

# Class05R.R

abigailjaquish

2021-10-13

```
#Class 05: Data Visualization 10-13-21  
#Today we are going to use ggplot2 package  
  
#first you need to load the package/library before you can use it  
library(ggplot2)  
ggplot(cars)
```

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
```

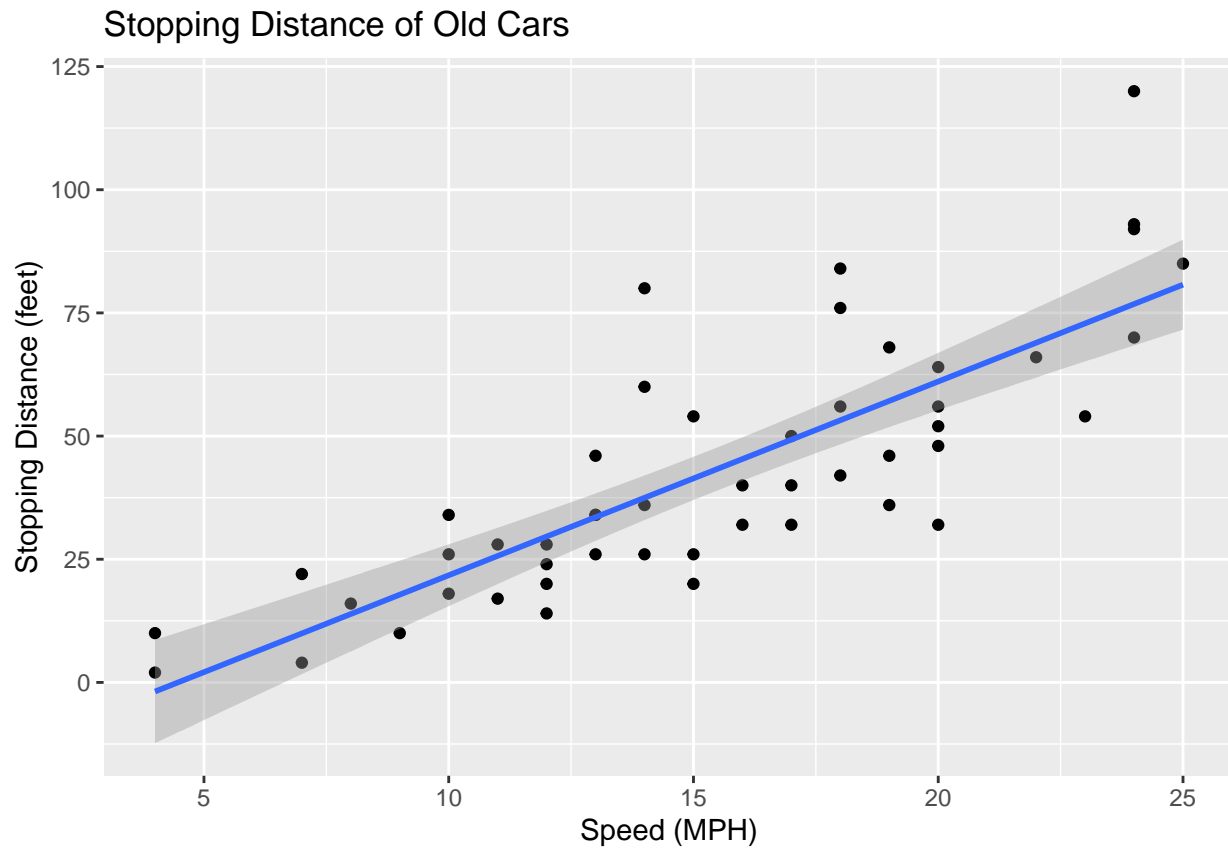
```
#We will use this inbuilt "cars" dataset first
head(cars)
```

```
##   speed dist
## 1     4     2
## 2     4    10
```

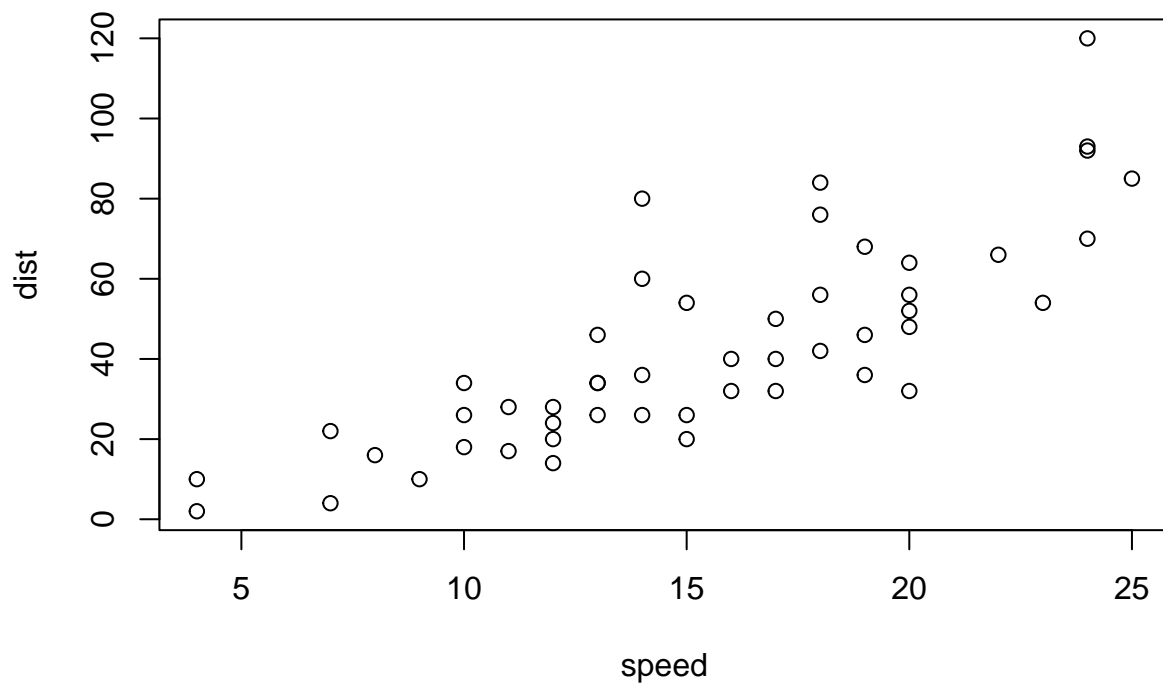
```
## 3    7    4
## 4    7   22
## 5    8   16
## 6    9   10
```

```
#All ggplots have at least 3 layers
# data + aes (aesthetics) + geoms (geometries)
#labs = different labels for features
#geom_line, geom_smooth (method="lm") to make it linear, grey is STerror
ggplot(data=cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm") +
  labs(title="Stopping Distance of Old Cars", x= "Speed (MPH)",
        y="Stopping Distance (feet)")
```

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



```
#Side-note: ggplot is not the only graphics system
# a very popular one is good old "base" R graphics
plot(cars)
```



*#Lab Project Week4*

```
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
```

*#Q. How many genes in the dataset?*

```
nrow(genes)
```

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

*#What are the column names and how many there are?*

```
colnames(genes)
```

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

```
ncol(genes)
```

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

```
#How many upregulated genes in the State column?
```

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

```
##
```

```
##      down  unchanged      up
```

```
##      72      4997      127
```

```
#What fraction of total genes is upregulated?
```

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

```
##
```

```
##      down  unchanged      up
```

```
##      1.39      96.17      2.44
```

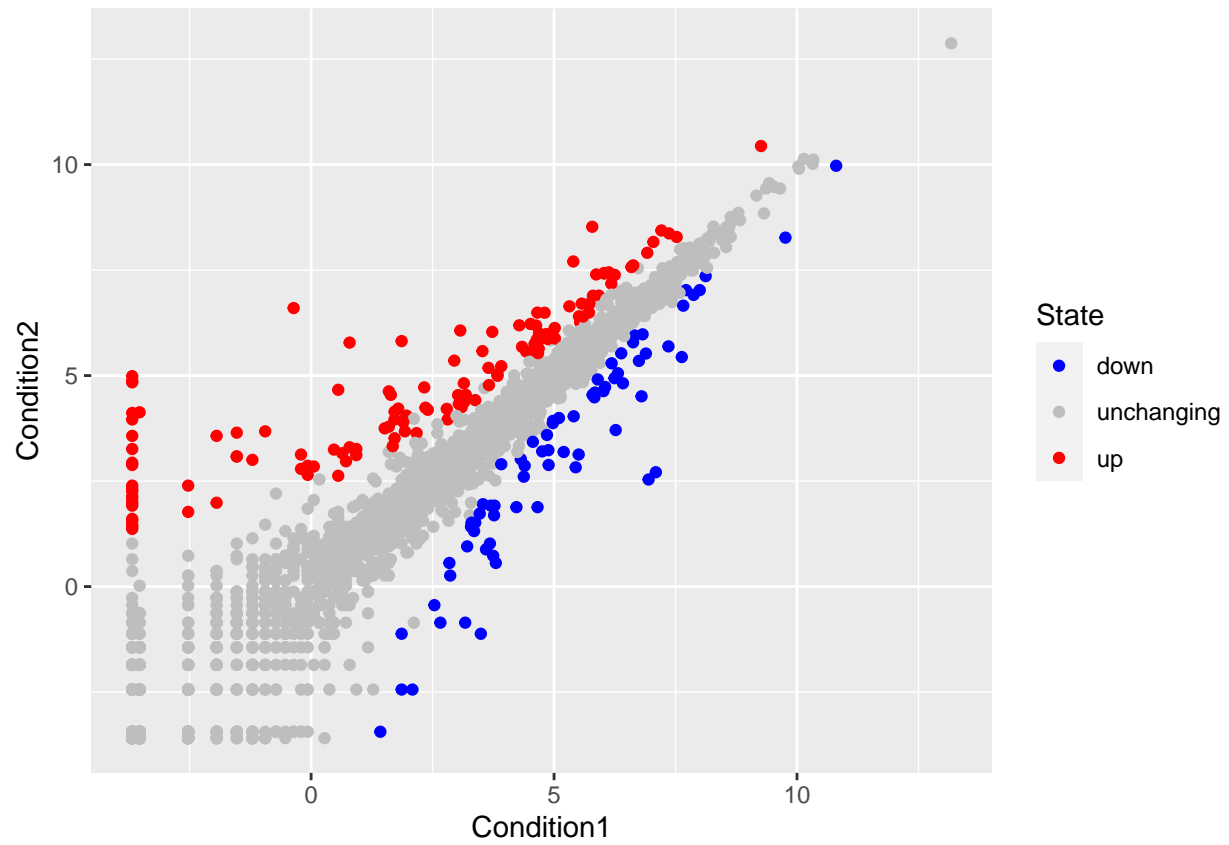
```
#Use the genes dataset in ggplot function
```

```
#set the x and y aesthetic mappings to the Condition1 and Condition2
```

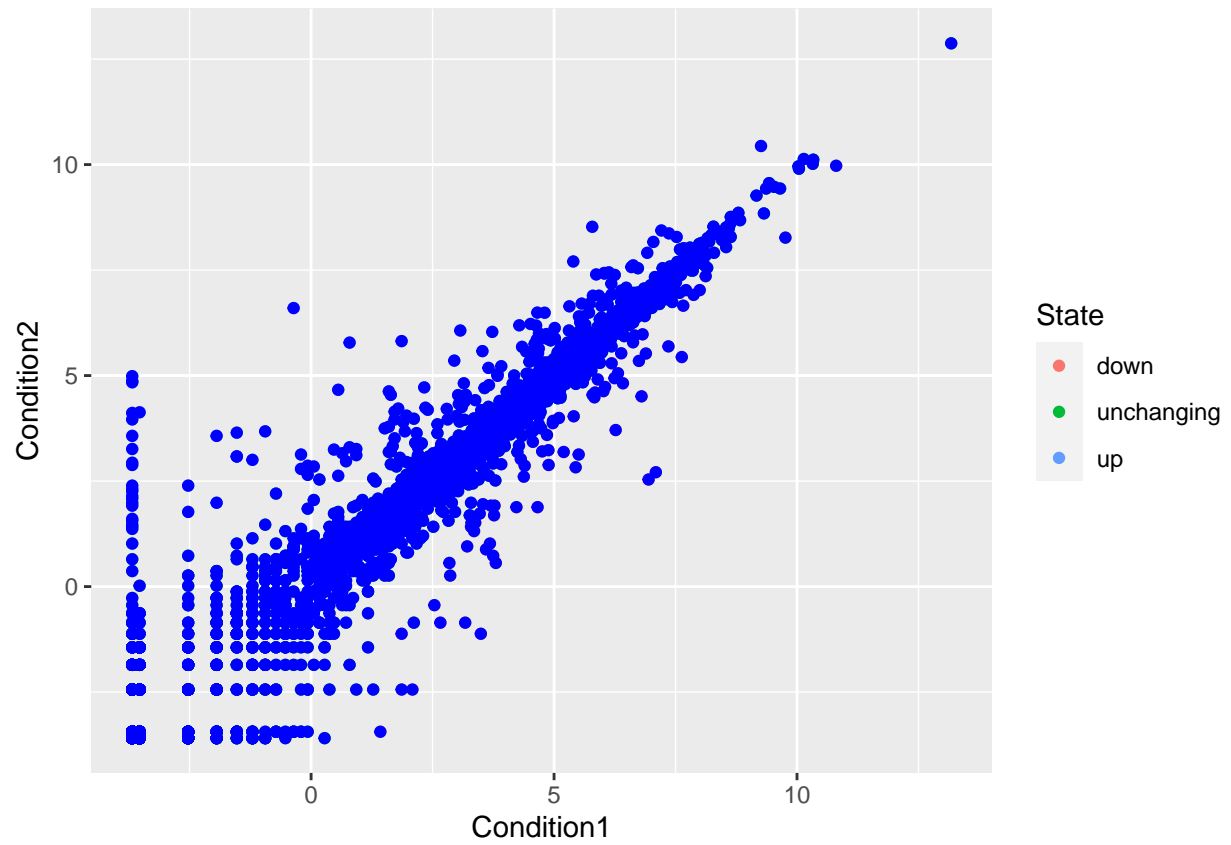
```
p <- ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) +  
  geom_point()
```

```
#Adding new colors
```

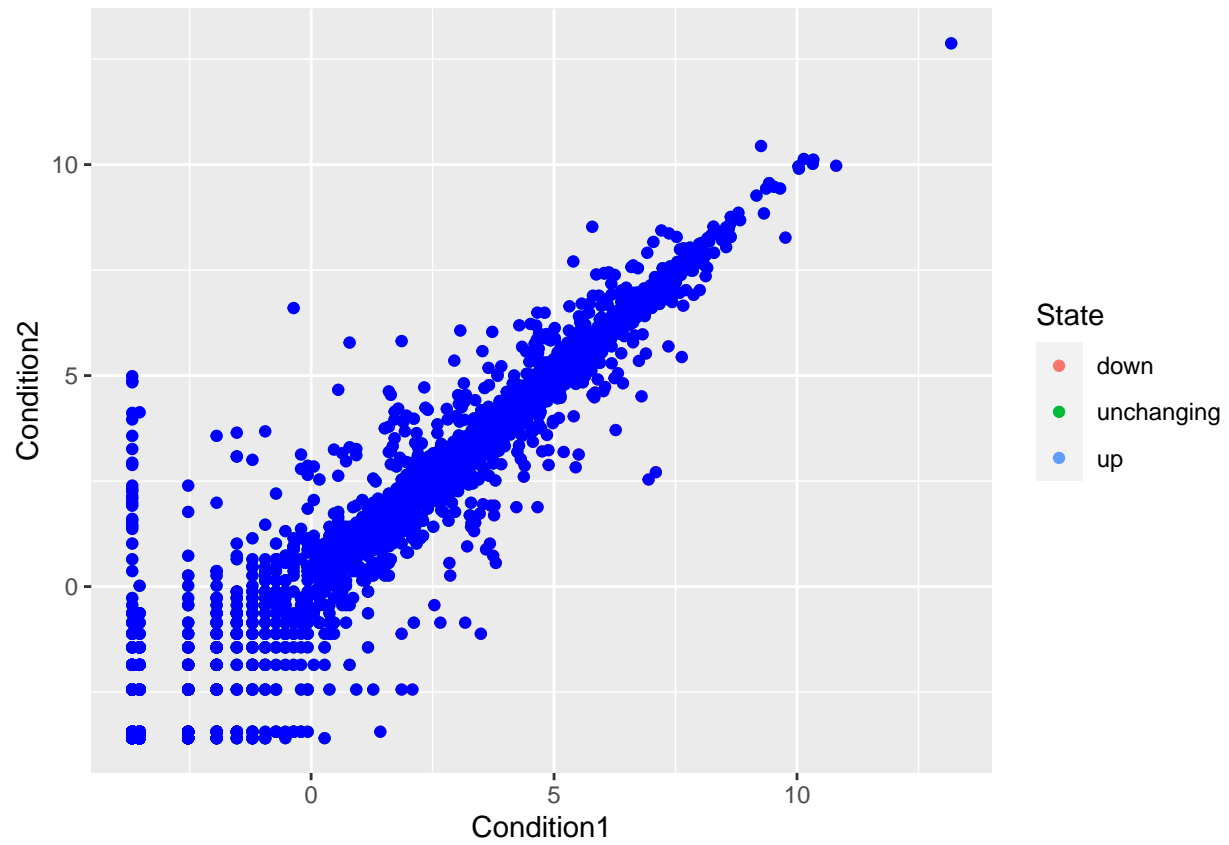
```
p + scale_colour_manual( values=c("blue", "gray", "red"))
```



```
p + geom_point(col="blue")
```

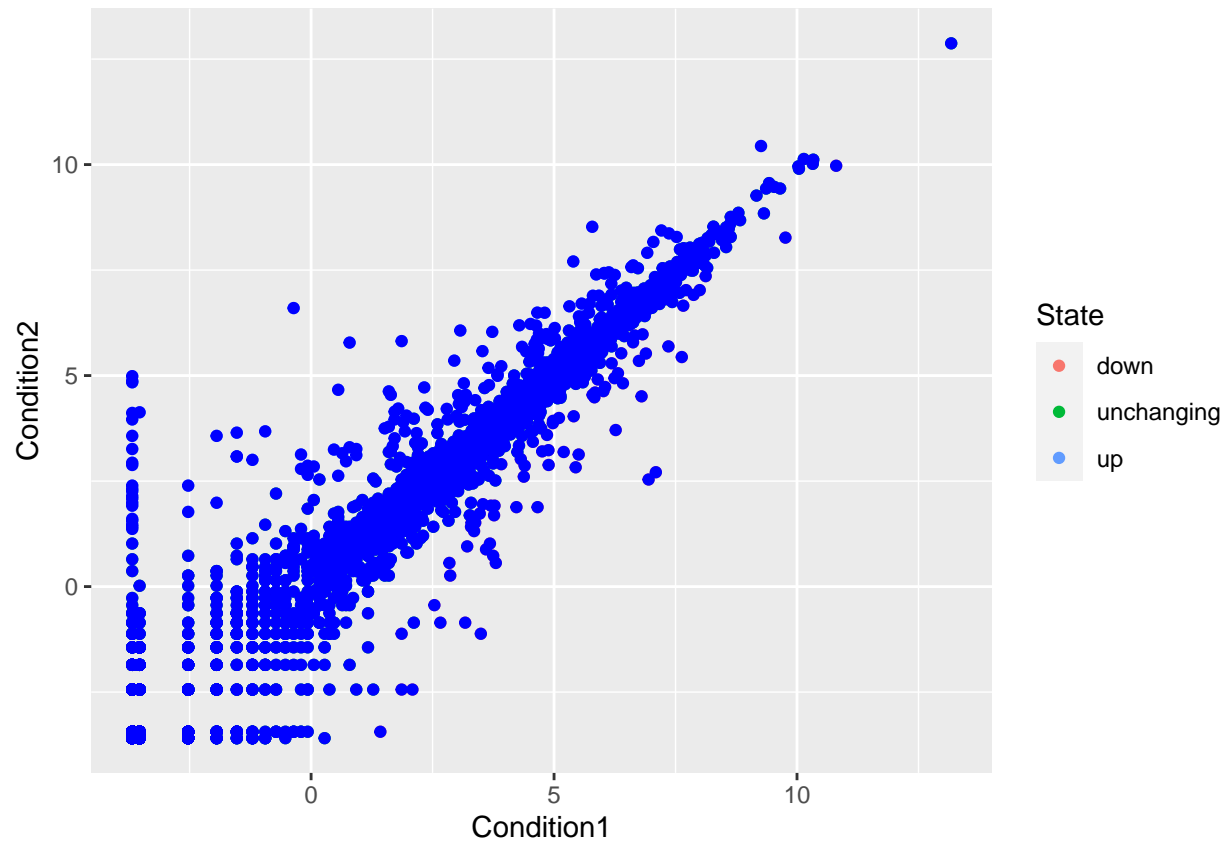


```
p + aes(col=State) + geom_point(col="blue")
```



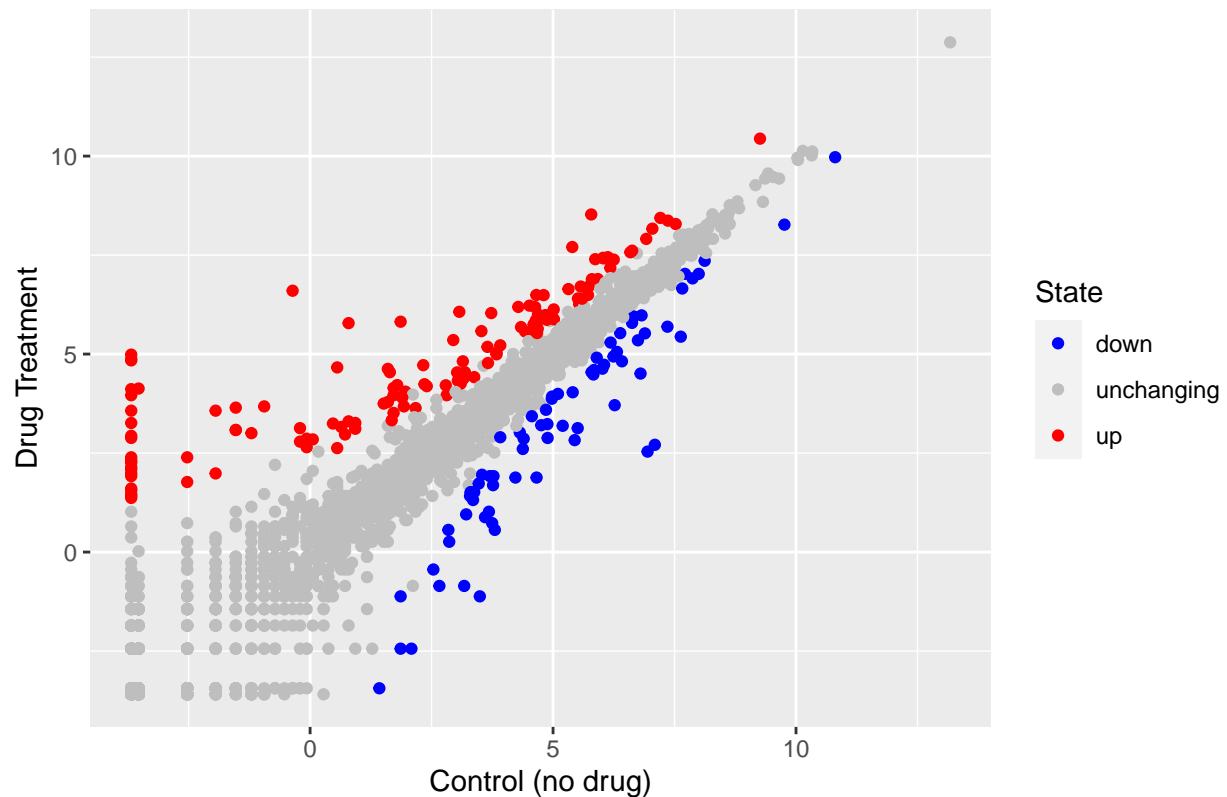
```
p + geom_point(col="blue") + aes(col=State)
```





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

## Gene Expression Changes Upon Drug Treatment

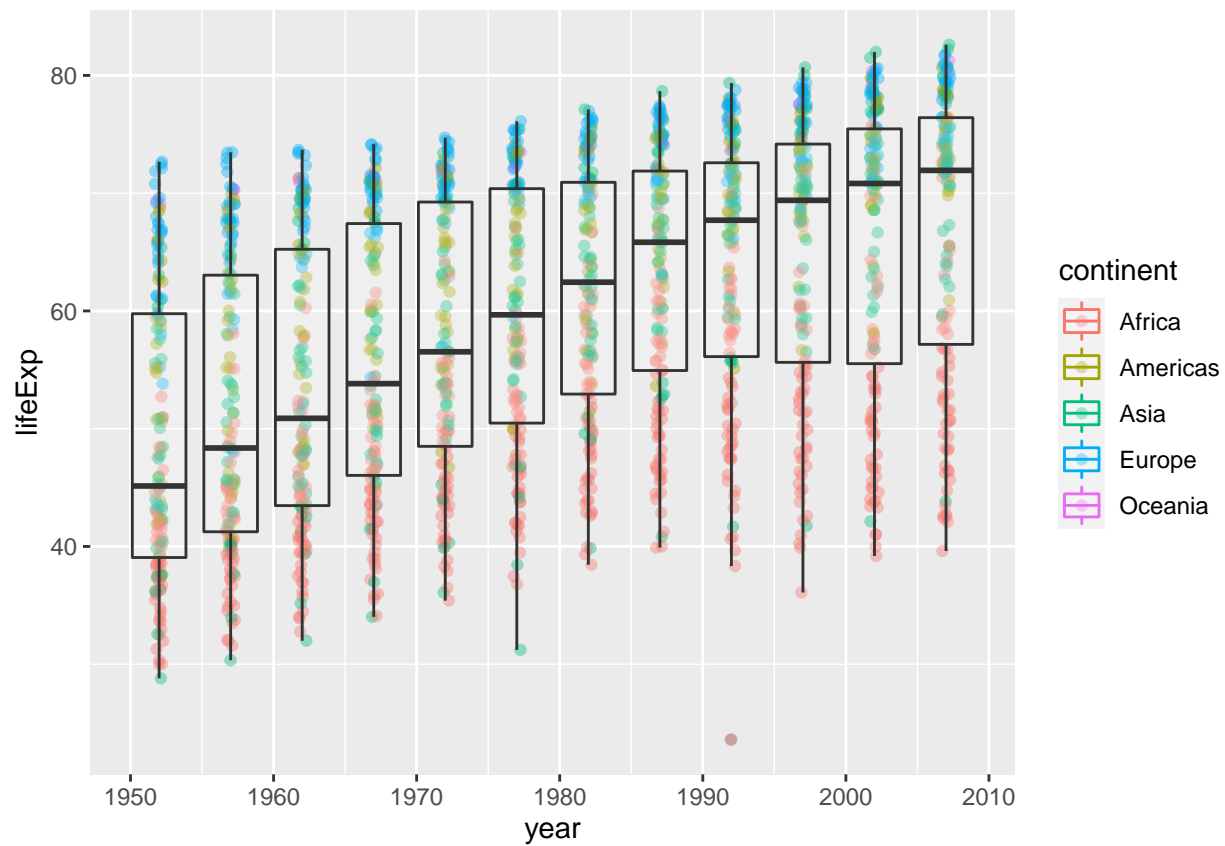


```
#6. Optional--Let's explore gapminder (a dataset)
#install.packages("gapminder")
library(gapminder)
# File location online
#url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.tsv"

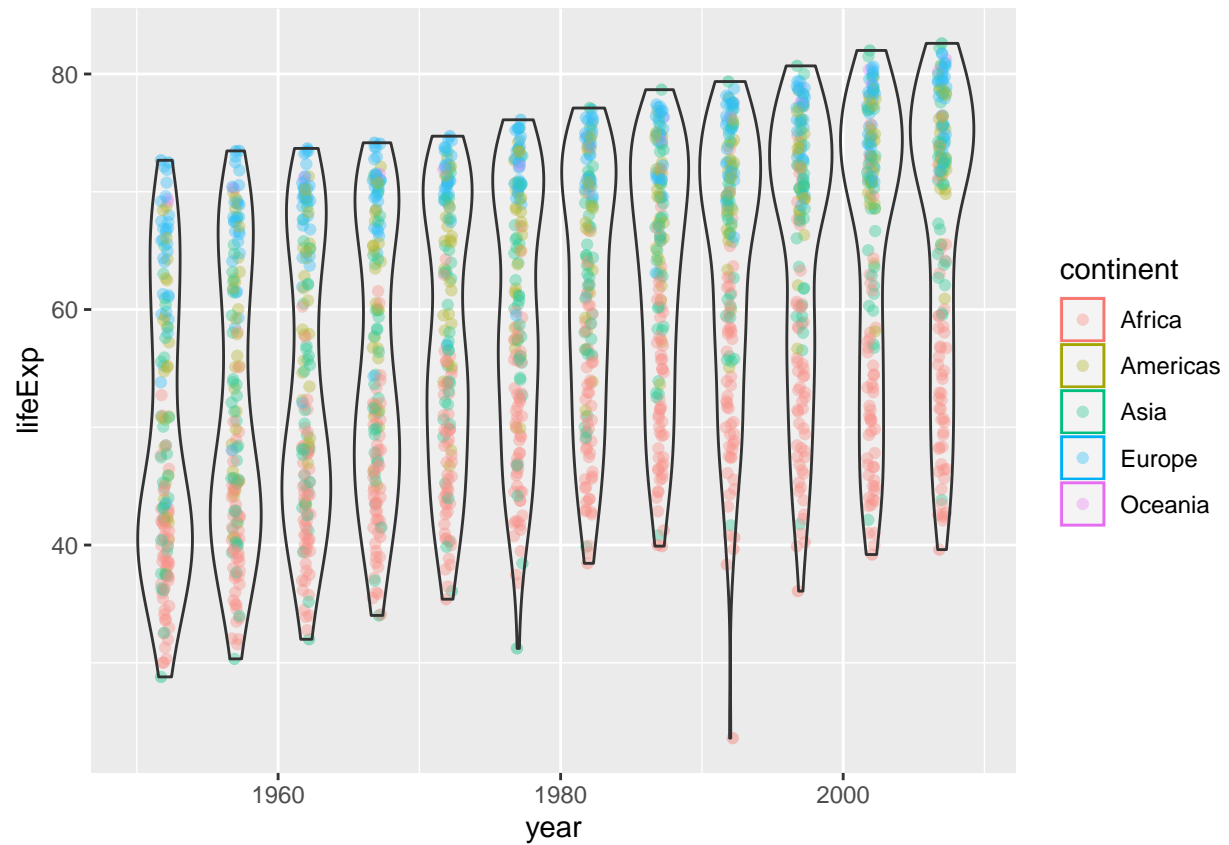
#gapminder <- read.delim(url)
library(gapminder)
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country    continent  year lifeExp    pop gdpPercap
##   <fct>      <fct>    <int> <dbl>    <int>    <dbl>
## 1 Afghanistan Asia      1952  28.8  8425333    779.
## 2 Afghanistan Asia      1957  30.3  9240934    821.
## 3 Afghanistan Asia      1962  32.0 10267083    853.
## 4 Afghanistan Asia      1967  34.0 11537966    836.
## 5 Afghanistan Asia      1972  36.1 13079460    740.
## 6 Afghanistan Asia      1977  38.4 14880372    786.
```

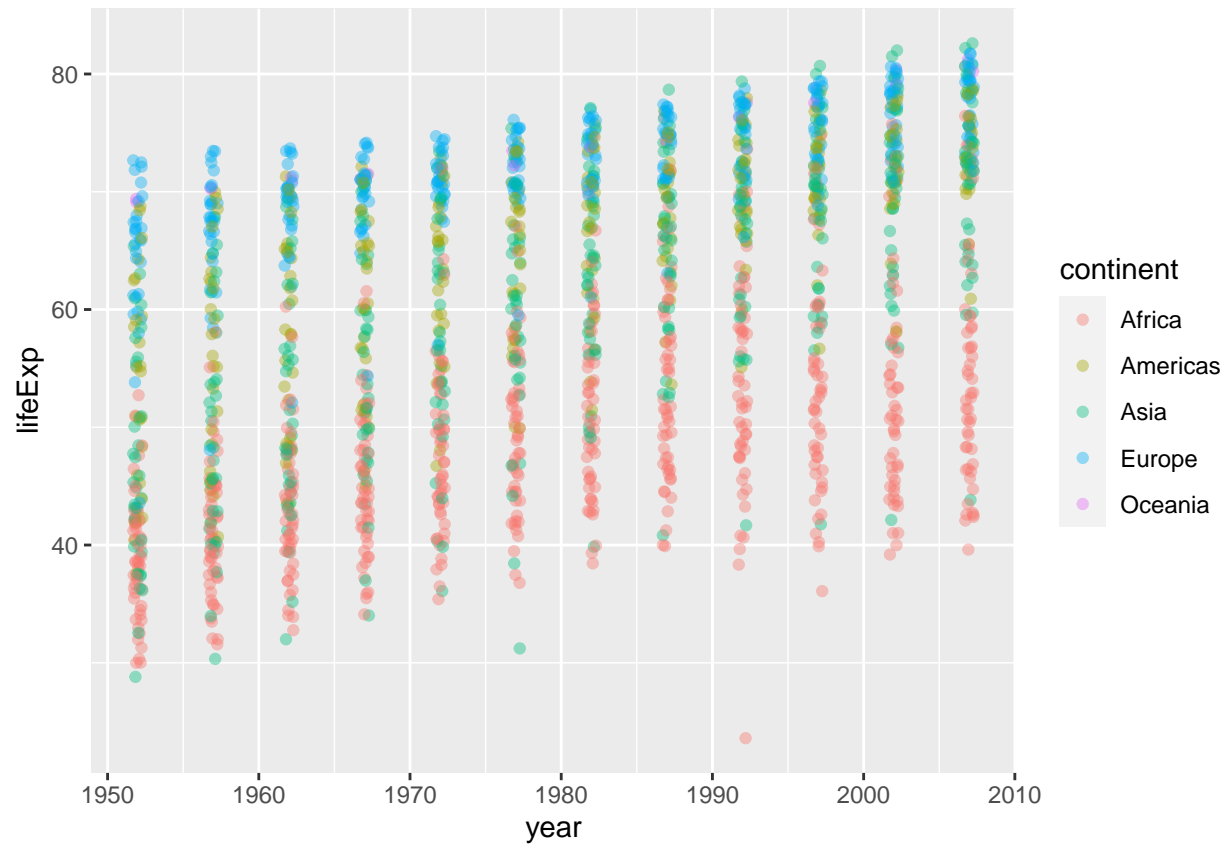
```
#Lets make a new plot of year vs lifeexp
#alpha=transparency
ggplot(gapminder) +
  aes(x=year, y=lifeExp, col=continent) +
  geom_jitter(width=0.3,alpha=0.4) +
  geom_boxplot(aes(group=year), alpha=0.2)
```



```
ggplot(gapminder) +  
  aes(x=year, y=lifeExp, col=continent) +  
  geom_jitter(width=0.3,alpha=0.4) +  
  geom_violin(aes(group=year), alpha=0.2)
```



```
ggplot(gapminder) +
  aes(x=year, y=lifeExp, col=continent) +
  geom_jitter(width=0.3,alpha=0.4)
```



```
#geom_violin(aes(group=year), alpha=0.2,  
              #draw_quantiles=0.5  
  
#Install the plotly  
#install.packages("plotly")  
  
#Plot Animation  
#install.packages("gifski")  
#install.packages("gganimate")
```