

# BGGN213 Class 05: Data Visualization

Jibin Zhang, PID: A53300326

10/14/2021

```
# Class 05: Data Visualization

# Today we are going to use ggplot? package

# First we need to load the package!
# install.packages("ggplot2")
library(ggplot2)

# 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
```

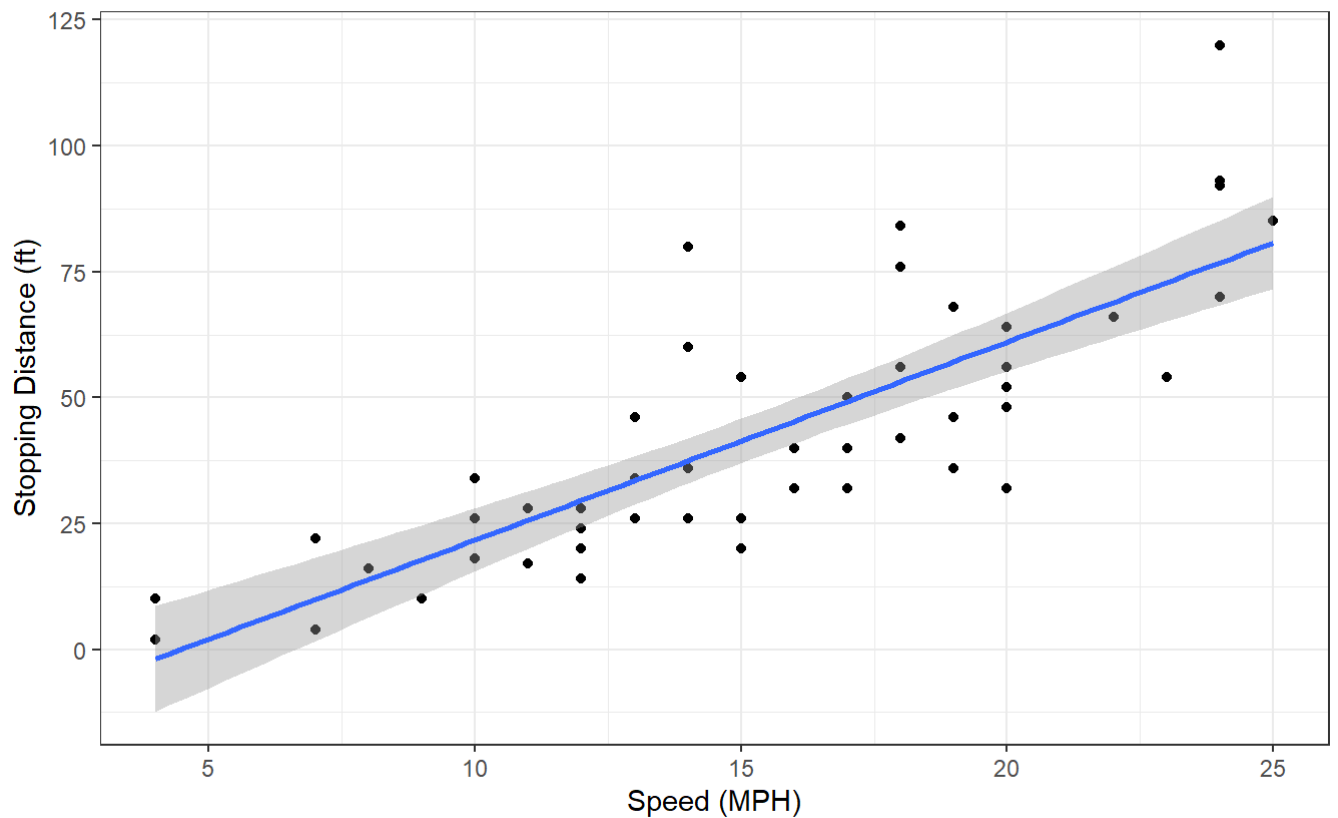
```
# All ggplots have at least 3 layers
# data + aes + geoms

library(ggplot2)
ggplot(data=cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm") +
  labs(title="Speed and Stopping Distances of Cars",
        x="Speed (MPH)",
        y="Stopping Distance (ft)",
        subtitle = "Your informative subtitle text here",
        caption="Dataset: 'cars'") +
  geom_smooth(method="lm", se=FALSE)+
  theme_bw()
```

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

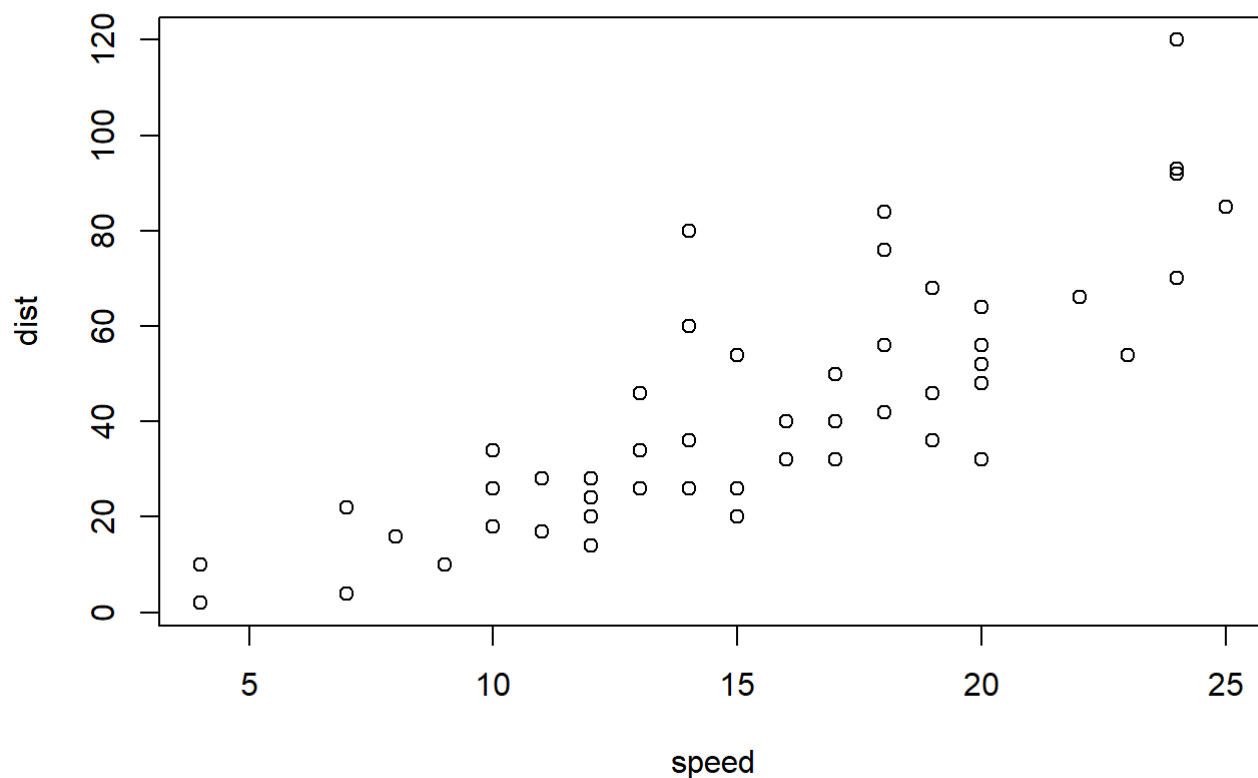
## Speed and Stopping Distances of Cars

Your informative subtitle text here



Dataset: 'cars'

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



```
# Plot some gene expression results.
# Dataset is online in tab separated format so we
# use the read.delim() function to import into R

url <- "https://bioboot.github.io/bimml43_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 this dataset?
nrow(genes)
```

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

```
ncol(genes)
```

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

```
# Q: How many genes are "up"?
table(genes$State)
```

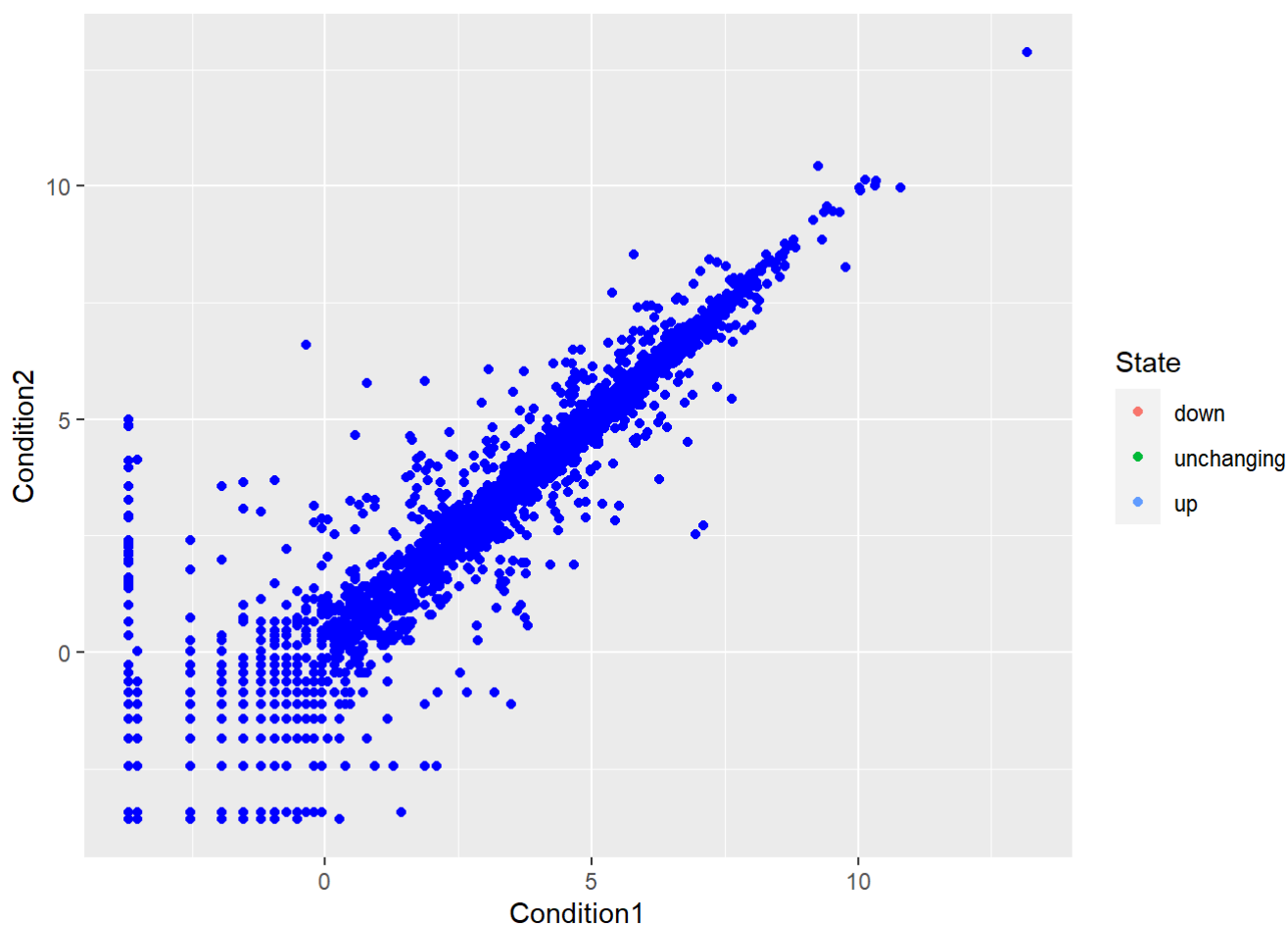
```
##
##      down  unchanged      up
##      72      4997      127
```

```
# Q: What % are up?
round(table(genes$State)/nrow(genes)*100, 3)
```

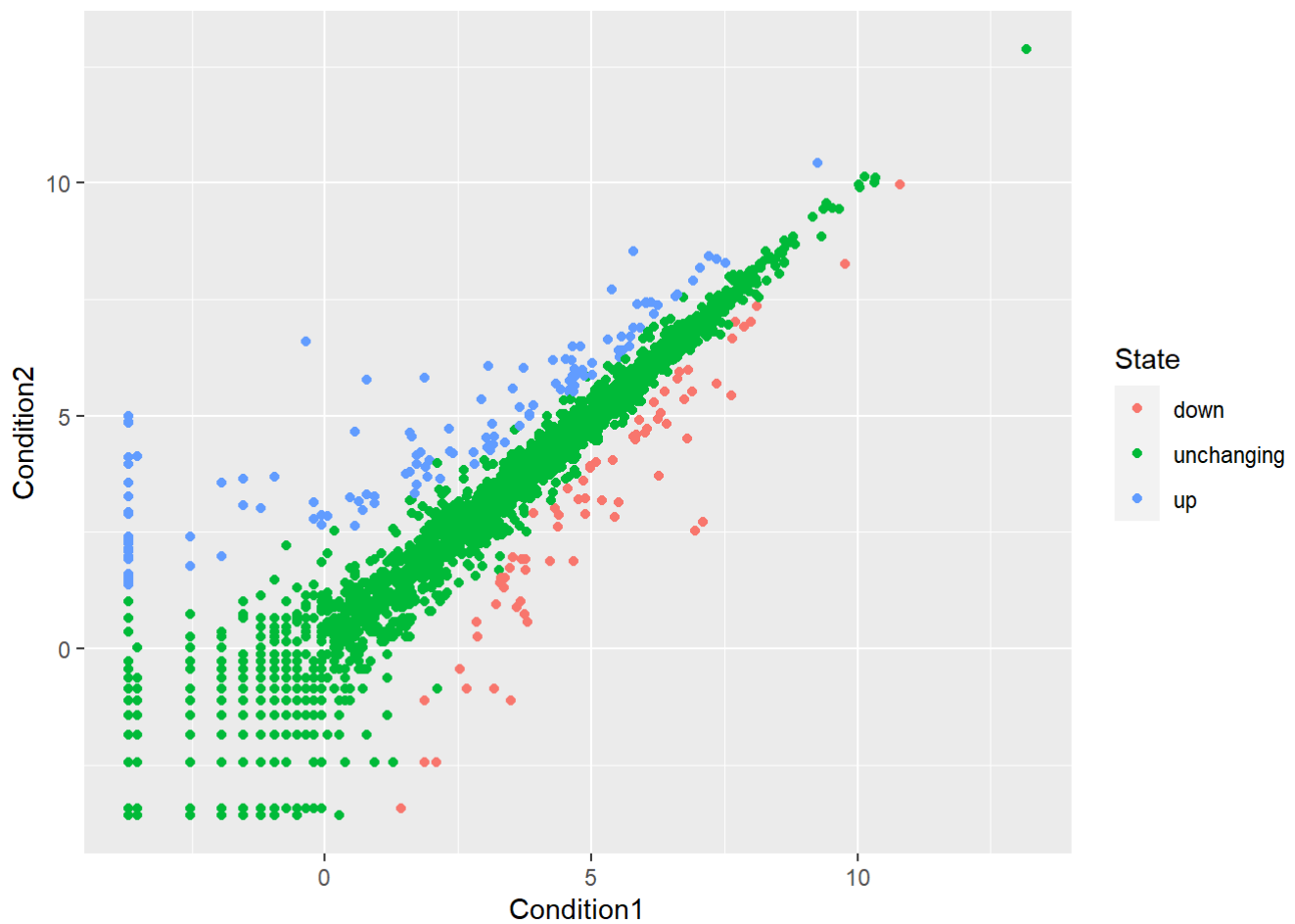
```
##
##      down  unchanged      up
##      1.386      96.170      2.444
```

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

# Yuk!
p + geom_point(col = "blue")
```

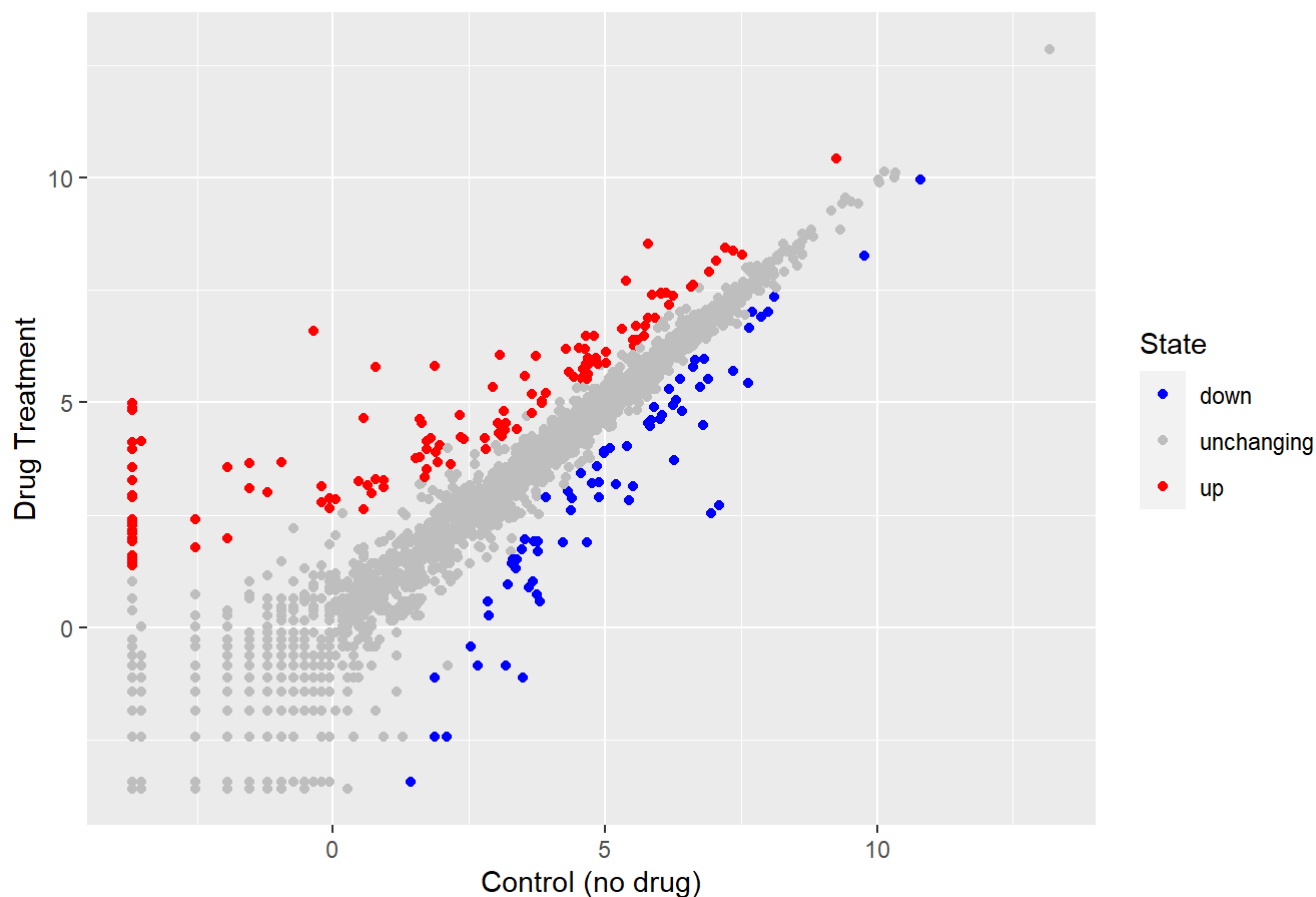


```
# Nicer color by State column in data
p + aes(col=State)
```



```
# I like it but not the default colors, let's change them
p + scale_colour_manual( values=c("blue","gray","red") ) +
  labs(title = "Gene Expression Changes Upon Drug Treatment",
        x = "Control (no drug)",
        y = "Drug Treatment")
```

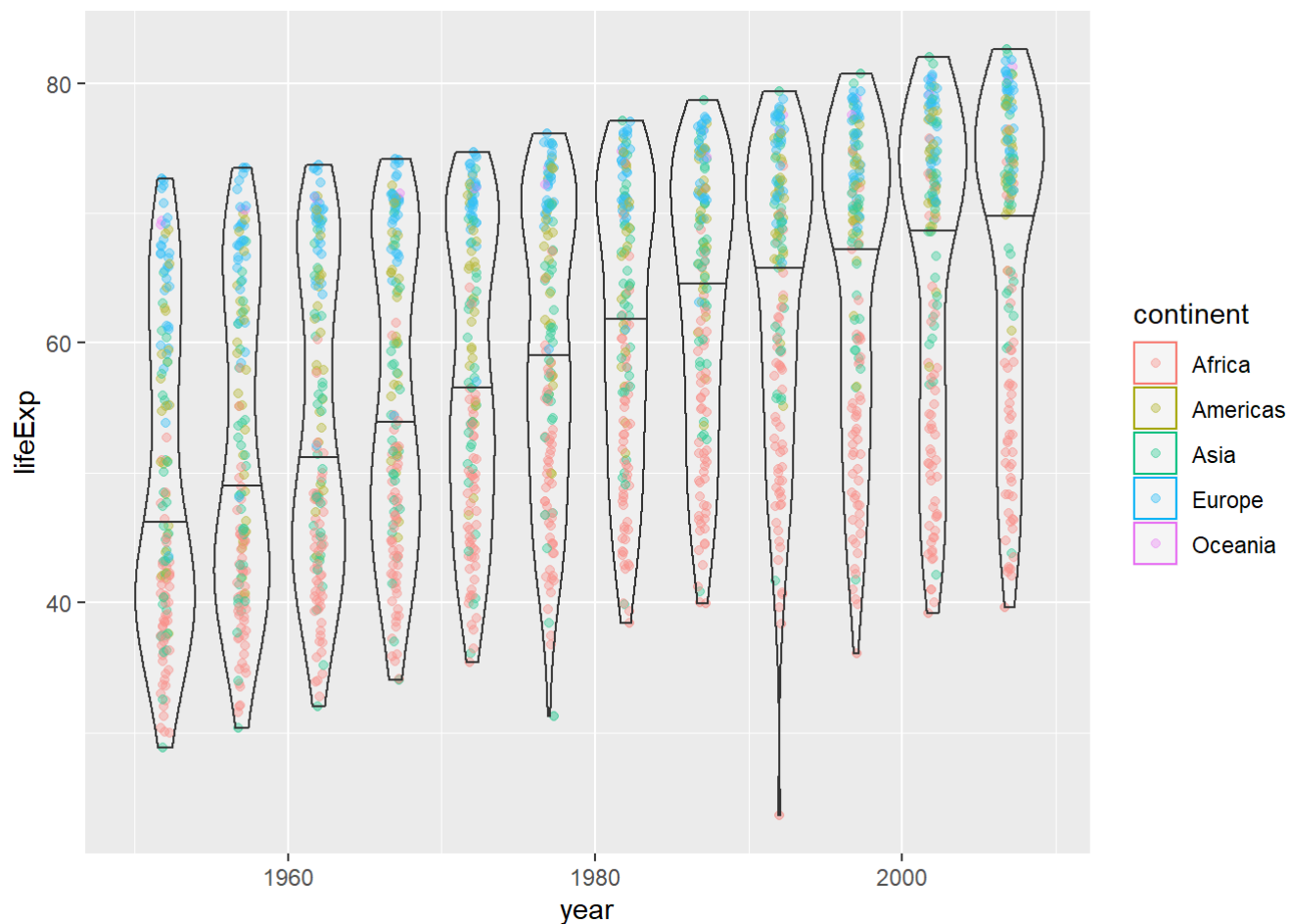
## Gene Expression Changes Upon Drug Treatment



```
# 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.
```

```
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")
library(plotly)
```

```
##
## 载入程辑包: 'plotly'
```

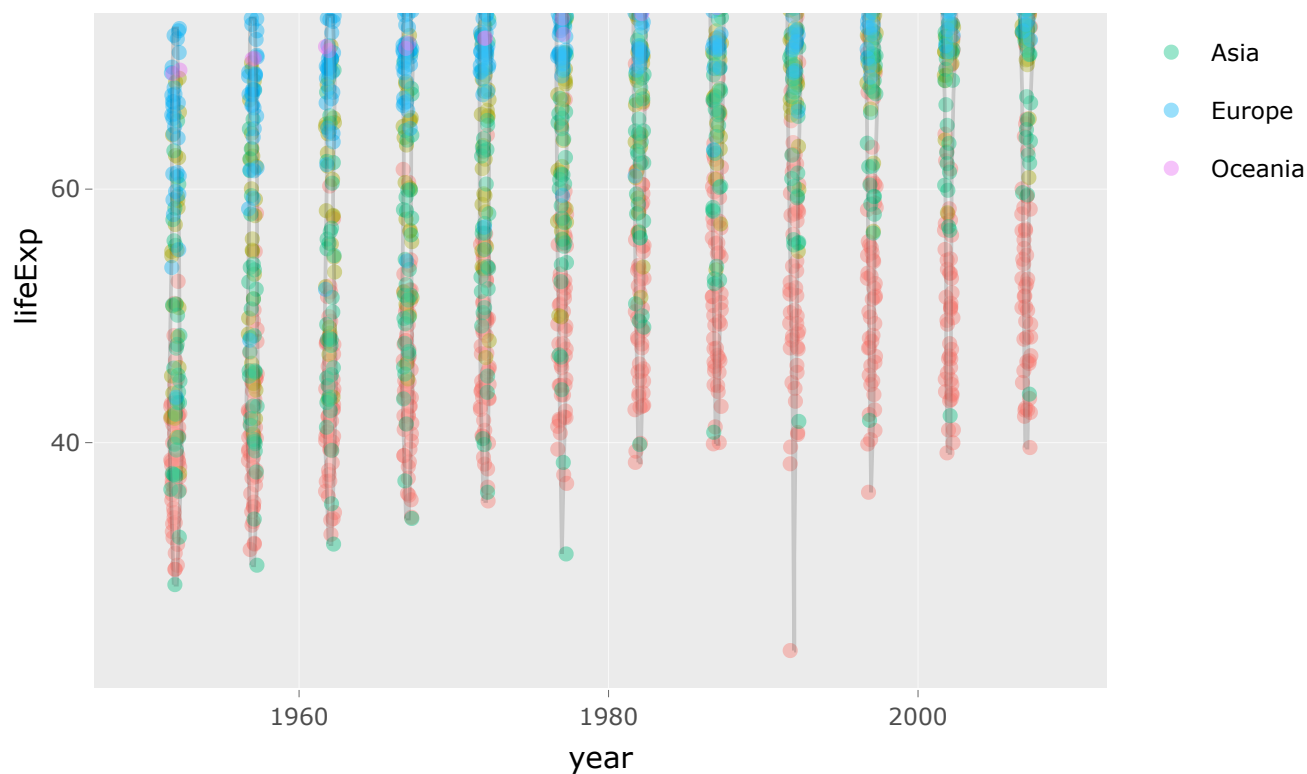
```
## The following object is masked from 'package:ggplot2':
##
##   last_plot
```

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

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

```
ggplotly()
```





```
# install.packages("dplyr")
# uncomment to install if needed
library(dplyr)
```

```
##
## 载入程辑包: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
gapminder_2007 <- gapminder %>% filter(year==2007)
```