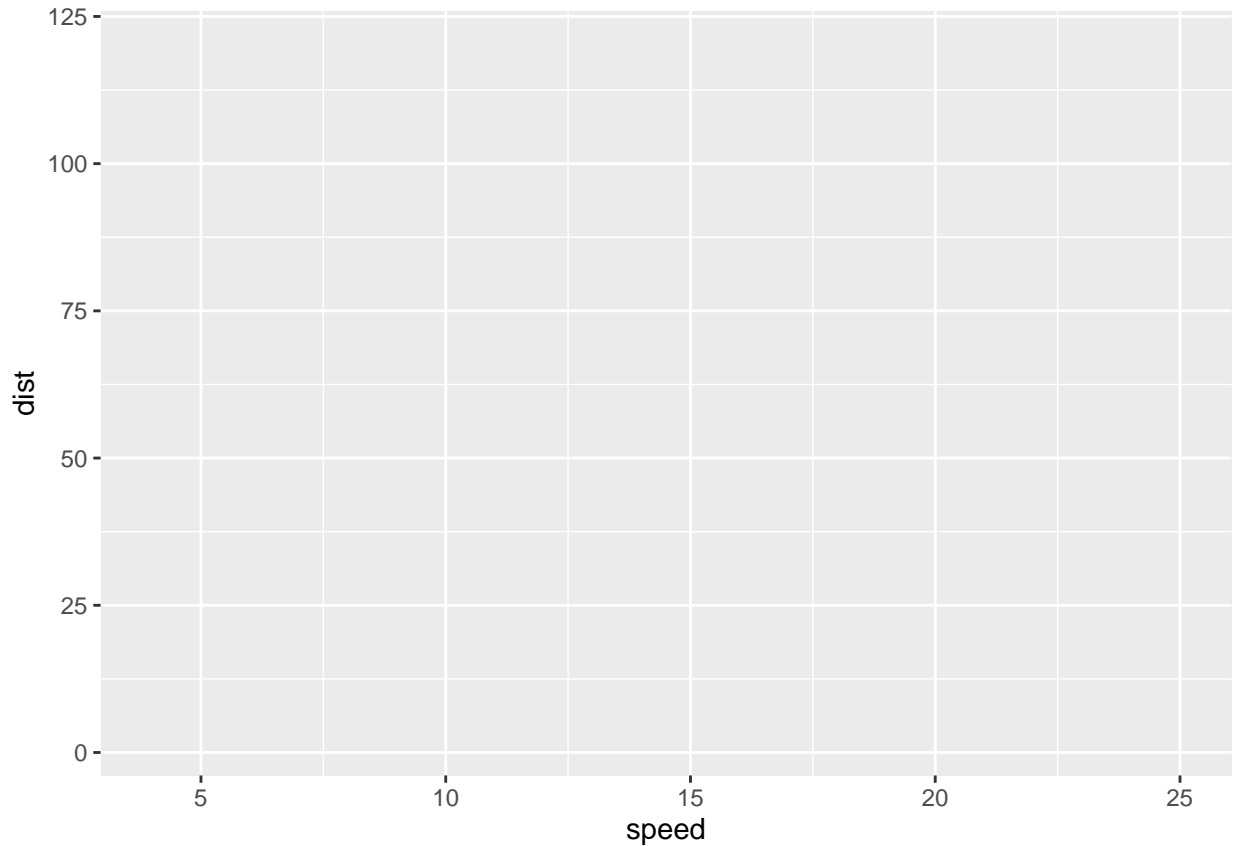


class5-1.R

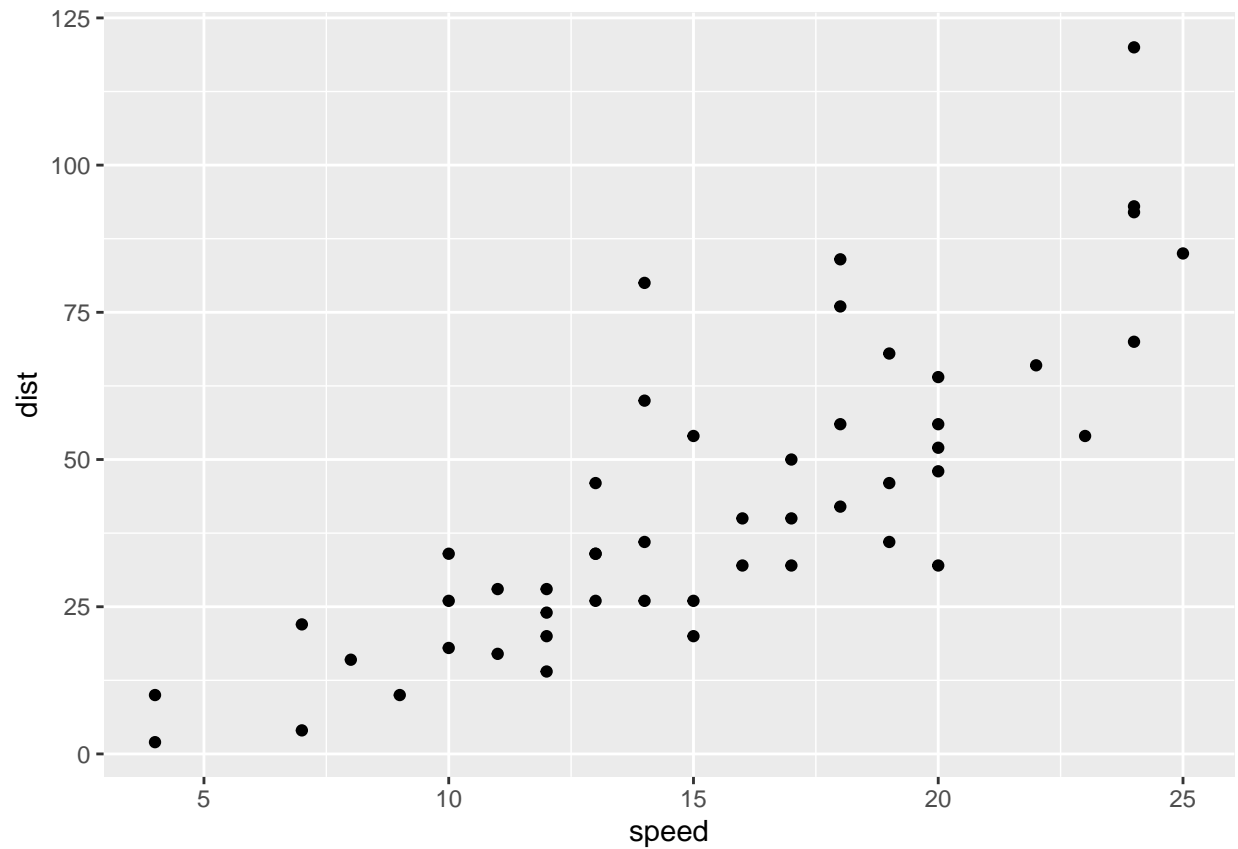
dudu

2022-02-04

```
#Class 5 ggPlot learning, always remember to library(ggplot2) first.  
library(ggplot2)  
# Open ggplot(cars) and add aes  
ggplot(cars) +  
  aes(x=speed, y=dist)
```

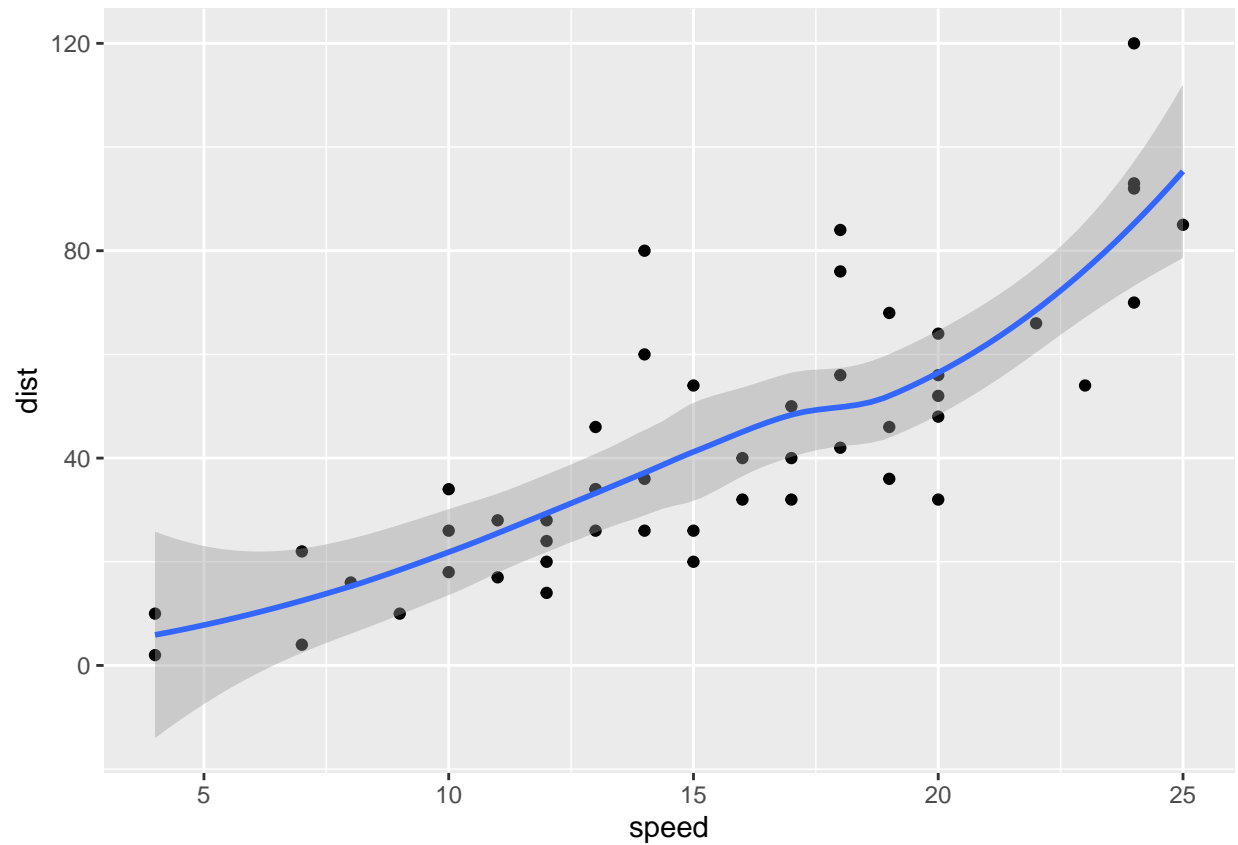


```
## we need to add one of ggplot's geometric layers (or geoms) to define how we want to visualize our data  
## We can use geom_line(), geom_col(), etc  
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point()
```



```
## let's make it smooth. What is smooth? Aids the eye in seeing patterns in the presence of overplotting
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth()
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```

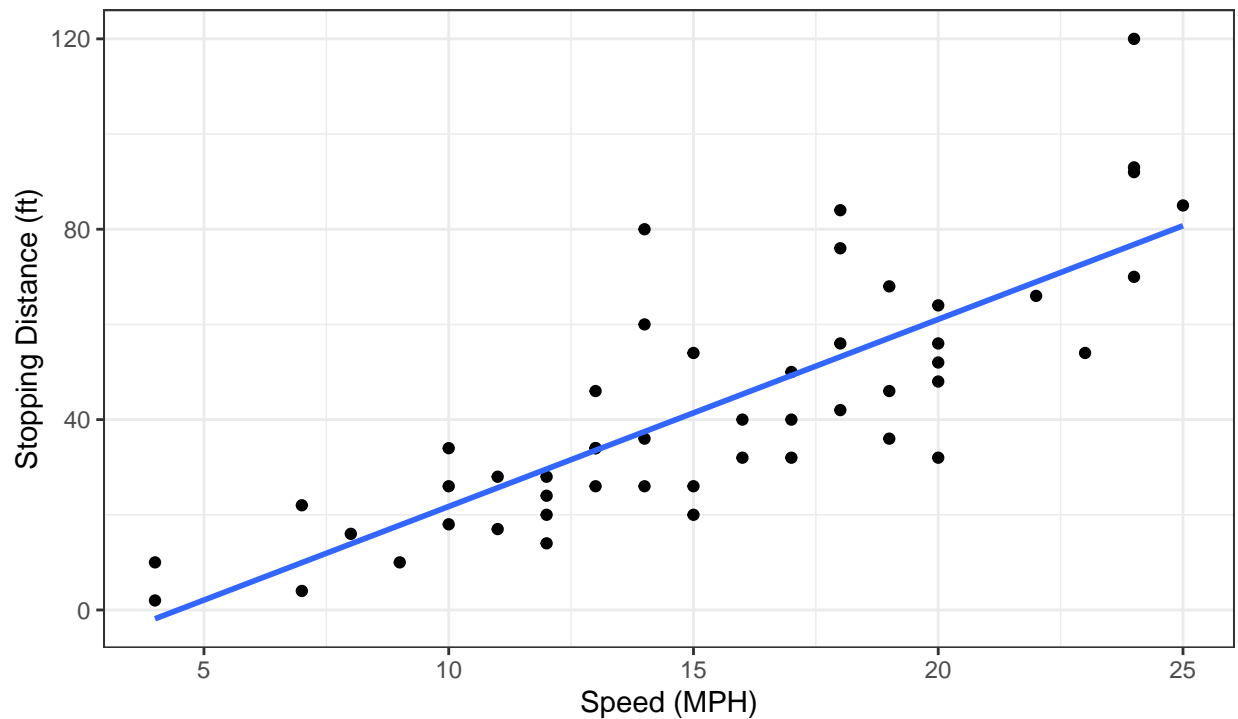


```
## add labs() function and changing the plot look to "black & white" theme by adding the theme_bw() fun
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  labs(title="Speed and Stopping Distances of Cars",
        x="Speed (MPH)",
        y="Stopping Distance (ft)",
        subtitle = "Your informative subtitle text here",
        caption="Dataset: 'cars'") +
  geom_smooth(method="lm", se=FALSE) +
  theme_bw()
```

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

Speed and Stopping Distances of Cars

Your informative subtitle text here



Dataset: 'cars'

```
##Adding more plot aesthetics through aes()
##Adjust the point size of a scatter plot using the size parameter.
##Change the point color of a scatter plot using the color parameter.
##Set a parameter alpha to change the transparency of all points.

#another exercise
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
```

```
colnames(genes)
```

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

```
ncol(genes)
```

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

```
## table the state in the genes database
```

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

```
##
```

```
##      down  unchanged      up
```

```
##      72      4997     127
```

```
## the round( ,2 ) here means 2 digital after the decimal point
```

```
round( table(genes$State)/nrow(genes) * 100, 2 )
```

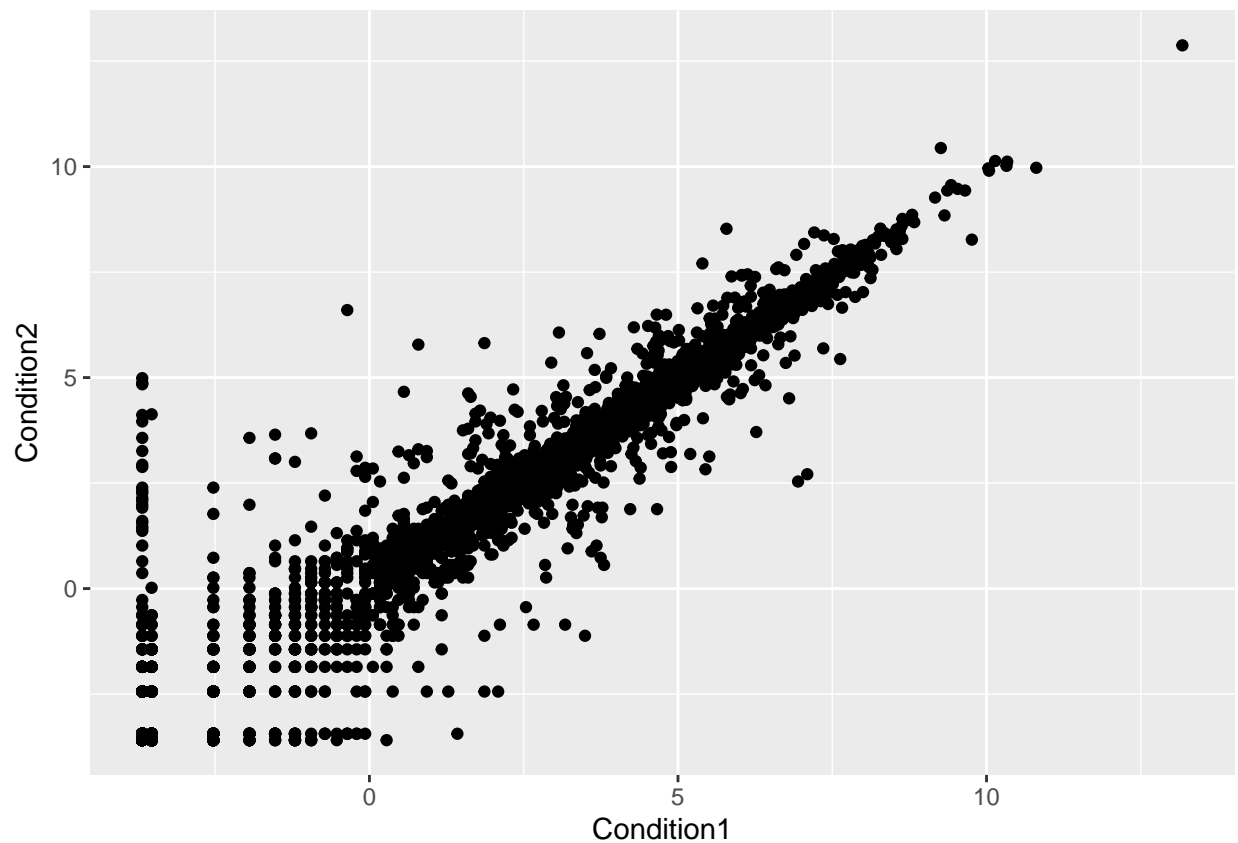
```
##
```

```
##      down  unchanged      up
```

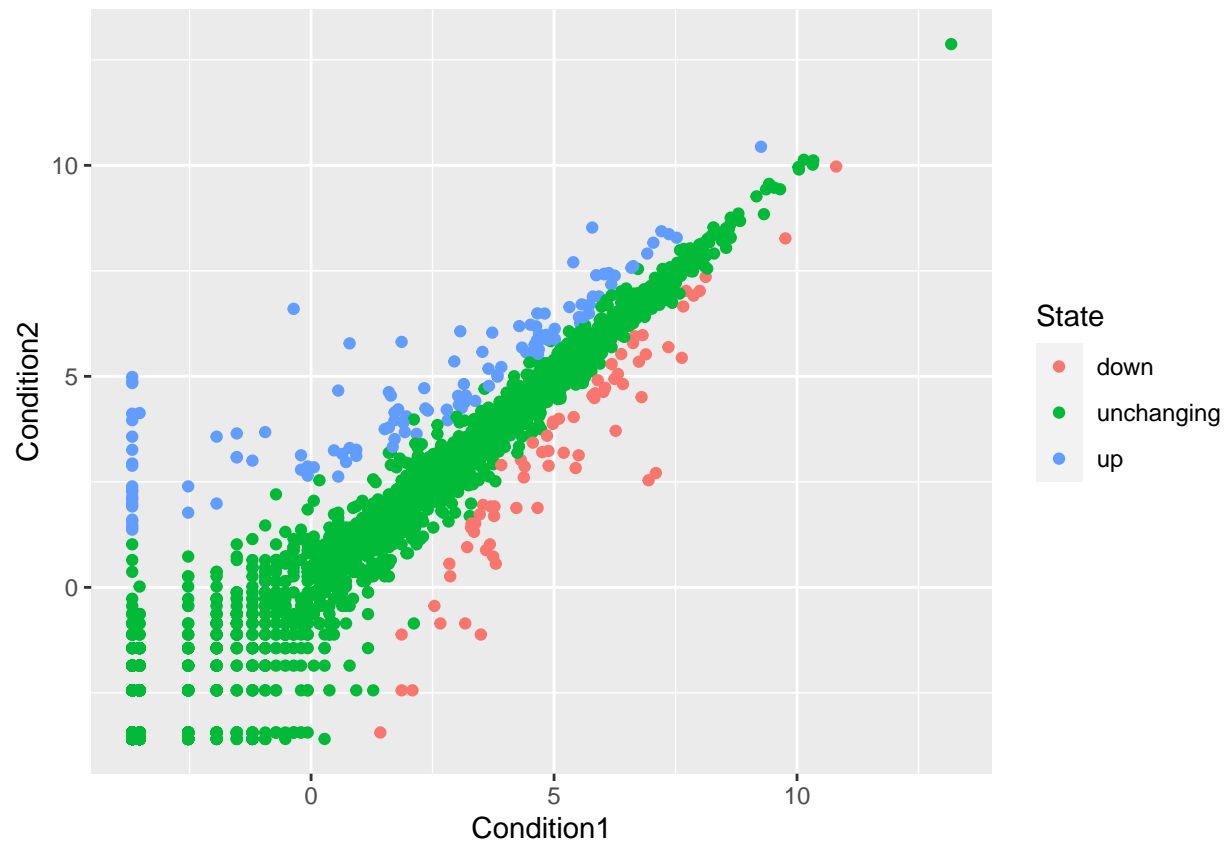
```
##      1.39      96.17      2.44
```

```
## make a basic plot first
```

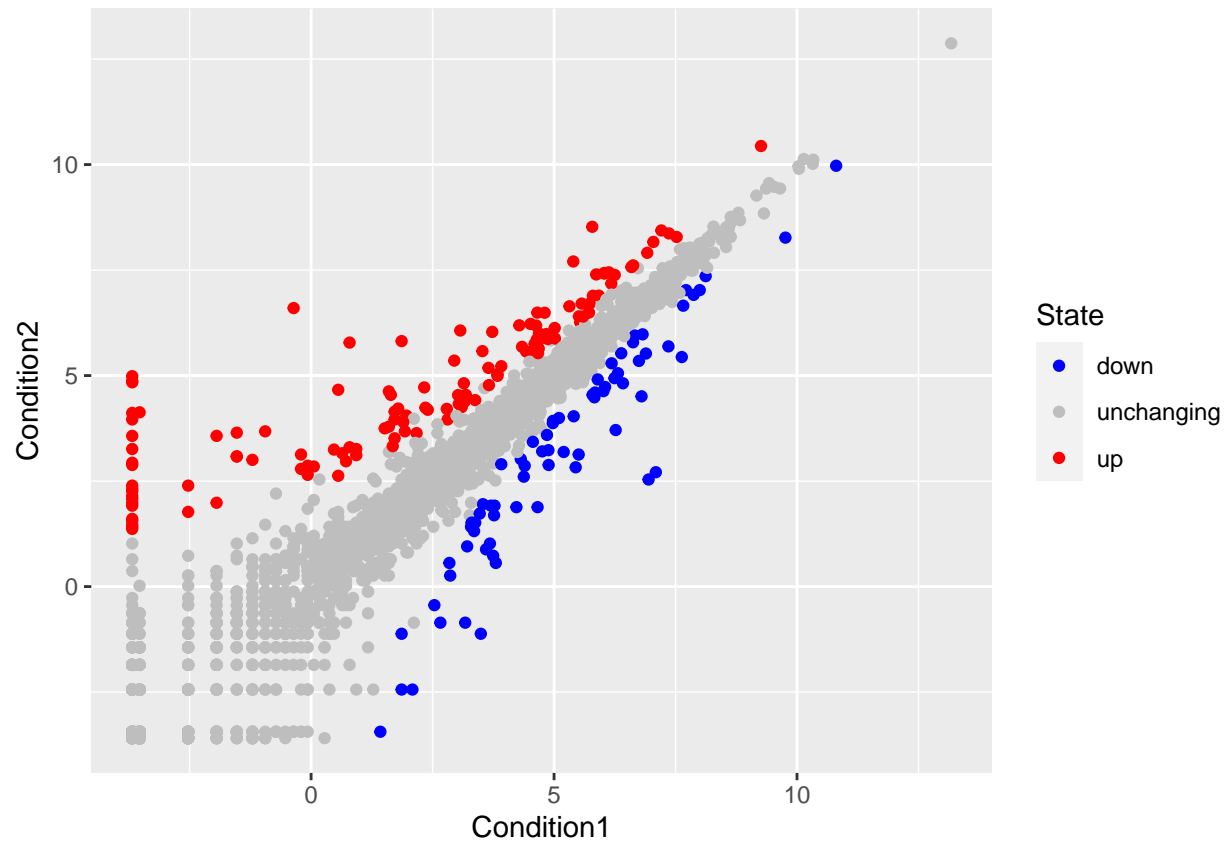
```
ggplot(genes) +  
  aes(x=Condition1, y=Condition2) +  
  geom_point()
```



```
## name this p, and add color for the dots according to the State
p <- ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point()
p
```

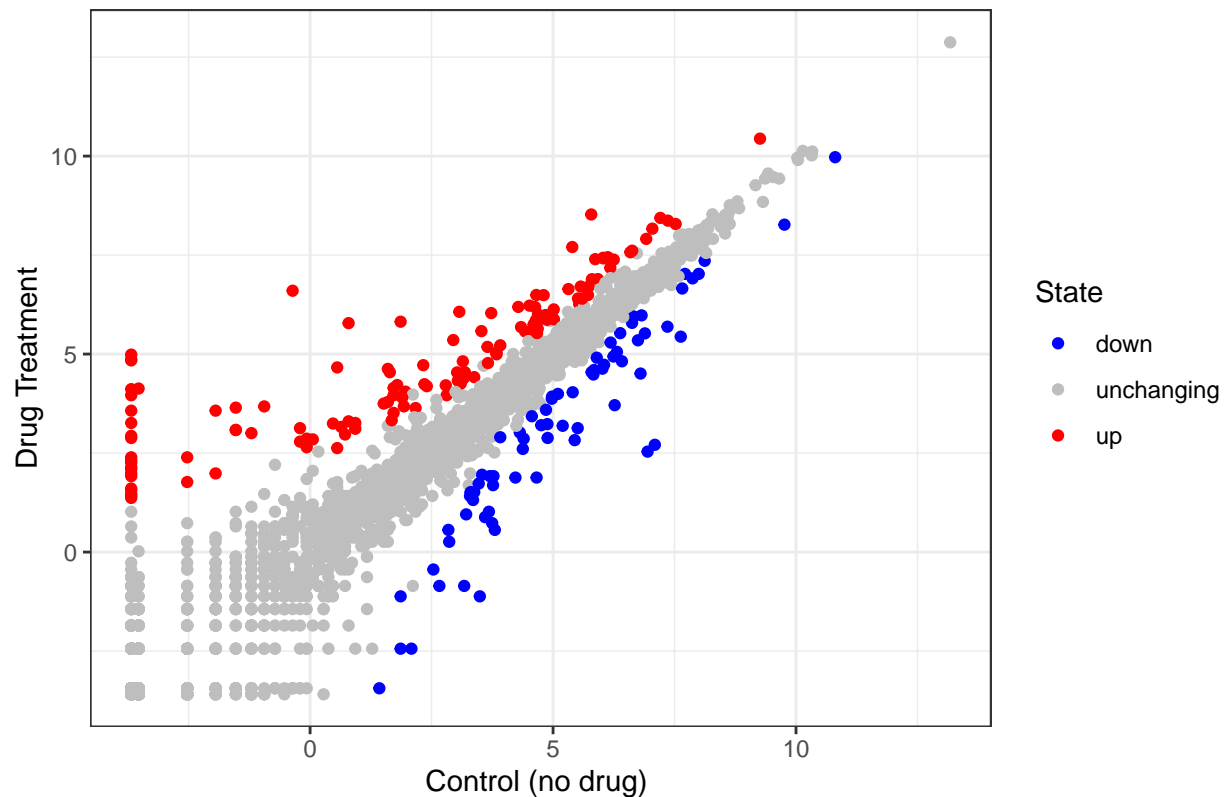


```
##change color for the dots manually
p + scale_colour_manual( values=c("blue","gray","red") )
```



```
## add labels and change color of theme
p + scale_colour_manual(values=c("blue","gray","red")) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
        x="Control (no drug) ",
        y="Drug Treatment") + theme_bw()
```

Gene Expression Changes Upon Drug Treatment

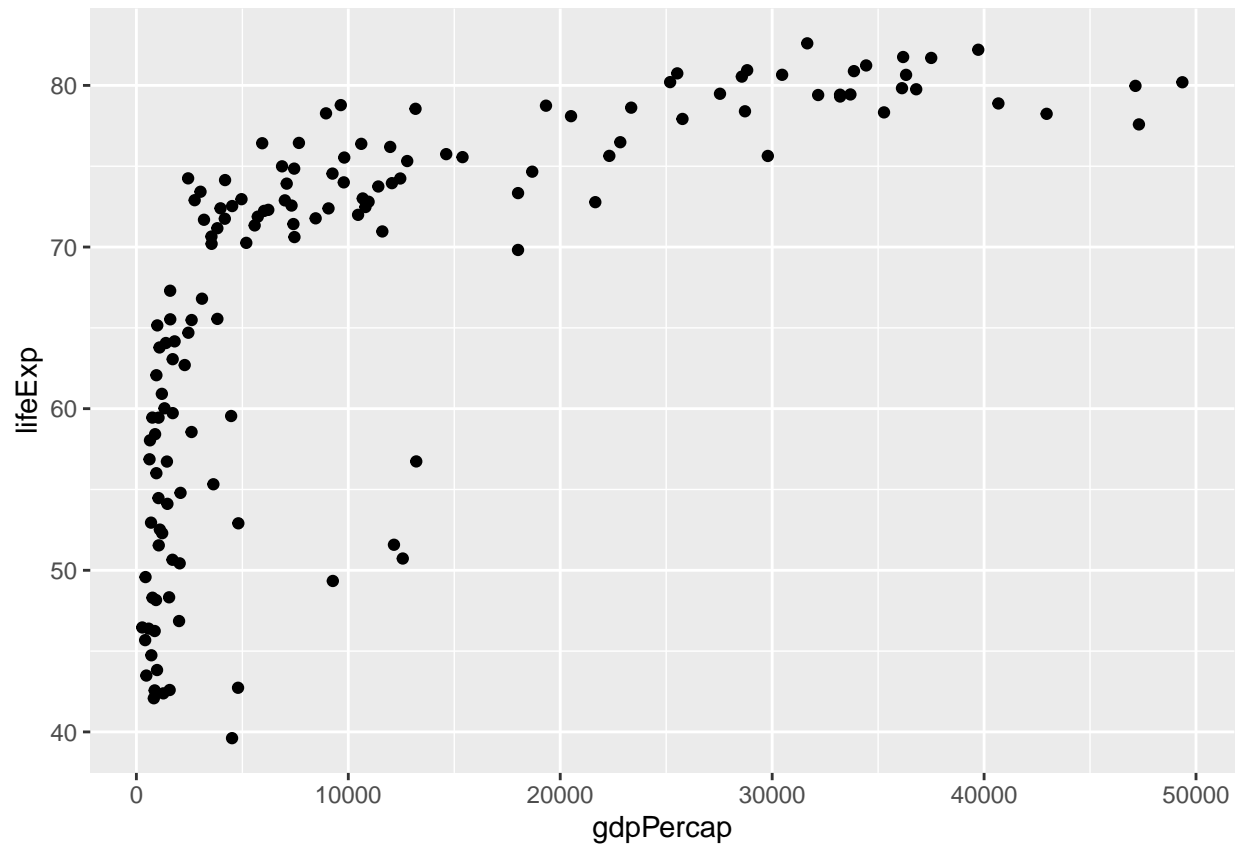


```
# what is more??  
library(gapminder)  
# only want 2007?  
library(dplyr)
```

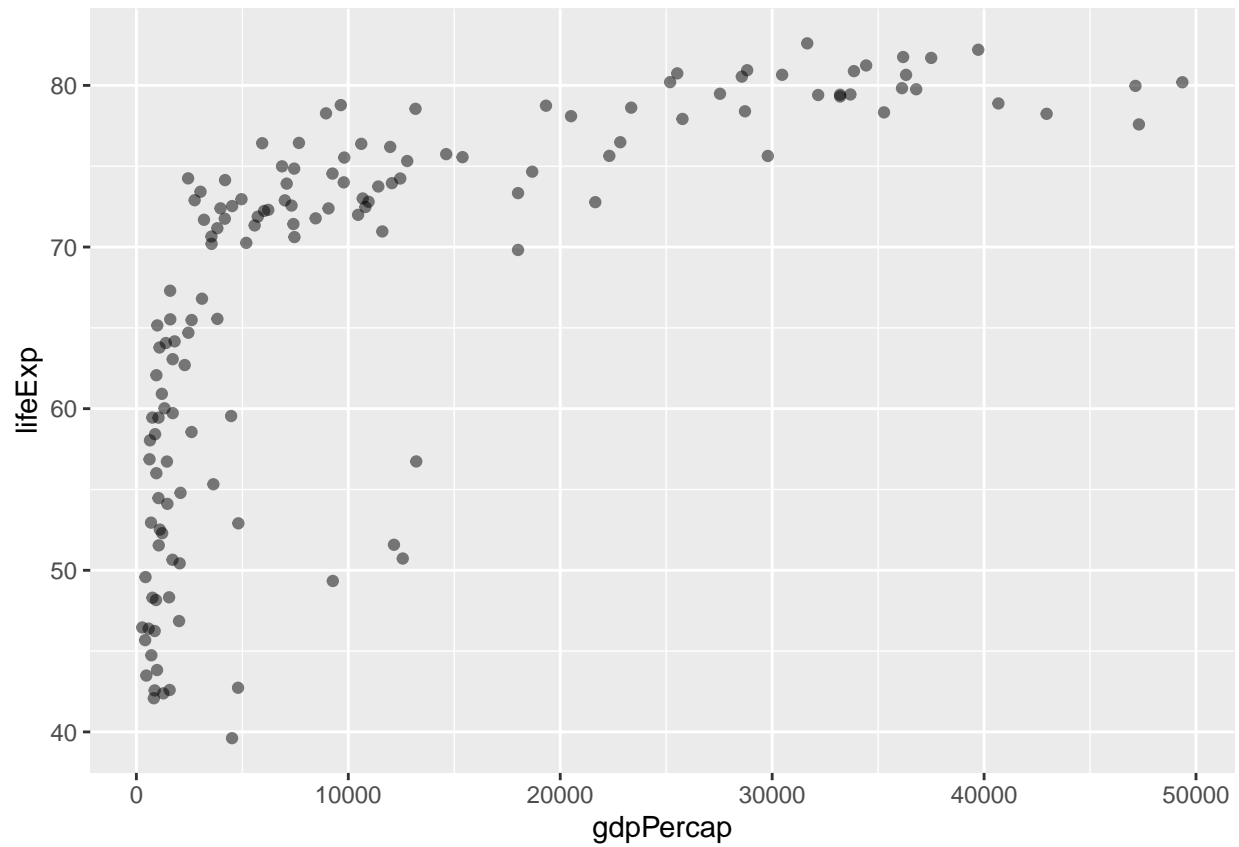
```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

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

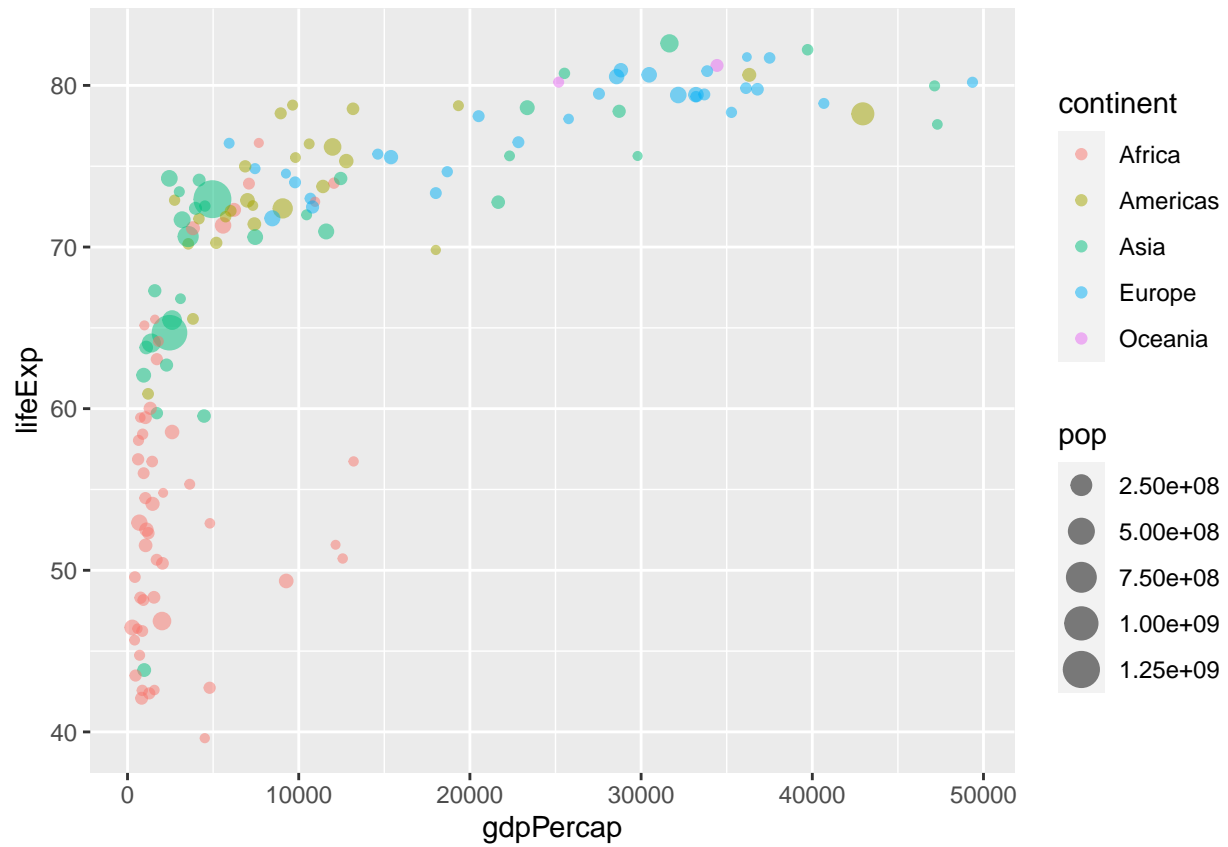
```
#Let's consider the gapminder_2007 dataset which contains the variables GDP per capita gdpPercap and li  
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y=lifeExp) +  
  geom_point()
```

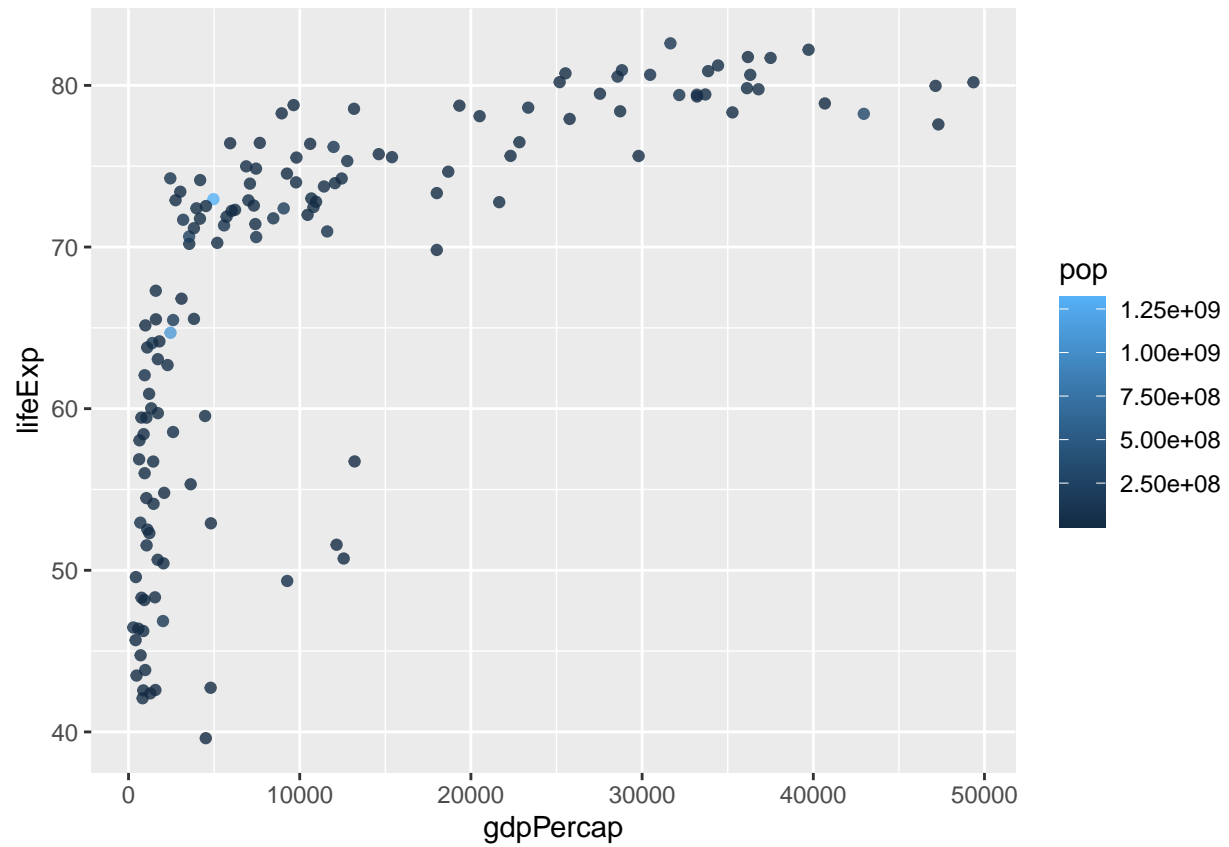
#on the top of each other? no worries! One useful approach here is to add an `alpha=0.4` argument to your
`ggplot(gapminder_2007) +`
 `aes(x=gdpPercap, y=lifeExp) +`
 `geom_point(alpha=0.5)`



```
# add the population pop (in millions) through the point size argument to aes() we can obtain a much ri  
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +  
  geom_point(alpha=0.5)
```

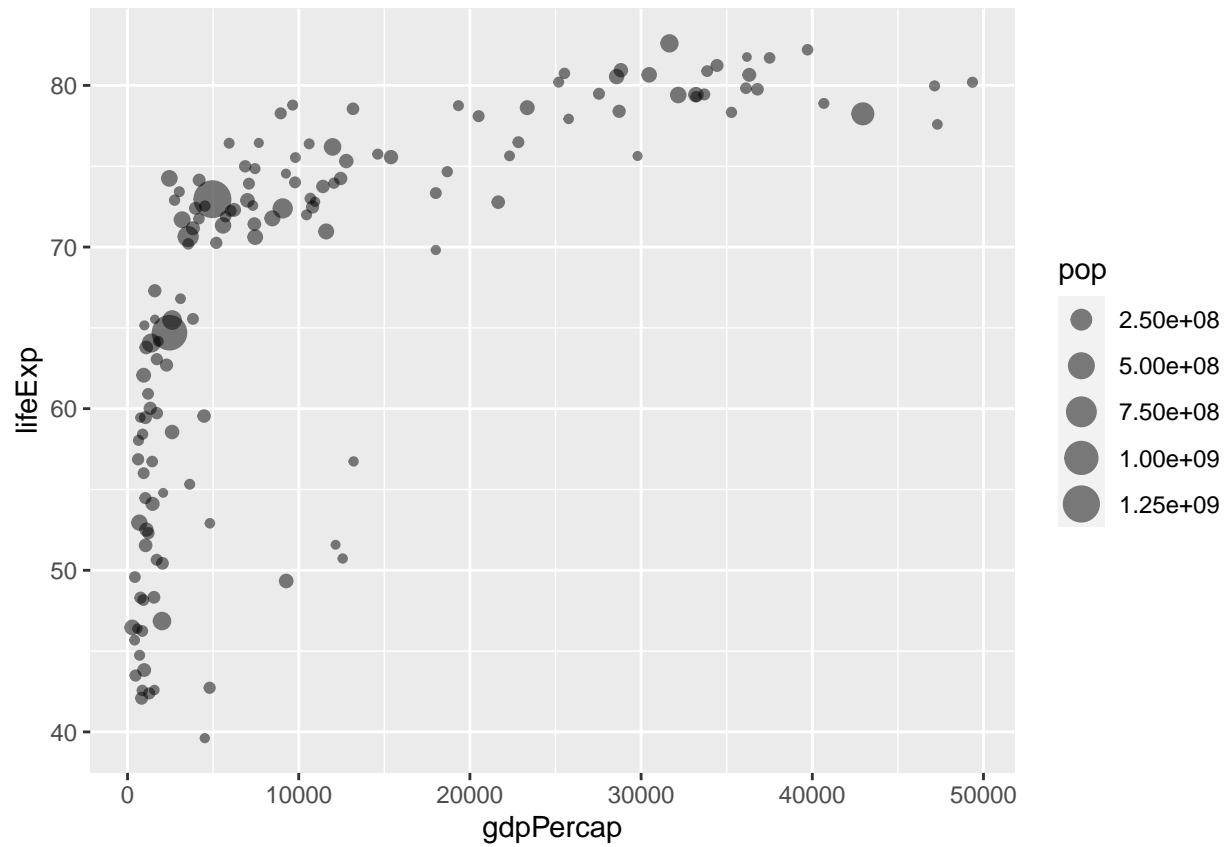


```
# we also can color the dots by the numeric variable population pop
ggplot(gapminder_2007) +
  aes(x = gdpPerCap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```

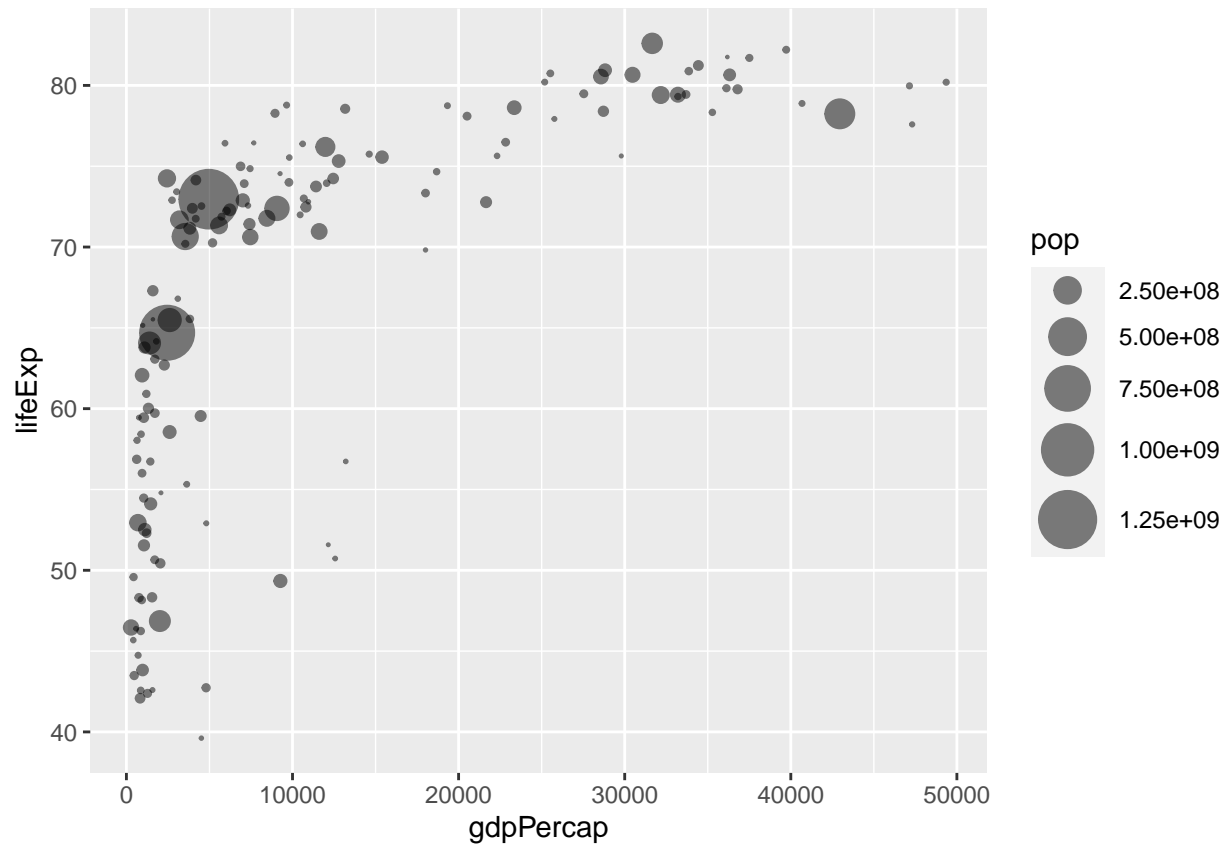


#I do not like using color to indicate the population in this case. let me change back to using the size

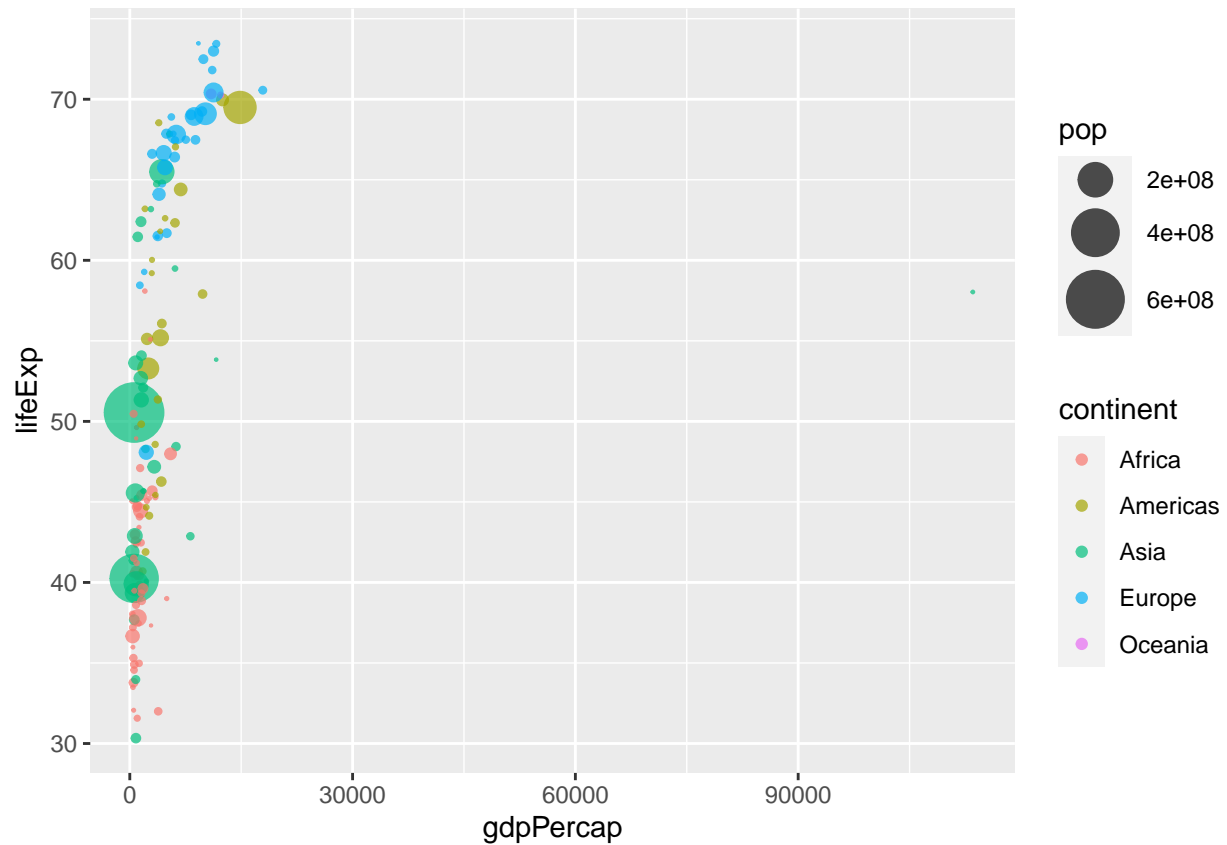
```
ggplot(gapminder_2007) +  
  aes(x = gdpPerCap, y = lifeExp, size = pop) +  
  geom_point(alpha=0.5)
```



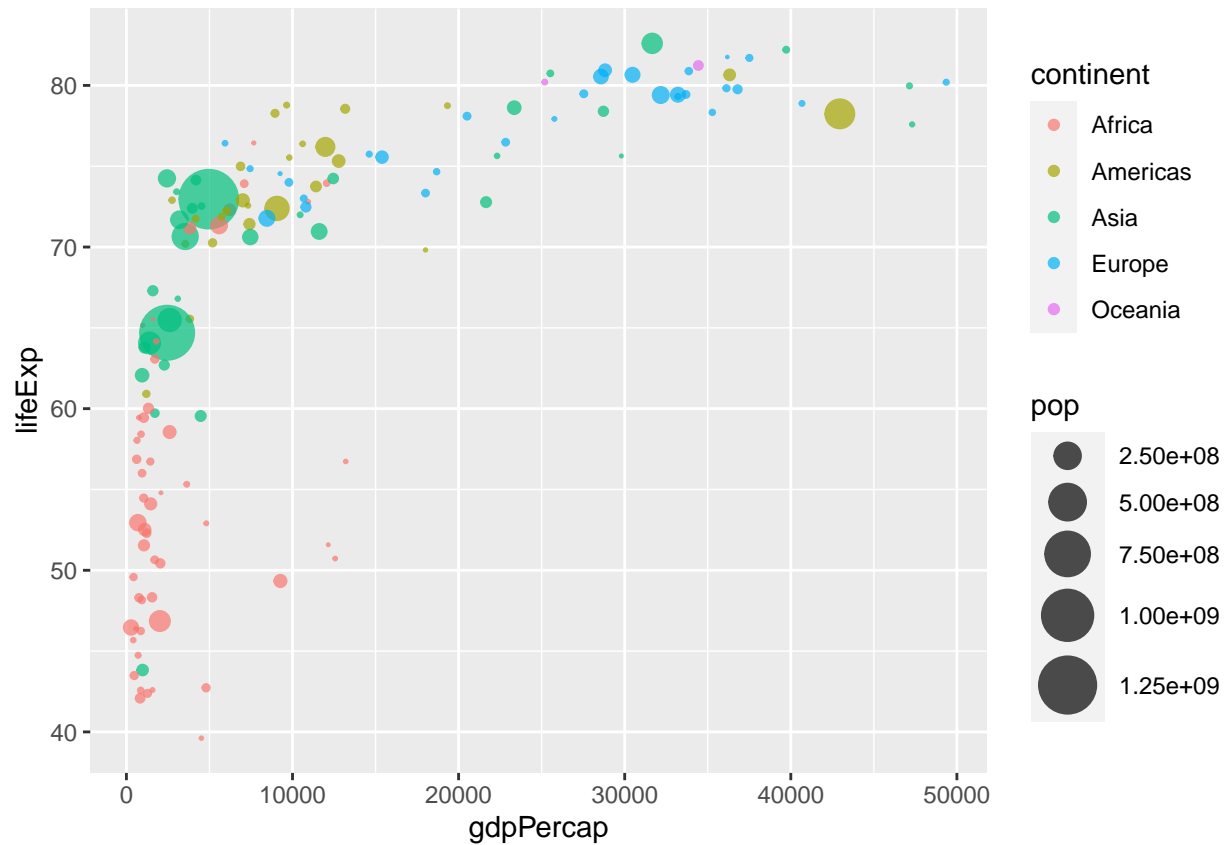
```
# This is not really reflecting the real size. let us use scale_size_area() function  
ggplot(gapminder_2007) +  
  geom_point(aes(x = gdpPerCap, y = lifeExp,  
                 size = pop), alpha=0.5) +  
  scale_size_area(max_size = 10)
```



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



```
gapminder_top5 <- gapminder %>% filter(year==2007)
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp,color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10)
```



```
# How to put them together?? multi-panel plots in R using facet_wrap()
gapminder_1957 <- gapminder %>% filter(year==1957 | year==2007)

ggplot(gapminder_1957) +
  geom_point(aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```