

Class 05: Data Visualization

Ani (A16647613)

Graphic Systems in R

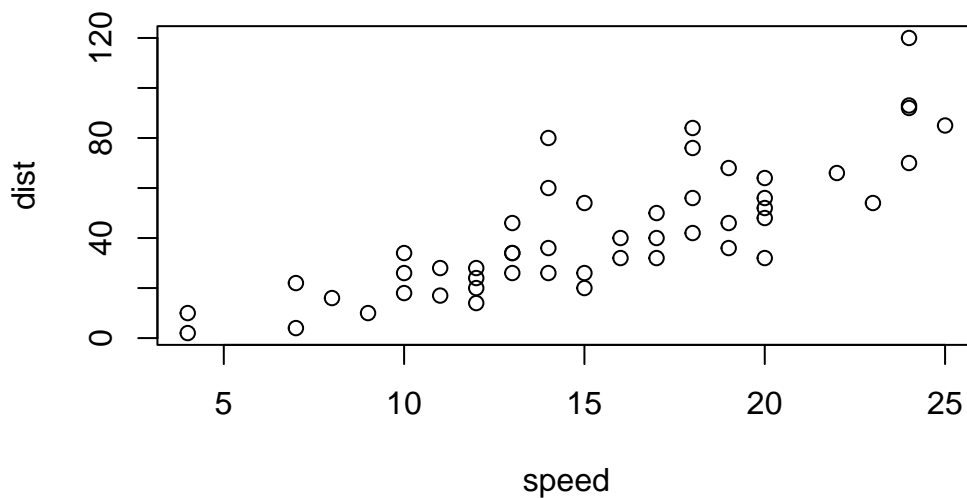
There are many graphics in R for making plots and figures

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()`.

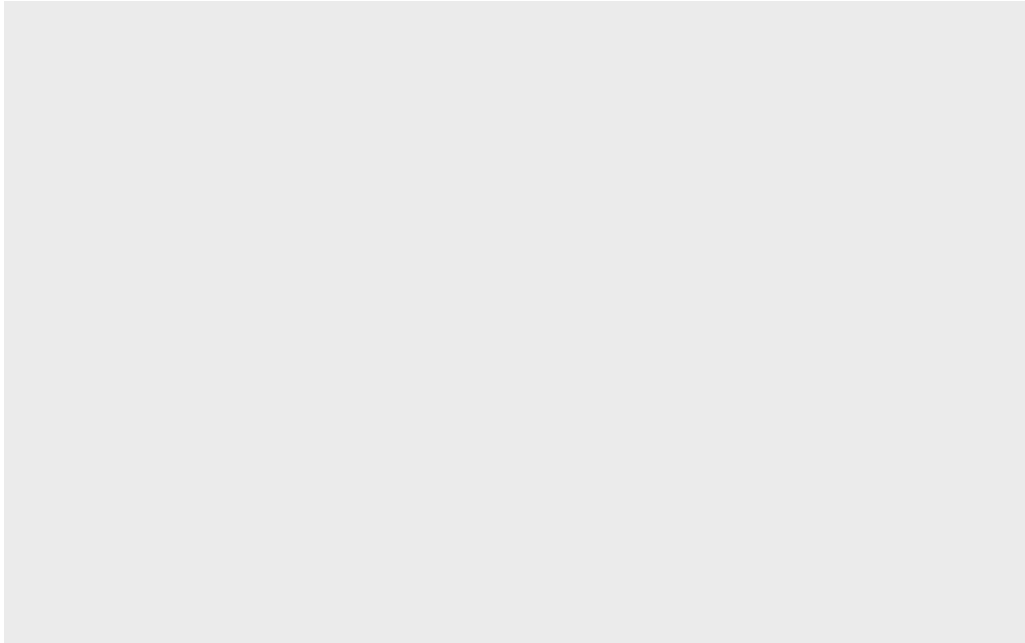
This is an add on package- ie. we need to install it. I installed it (like I install any package) with the `install.packages()` function.

```
plot(cars)
```



Before I can use the functions from a package I have to load 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) - aes (how the columns of data map to the plot aesthetics) - 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.

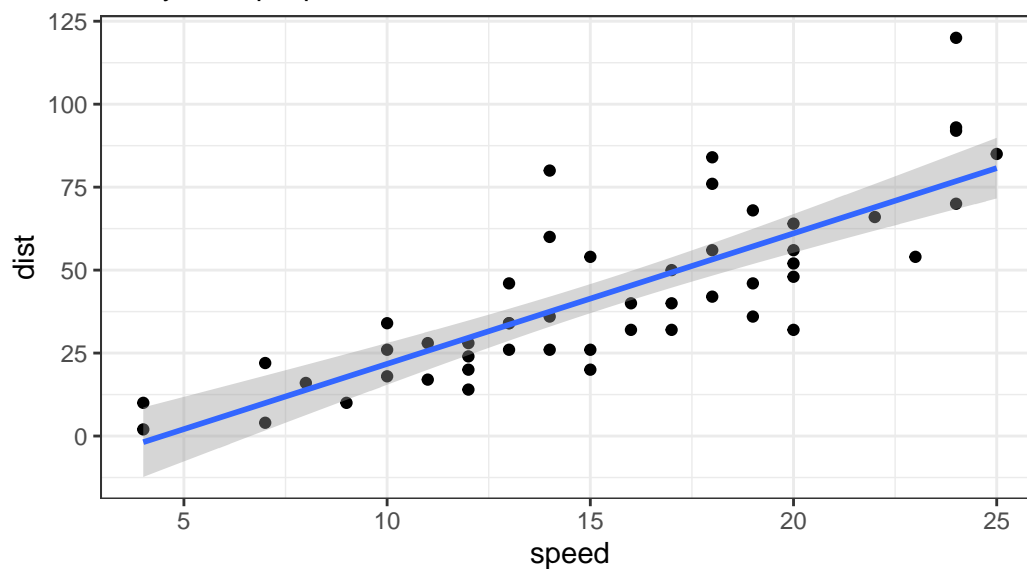
Add some more layers to our ggplot:

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

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

Stopping distance of old cars

A silly example plot



Section 6

Now, looking into dataset “genes”:

```
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

Table of column titled “State” to determine how many up-regulated genes there are:

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

down	unchanging	up
72	4997	127

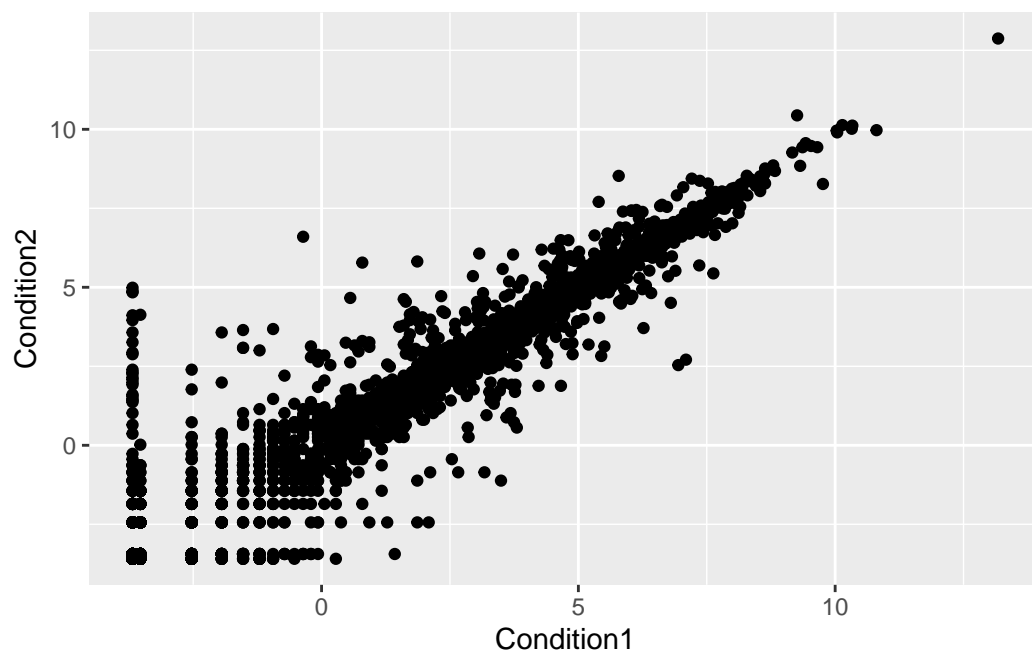
Total genes up-regulated in dataset

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

down	unchanging	up
1.39	96.17	2.44

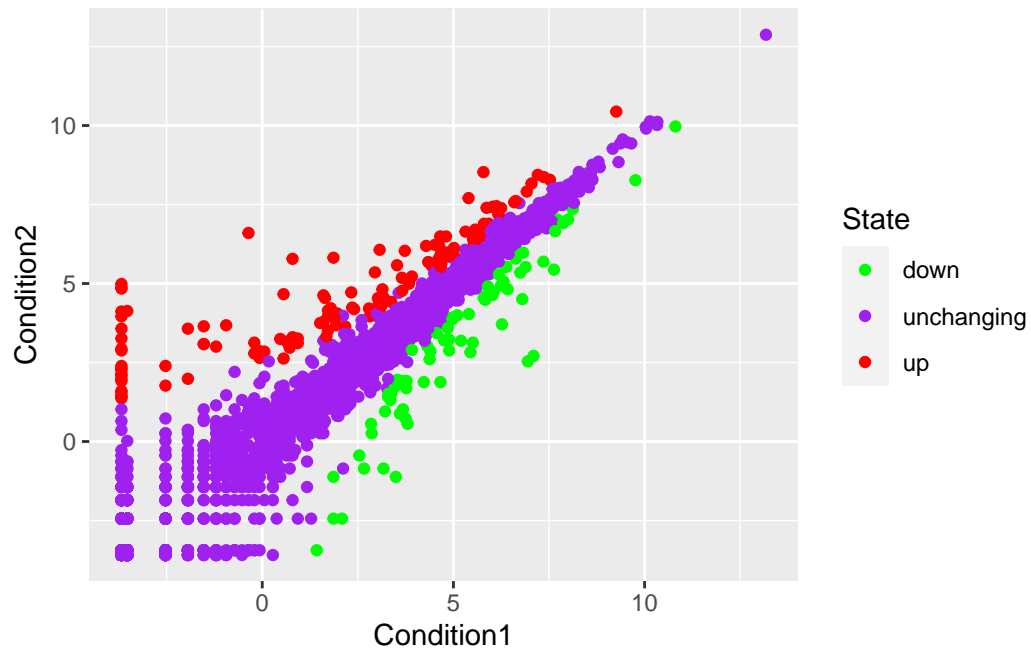
Plot of Genes in terms of Condition1 and Condition2

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



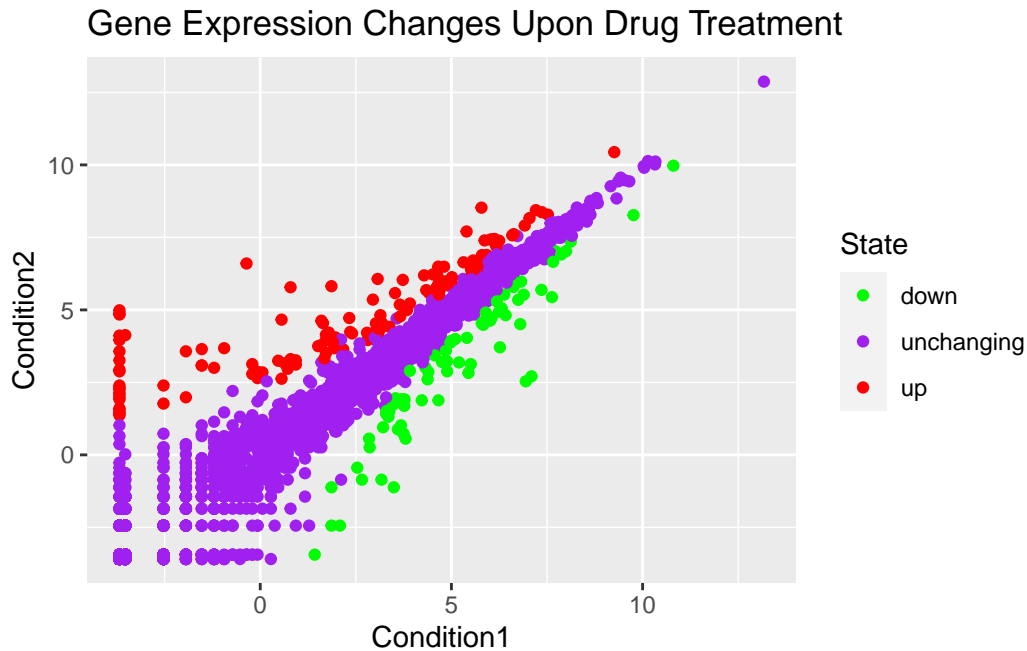
Plot of Genes in terms of Condition1 and Condition2 (Enhanced based on the conditions of the State Column)

```
ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point() +
  scale_color_manual(values= c("green", "purple", "red"))
```



Now time to add the proper title:

```
ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point() +
  scale_color_manual(values= c("green", "purple", "red")) +
  labs(title="Gene Expression Changes Upon Drug Treatment")
```



Section 7

```
url <-"https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.t
gapminder <- read.delim(url)
head(gapminder)
```

	country	continent	year	lifeExp	pop	gdpPercap
1	Afghanistan	Asia	1952	28.801	8425333	779.4453
2	Afghanistan	Asia	1957	30.332	9240934	820.8530
3	Afghanistan	Asia	1962	31.997	10267083	853.1007
4	Afghanistan	Asia	1967	34.020	11537966	836.1971
5	Afghanistan	Asia	1972	36.088	13079460	739.9811
6	Afghanistan	Asia	1977	38.438	14880372	786.1134

Now, we are going to consider the dataset values specifically from 2007, after downloading the “dplyr” binary package.

```
library(dplyr)
```

Attaching package: 'dplyr'

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

`filter`, `lag`

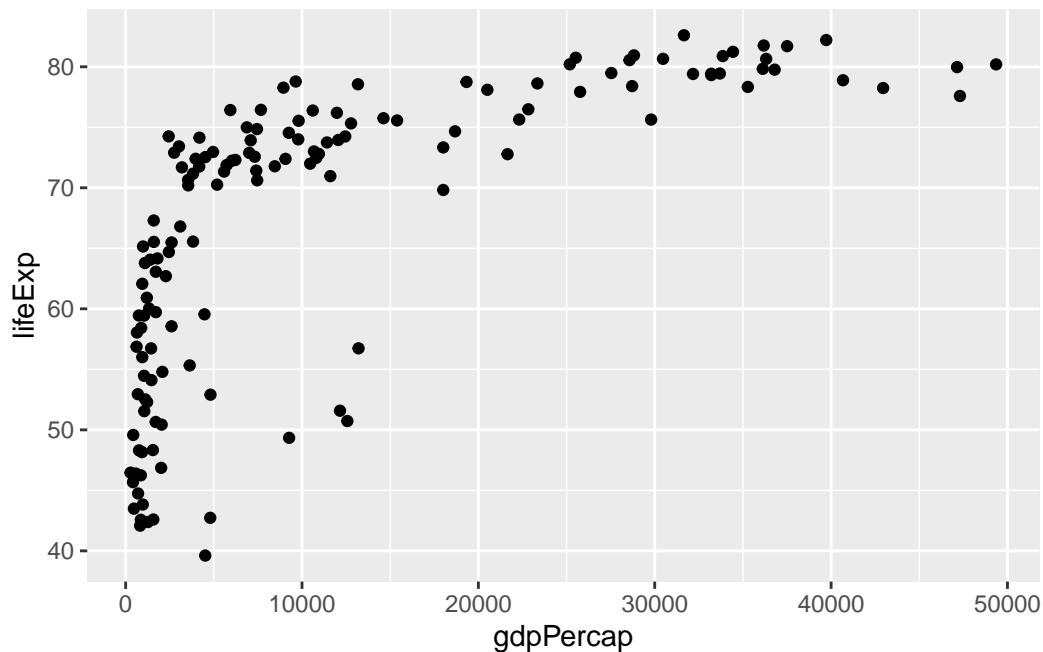
The following objects are masked from 'package:base':

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

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

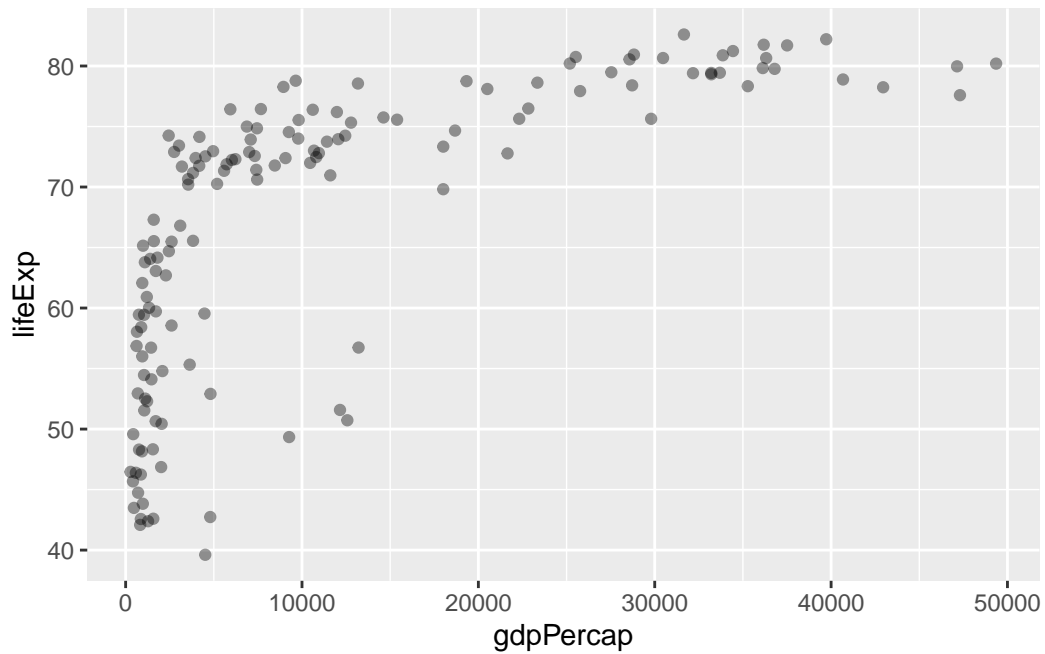
The first ggplot of gapminder_2007 in terms of the gdpPercap and Life Expectancy

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



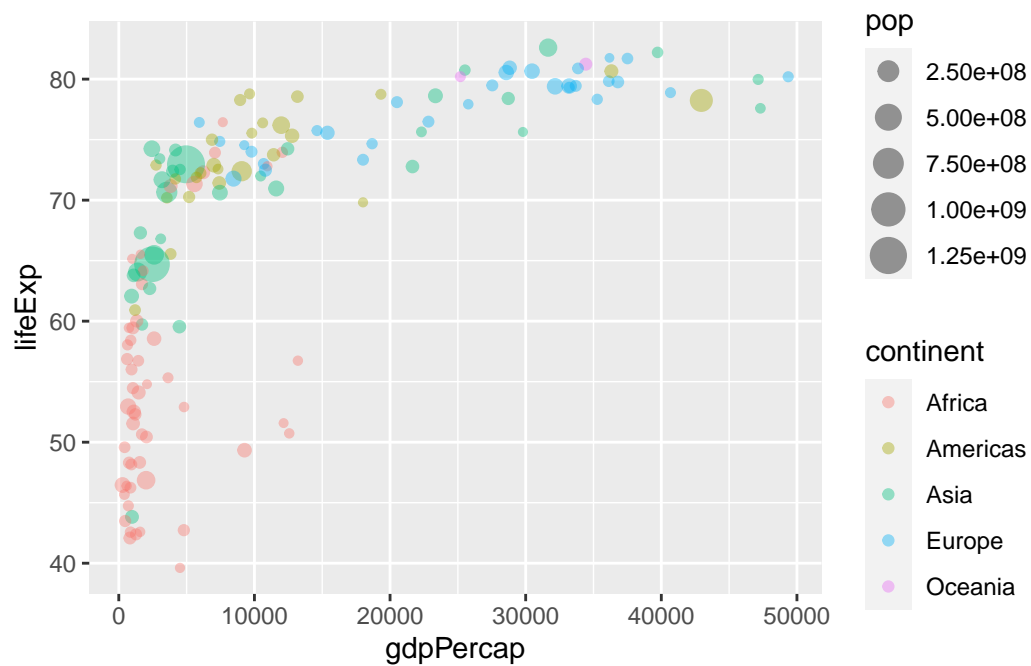
It will be easier to see the points more clearly with transparency, as added to the following sequence by including `alpha=0.4` to `geom_point()`.


```
ggplot(gapminder_2007)+aes(gdpPercap, y=lifeExp) +  
geom_point(alpha=0.4)
```



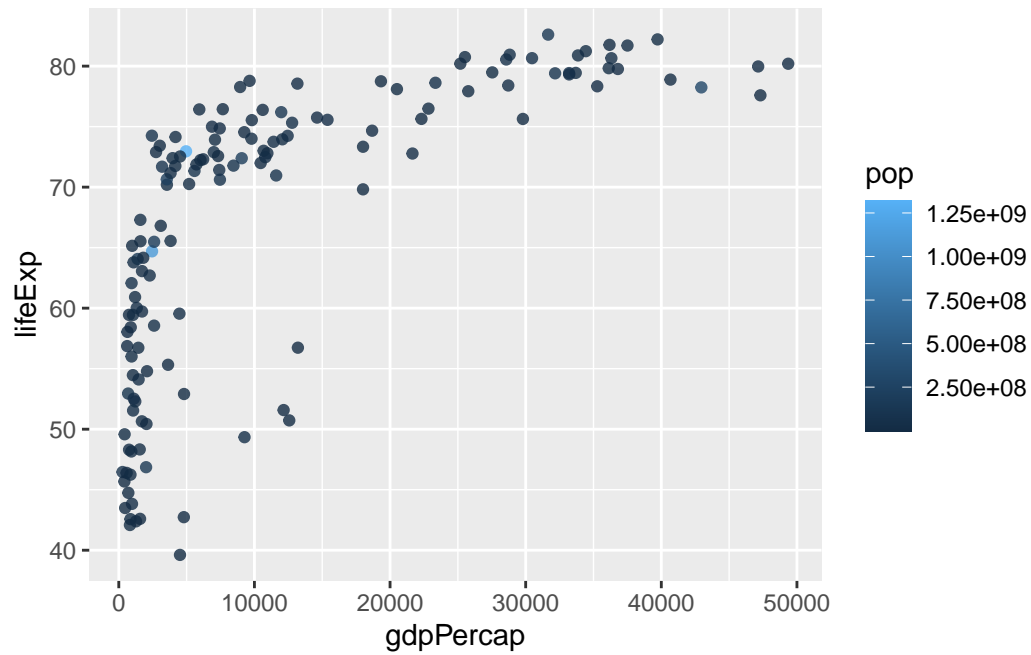
We are going to separate each point on the plot by size in the parameters of **population** and color in the parameters of **continent**.

```
ggplot(gapminder_2007) +  
aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +  
geom_point(alpha=0.4)
```



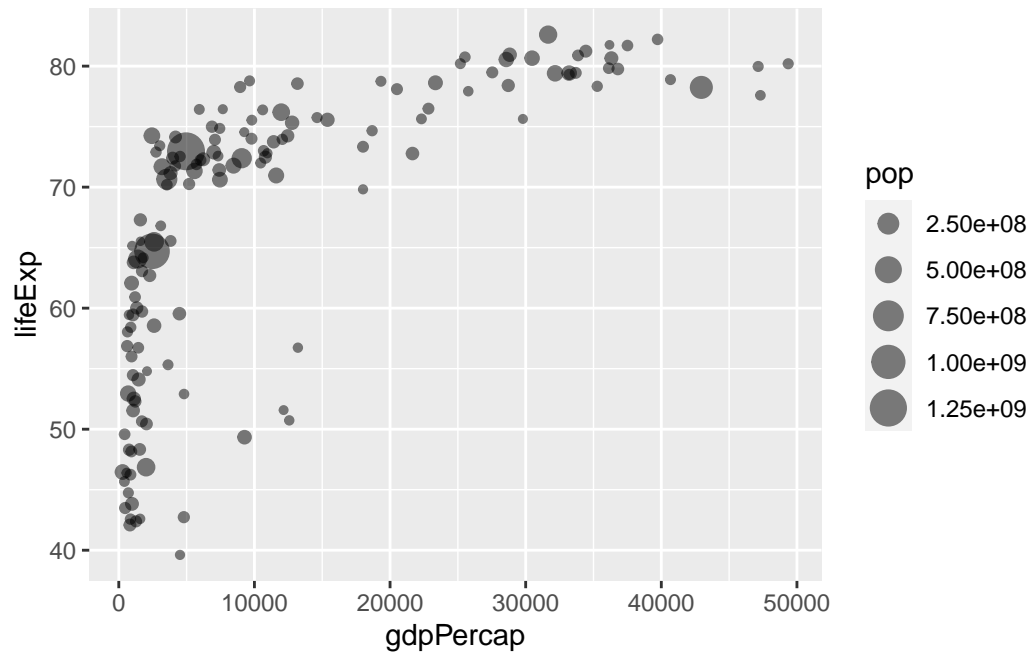
Points based on numeric variable population pop:

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

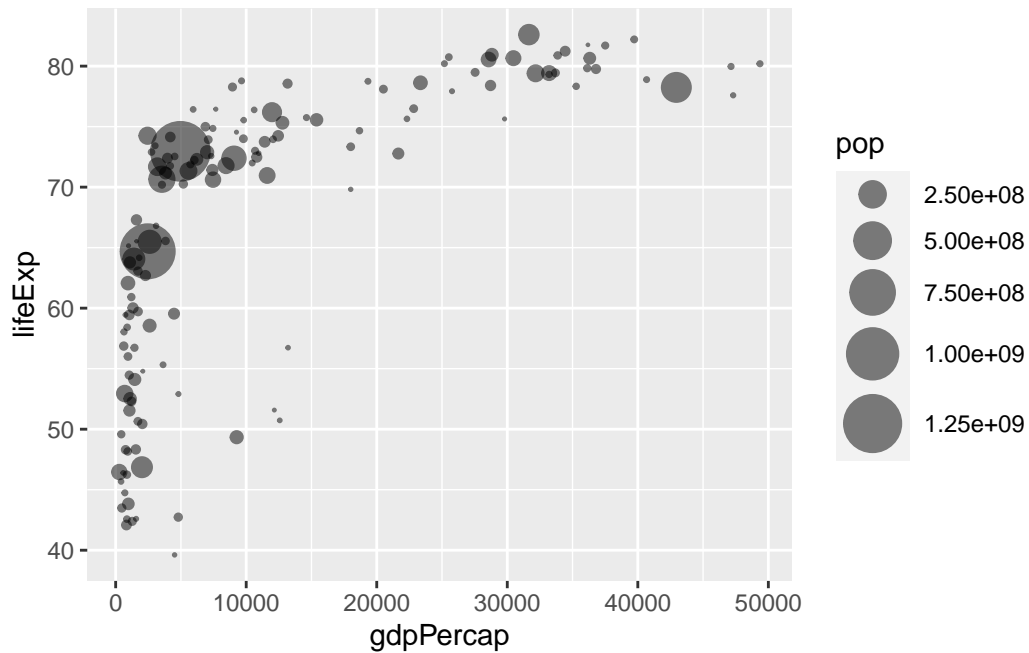


To adjust point size based on `size = pop` of each country and the rest of the adjustments:

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



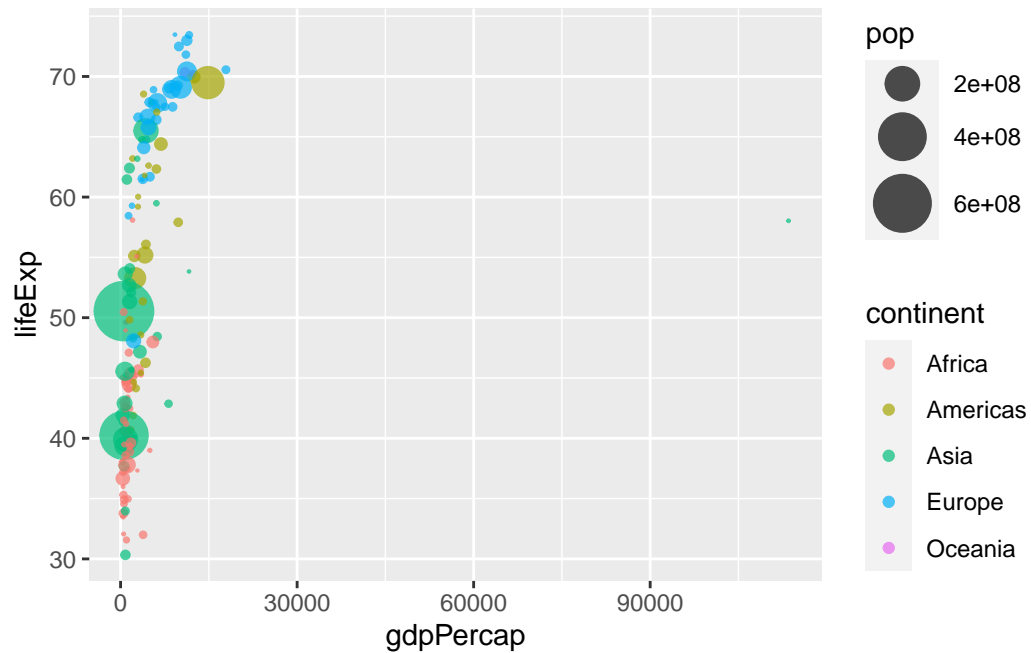
```
ggplot(gapminder_2007) +  
  geom_point(aes(x = gdpPercap, y = lifeExp, size = pop),  
    alpha=0.5) + scale_size_area(max_size = 10)
```



Now, to create a plot for the dataset values for the year 1957

```
gapminder_1957 <- gapminder %>% filter(year==1957)

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



Comparing the gdpPercap and lifeExp of populations in the continents in 1957 versus 2007:

```
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)
```

