

Class 05: Data Visualization with GGLOT

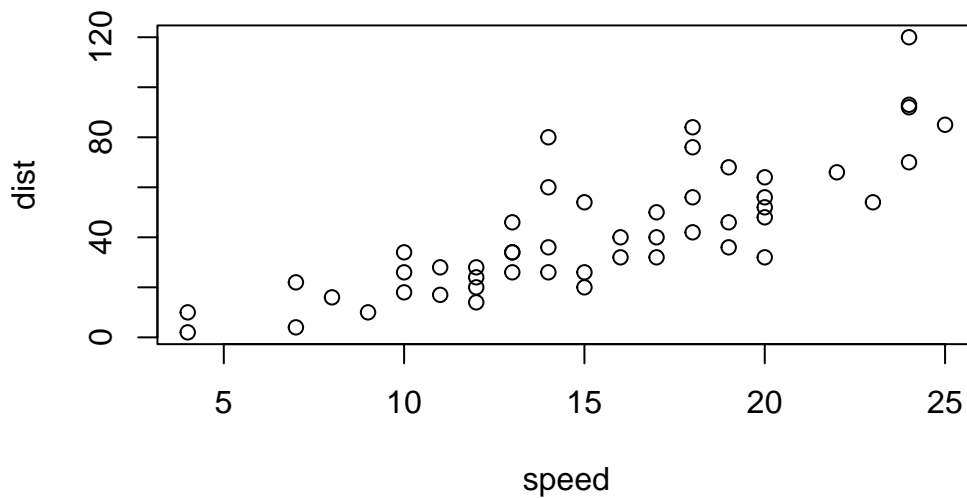
Angela Liu

Plotting in R

R has multiple plotting and graphic systems. The most popular of which is **ggplot2**.

We have already played with “base” R graphics. This comes along with R “out of the box”.

```
plot(cars)
```



Compared to base R plots, ggplot is much more verbose - I need to write more code to get simple plots like the above.

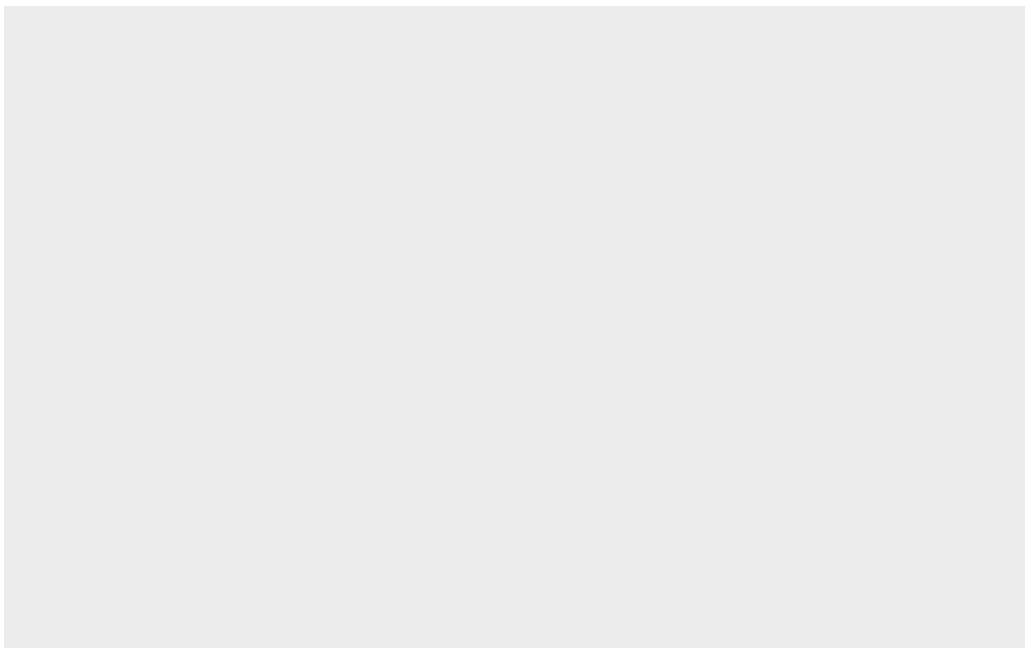
To use ggplot, we need to install the ggplot2 package. To install any package in R, use the `install.packages()` command along with the package name.

The install is a one time only requirement. The package is now on the computer and does not need to be re-installed.

However, ggplot cannot just be used without loading it up in a `library()` call.

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

```
ggplot(cars)
```

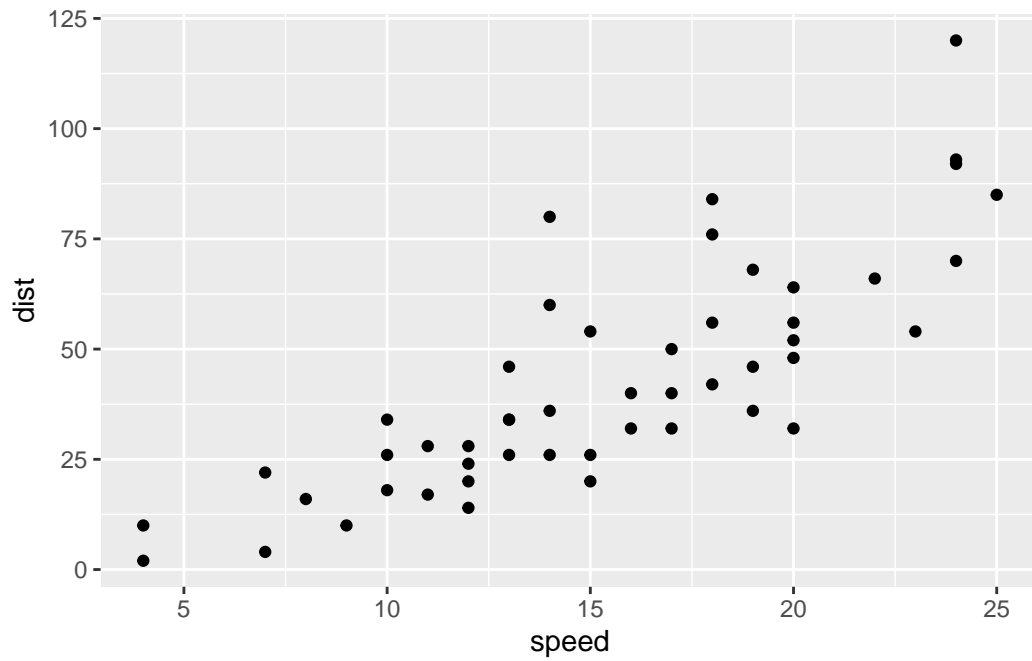


All ggplot figures need at least three things:

1. data (this is the data.frame with our numbers)
2. aesthetics (“aes”, how our data maps to the plot like x-axis, y-axis, colors, etc)
3. geoms (lines, points, columns, etc)

```
bb <- ggplot(data=cars) +  
  aes(x=speed, y=dist) +  
  geom_point()
```

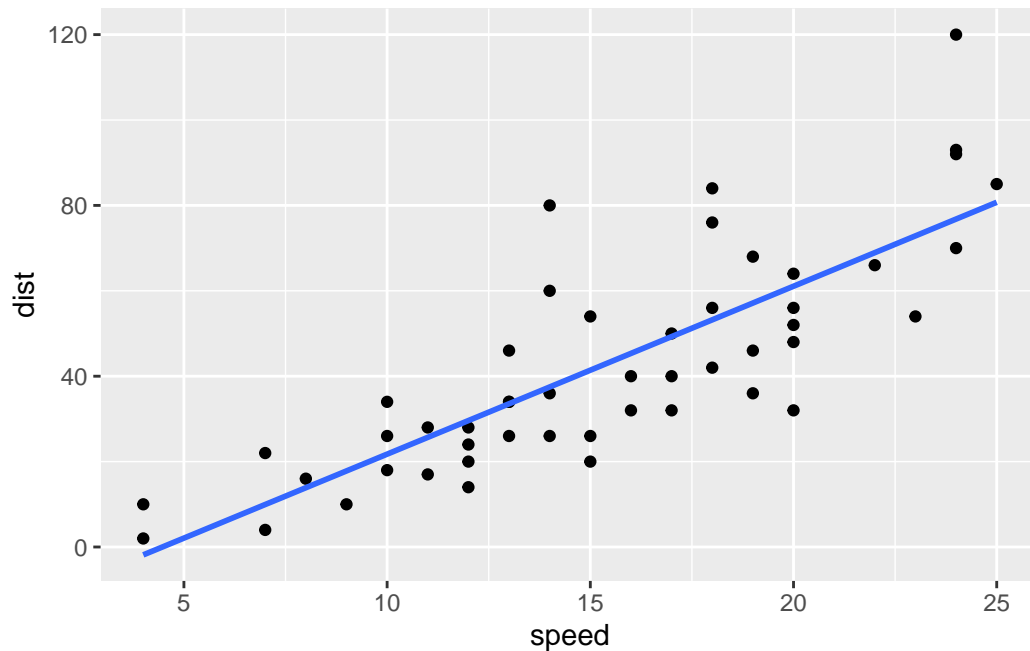
```
bb
```



I want a trendline to show the relationship between speed and stopping distance.

```
bb + geom_smooth(method = "lm", se = FALSE)
```

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



Genes

Downloading the data set and displaying the first few rows of the data:

```
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
```

```
ncol(genes)
```

```
[1] 4
```

```
state <- table(genes$State) #get table to find number of 'up' regulated genes
state
```

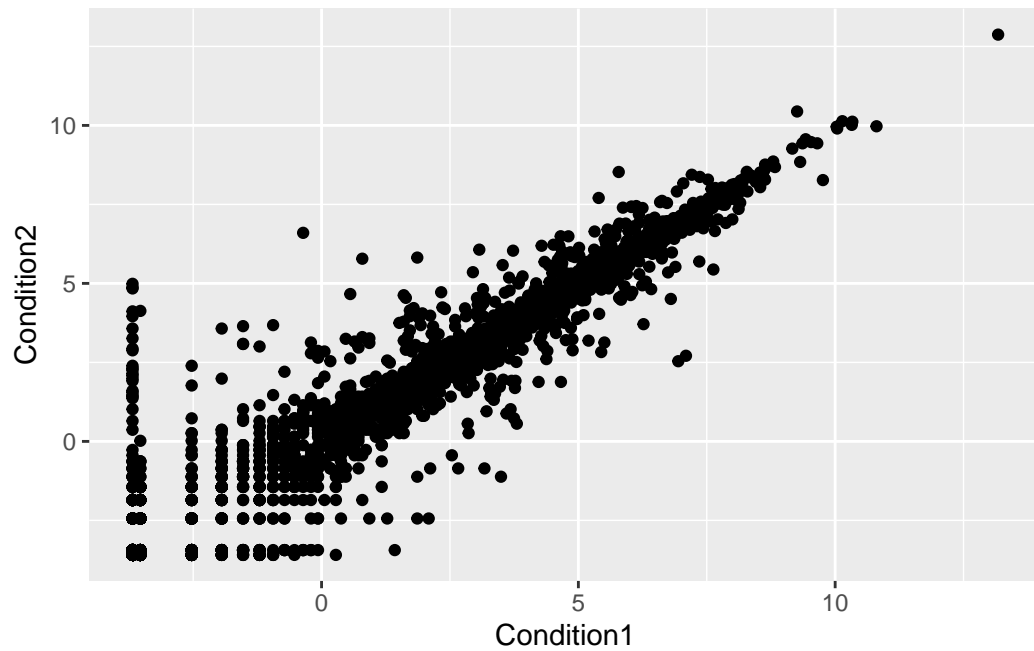
down	unchanging	up
72	4997	127

```
round(state/nrow(genes) * 100, 2) #to find fraction of up-regulated genes out of total, rounded
```

down	unchanging	up
1.39	96.17	2.44

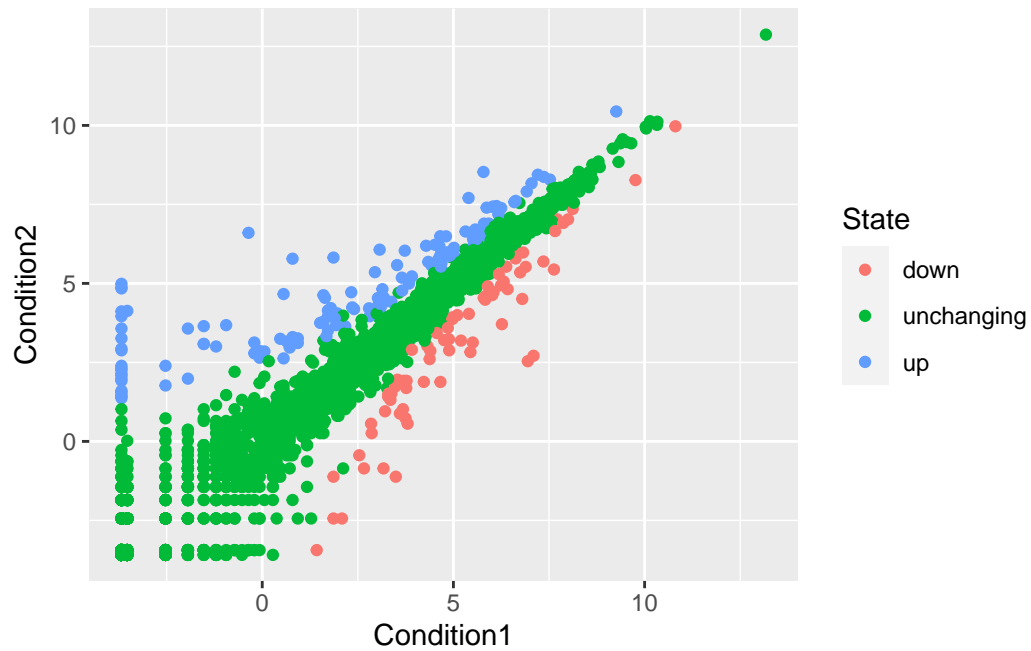
Making a basic scatter plot:

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



There's extra information in the data set – in the **State** column that shows if there is a statistically significant difference in expression values between conditions.

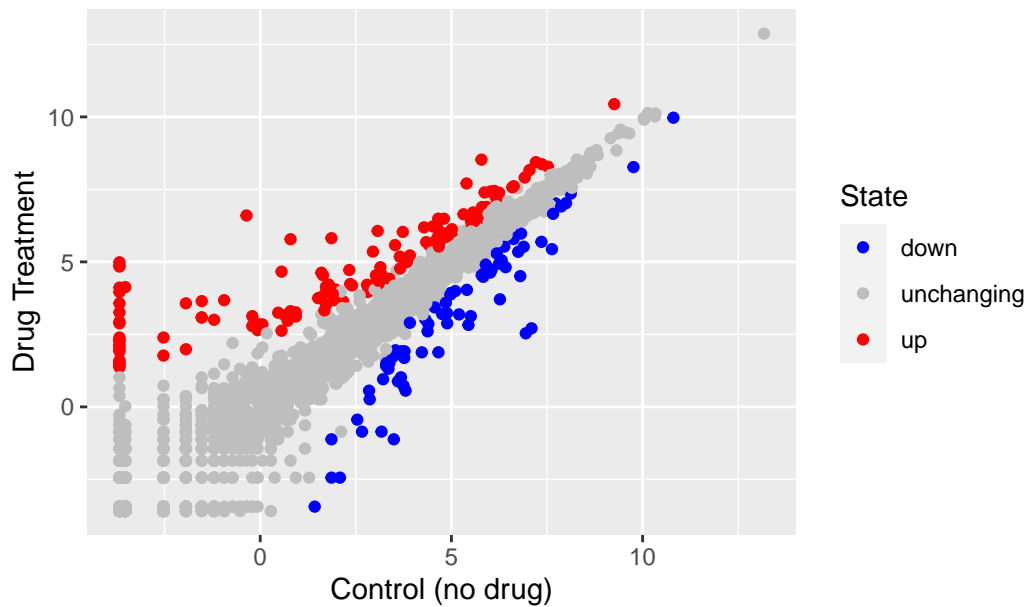
```
p <- ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) +  
  geom_point()  
p
```



To change the colors in this plot with `scale_colour_manual` and add some better labels with the `labs()` function:

```
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



Gapminder

```
#install.packages("gapminder")
library(gapminder)
#install.packages("dplyr")
library(dplyr)
```

Attaching package: 'dplyr'

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

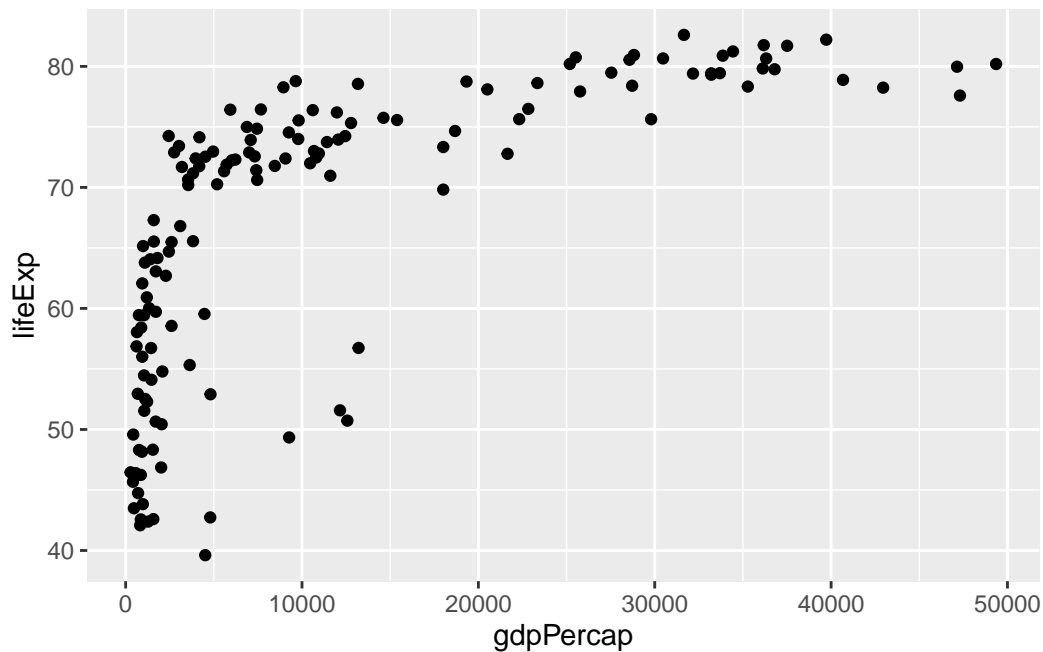
filter, lag

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

intersect, setdiff, setequal, union

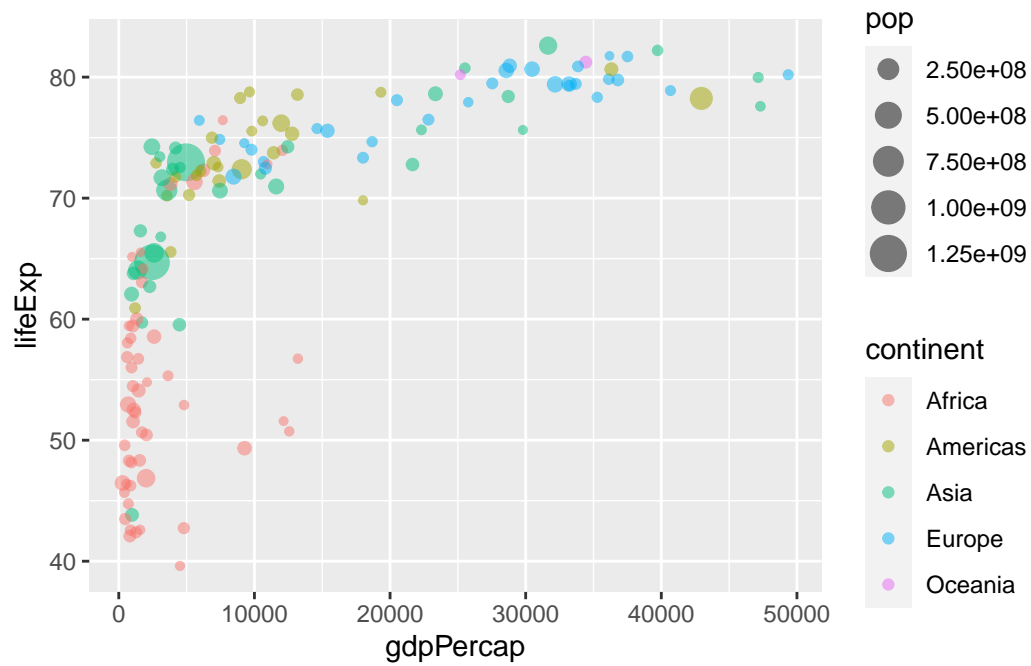

```
gapminder_2007 <- gapminder %>% filter(year==2007) #filtering for 2007 data

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



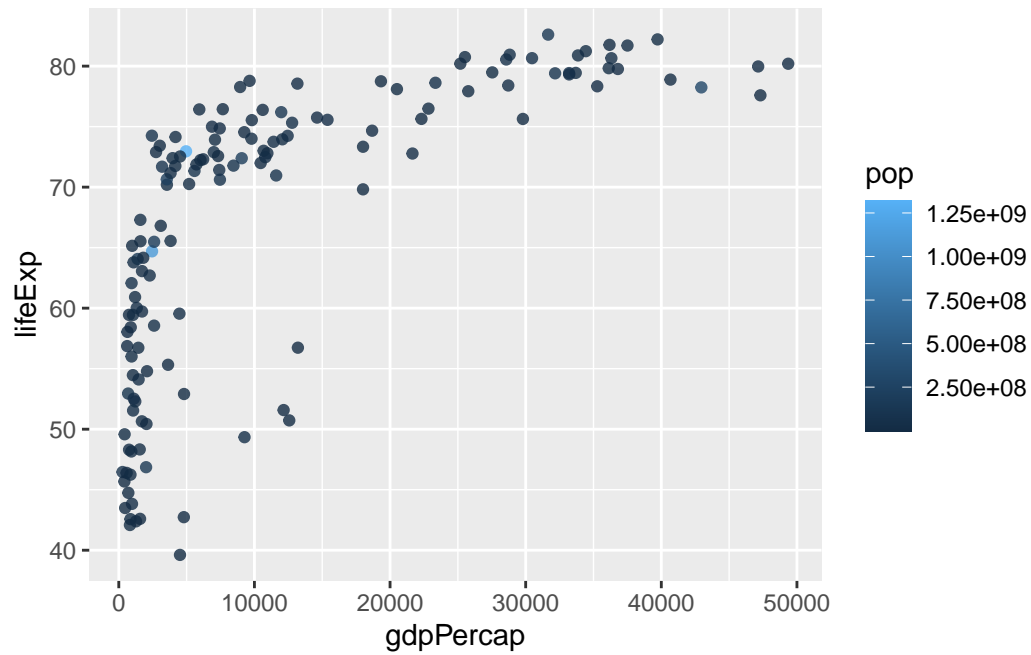
Some points on top of each other – make points more transparent. Add more variables

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



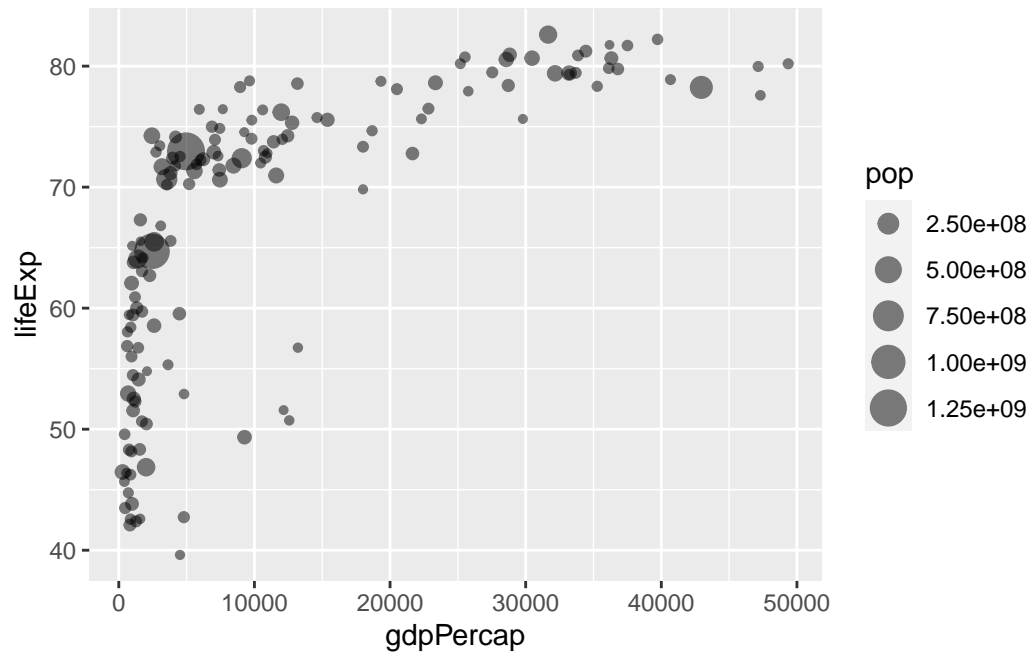
to color the points by population

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



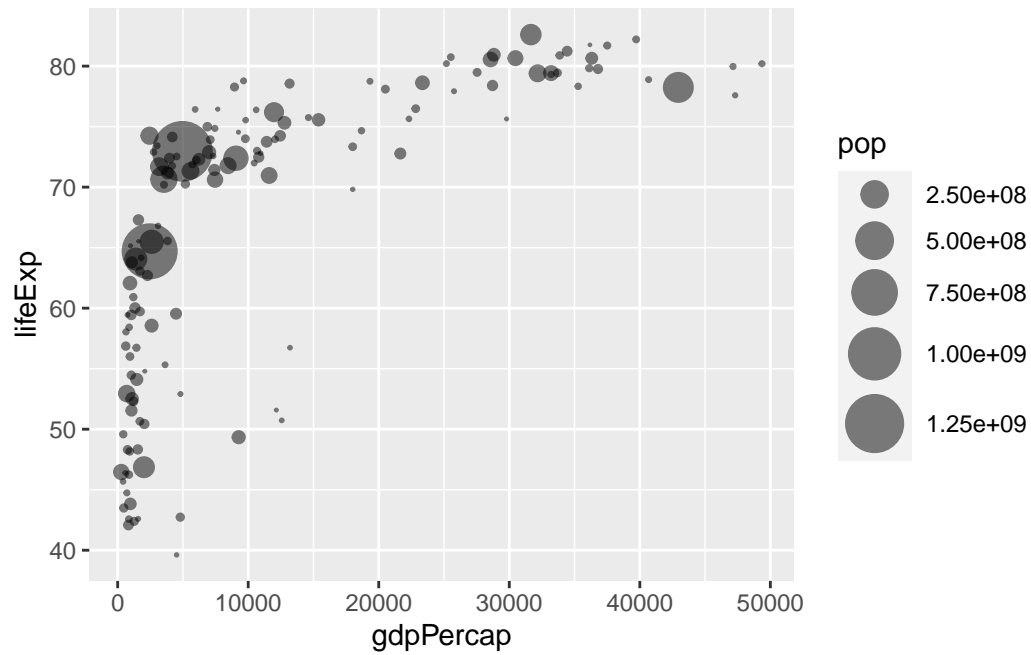
to make the points' size (small/big) based on population

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



to reflect the actual areas of the population so that the sizes of the points are proportional using `scale_size_area()`

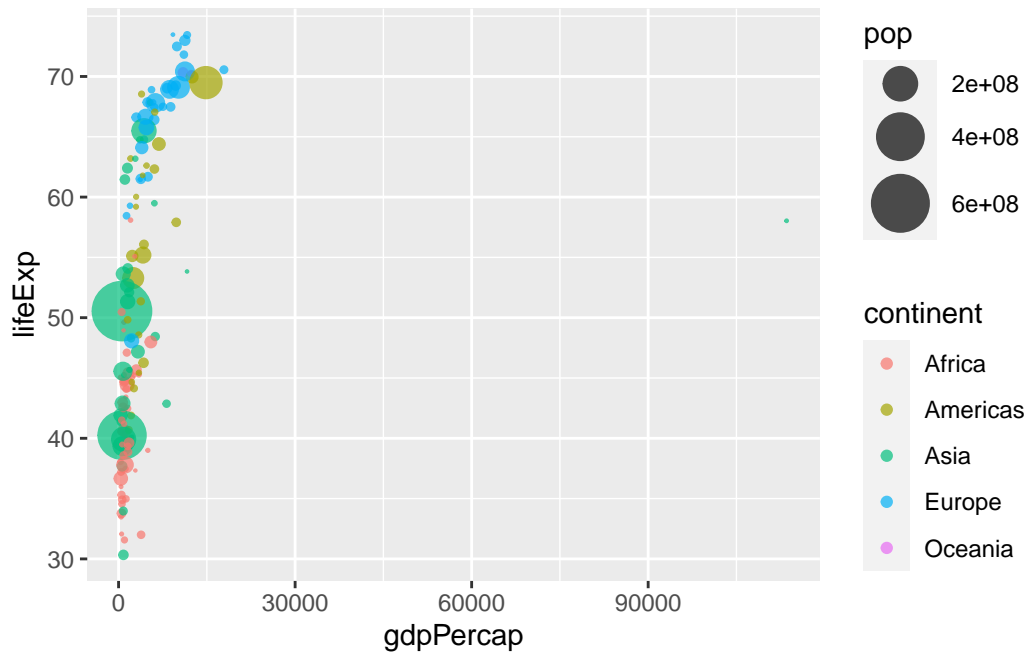
```
ggplot(gapminder_2007) +  
  geom_point(aes(x = gdpPerCap, y = lifeExp, size = pop), alpha = 0.5) +  
  scale_size_area(max_size = 10)
```



1957 Data

```
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)
```



Producing graphs side by side

```
gapminder_comp <- gapminder %>% filter(year==1957 | year==2007) #filter the date from 1957

ggplot(gapminder_comp) +
  geom_point(aes(x=gdpPerCap, y=lifeExp, color=continent, size=pop), alpha=0.7) +
  scale_size_area(max_size=10) +
  facet_wrap(~year) #compare the data from 1957 and 2007 side by side
```

