Class 5: Data Viisualization with ggplot

Fu-Hsuan Ko

Table of contents

Our first plot

R has base graphic

```
#ggplot(cars)
```

How would I plot this with ‘ggplot’? NO! We need to install package first We use “install_packeage()”

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

Before I can use this package I need to load it with a “library()” call

```
library(ggplot2)
```

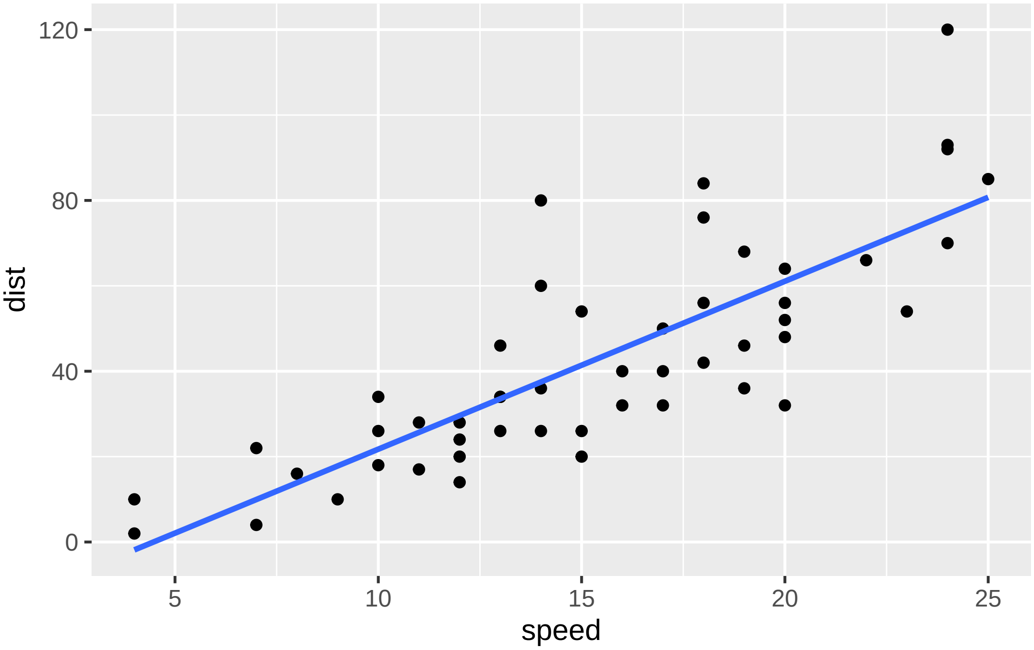
Every ggplot needs at least three layers”

- **Data** (i.e. the data.frame we have)
- **Aes** (the aesthetic mapping of our data to what we want to plot)
- **Geoms** (How we want to plot this stuff!)

add a straight line without SE

```
ggplot(data=cars)+
  aes(x=speed, y=dist)+
  geom_point()+
    geom_smooth(se=FALSE, method="lm")
```

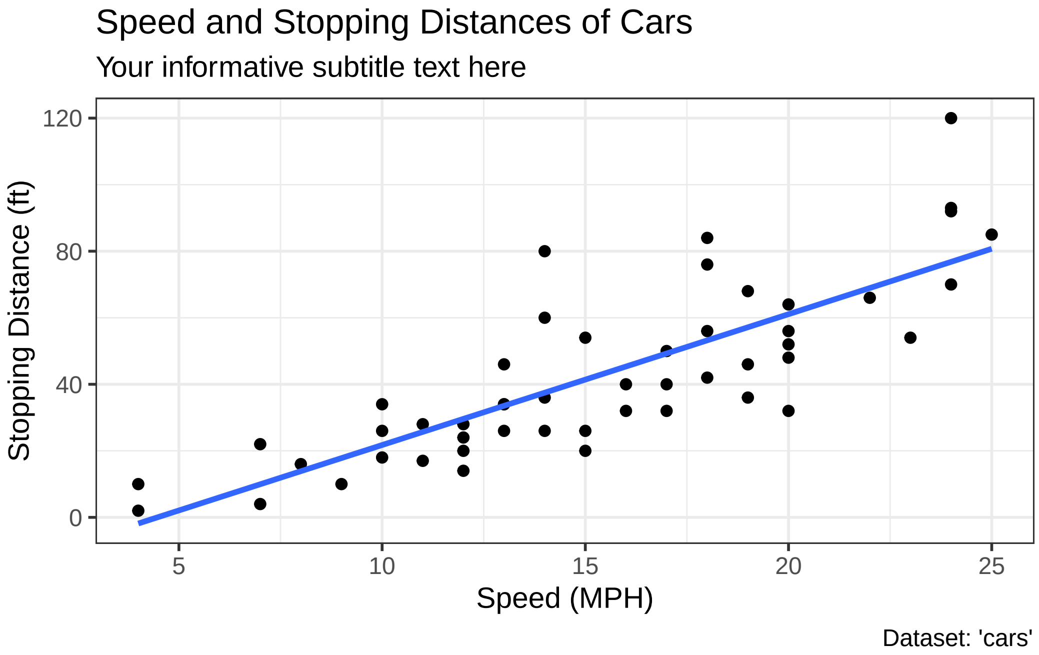
```
`geom_smooth()` using formula 'y ~ x'
```



add lable and change background

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

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



```
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. Use the nrow() function to find out how many genes are in this dataset. What is your
nrow(genes)
```

```
[1] 5196
```

```
# Q. Use the colnames() function and the ncol() function on the genes data frame to find o
colnames(genes)
```

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

```
#Q. Use the table() function on the State column of this data.frame to find out how many ‘
ncol(genes)
```

```
[1] 4
```

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

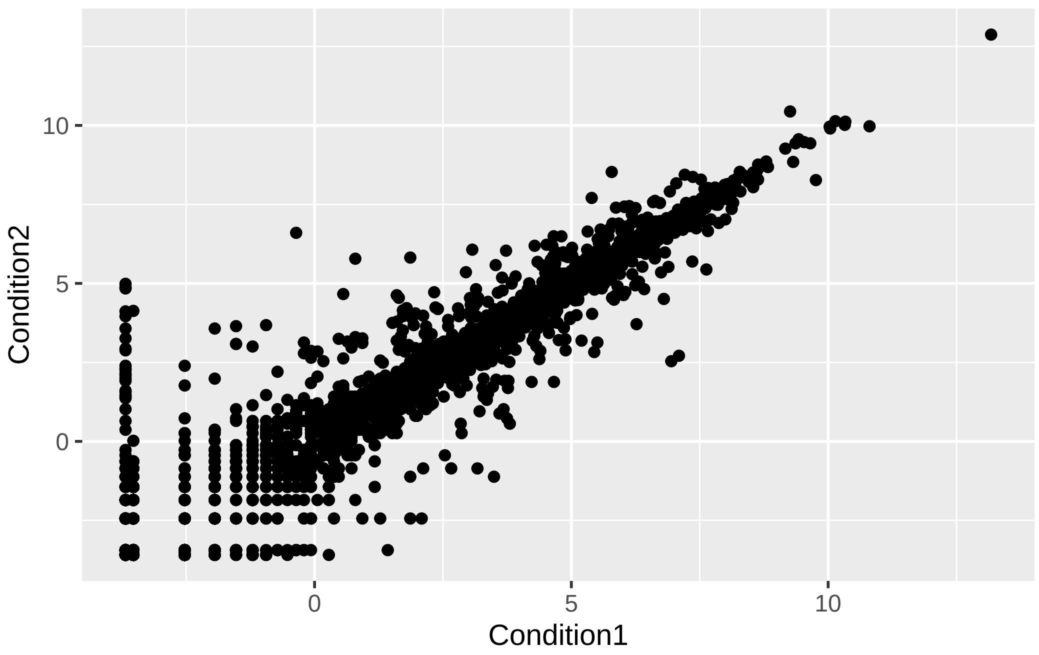
```

      down unchanging         up 
        72       4997        127 
```

```
# Using your values above and 2 significant figures. What fraction of total genes is up-re
round(table(genes$State)/nrow(genes)*100,2)
```

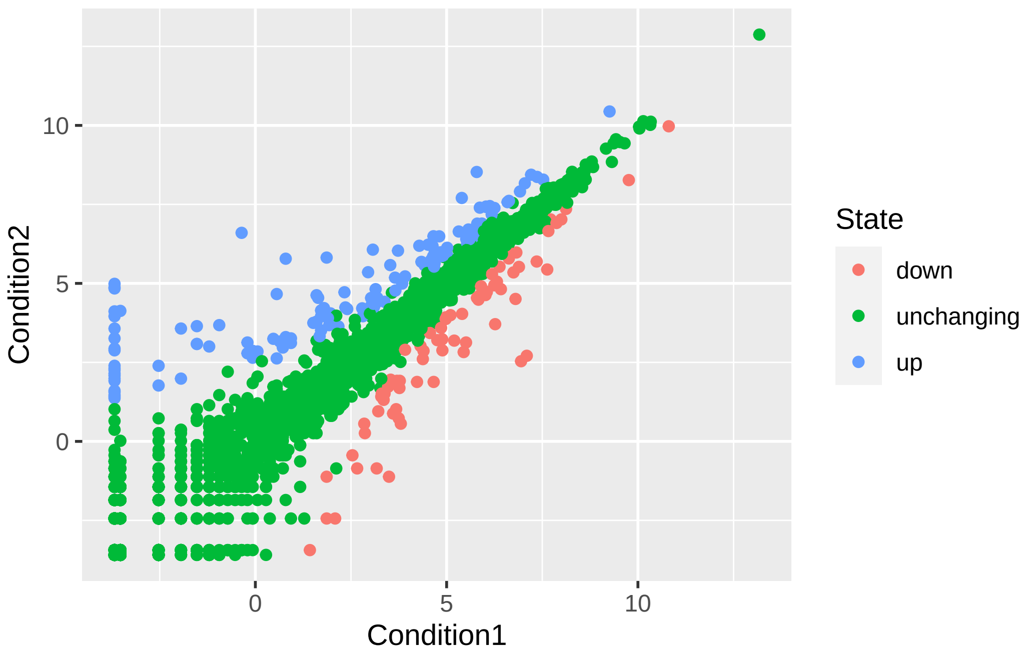
```

      down unchanging         up 
      1.39      96.17       2.44 
```

There are ‘r nrow(genes)’ genes in this data set

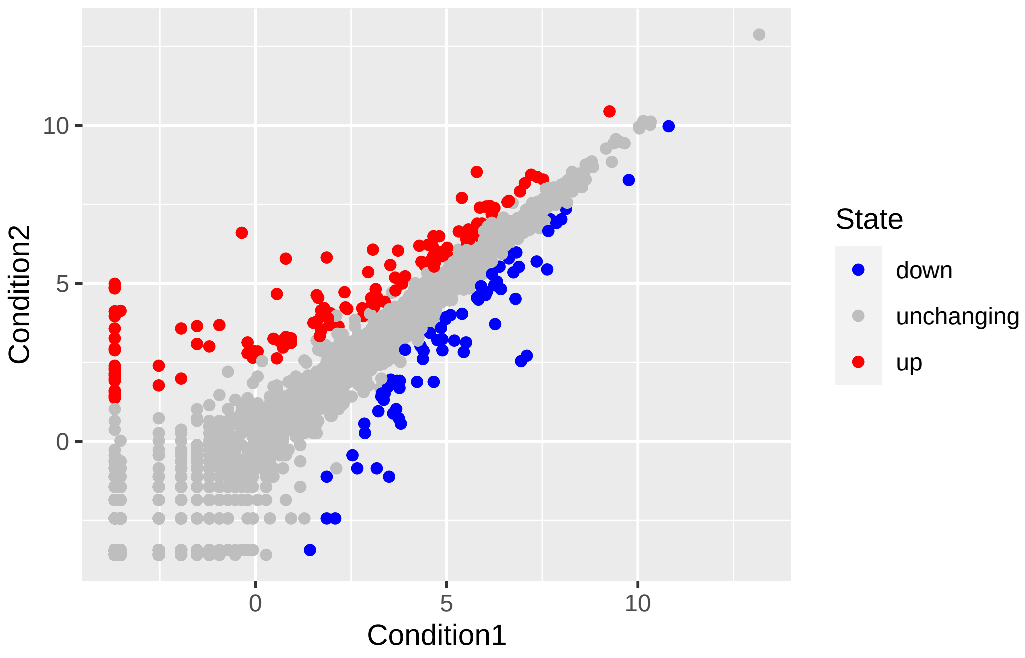
```
ggplot(genes) +
    aes(x=Condition1, y=Condition2) +
    geom_point()
```



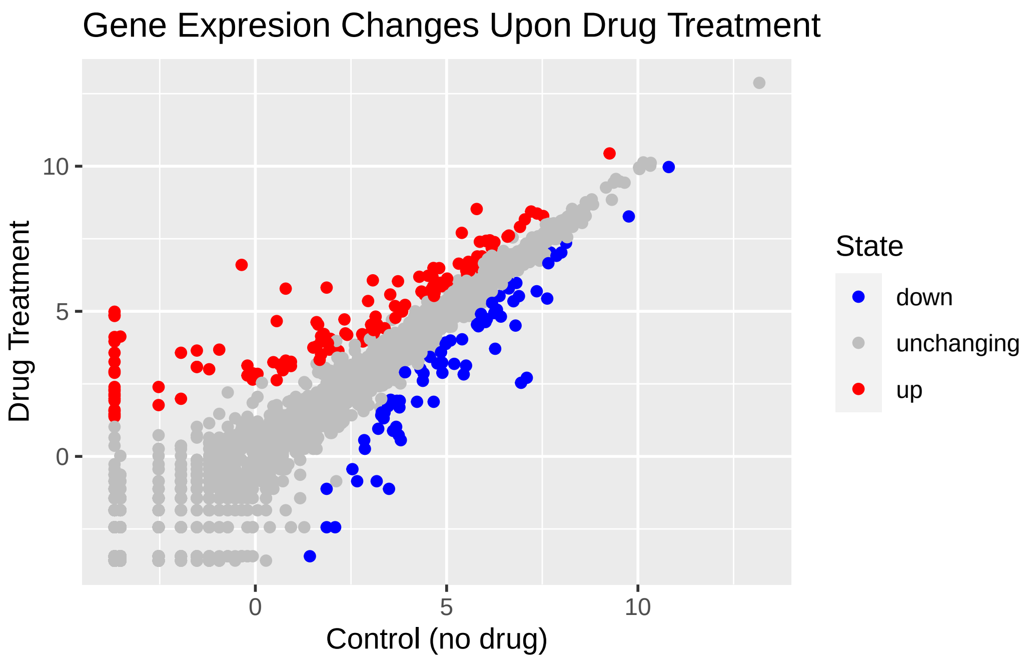
```
p <- ggplot(genes) +
    aes(x=Condition1, y=Condition2, col=State) +
    geom_point()
p
```



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



```r
#Q. Nice, now add some plot annotations to the p object with the labs() function so your p
p + scale_colour_manual( values=c("blue","gray","red") )+
  labs(title="Gene Expresion Changes Upon Drug Treatment", x="Control (no drug)", y="Drug
```



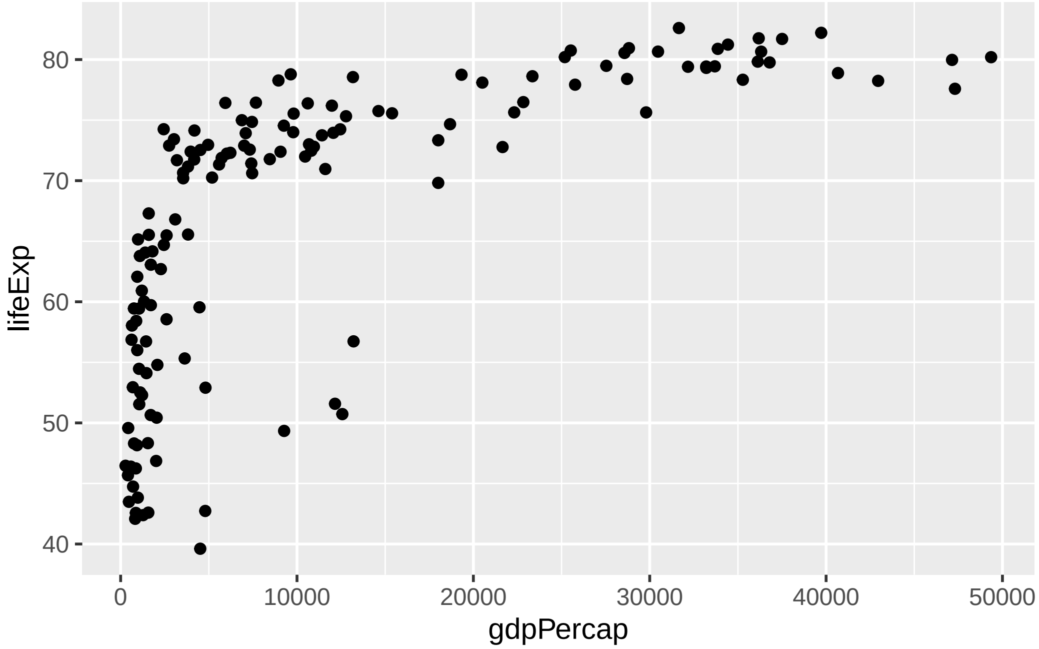
```r
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.

gapminder <- read.delim(url)
```

```r
# install.packages("dplyr")  ## un-comment to install if needed
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':
```

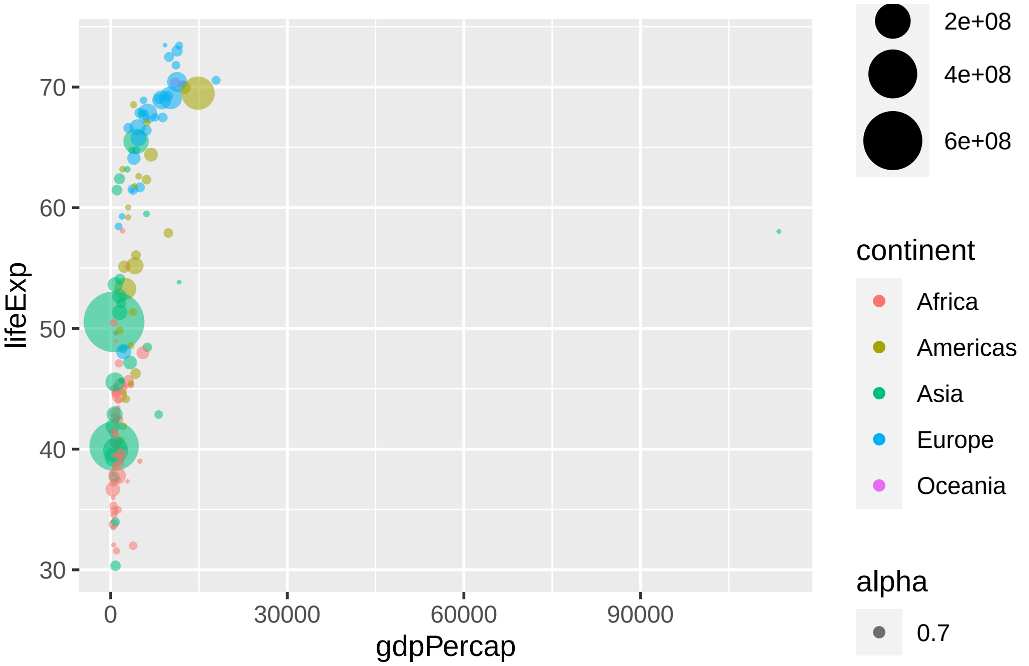
```
intersect, setdiff, setequal, union
```

```
gapminder_2007 <- gapminder %>%
  filter(year==2007)
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point()
```



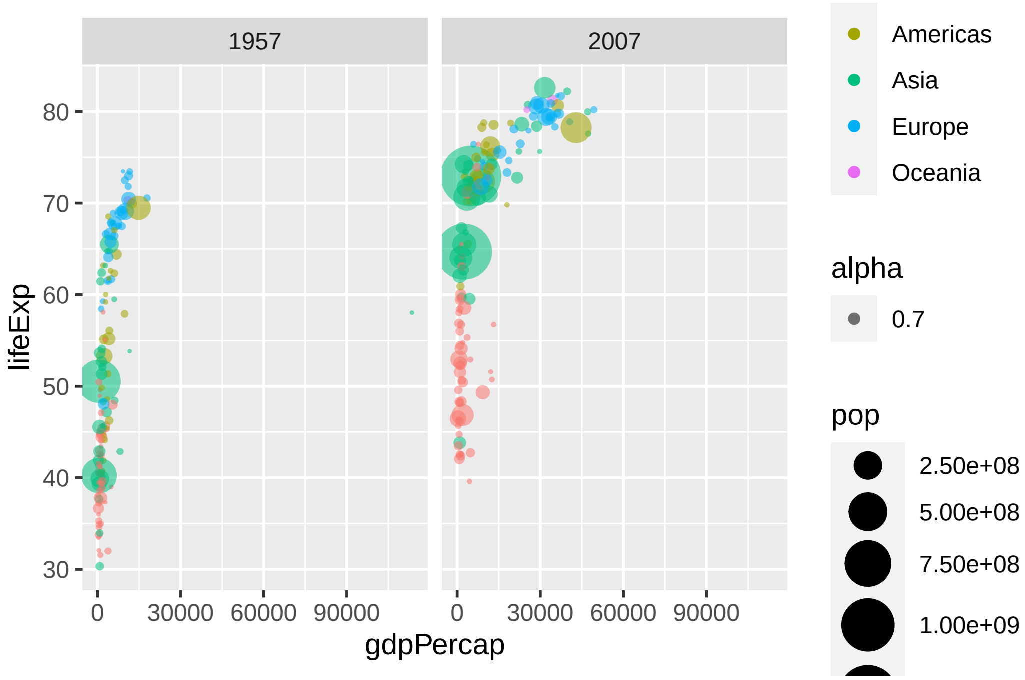
Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

```
gapminder_1957 <- gapminder %>%
  filter(year==1957)
ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, color=continent,size = pop, alpha=0.7) +
  geom_point() +
  scale_size_area(max_size = 10)
```



Q. Do the same steps above but include 1957 and 2007 in your input dataset for ggplot(). You should now include the layer facet_wrap(~year) to produce the following plot:

```
gapminder_1957 <- gapminder %>%
  filter(year==1957|year==2007)
ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, color=continent,size = pop, alpha=0.7) +
  geom_point() +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```


1957
2007
lifeExp
80
70
60
50
40
30
0
30000
60000
90000
gdpPercap
Americas
Asia
Europe
Oceania
alpha
0.7
pop
2.50e+08
5.00e+08
7.50e+08
1.00e+09