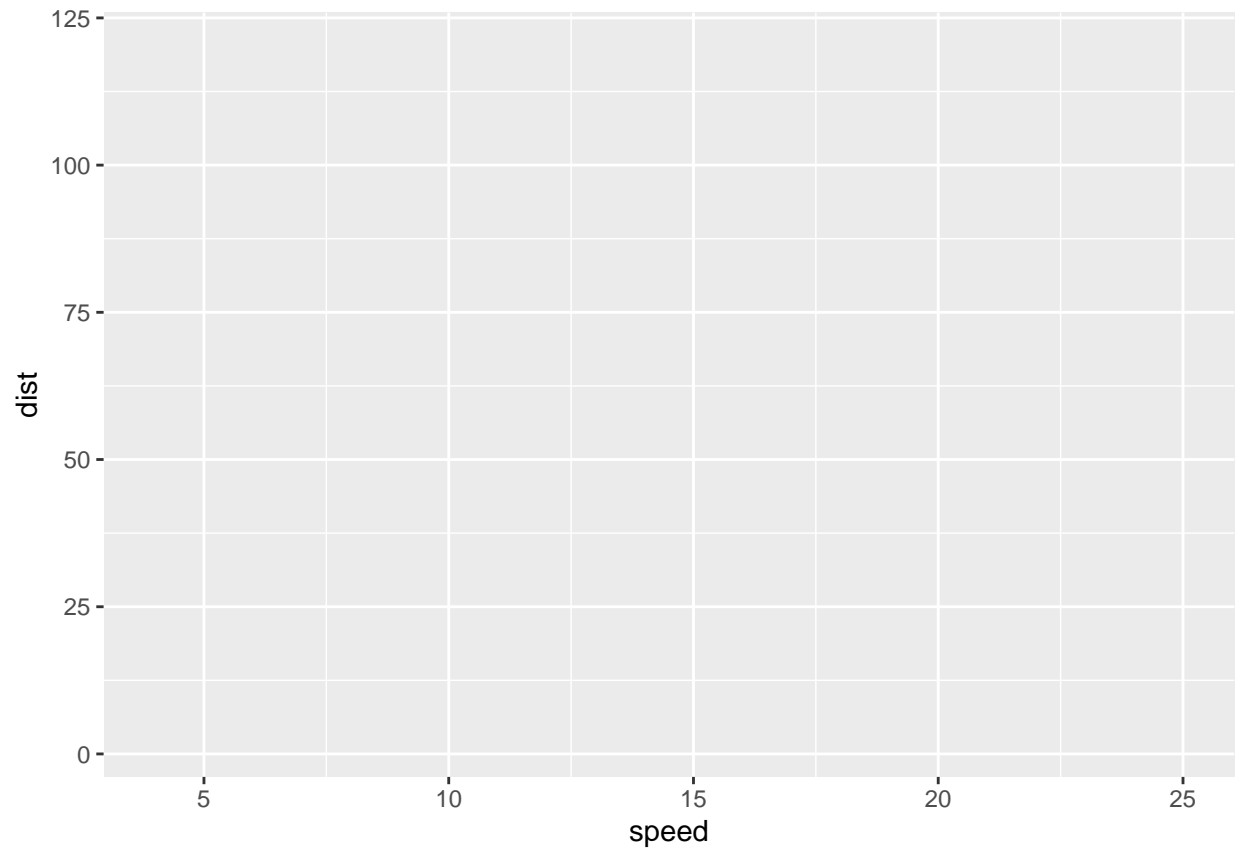
```
# install ggplot using install.packages("ggplot2")
```

```
#then, load ggplot install  
library(ggplot2)
```

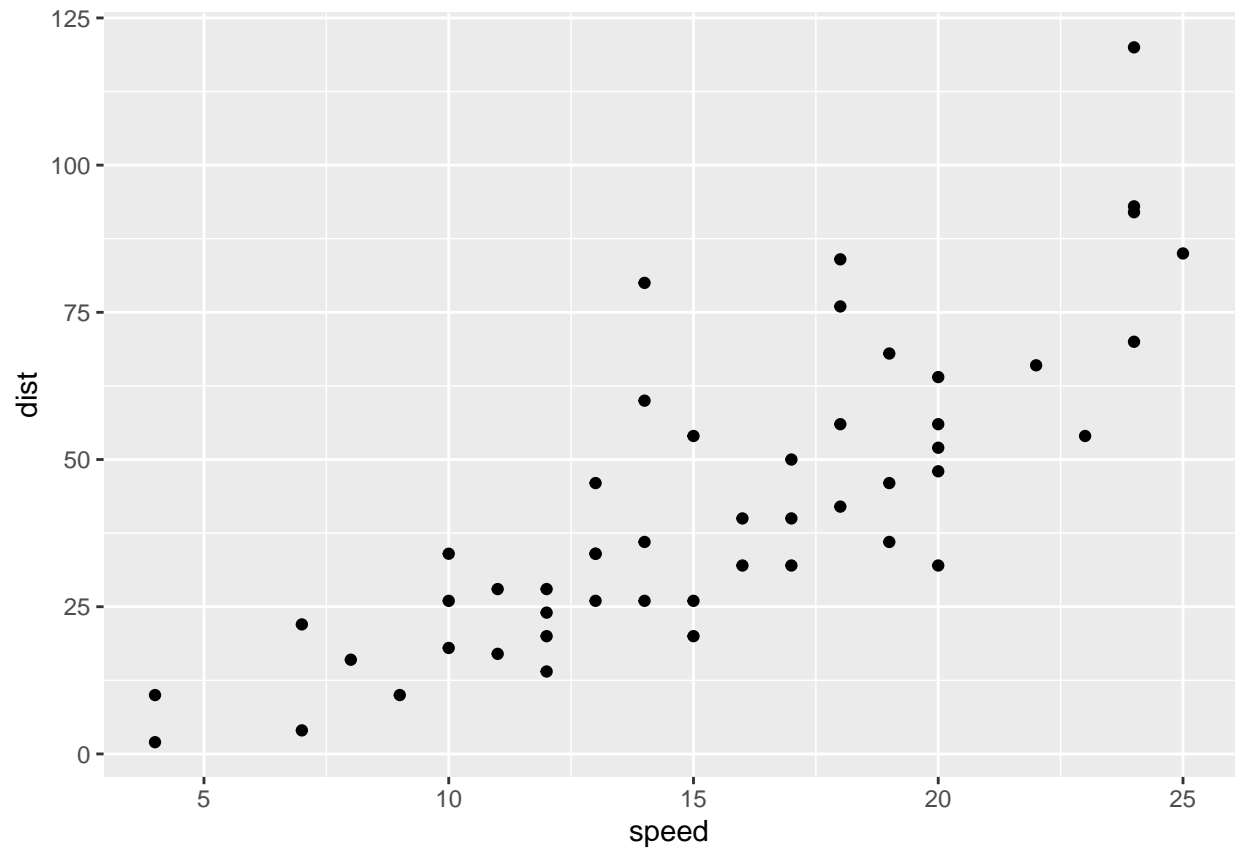
```
# assign data set to ggplot  
ggplot(cars)
```



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

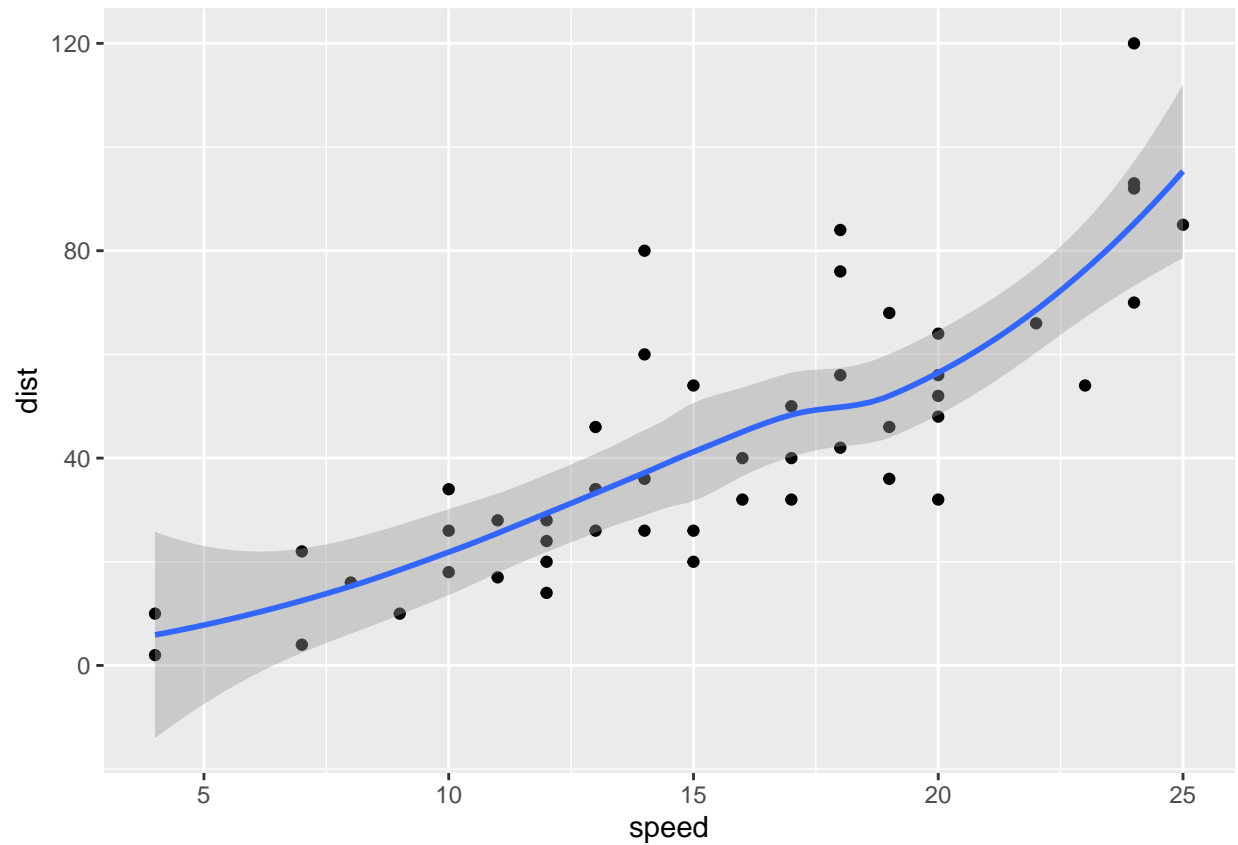


```
# add points/lines/columns (geom_point or line or col)
ggplot(data=cars) + aes(x=speed, y=dist) + geom_point()
```

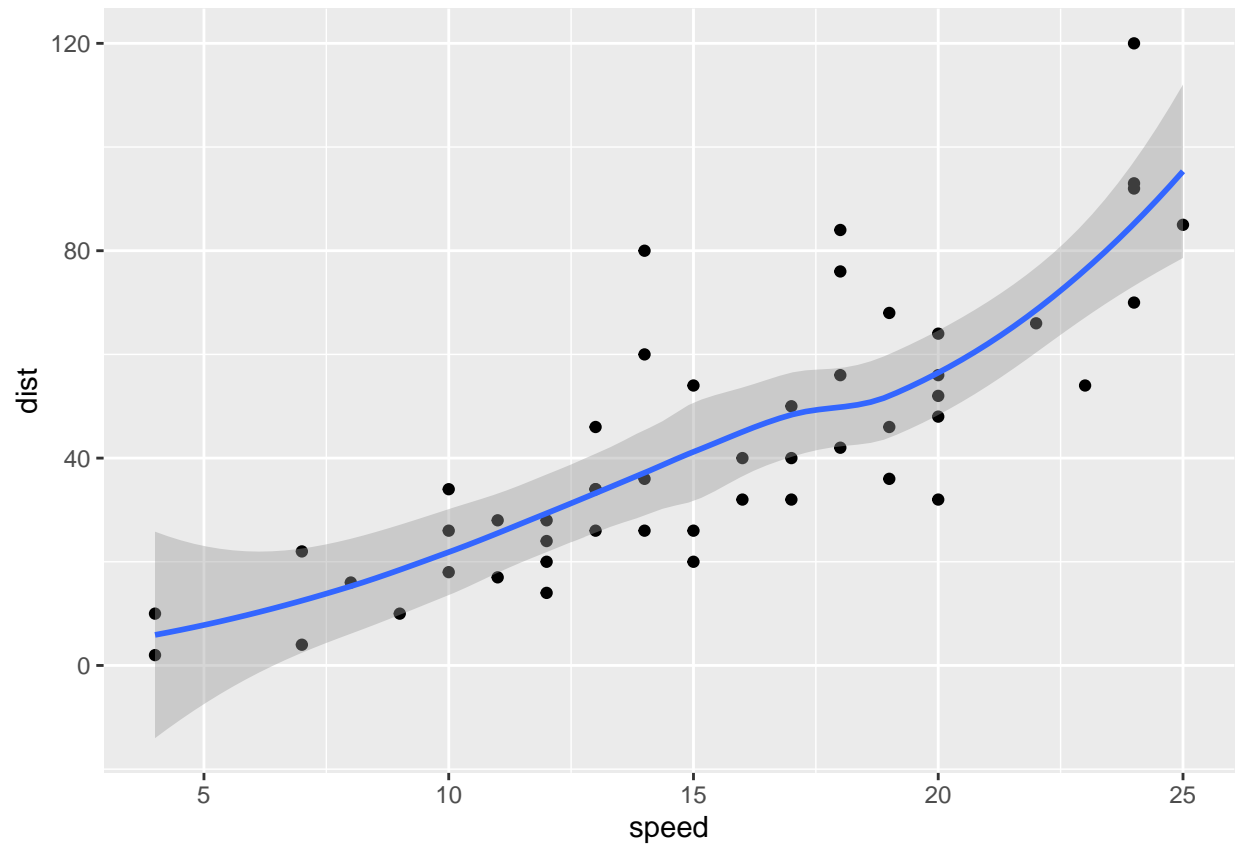


```
# add trend line  
ggplot(data=cars) + aes(x=speed, y=dist) + geom_point() + geom_smooth()
```

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



```
# assign variable to plot for ease of recall  
p <- ggplot(data=cars) + aes(x=speed, y=dist) + geom_point() + geom_smooth()  
p  
  
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```

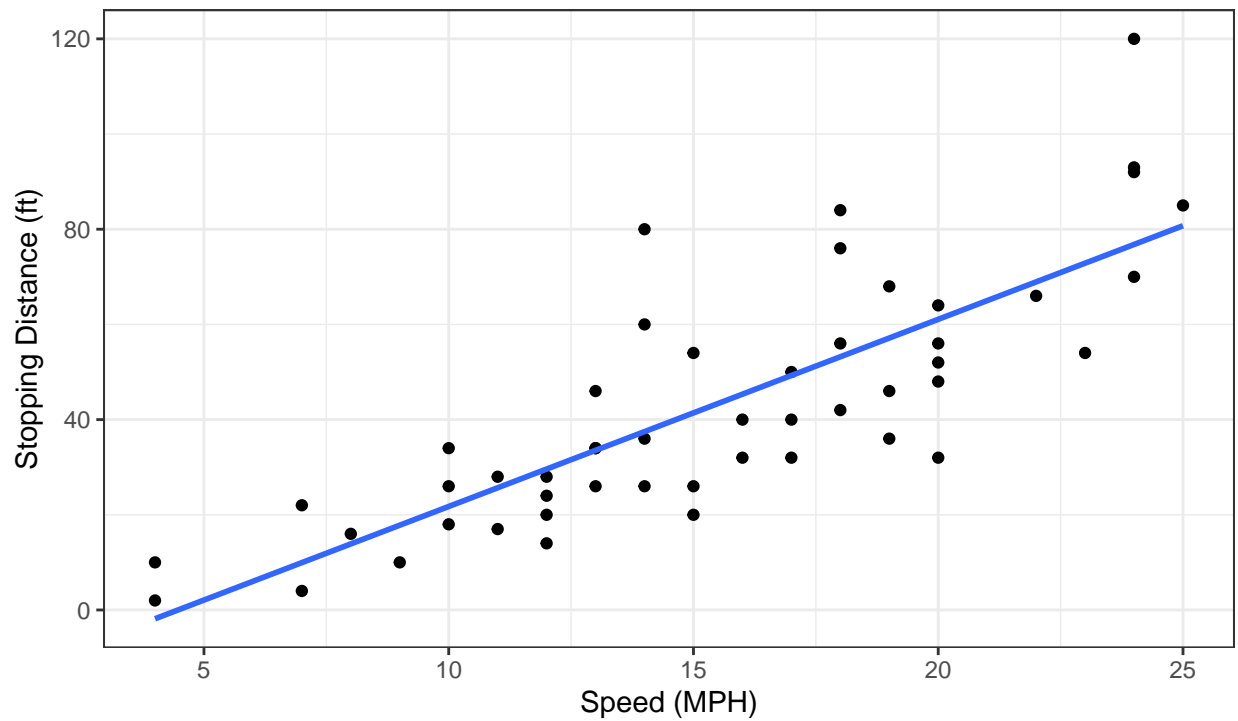


```
# add label annotations (labs()), and black and white theme using theme_bw()
ggplot(cars) + aes(x=speed, y=dist) + geom_point() +
  labs(title="Speed and Stopping Distances of Cars", x="Speed (MPH)", y="Stopping Distance (ft)",
        subtitle = "My informative subtitle text here",
        caption="Dataset: 'cars'") +
  geom_smooth(method="lm", se=FALSE) +
  theme_bw()
```

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

## Speed and Stopping Distances of Cars

My informative subtitle text here



Dataset: 'cars'

```
labelled_p <- 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 = "My informative subtitle text here",  
        caption="Dataset: 'cars'") +  
  geom_smooth(method="lm", se=FALSE) +  
  theme_bw()
```

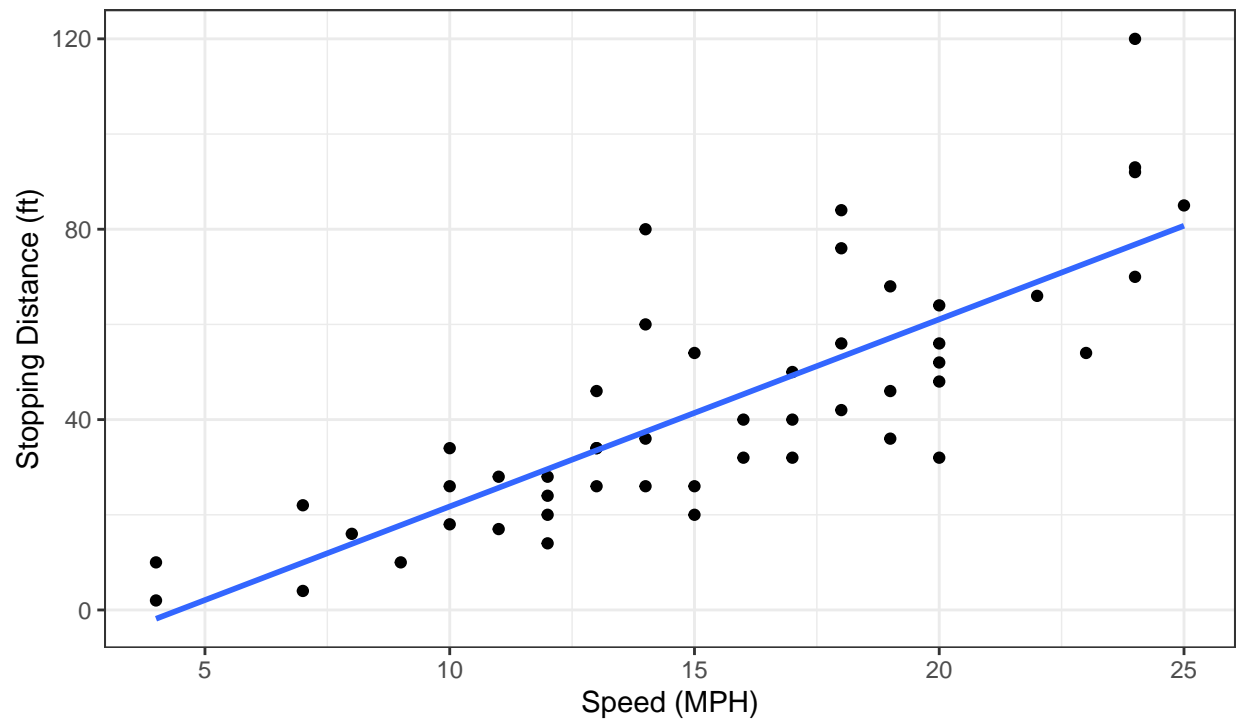
```
labelled_p
```

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



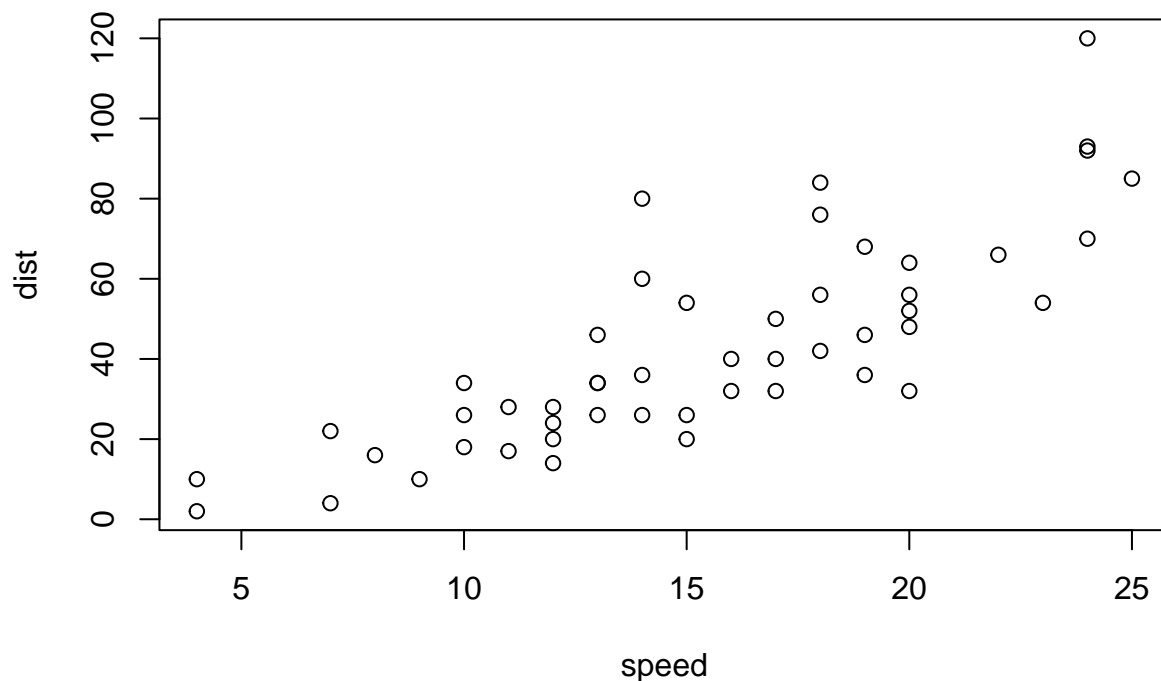
## Speed and Stopping Distances of Cars

My informative subtitle text here



Dataset: 'cars'

```
#Base graphics is shorter method of graphing  
plot(cars)
```



```
#let's get into more advanced aesthetics
#Adjust the point size of a scatter plot using the (size) parameter.
#Change the point color of a scatter plot using the (color) parameter.
#Set a parameter (alpha) to change the transparency of all points.

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
```

```
#find number of rows in data set
nrow(genes)
```

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

```
#find number of columns and column names
ncol(genes)
```

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

```
colnames(genes)
```

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

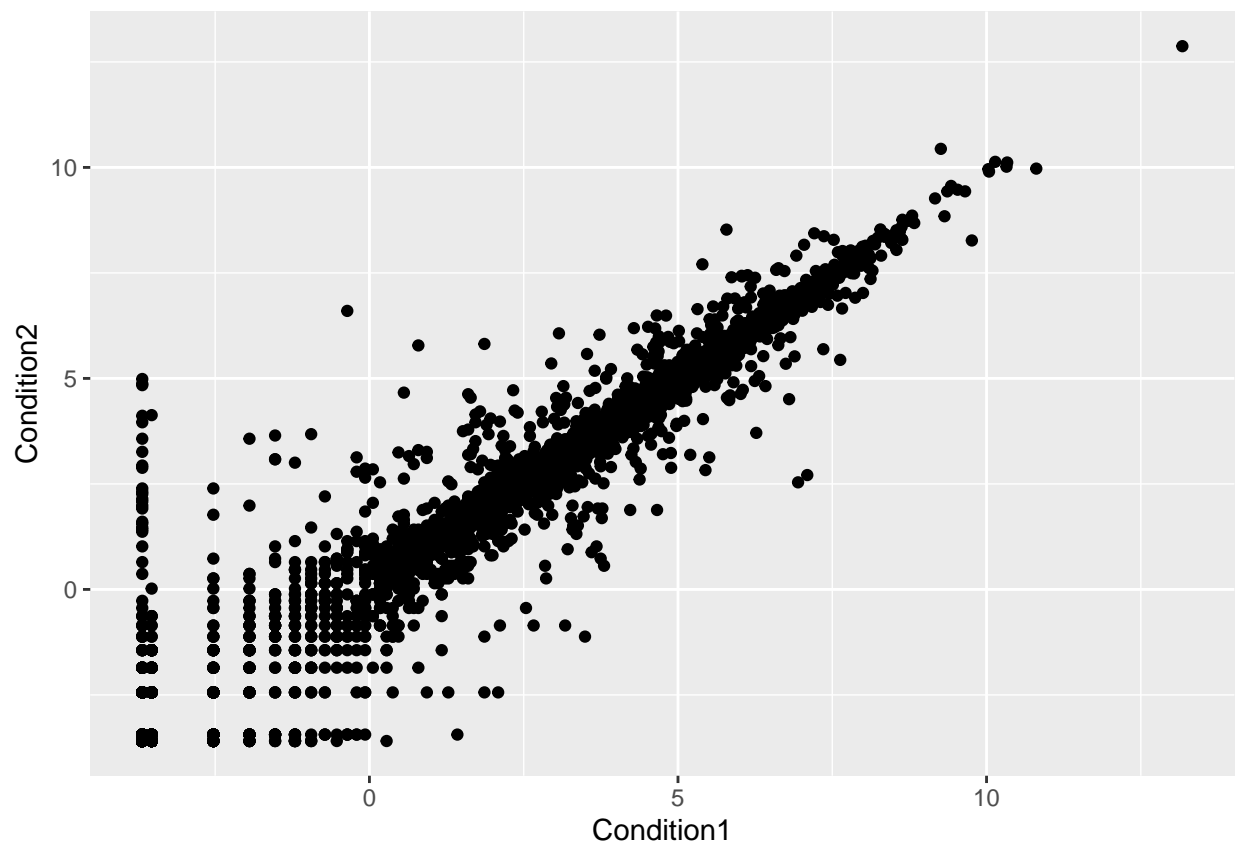
```
#use table() function on the State column name to find how many upregulated genes there are in this data  
table(genes$State)
```

```
##  
##      down  unchanged      up  
##       72      4997     127
```

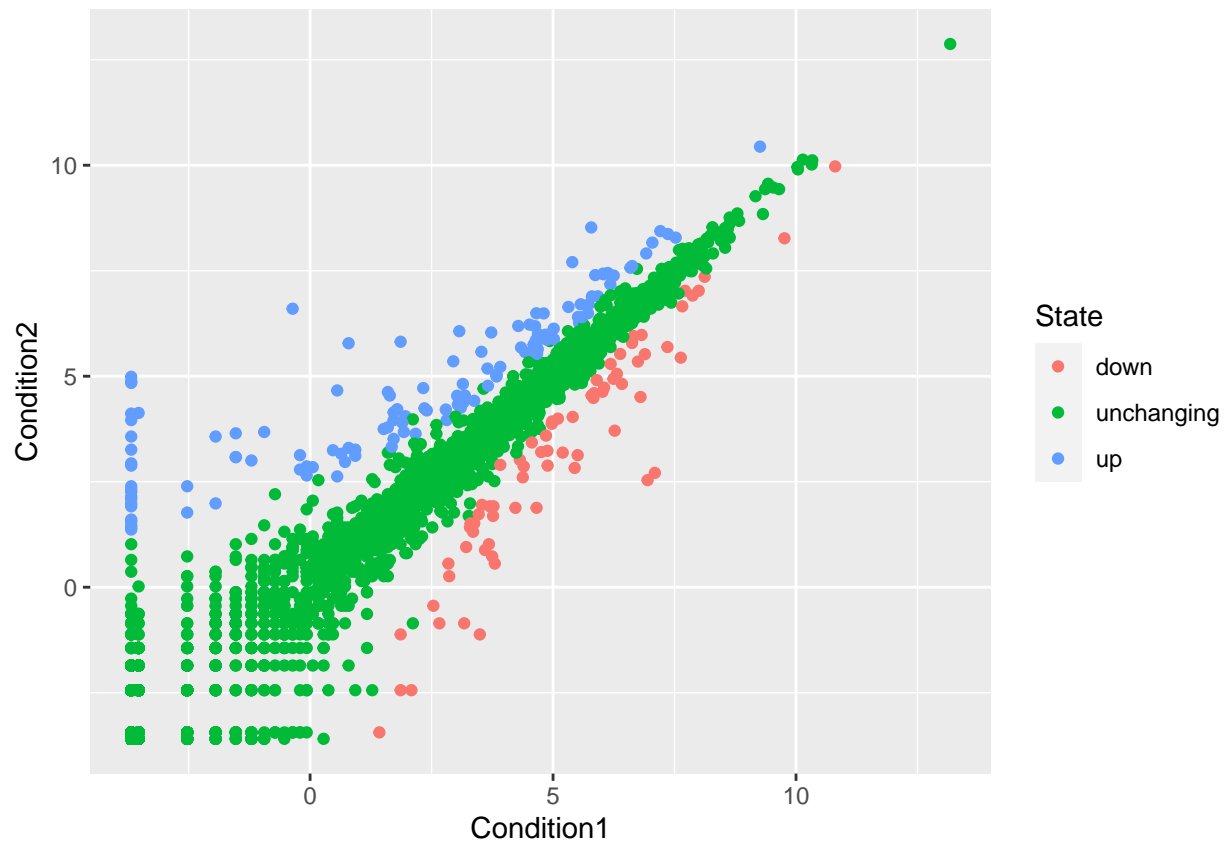
```
#fraction of upregulated over total  
round(table(genes$State)/nrow(genes) * 100, 2 )
```

```
##  
##      down  unchanged      up  
##     1.39     96.17     2.44
```

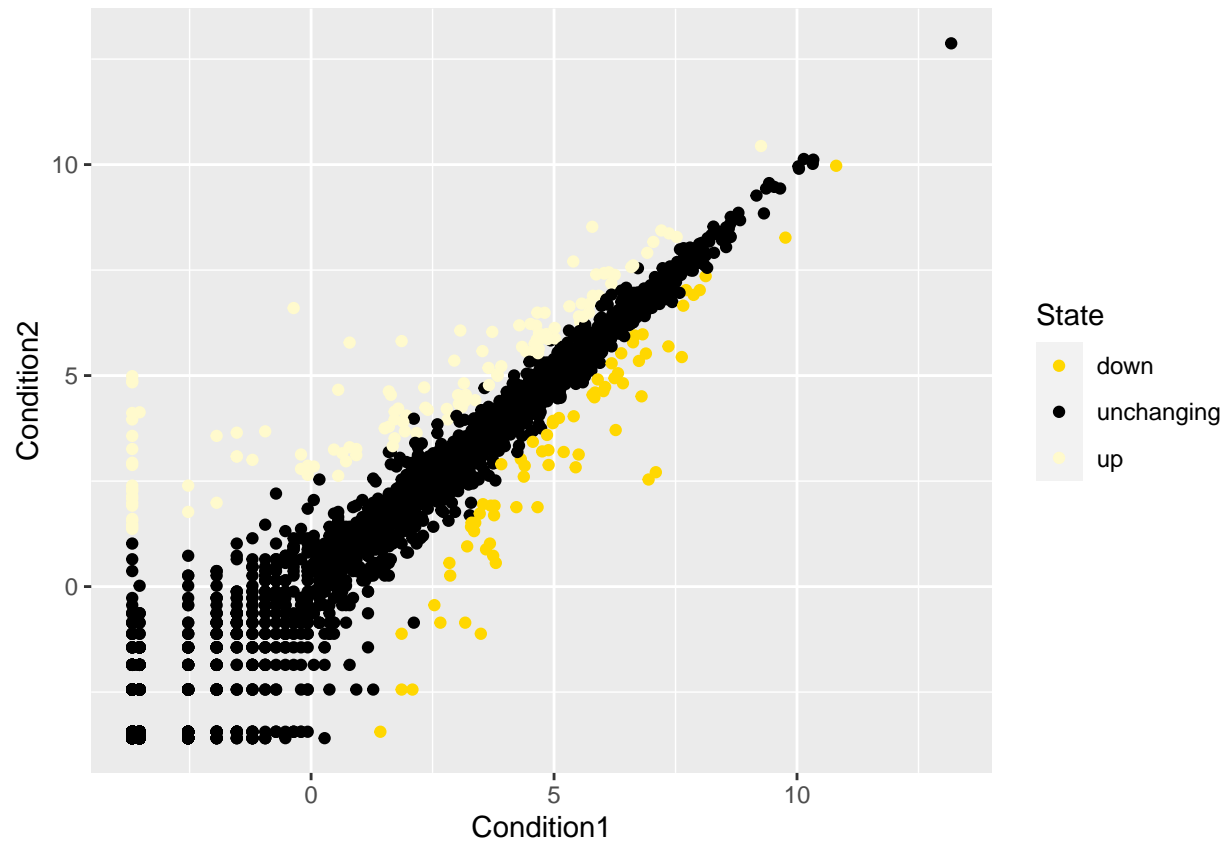
```
#create scatter plot with genes data.frame and ggplot and add respective aesthetics  
ggplot(genes) + aes(x=Condition1, y=Condition2) + geom_point()
```



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

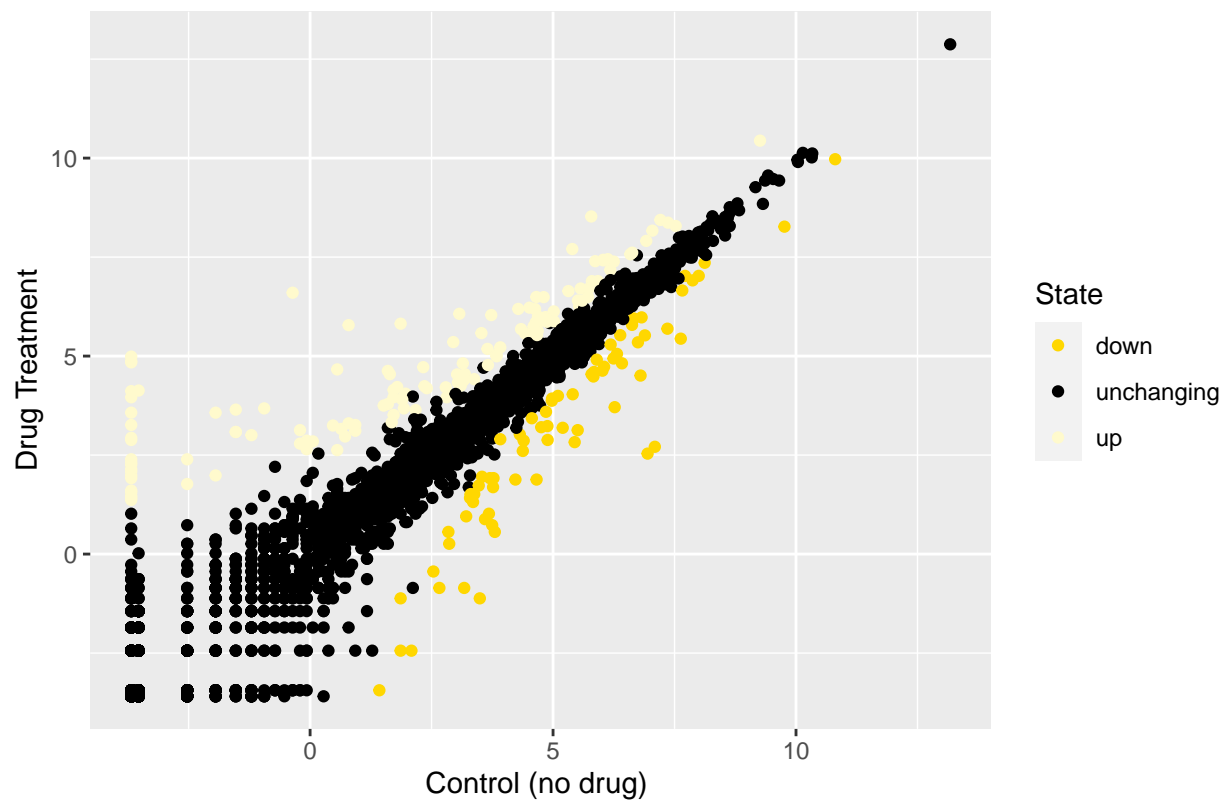


```
#change colors
p + scale_colour_manual( values=c("gold","black","lemonchiffon") )
```



```
#add some plot annotations to the p object with the labs() function
p + scale_colour_manual(values=c("gold","black","lemonchiffon")) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
        x="Control (no drug) ",
        y="Drug Treatment")
```

## Gene Expression Changes Upon Drug Treatment



```
#assign final plot to object
p_labelled_genes <- p + scale_colour_manual(values=c("gold", "black", "navy")) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
        x="Control (no drug) ",
        y="Drug Treatment")
p_labelled_genes
```

Gene Expression Changes Upon Drug Treatment

