class5.R

sanluc

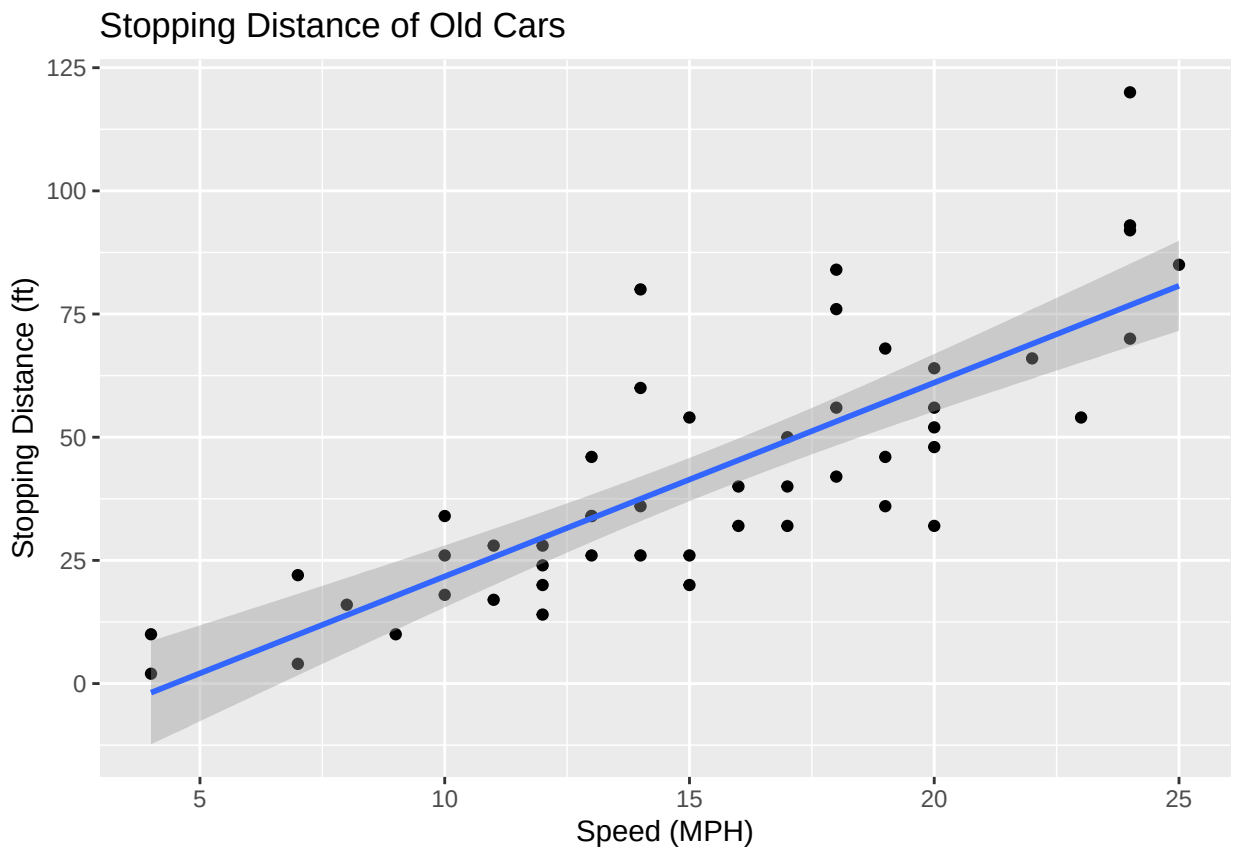
2021-10-13

```
#class 05: Data Visualization

# Today we are going to use ggplot2 package

# Remember to load the package from the library
# If you haven't download it,
# Enter the command to download in the console,
# Not in the script, or else it will download every single time

library(ggplot2)

# We are using the inbuilt dataset "cars" to work on today ggplot assignment.
head(cars)
```

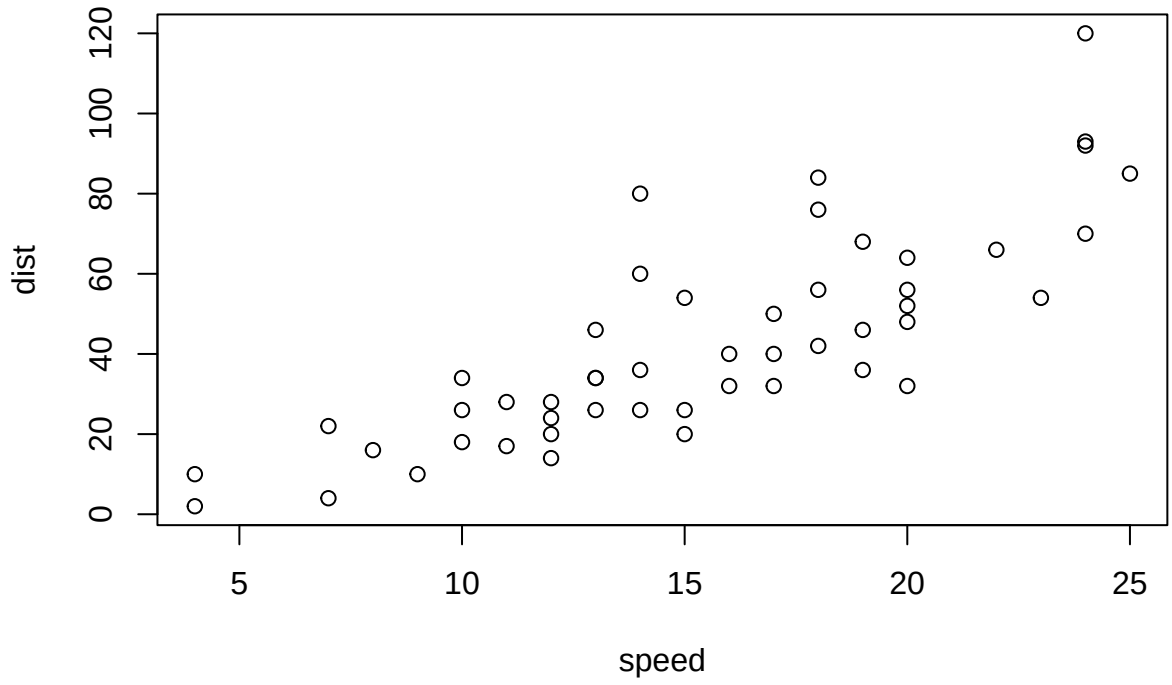
```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```
# All ggplots have at least 3 layers
# dataset + aes + geoms function

ggplot(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 (ft)")
```

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




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




```
# However ggplot is still easier to use

#Extra
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
```

```
# To look at data.frame genes
View(genes)

# Number of rows in data.frame genes
nrow(genes)
```

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

```r
# Columns names
colnames(genes)
```

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

```r
# Number of column in data.frame genes
ncol(genes)
```

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

```r
# To find the total number of genes according to its state
table(genes$State)
```

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

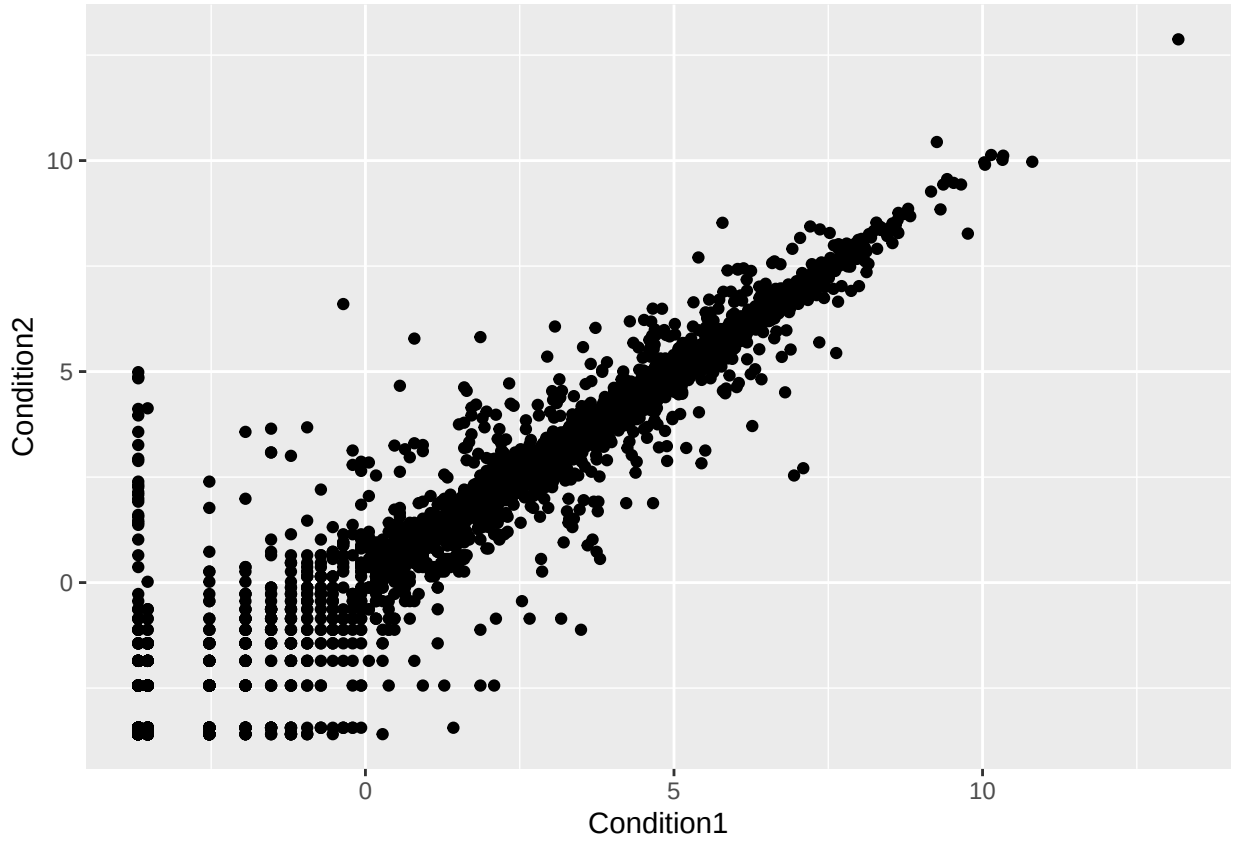
```r
# To find the percentage of up genes
127/5196
```

```
## [1] 0.02444188
```

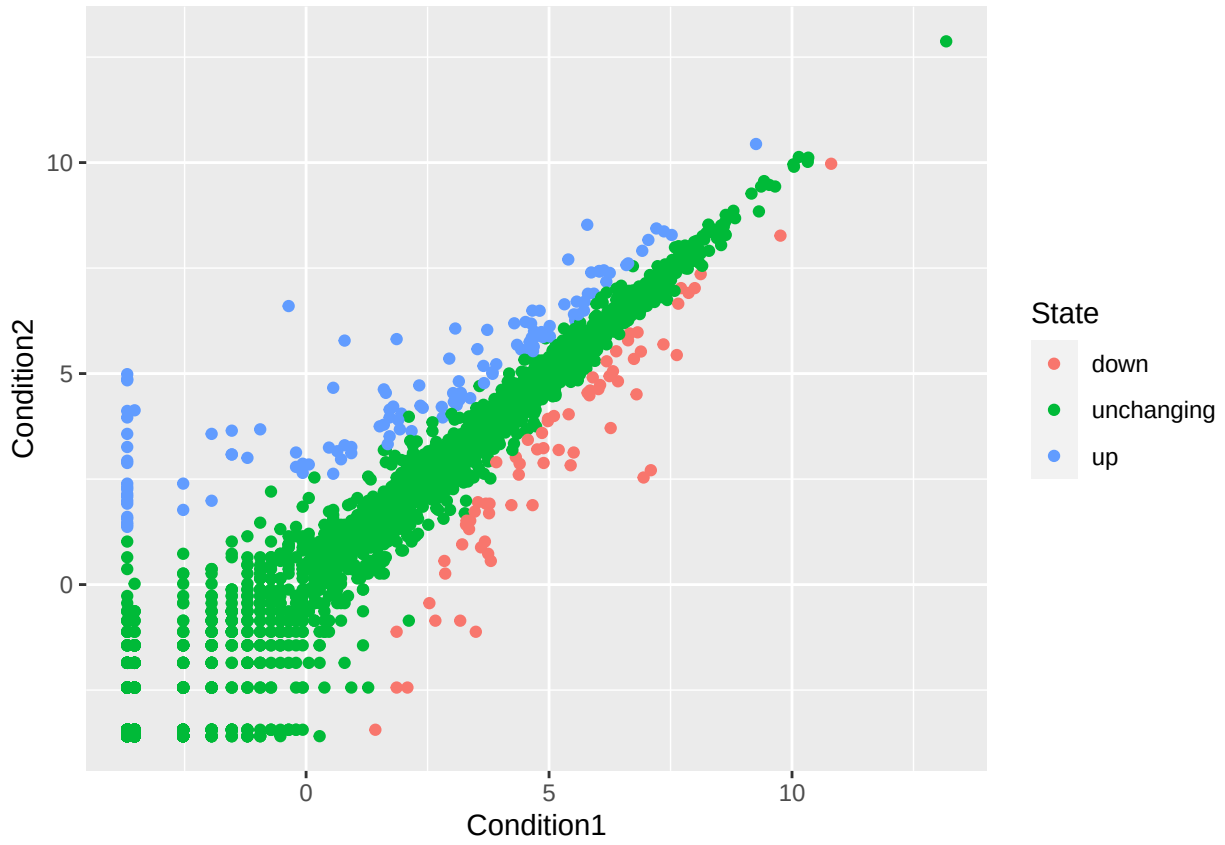
```r
  # We can use the math function like above, or use table
round( table(genes$State)/nrow(genes) * 100, 3)
```

```
##
##       down unchanging         up
##      1.386     96.170      2.444
```

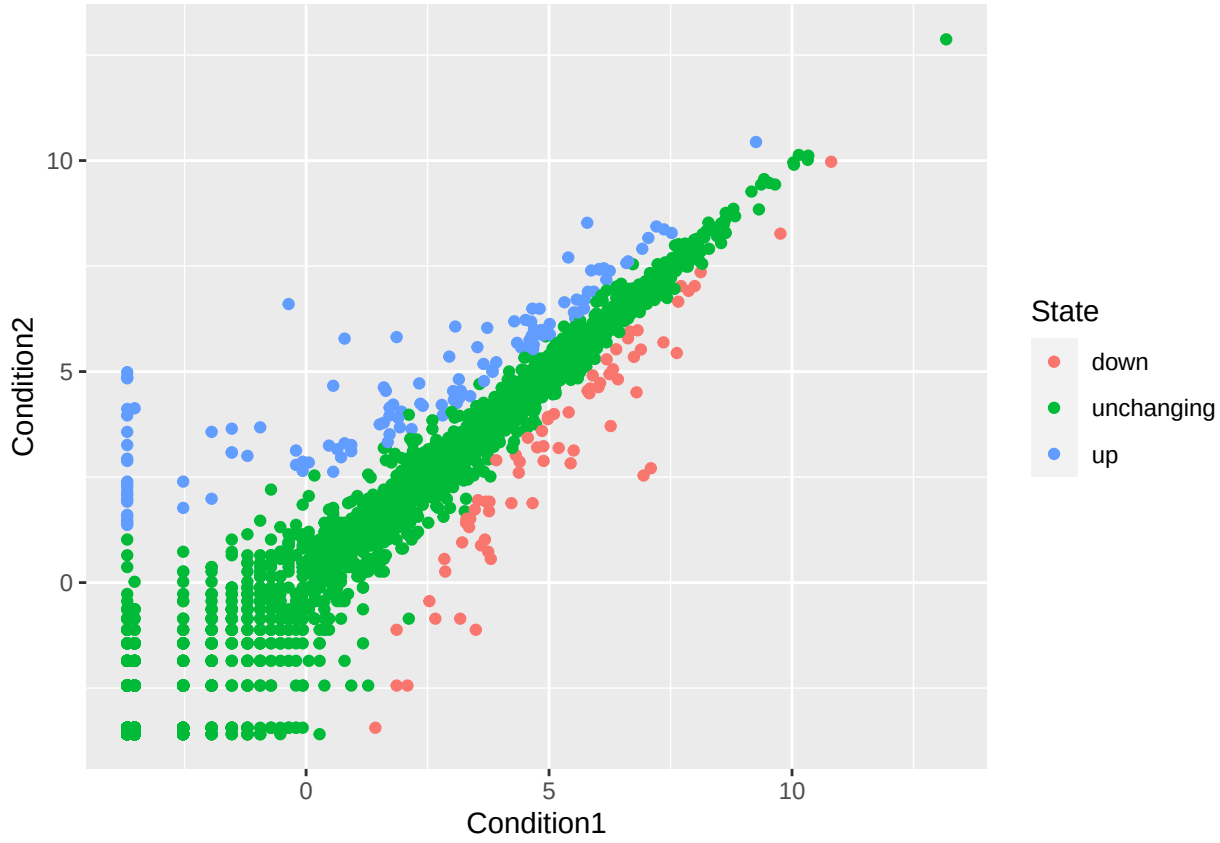
```r
# Making figure- simple dot plot
ggplot(genes) +
  aes(x=Condition1,y=Condition2) +
  geom_point()
```



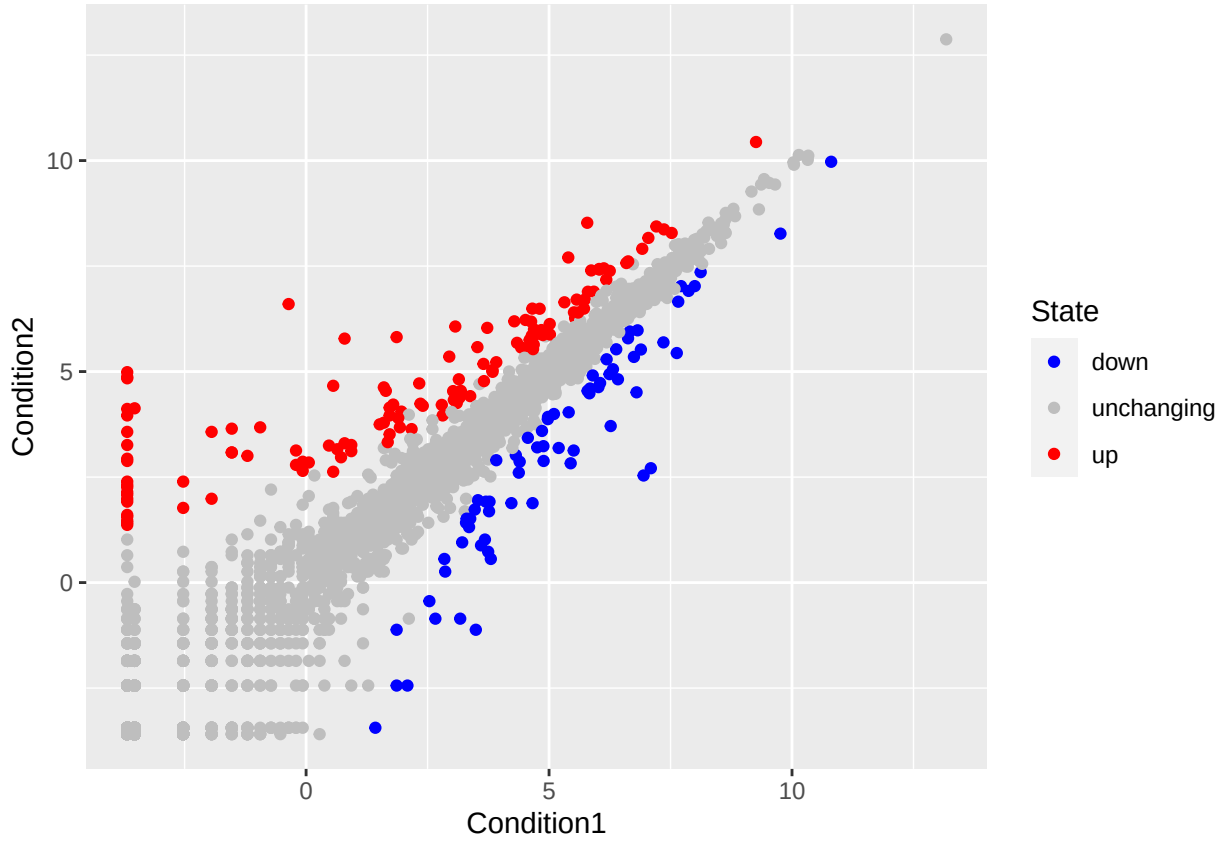

```
# Making figure- color by state
ggplot(genes) +
  aes(x=Condition1,y=Condition2, col=State) +
  geom_point()
```



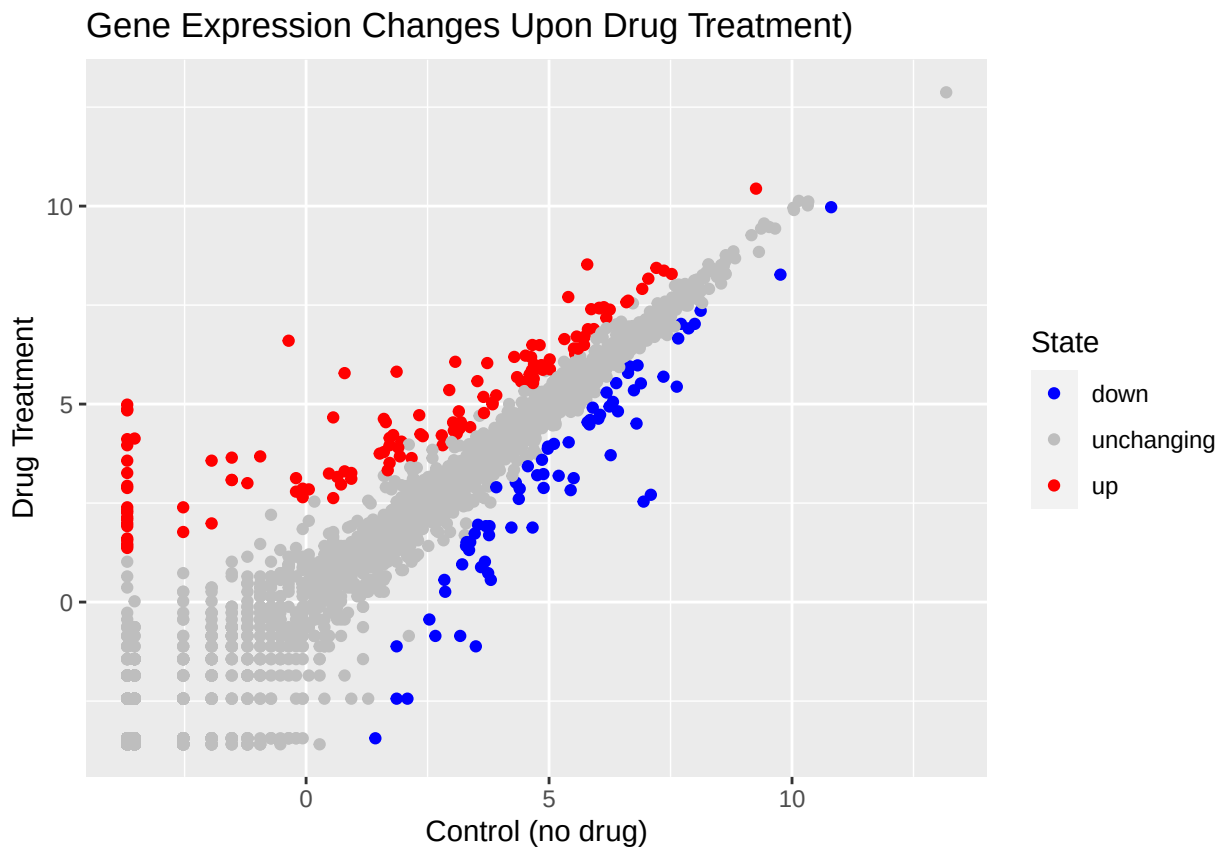

```
  # We can also add it outside (as a separate function)
ggplot(genes) +
  aes(x=Condition1,y=Condition2) +
  aes(col=State) +
  geom_point()
```




```
# Making figure- changing colors manually
ggplot(genes) +
  aes(x=Condition1,y=Condition2,col=State) +
  geom_point() +
  scale_color_manual( values=c("blue", "gray","red"))
```



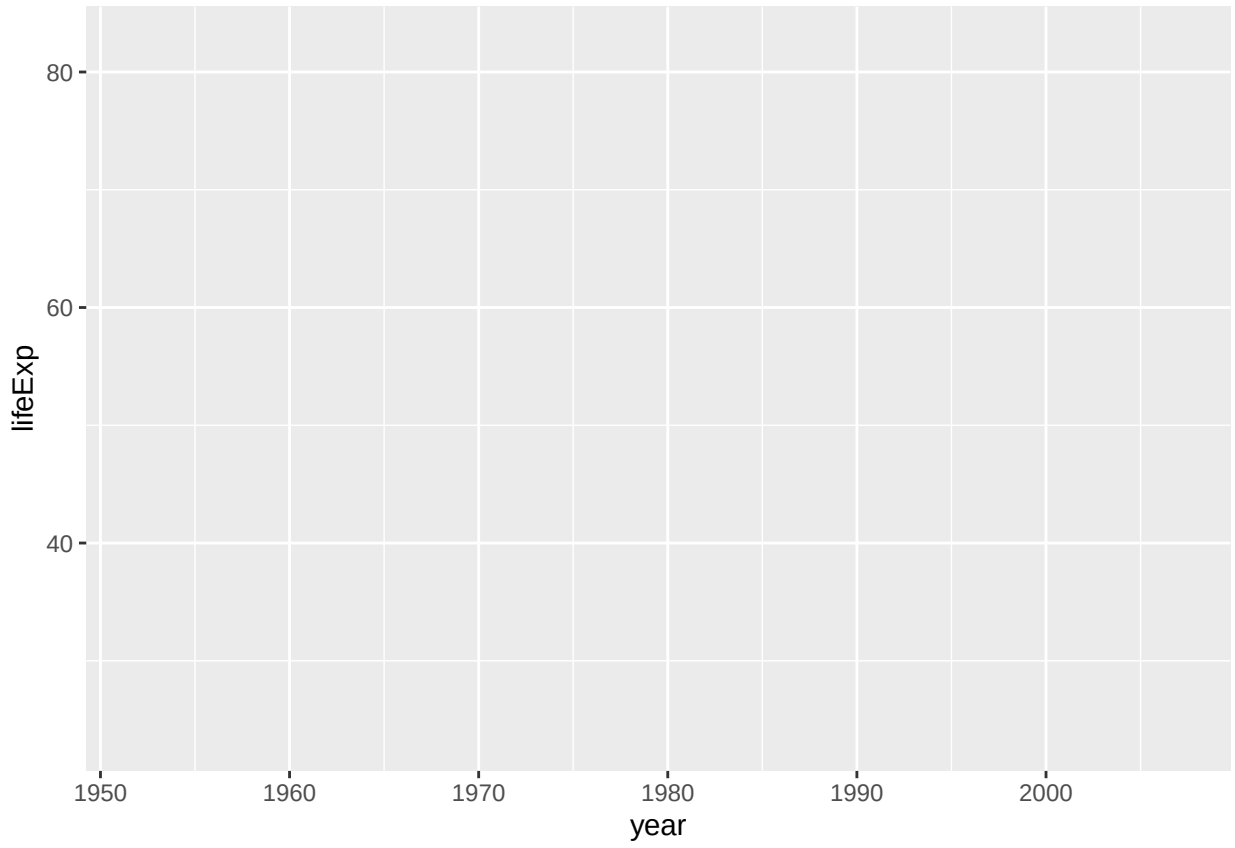

```
# Making figure- adding labels

ggplot(genes) +
  aes(x=Condition1,y=Condition2,col=State) +
  geom_point() +
  scale_color_manual( values=c("blue", "gray","red")) +
  labs(title="Gene Expression Changes Upon Drug Treatment)", x="Control (no drug)", y="Drug Treatment")
```



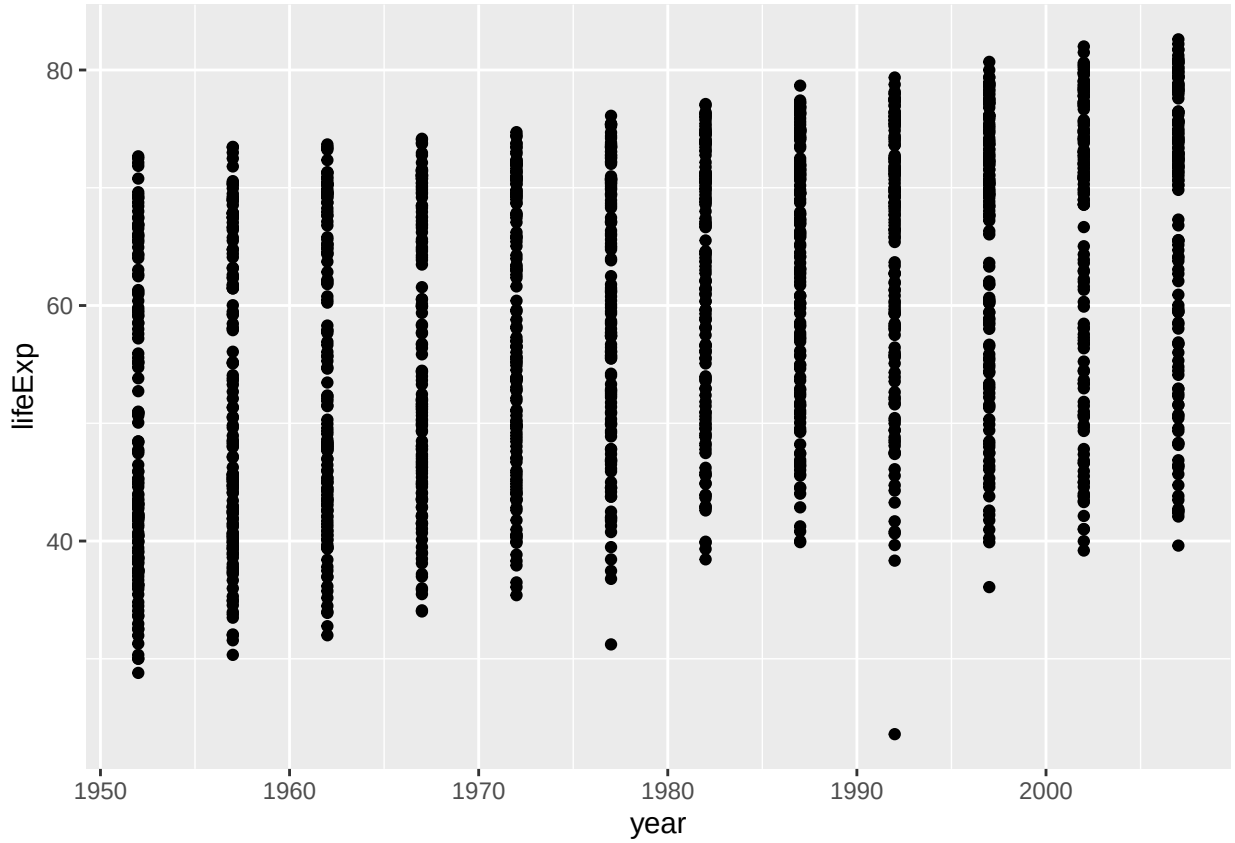

```
# Let's explore the gapminder dataset
# install.packages("gapminder")
library(gapminder)
head(gapminder)
```

```
## # 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.
```

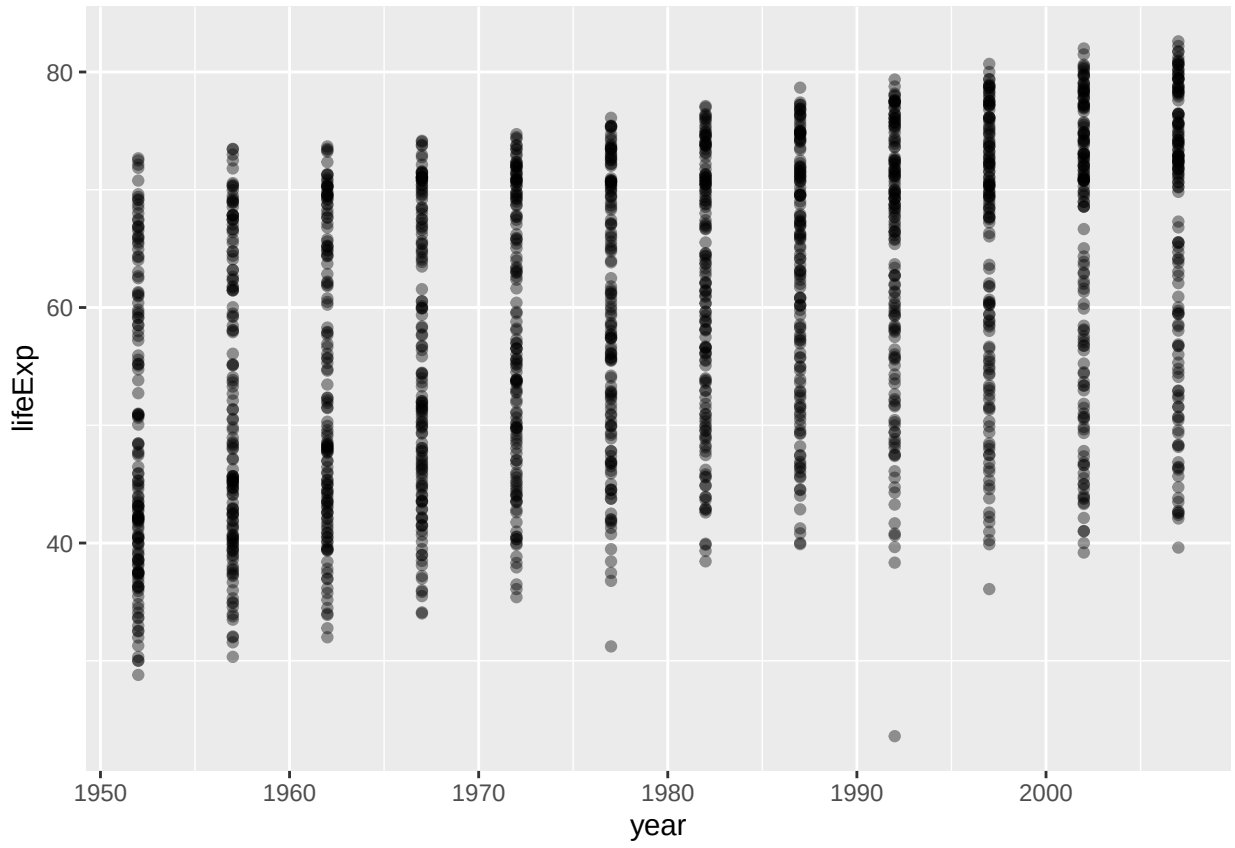
```
# Let's make a new plot of year vs lifeExp (with no data yet)
ggplot(gapminder) +
  aes(x=year, y=lifeExp)
```



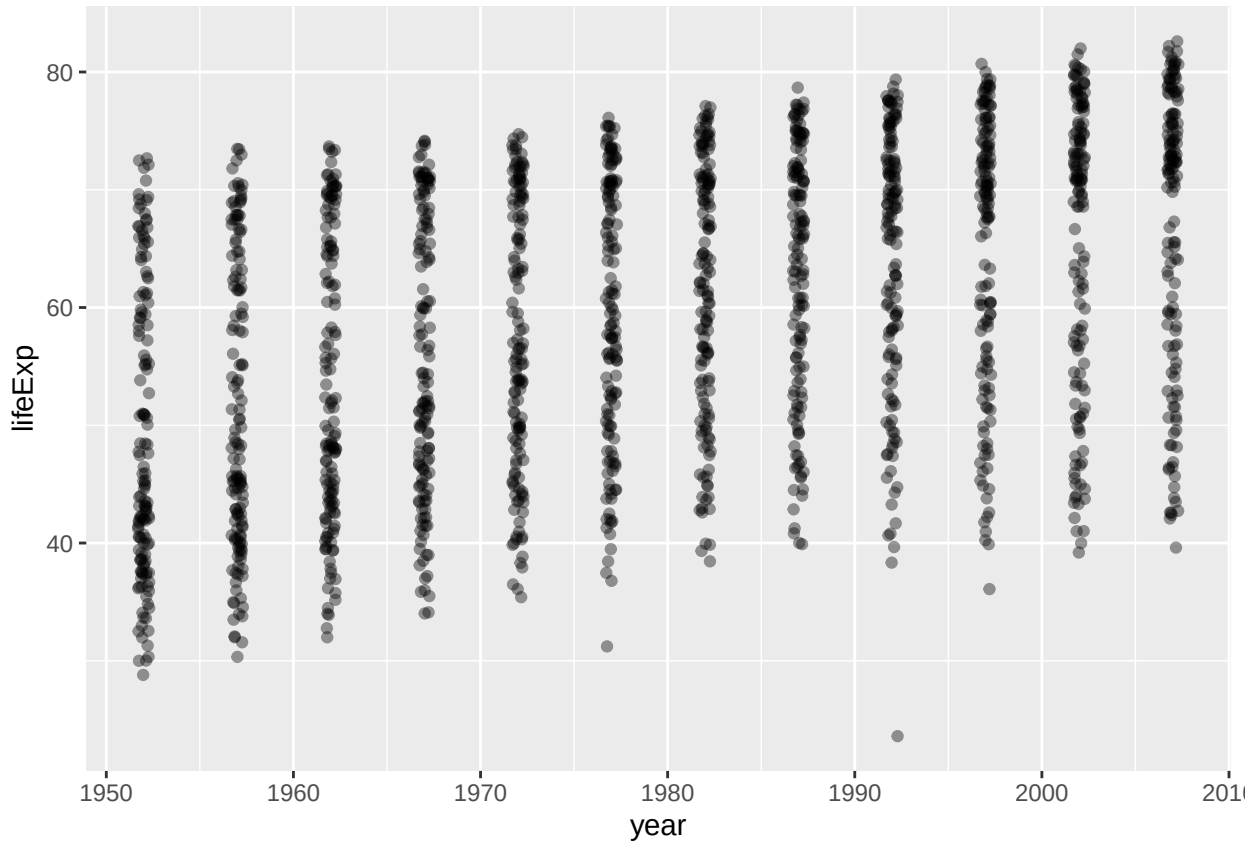

```
# To add data in (simple dot plot)
ggplot(gapminder) +
  aes(x=year, y=lifeExp) +
  geom_point()
```



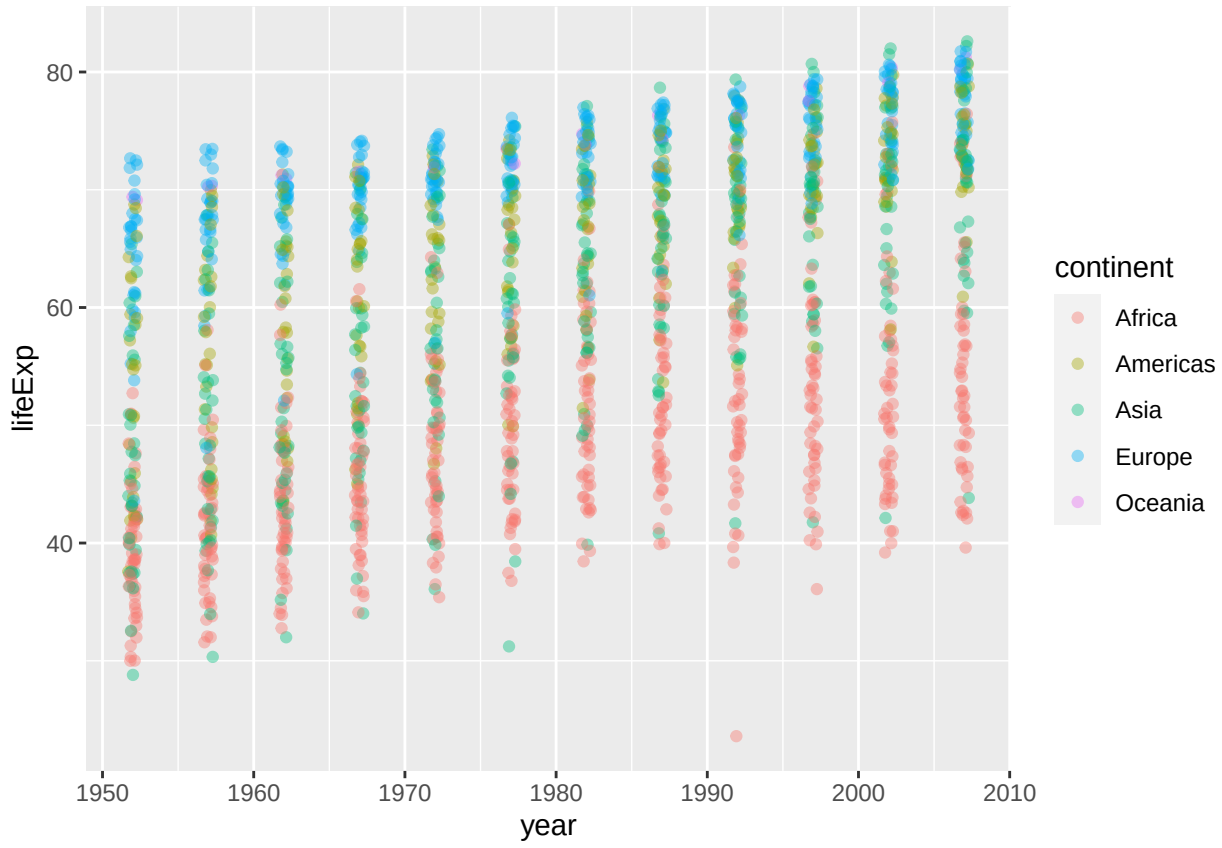

```
# It's hard to see, so maybe we should change the transparency using alpha
ggplot(gapminder) +
  aes(x=year, y=lifeExp) +
  geom_point(alpha=0.4)
```



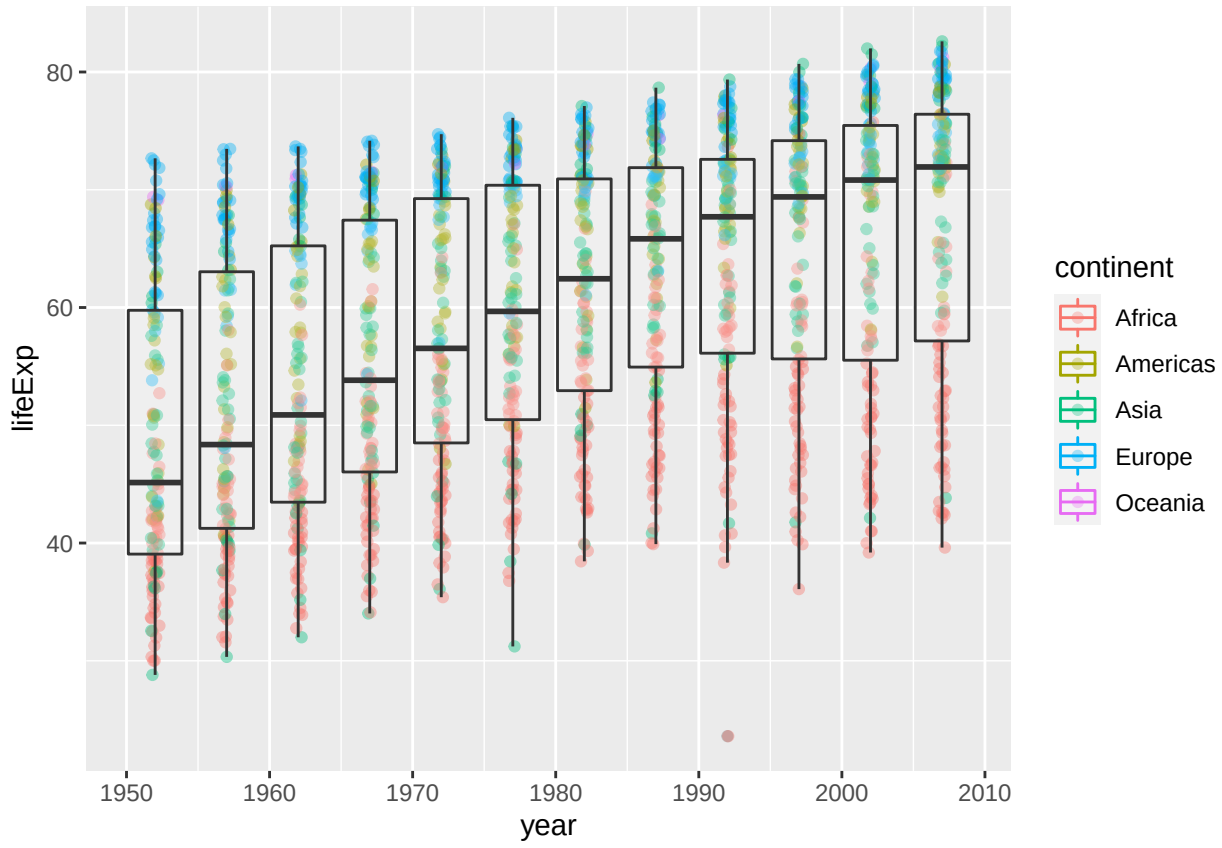

```
# It's still really hard to see, so maybe we can try something else?
# Maybe we can try jitter, which is for a lot of overlapping/clustered data
ggplot(gapminder) +
  aes(x=year, y=lifeExp) +
  geom_jitter(width = 0.3, alpha = 0.4)
```



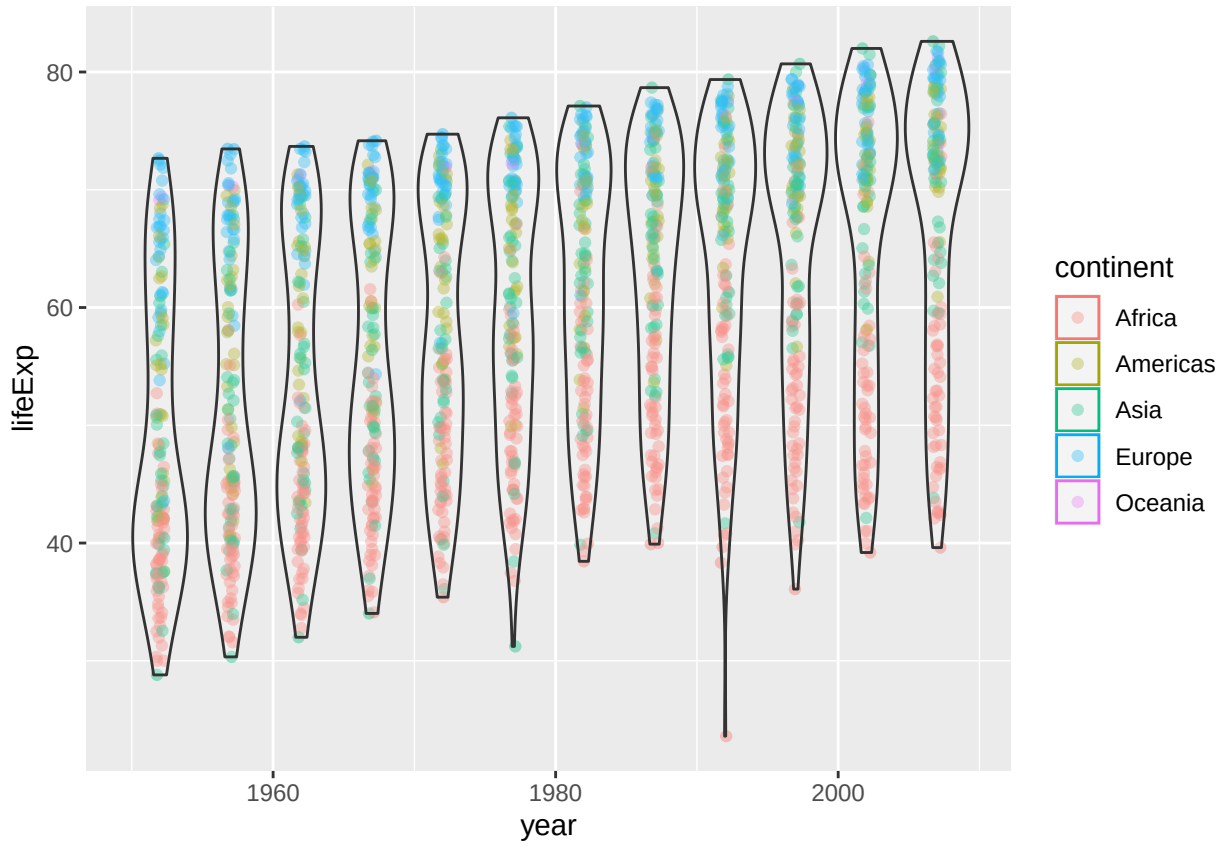

```
# We can also add color in for easy visualization- defining by a different category
ggplot(gapminder) +
  aes(x=year, y=lifeExp, col=continent) +
  geom_jitter(width = 0.3, alpha = 0.4)
```



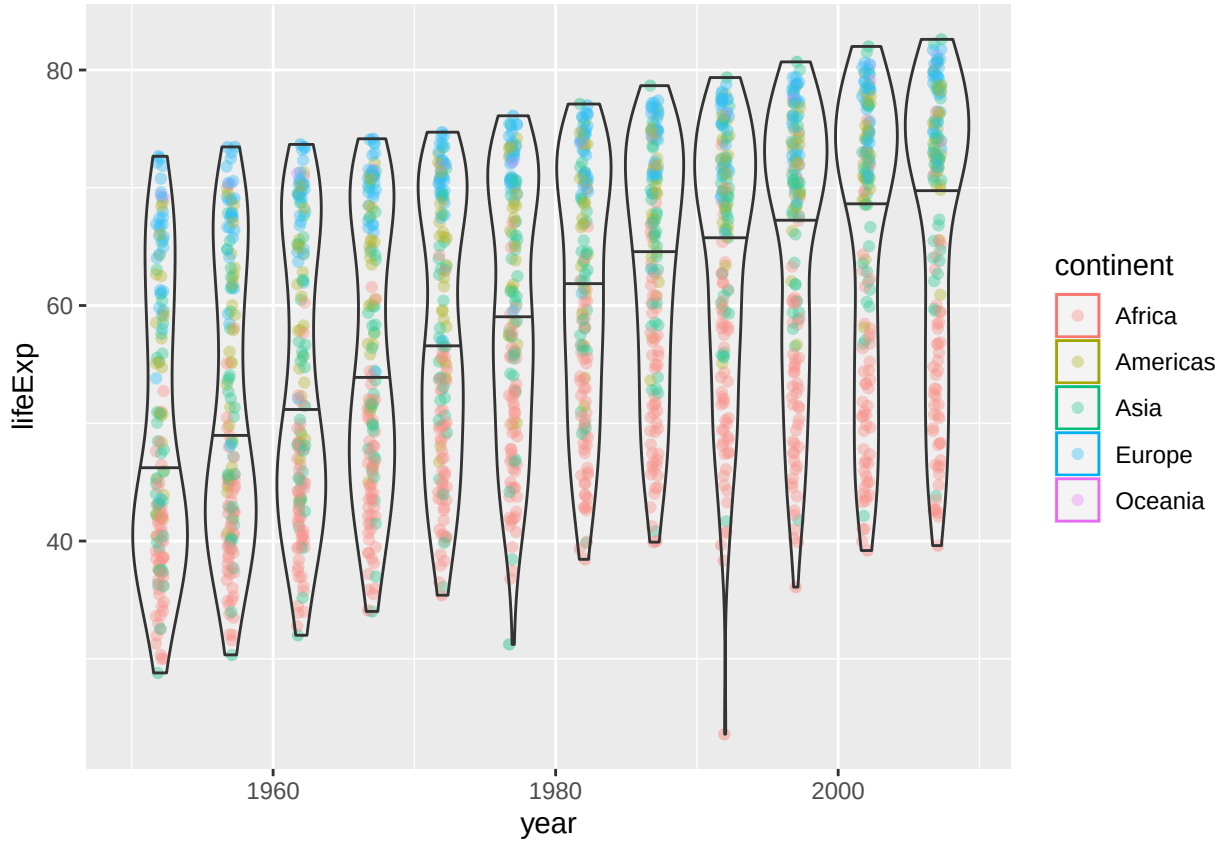

```
#Let's try a boxplot and change it transparency
ggplot(gapminder) +
  aes(x=year, y=lifeExp, col=continent) +
  geom_jitter(width = 0.3, alpha = 0.4) +
  geom_boxplot(aes(group=year),alpha=0.2)
```




```
# let's try the violin plot with transperency
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)
```




```
p <- ggplot(gapminder) +
  aes(x=year, y=lifeExp, col=continent)

# let's try the violin plot with transperency and median
p + geom_jitter(width = 0.3, alpha=0.4) +
  geom_violin(aes(group=year),alpha=0.2,draw_quantiles=0.5)
```




```
# Install the plotly package
 #install.packages("plotly")
 #library(plotly)
 #ggplotly()
```