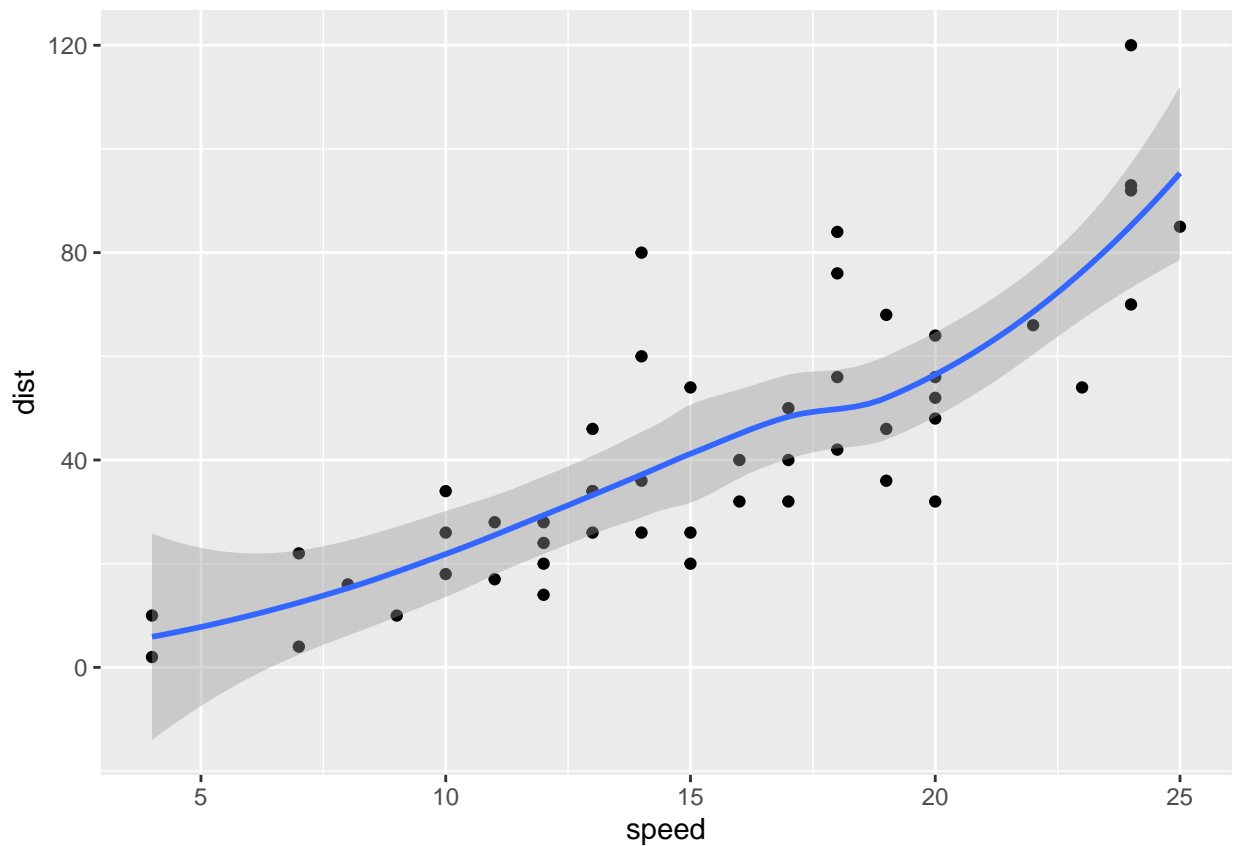


# Class 05 Data Visualization

Dominique Lie - A15470100

October 12, 2021

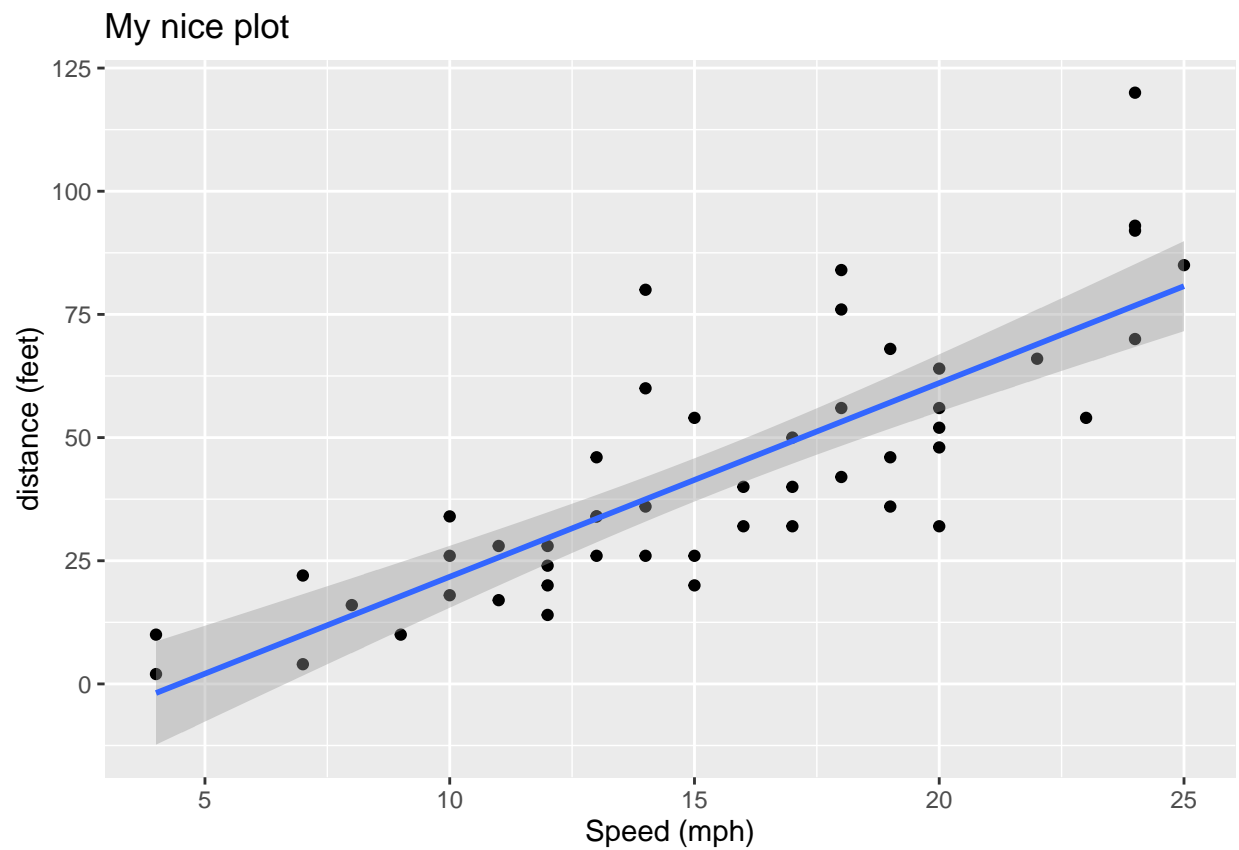
```
# Class 05 Data Visualization  
  
# Lets start with a scatterplot  
# Before we can use ggplot2, we need to load it up!  
# install.packages(ggplot2) before library(ggplot2)  
  
library(ggplot2)  
  
# Every ggplot has a data + aes + geoms  
ggplot(data = cars) + aes (x = speed, y = dist) + geom_point() + geom_smooth()  
  
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



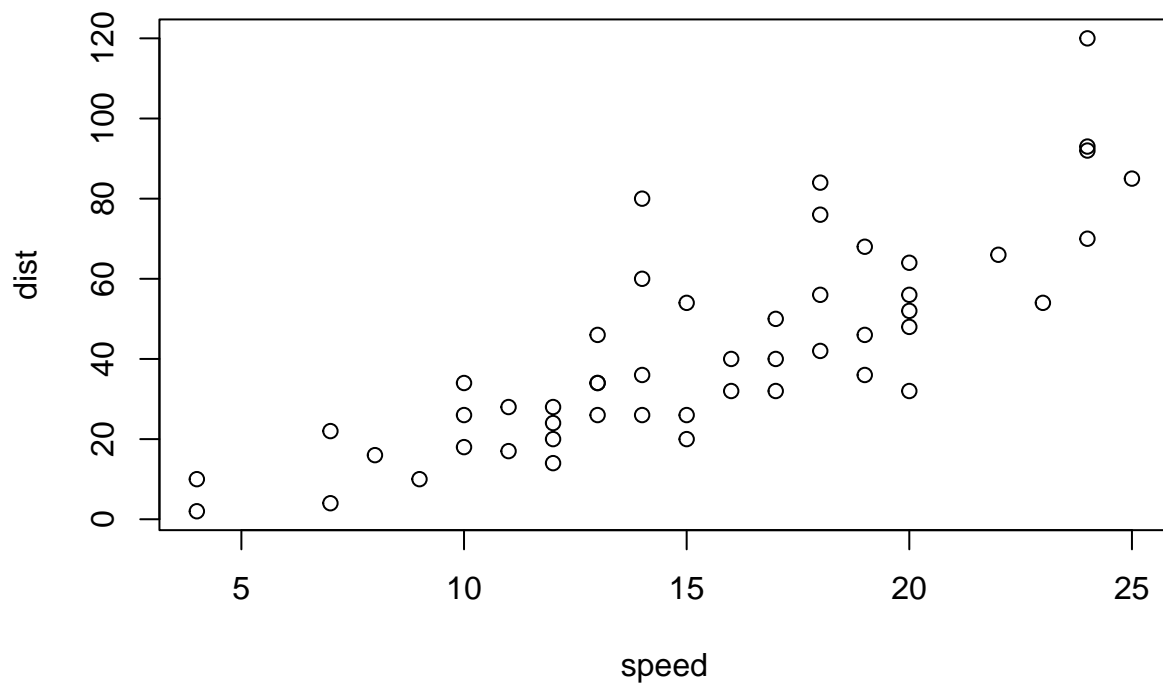
```
# Change to linear model
# Save plot as p
p <- ggplot(data = cars) + aes (x = speed, y = dist) + geom_point() + geom_smooth(method = "lm")

p + labs(title = "My nice plot", x = "Speed (mph)", y = "distance (feet)")
```

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



```
#Base graphics is shorter
plot(cars)
```



```
#Adding more plot aesthetics (size, color, alpha)
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
```

```
# Function to find out how many genes are in dataset
nrow(genes)
```

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

```
# colnames() for column names and ncol() for number of columns
colnames(genes)
```

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

```
ncol(genes)
```

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

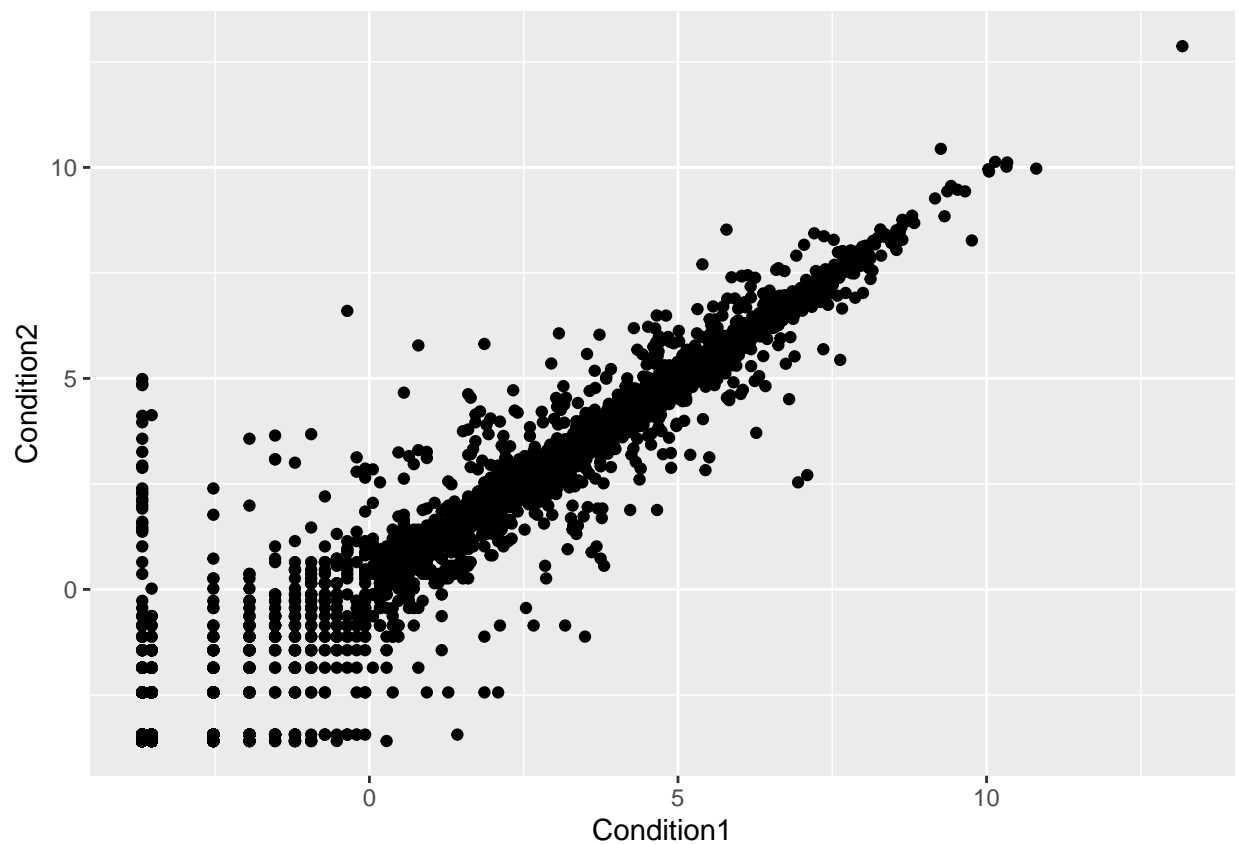
```
# table() for "state"upregulated" genes - just to access list genes["State"] or genes$State  
table(genes$State)
```

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

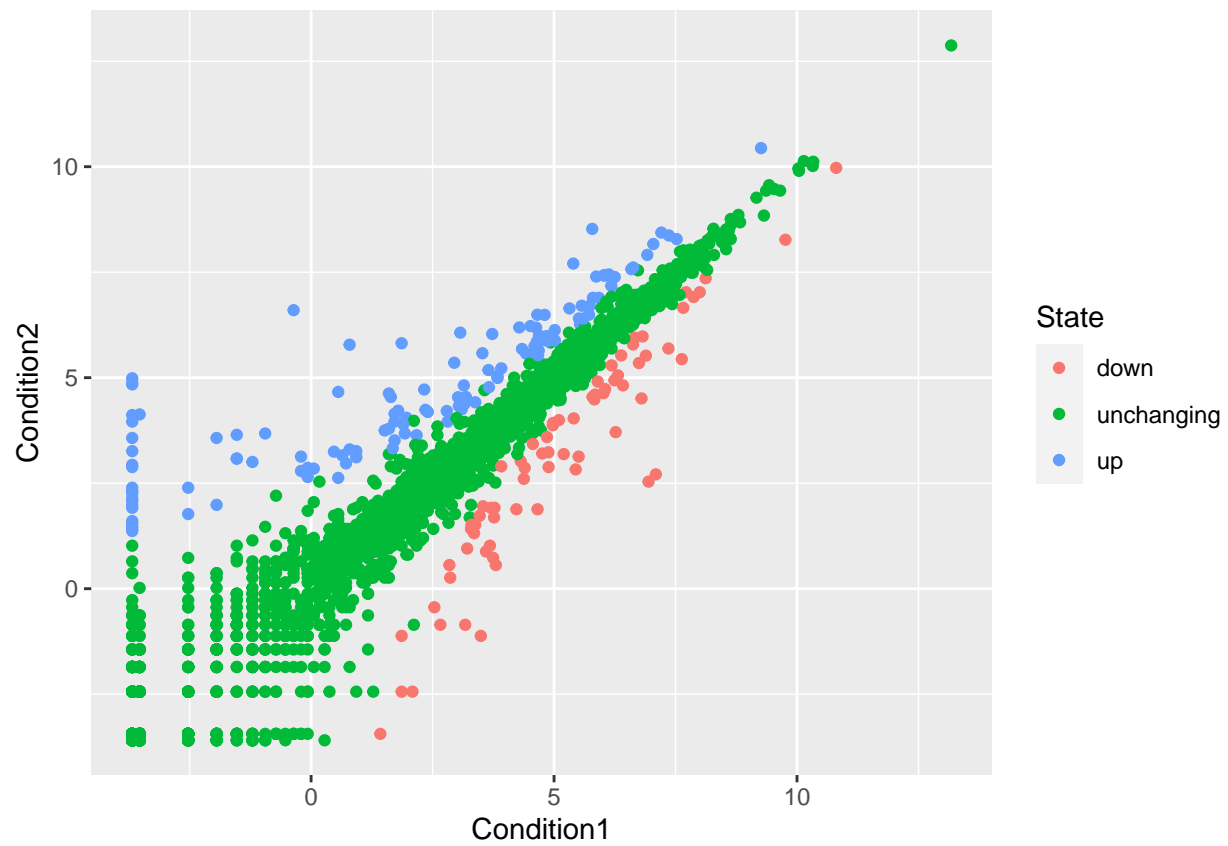
```
# Q. What % are up/down? round function for significant figures (,2) means two sigfig  
prec <- table(genes$State) / nrow(genes) * 100  
round(prec, 2)
```

```
##  
##      down  unchanged      up  
##      1.39      96.17     2.44
```

```
# Make scatterplot with new data genes  
ggplot( data = genes) + aes( x = Condition1, y = Condition2, ) + geom_point()
```

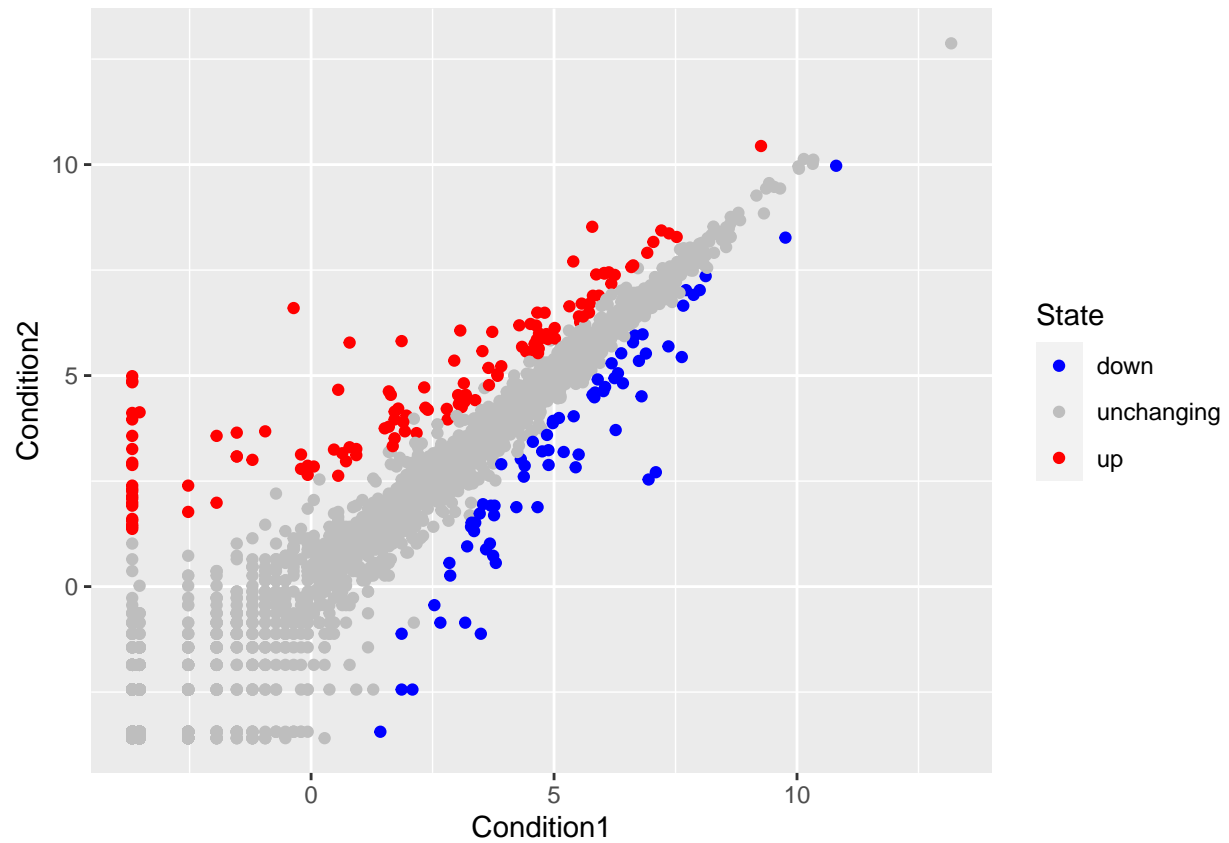


```
# Add state column and color
ggplot( data = genes) + aes( x = Condition1, y = Condition2, col = State ) + geom_point()
```



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

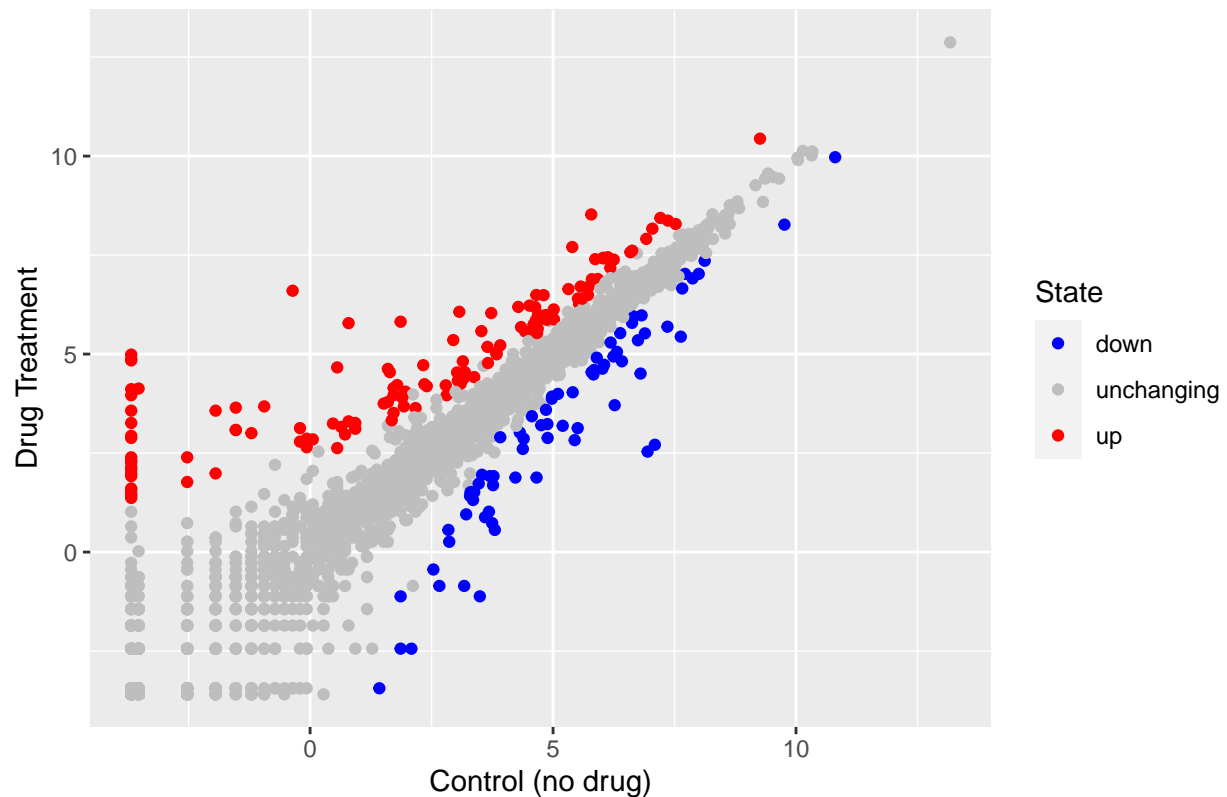
# Change default colors by adding another layer
p + scale_colour_manual( values = c("blue", "gray", "red"))
```



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

# Add plot notations to change name
p + labs( title = "Gene Expression Changes Upon Drug Treatment", x = "Control (no drug)", y = "Drug Tre
```

## Gene Expression Changes Upon Drug Treatment



```
# Optional gapminder set install.packages("gapminder")
library(gapminder)
```

```
# To focus on a single year install.packages("dplyr")
# Filter data frame
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

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

```
##
```

```
## filter, lag
```

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

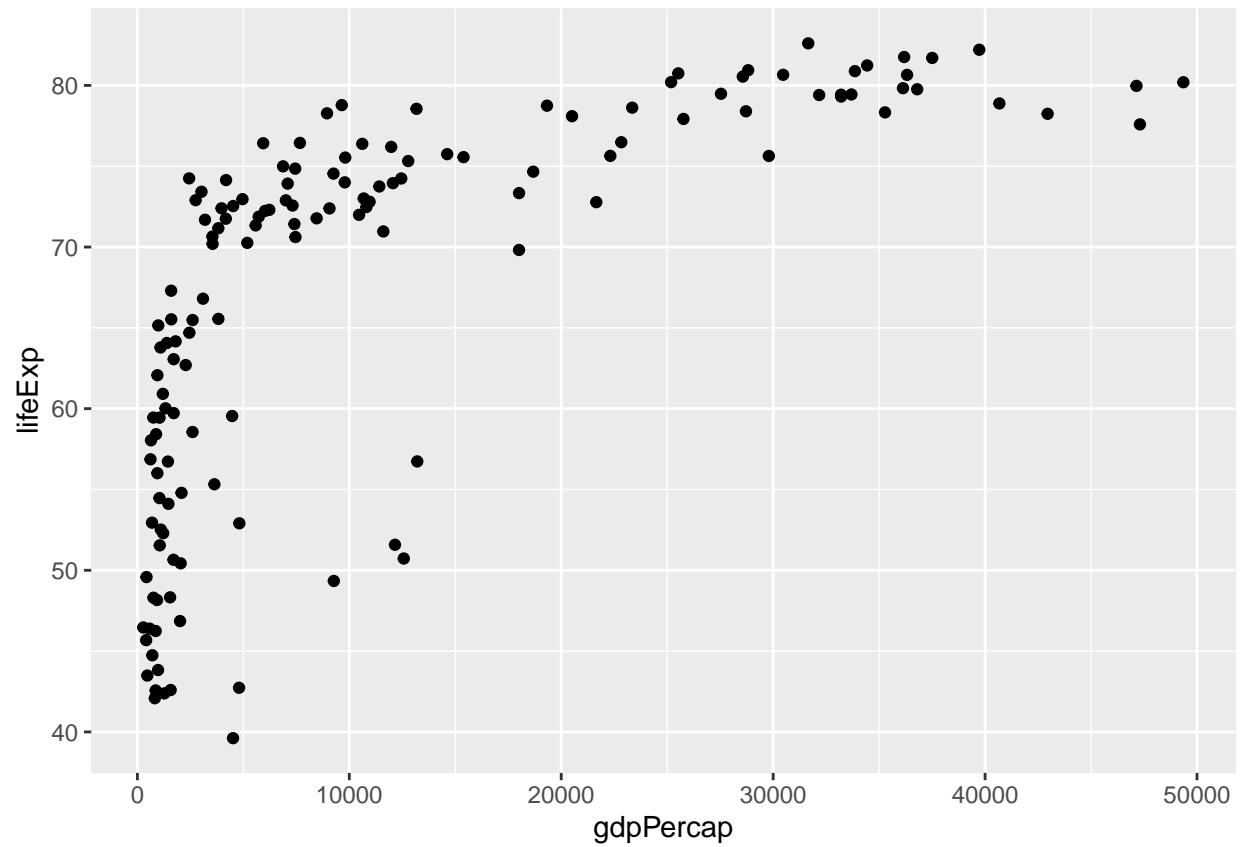
```
##
```

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

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

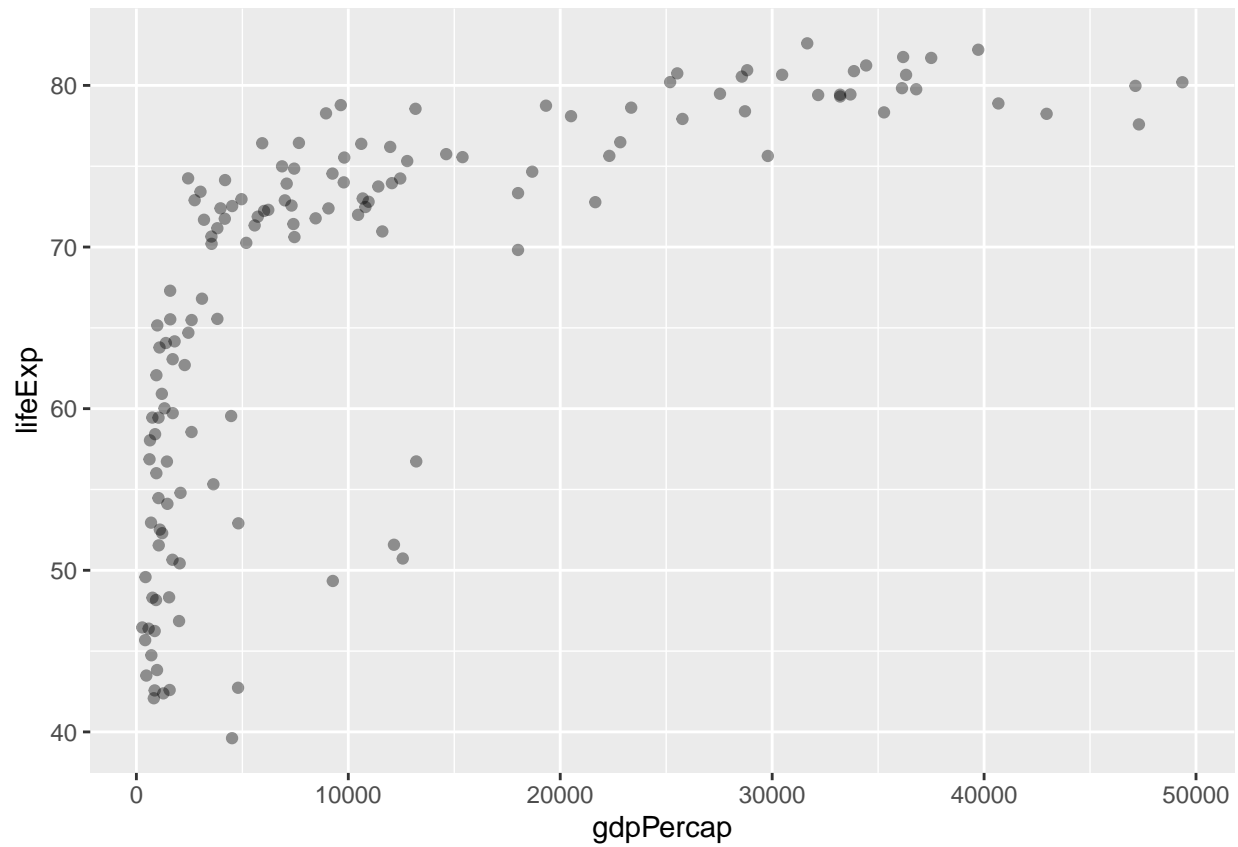
```
# basic scatterplot
```

```
ggplot( data = gapminder_2007) + aes( x= gdpPercap, y = lifeExp) + geom_point()
```

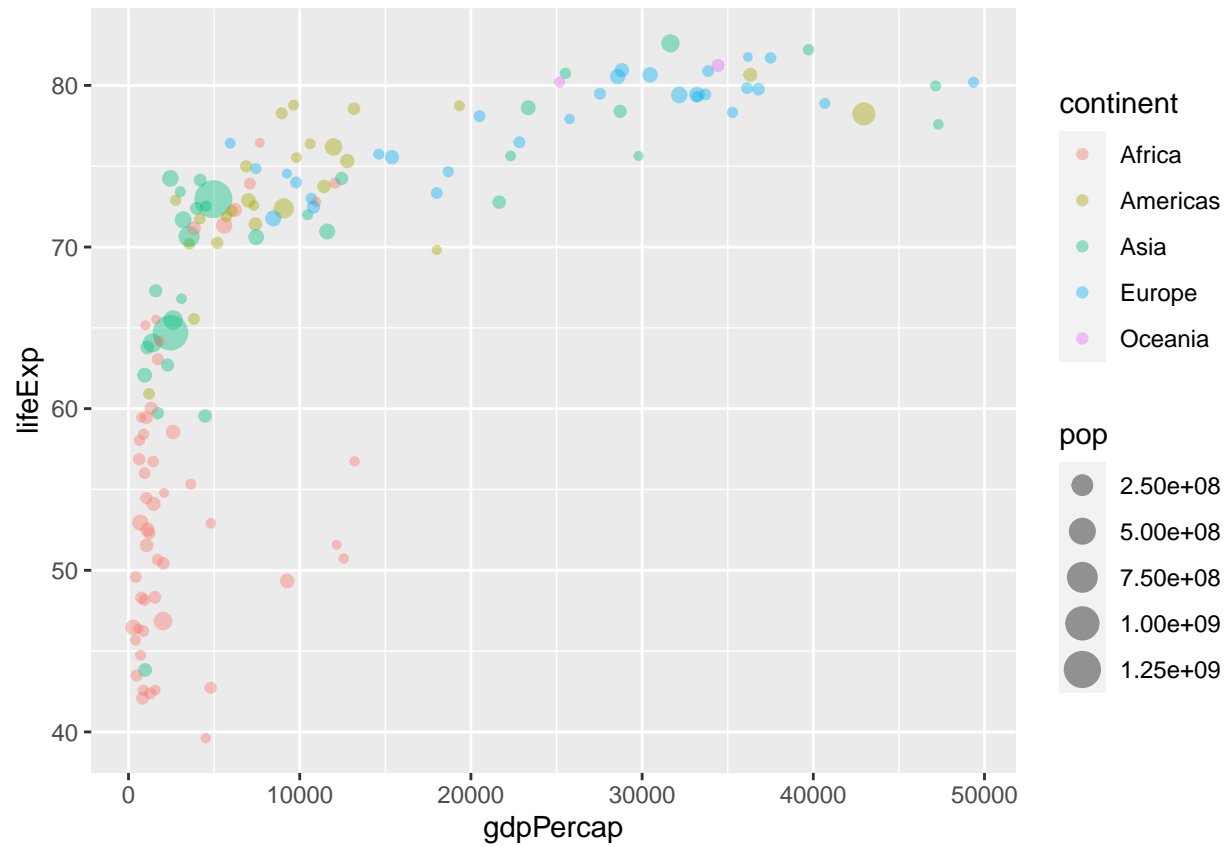


```
# make plot points transparent  
ggplot( data = gapminder_2007) + aes( x= gdpPercap, y = lifeExp) + geom_point(alpha = 0.4)
```

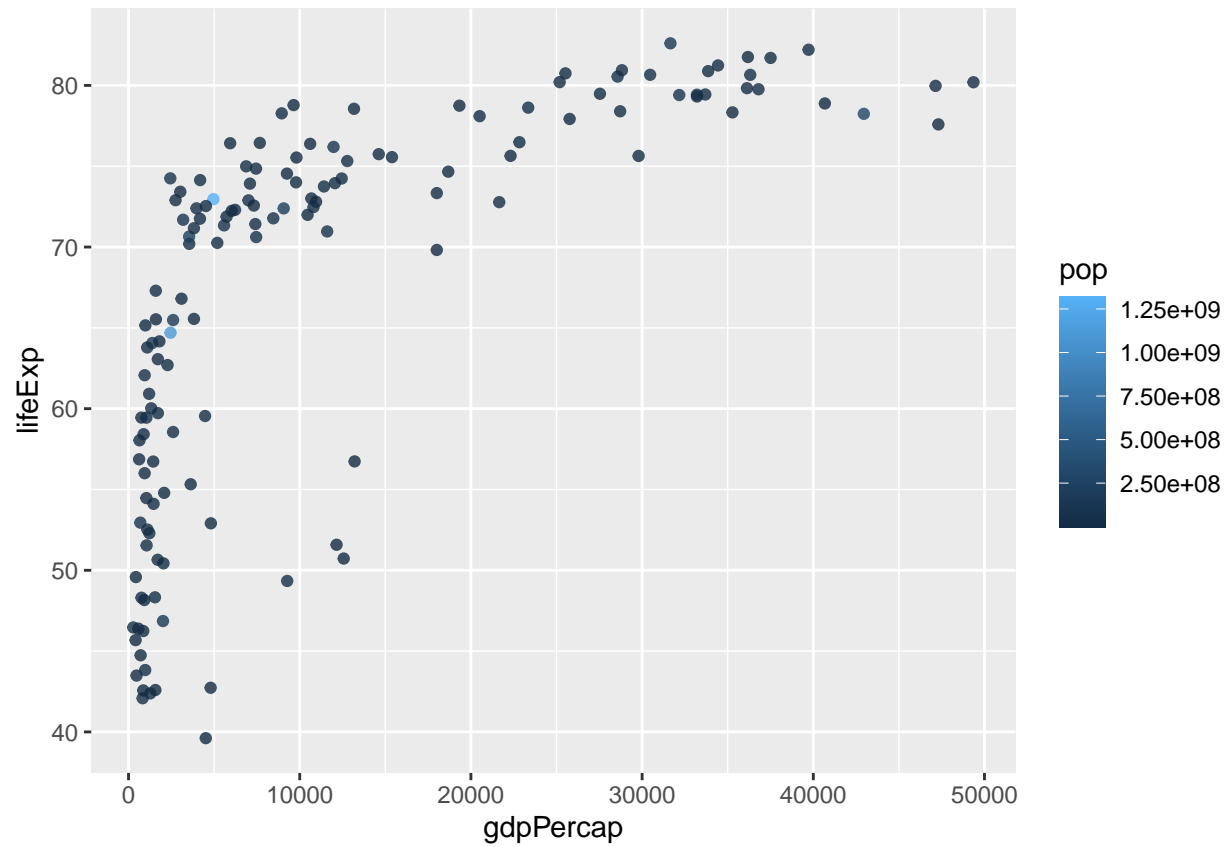




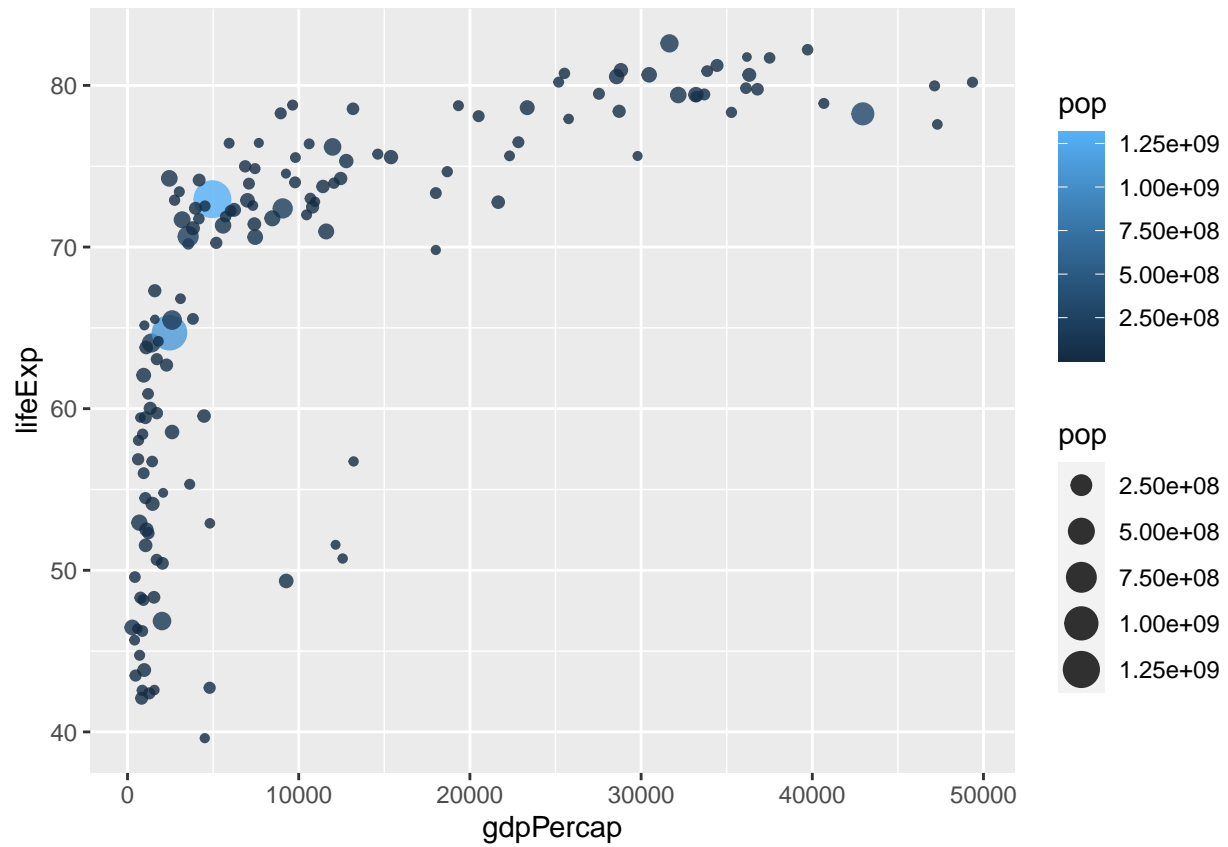
```
ggplot( data = gapminder_2007) + aes( x= gdpPercap, y = lifeExp, color = continent, size = pop) + geom_point()
```



```
ggplot(gapminder_2007) + aes( x = gdpPercap, y = lifeExp, color = pop) + geom_point(alpha = 0.8)
```



```
ggplot(gapminder_2007) + aes( x = gdpPercap, y = lifeExp, color = pop, size = pop) + geom_point(alpha =
```



```
ggplot(gapminder_2007) + aes( x = gdpPercap, y = lifeExp, color = pop, size = pop) + geom_point(alpha =
```

