

Class 05: Data Visualization

Laurie Chang A16891192

Graphics systems in R

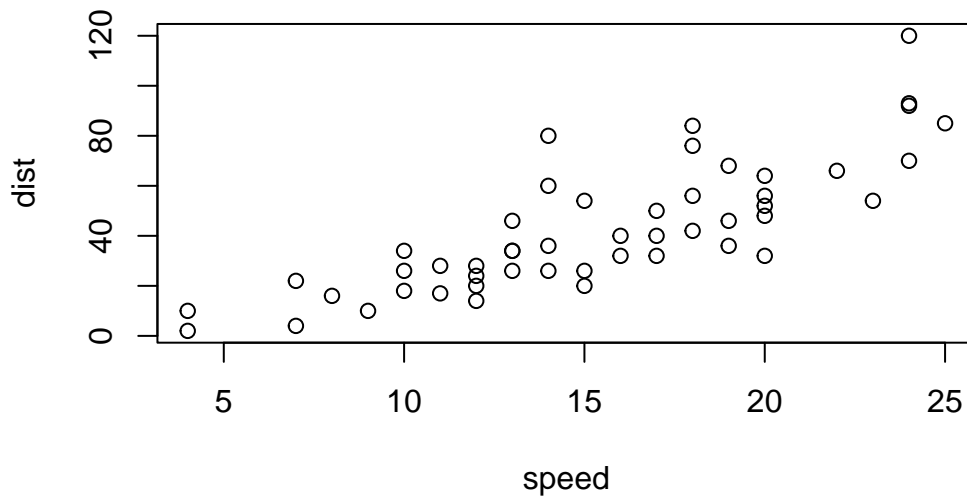
There are many graphics systems in R for making plots and figures.

We have already played a little with **base R** graphics and the `plot()` function.

Today, we will start learning about a popular graphics package called `ggplot2()`.

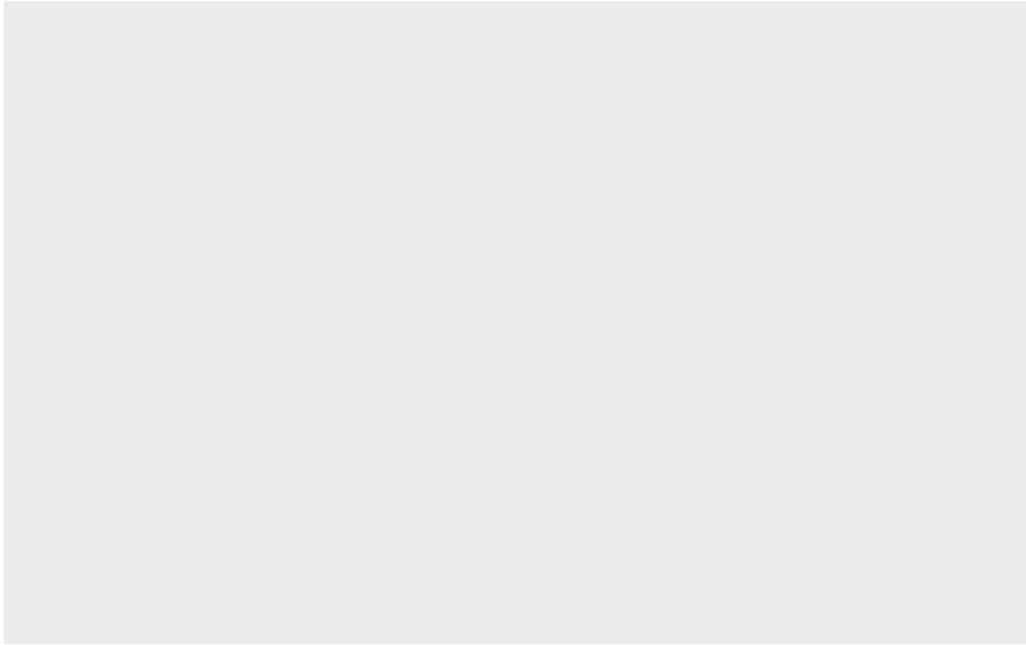
This is an add on package - i.e. we need to install it. I install it (like I install any packing) with `install.package()` function.

```
plot(cars)
```



Before I can use the functions from a package, I have to load up the package from my “library”. We use the `library(ggplot2)` command to load it up.

```
library(ggplot2)
ggplot(cars)
```



Every ggplot is made up of at least 3 things:

- data (the numbers, etc. that will go into your plot)
- aesthetics (aes) (how the columns of data map to the plot aesthetics)
- geometries (geoms) (how the plot actually looks: points, bars, lines, etc.)

```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point()
```



For simple plots, ggplot is more verbose- it takes more code - than base R plot.

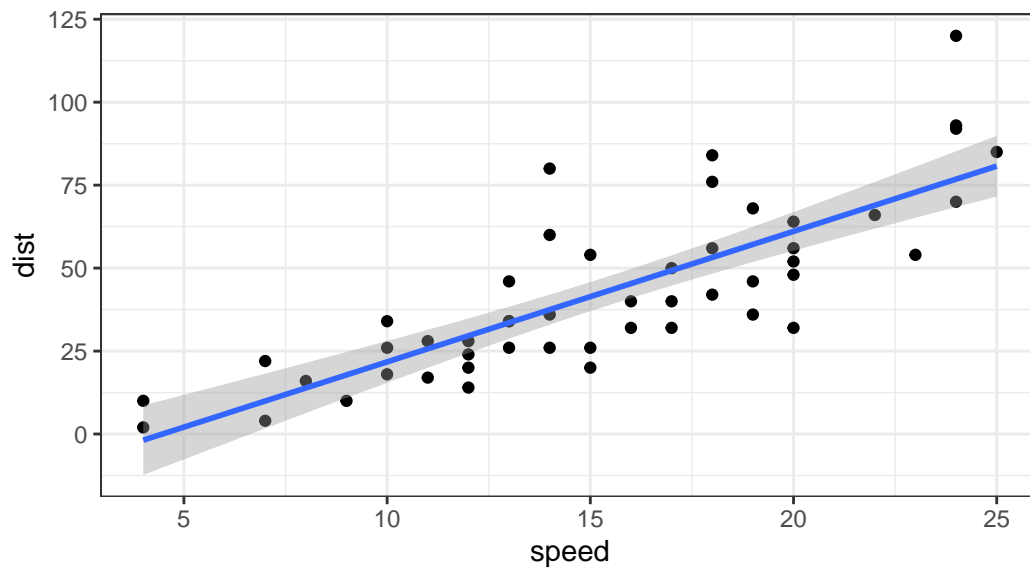
Add some more layers to our ggplot:

```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  geom_smooth(method= "lm") +  
  labs(title="Stopping distance of old cars", subtitle= "a silly simple plot") +  
  theme_bw()
```

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

Stopping distance of old cars

a silly simple plot

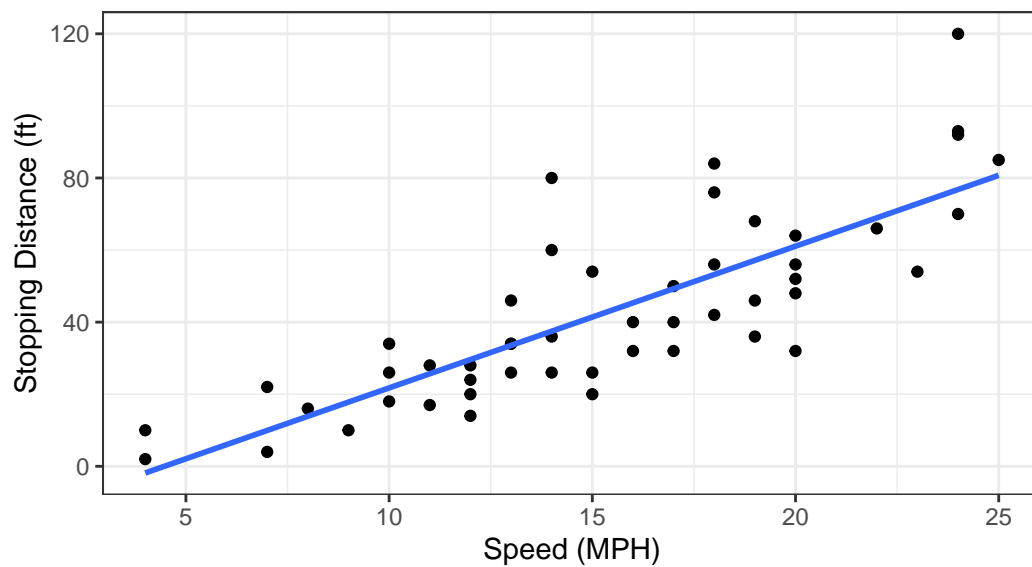


```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  geom_smooth(method= "lm", se = FALSE) +  
  labs(title="Stopping distance of old cars",  
        subtitle= "a silly simple plot",  
        x = "Speed (MPH)",  
        y = "Stopping Distance (ft)") +  
  theme_bw()
```

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

Stopping distance of old cars

a silly simple plot



#Adding more plot aesthetics through `aes()`

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
```

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

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

down	unchanging	up
72	4997	127

```
round(127/5196, 4)*100
```

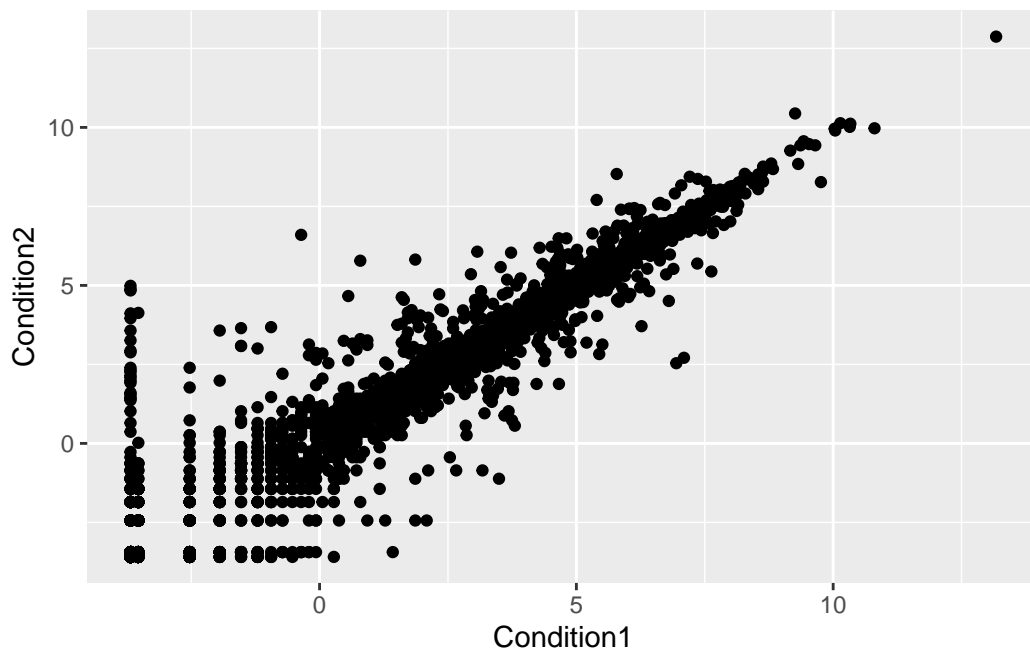
```
[1] 2.44
```

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

down	unchanging	up
1.39	96.17	2.44

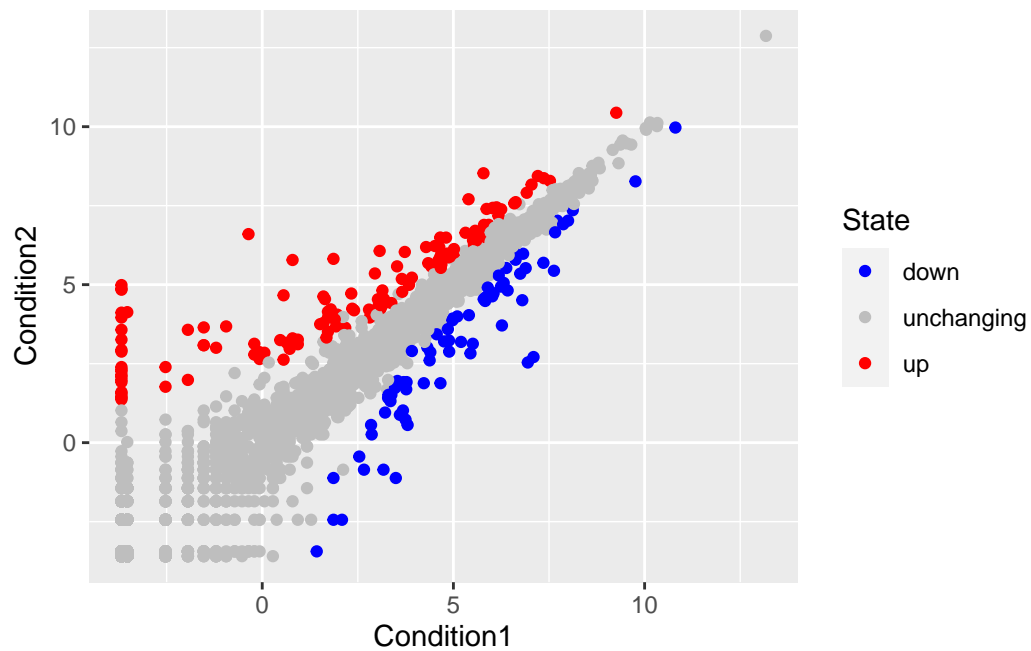
Basic scatterplot

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



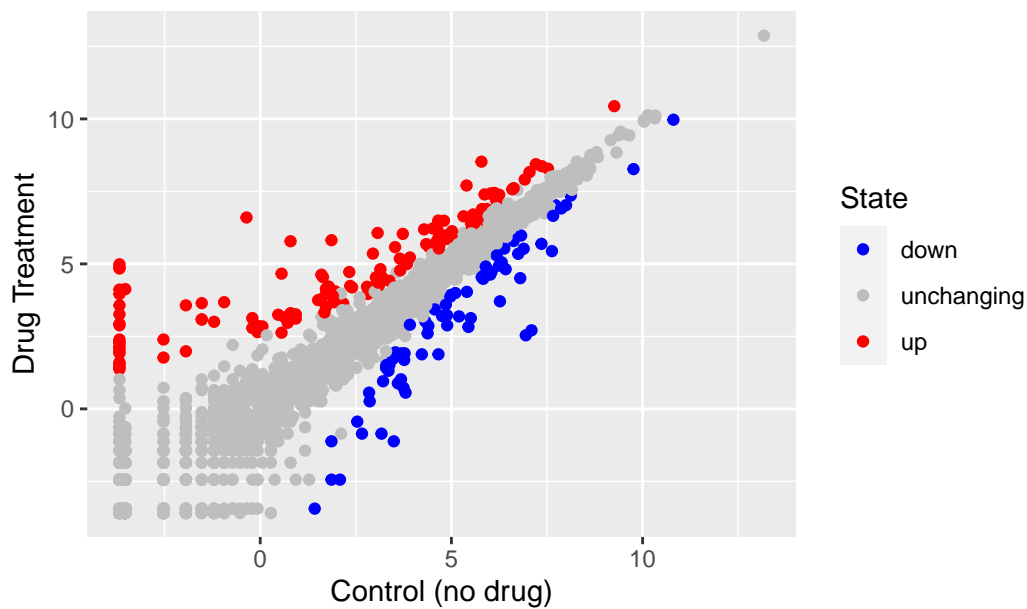
```
p <- ggplot(genes)+
  aes(x =Condition1, y = Condition2, col=State)+
  geom_point()

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



```
p + scale_color_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



Going Further

```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.  
library(dplyr)
```

Attaching package: 'dplyr'

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

filter, lag

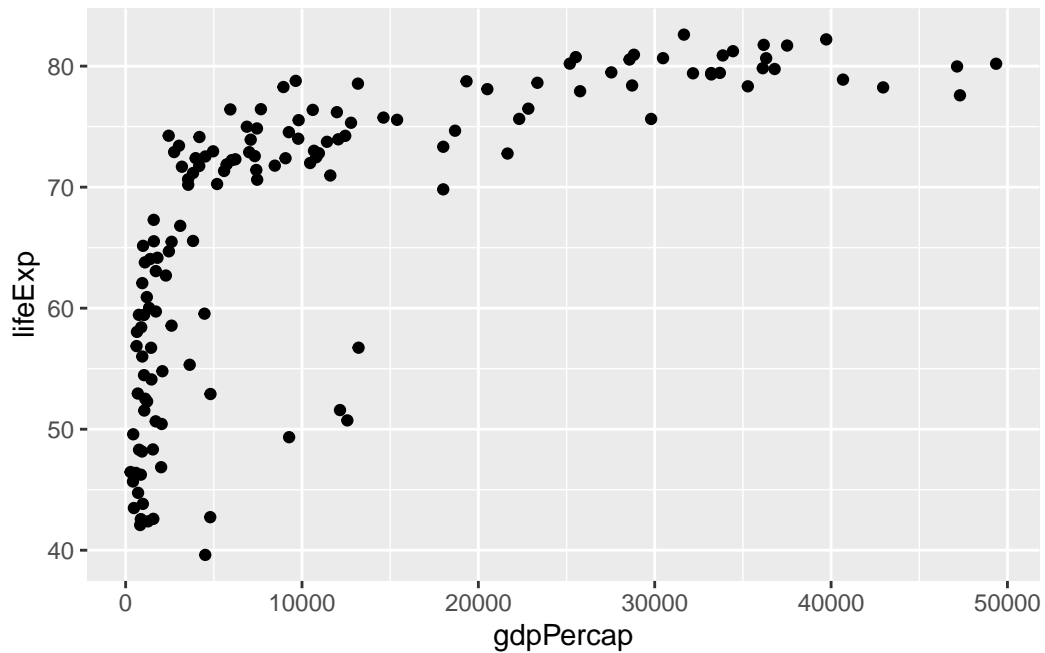
The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

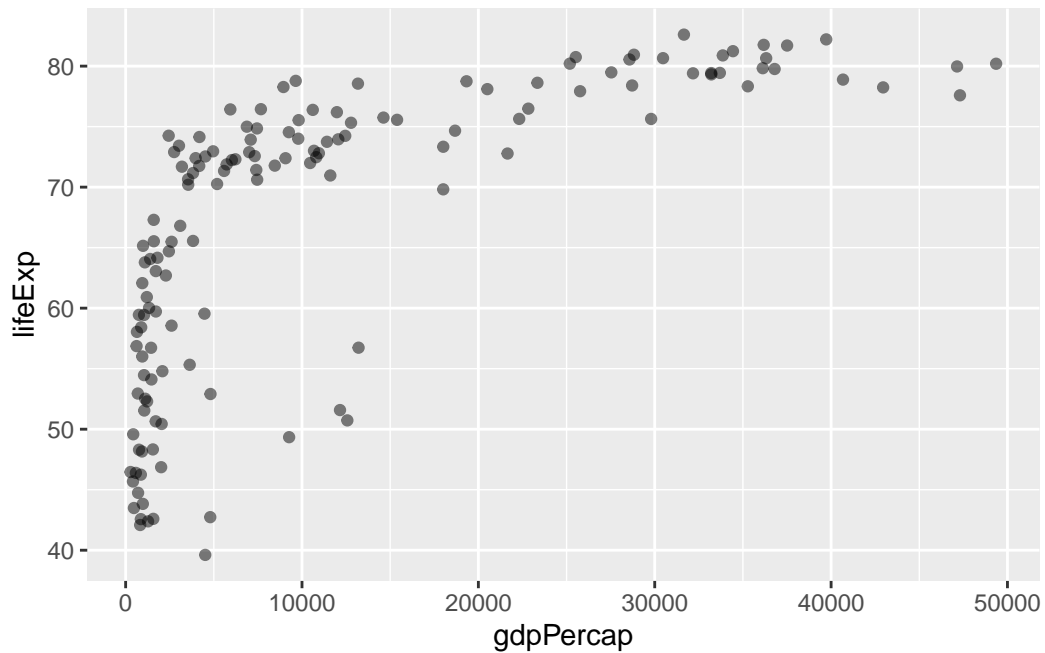
```
gapminder <- read.delim(url)  
gapminder_2007 <- gapminder %>% filter(year==2007)
```



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



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

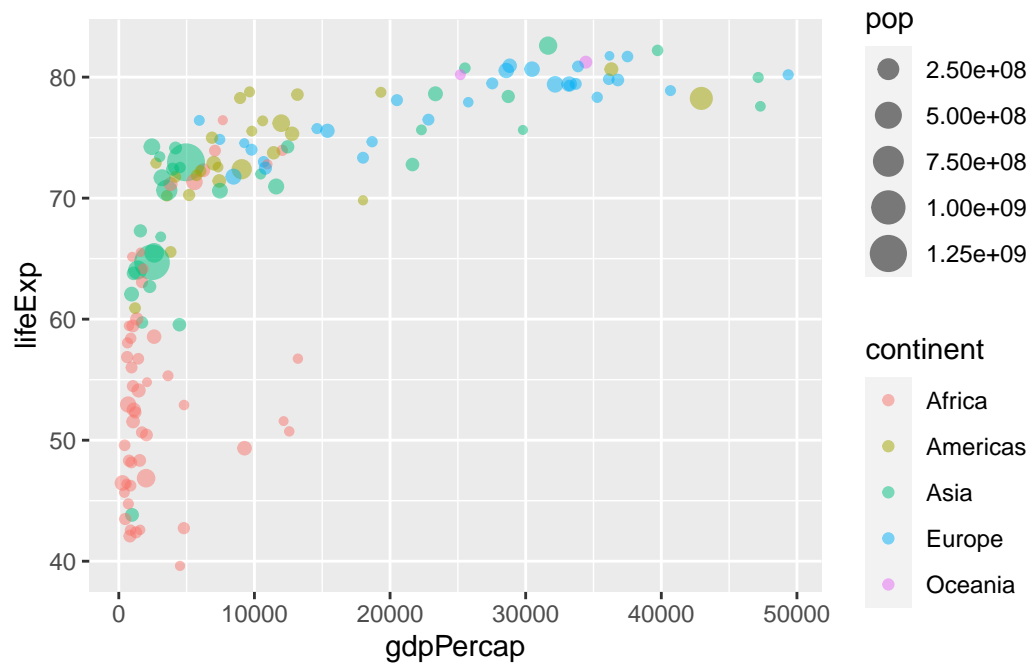


adding more variables to `aes()`

mapping the `continent` variable to point `color` aesthetic and the population `pop` (in millions) through the point `size` argument to `aes()`

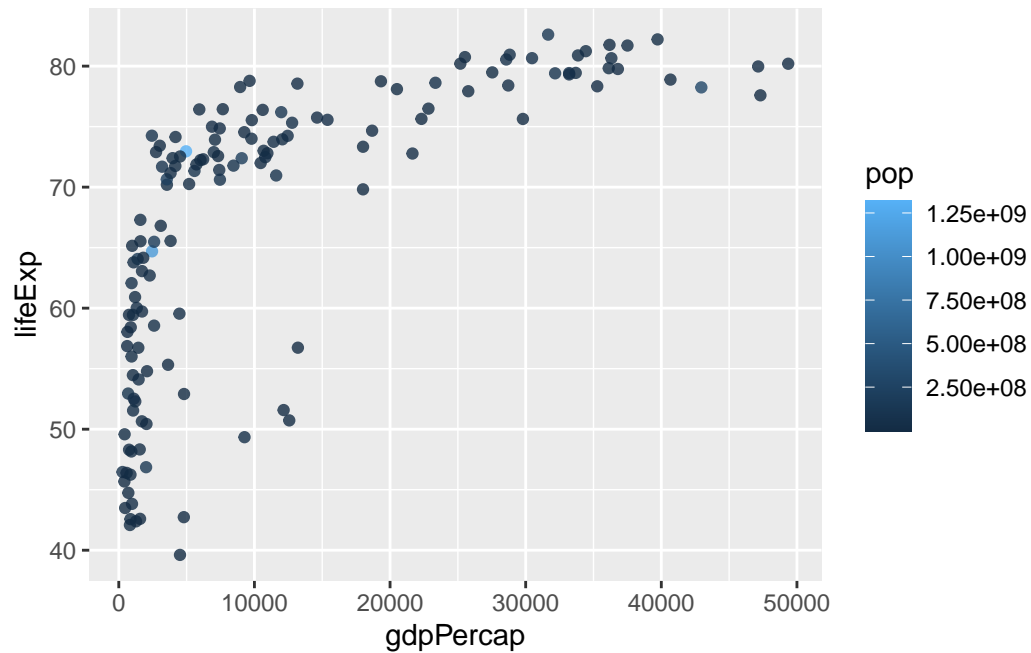
plot includes 4 different variables from data set

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



color the points by the numeric variable population pop

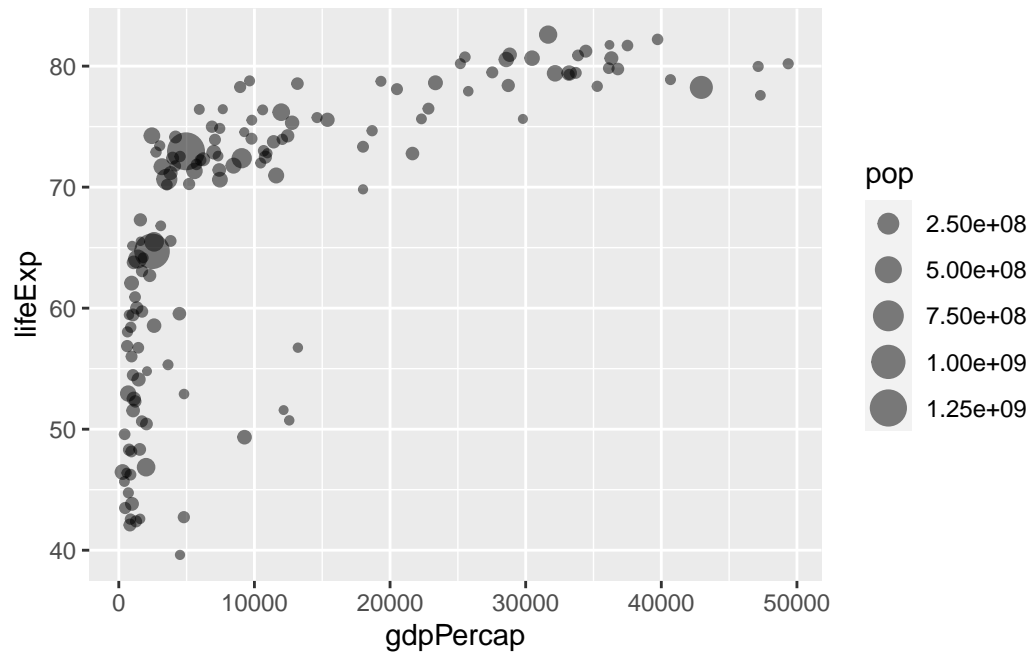
```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



- now a continuous scale as seen in legend
- light blue points now the countries with the highest population number (China and India)

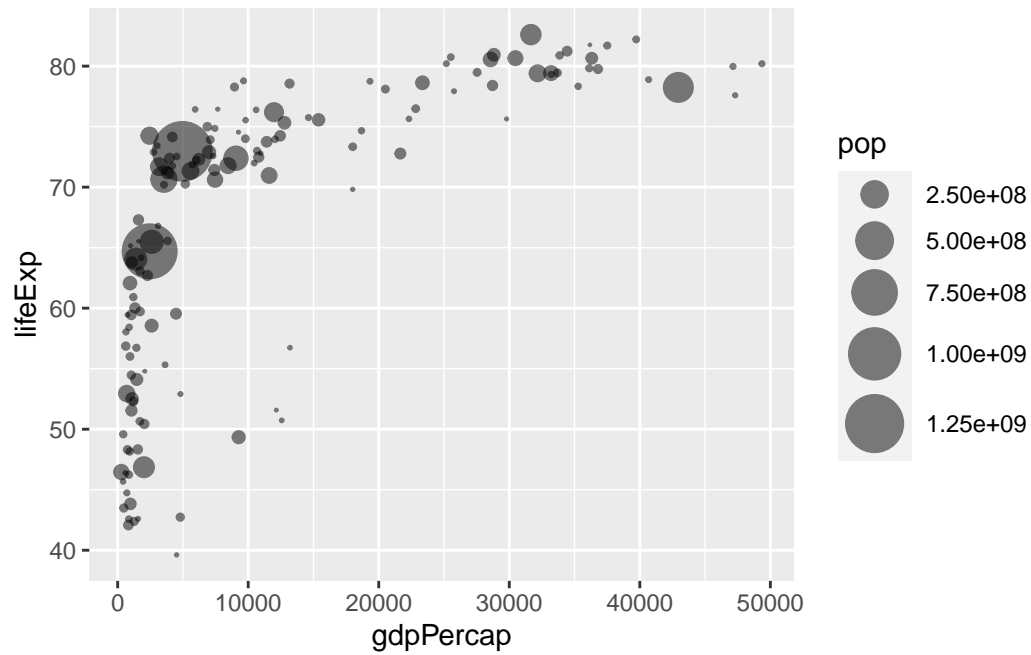
Adjusting point size

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



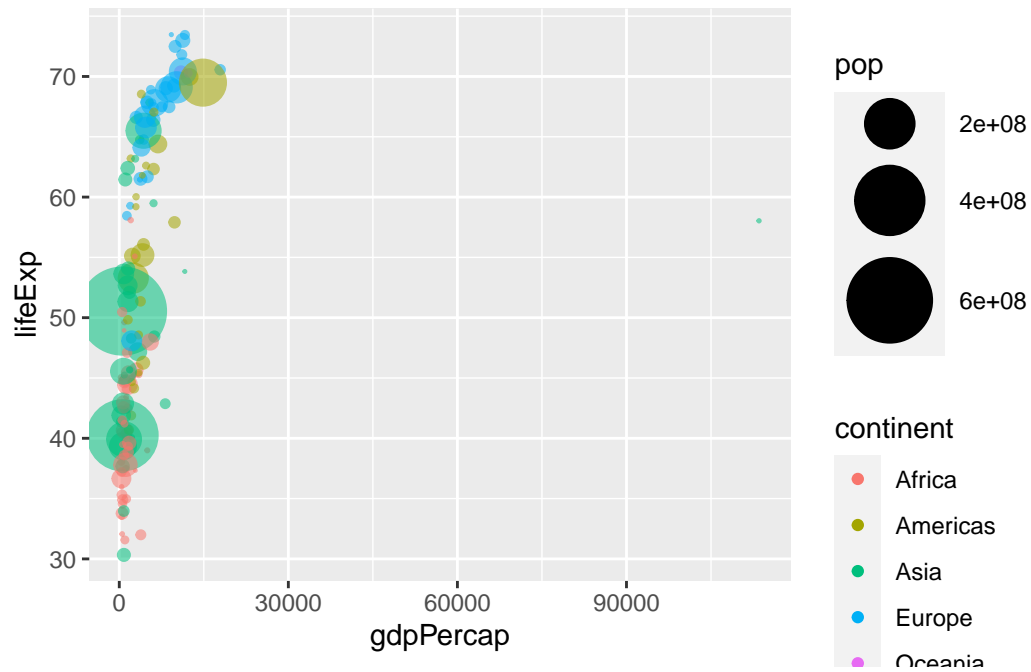
- point sizes in the plot do not clearly reflect the population differences in each country (sizes are not proportional)
- point sizes binned by default

```
ggplot(gapminder_2007) +  
  geom_point(aes(x = gdpPercap, y = lifeExp,  
                 size = pop), alpha=0.5) +  
  scale_size_area(max_size = 10)
```



1957 plot

```
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 = 15)
```



```
gapminder_combined <- gapminder %>% filter(year==1957 | year==2007)
ggplot(gapminder_combined)+
  geom_point(aes(x = gdpPercap, y= lifeExp, color= continent, size = pop), alpha = 0.7) +
  scale_size_area(max_size = 15) +
  facet_wrap(~year)
```

