

# class05.R

maywu

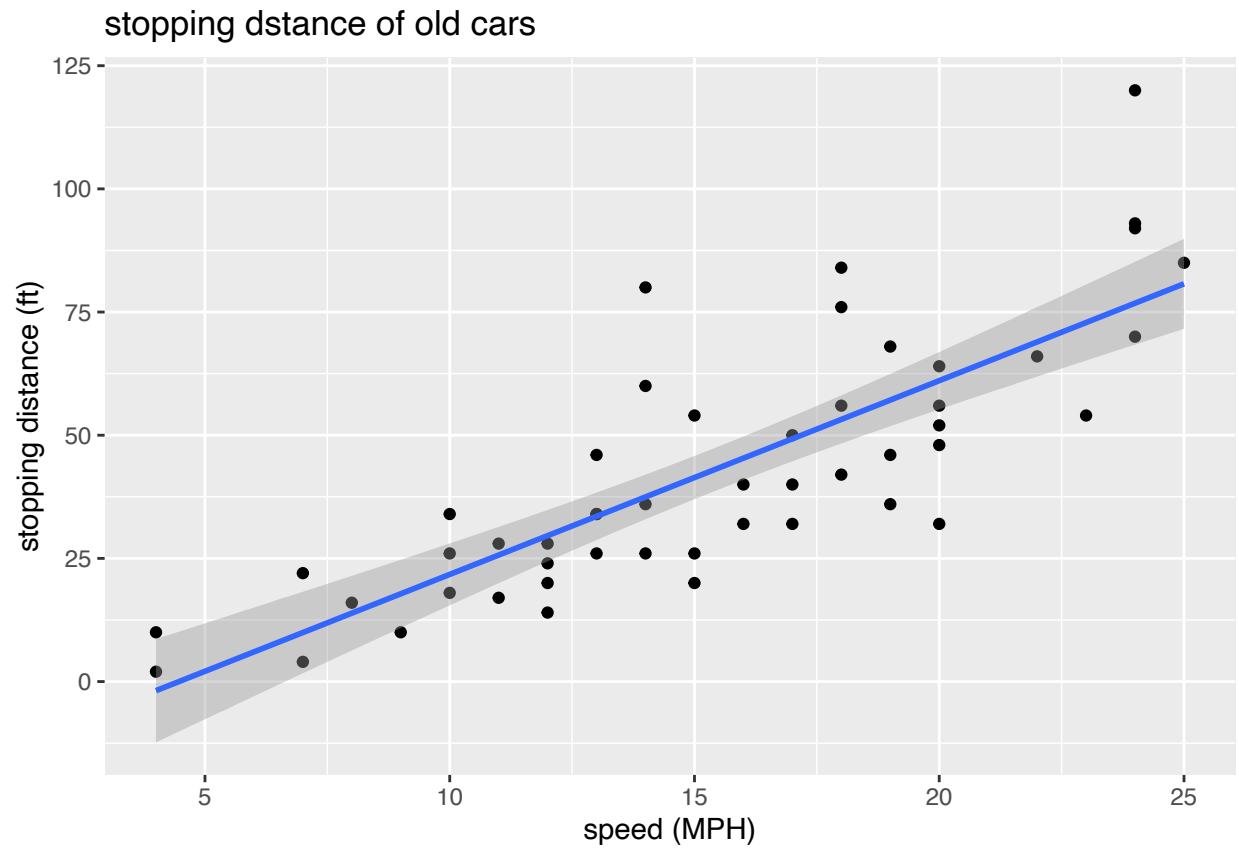
2021-10-13

```
#Class 05: Data visualization  
# use ggplot2 package  
library(ggplot2) # load the package  
head(cars)
```

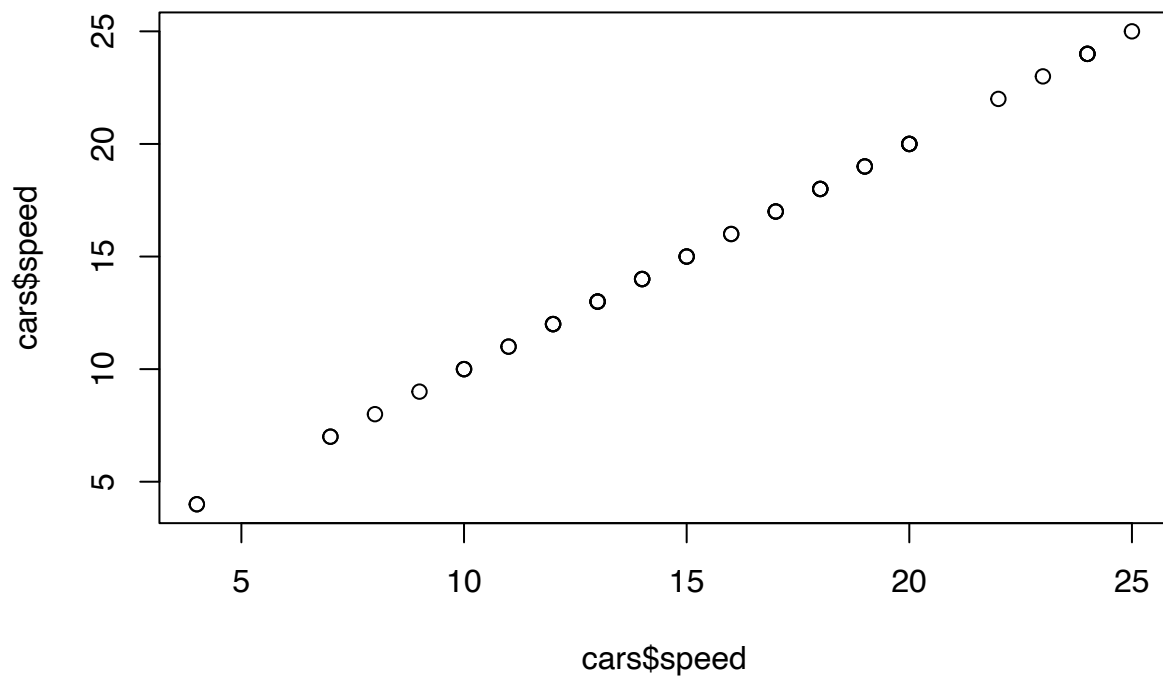
```
##   speed dist  
## 1     4    2  
## 2     4   10  
## 3     7    4  
## 4     7   22  
## 5     8   16  
## 6     9   10
```

```
# all ggplots have at least 3 layers  
#data + aes + geoms  
ggplot(data = cars) + aes(x = speed, y = dist) +  
  geom_point() +  
  # geom_line() +  
  geom_smooth(method = "lm") +  
  labs(title = "stopping distance of old cars",  
        x = "speed (MPH)",  
        y = "stopping distance (ft)")
```

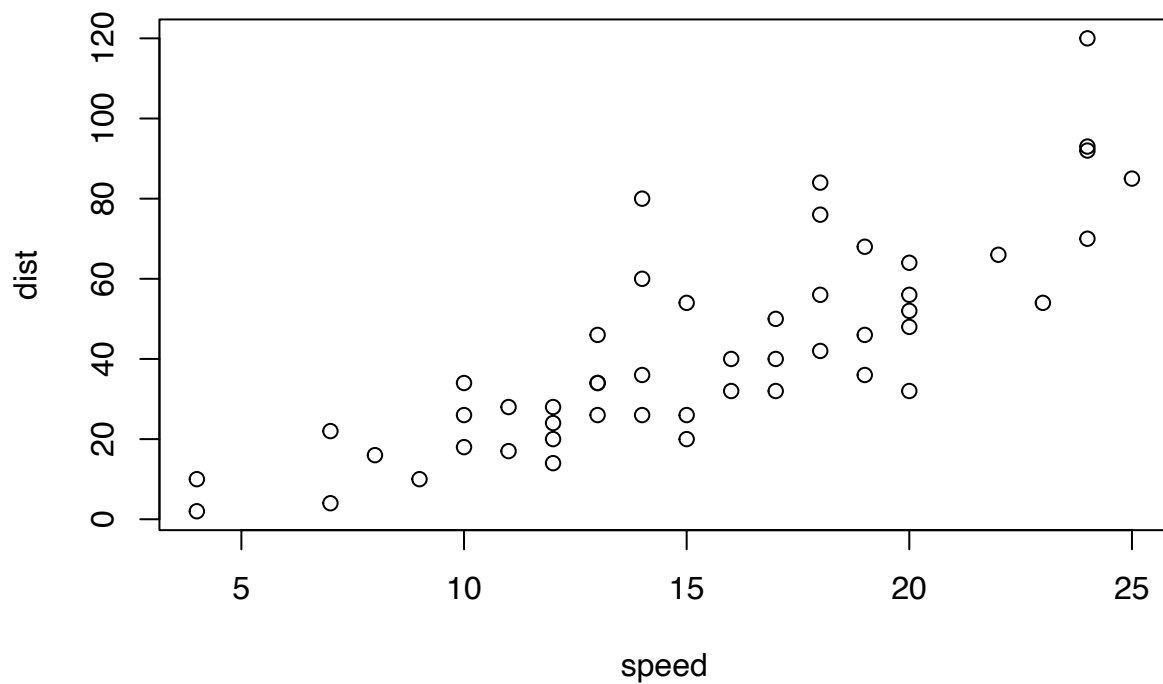
```
## 'geom_smooth()' using formula 'y ~ x'
```



```
# ggplot is nothe only graphic system  
plot(cars$speed, cars$speed)
```



```
plot(cars)
```



```
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
```

```
nrow(genes)
```

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

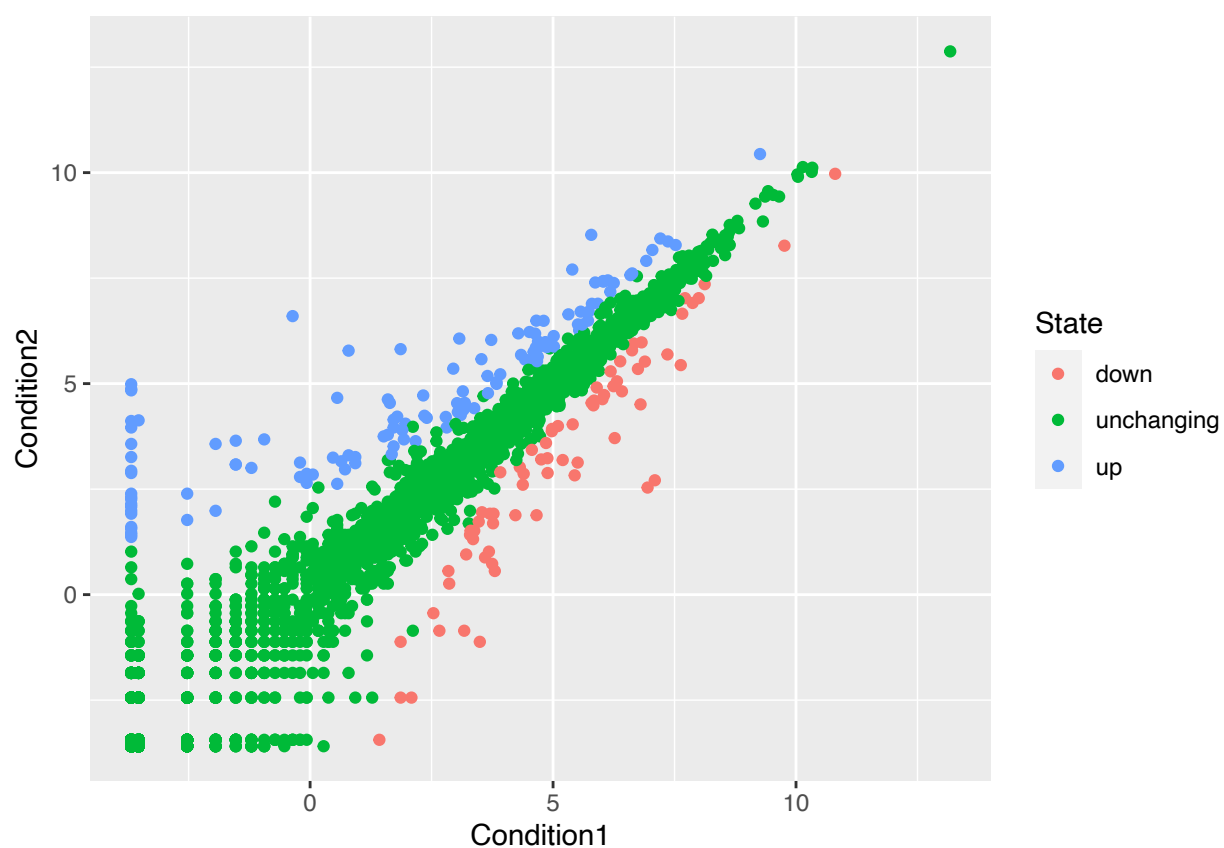
```
# how many genes are up?
table(genes$State)
```

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

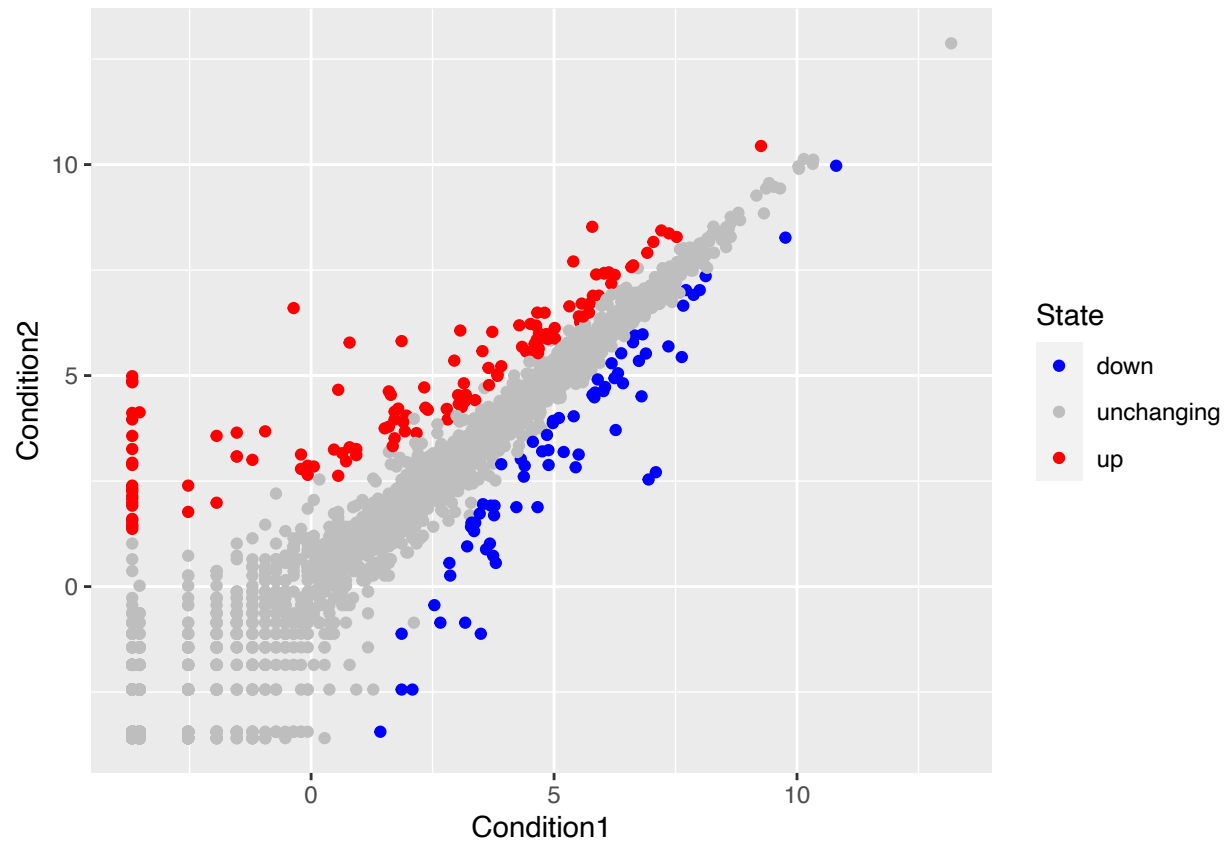
```
#what percentage of the genes are up?
# round() round up to whole number or certain digits
round(table(genes$State)/nrow(genes) *100, 3)
```

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

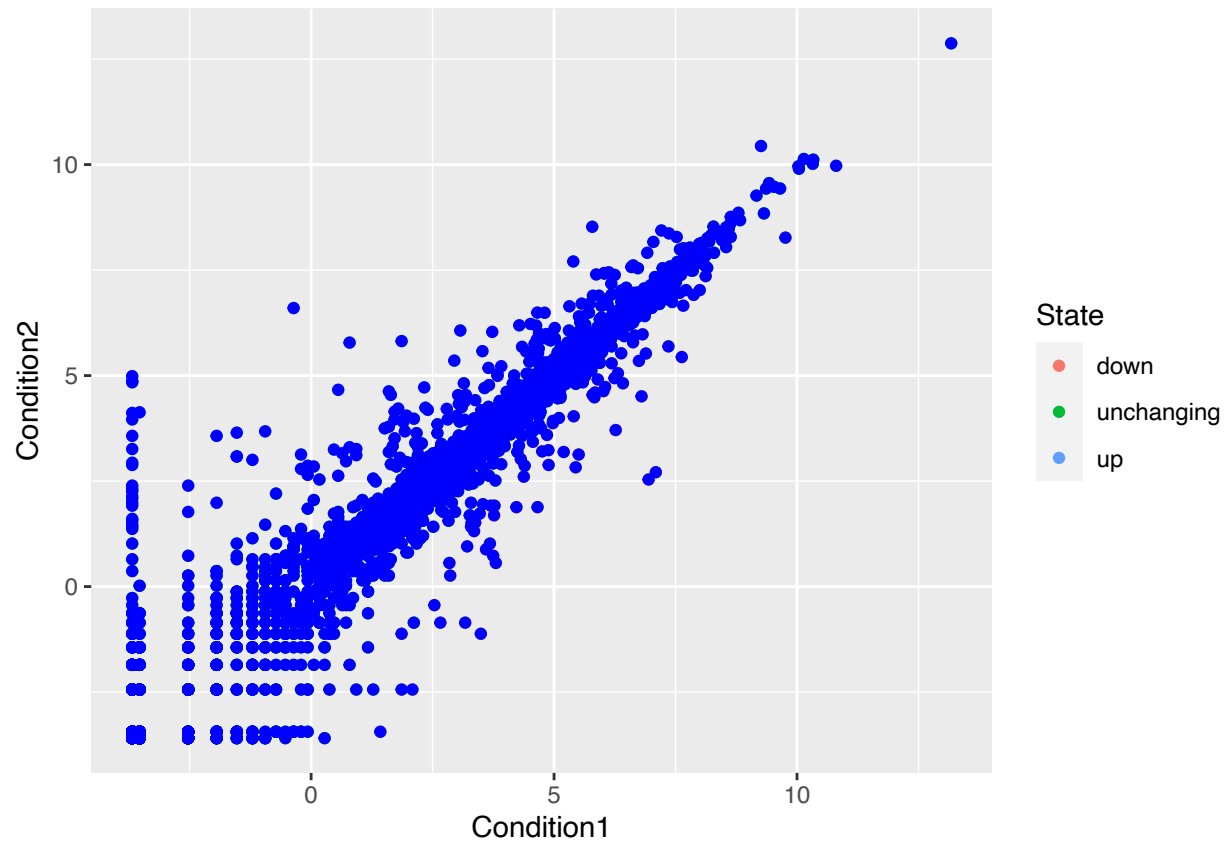
```
# make a figure
p <- ggplot(genes) + aes(x=Condition1, y=Condition2, col = State) +
  geom_point()
p
```



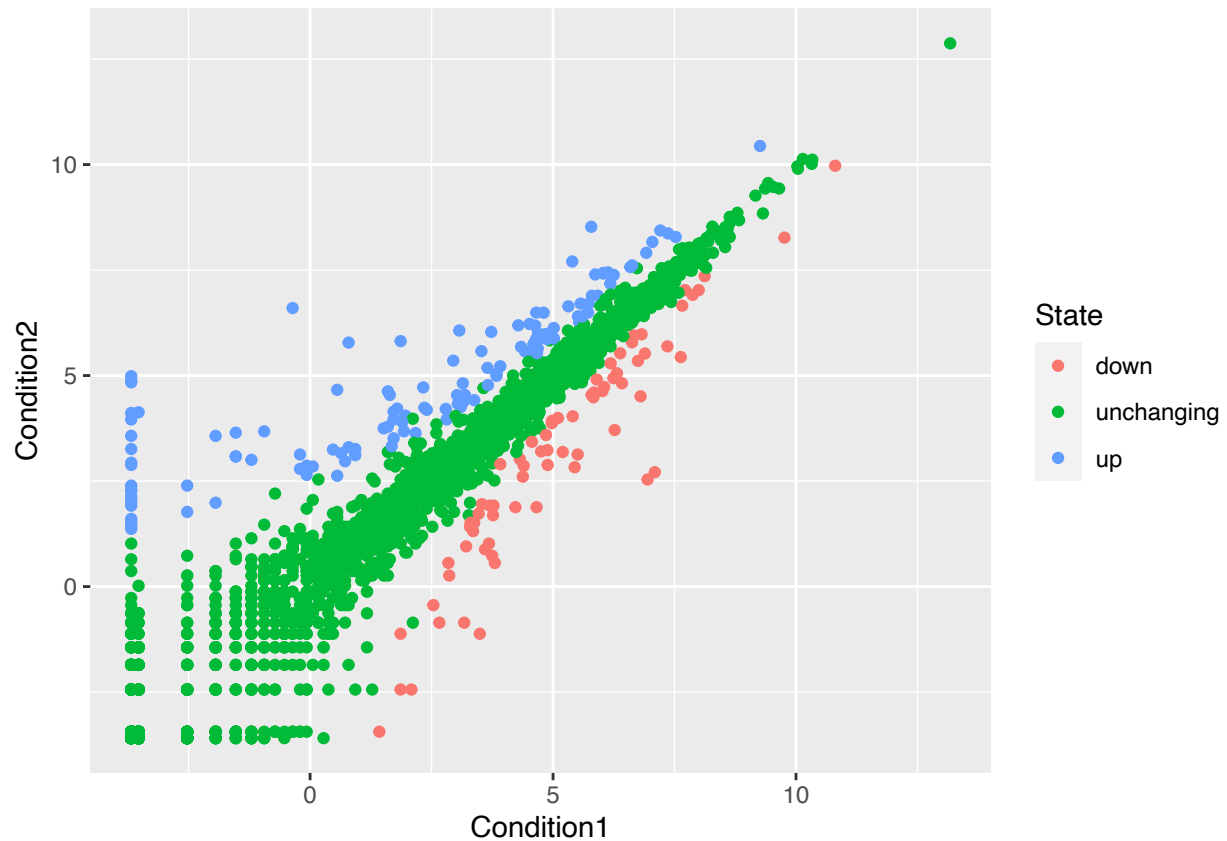
```
#change the color
p + scale_color_manual(values = c("blue", "grey", "red"))
```



```
# bad color  
p+ geom_point(col = "blue")
```



```
# nicer color  
p + aes(color = State)
```

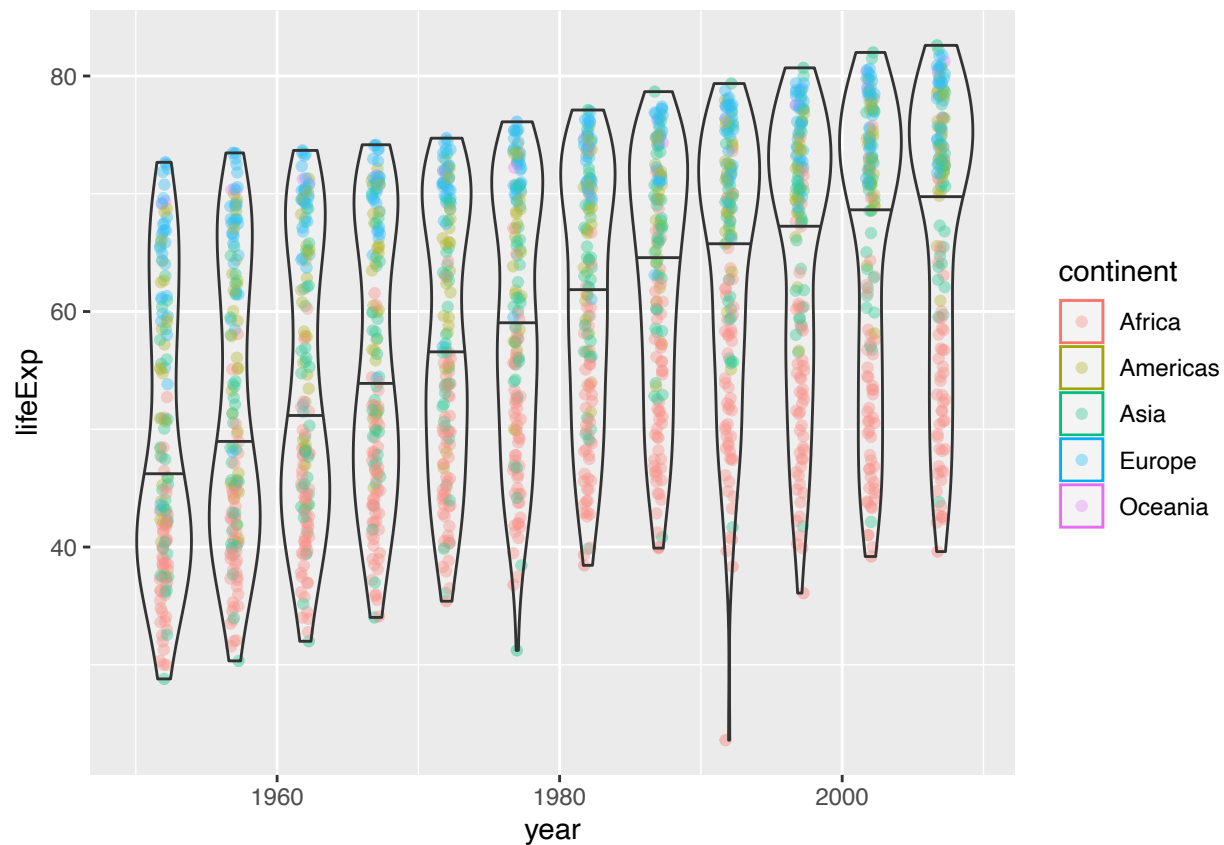


```
# explor the gapminder dataset
# install.packages("gapminder")
library(gapminder)
head(gapminder)
```

```
## # 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.
```

```
ggplot(gapminder,
       aes(year, lifeExp, col = continent)) +
  #geom_point(alpha = 0.4)+
  geom_jitter(width = 0.3, alpha = 0.4) +
  #geom_boxplot(alpha = 0.3, aes(group = year))
  geom_violin(aes( group = year), alpha = 0.2, draw_quantiles = 0.5)
```





```
# install the plotly
# install.packages("plotly")
# interactive plot
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##   last_plot
```

```
## The following object is masked from 'package:stats':
##
##   filter
```

```
## The following object is masked from 'package:graphics':
##
##   layout
```

```
#ggplotly()
```