Class05: Data Visulation w/ ggPlot

Scott MacLeod (PID: A16246401)

Graphics systems in R

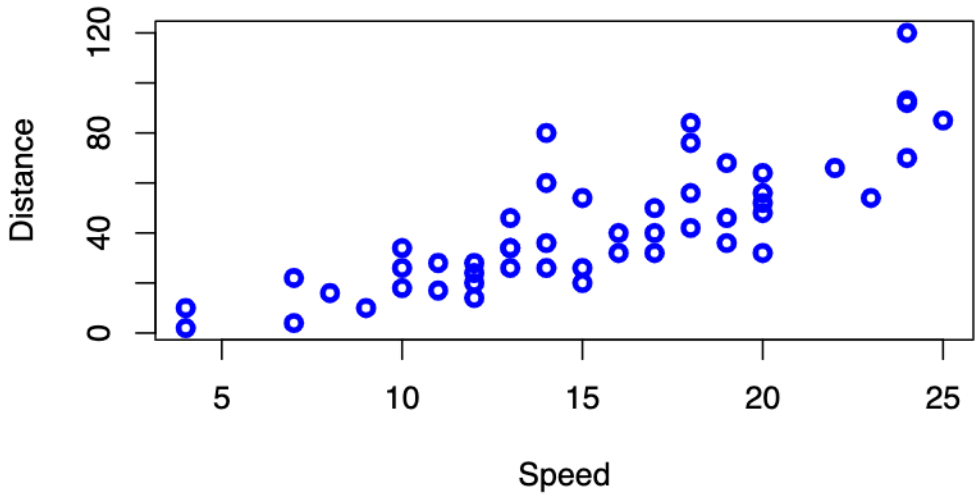
There are many graphics systems in R for making plots and figures.

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 package) with the `install.packages()` function.

```
plot(cars, xlab = "Speed", ylab = "Distance", col = "Blue", lwd = '3')
```



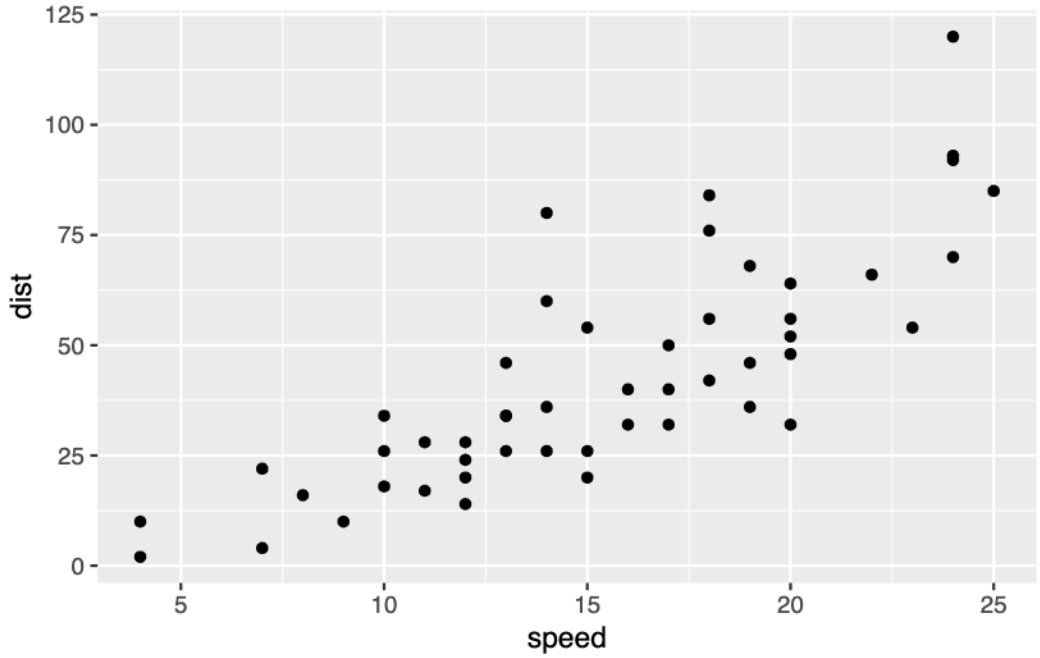
Before I can use the functions from a package I have to load up the package from my "library". We use the `library(ggplot2)` command to load it up.

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


Above is the initially loaded graph to get ggplot to work. This step is only required once.

Every ggplot is made up of at least **3** things: **1) data** (the numbers etc. that will go into your plot) **2) aes** (how the columns of data map to the plot aesthetics) **3) geoms** (how the lpot actually looks - i.e. - points, bars, lines, etc.)

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

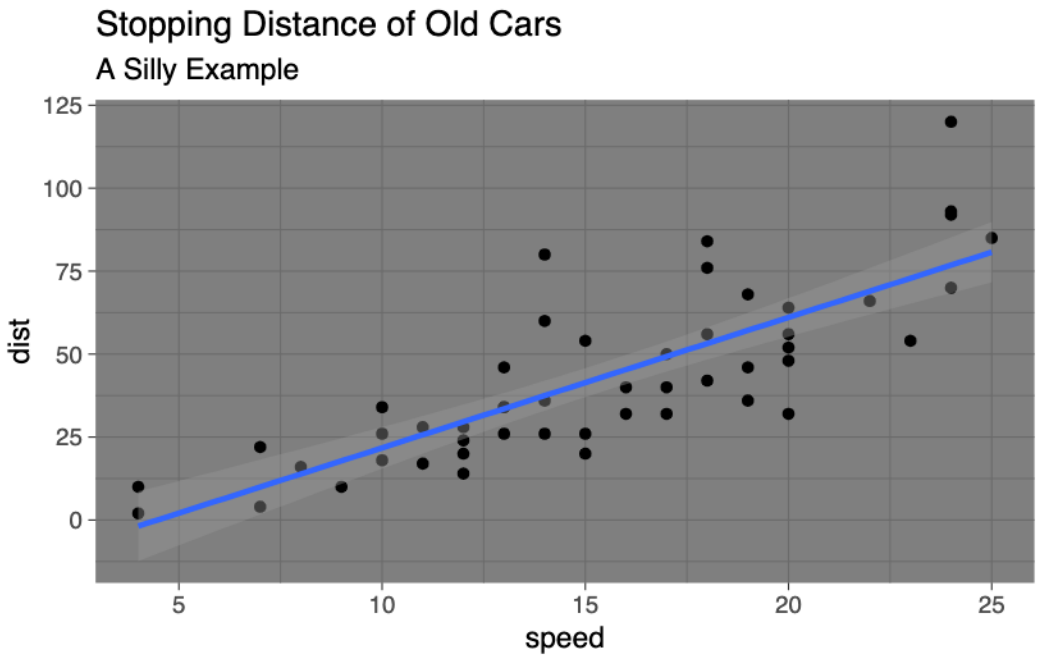


For simple plots, ggplot is more verbose - it takes more code - than base R plot.

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",
       xlab=("Speed"),
       ylab=("Distance")) +
  theme_dark()
```

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



Now we are going to work on a different project! First I need to download the data set!

```
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
```

Use the nrow() function to find out how many genes are in this dataset. What is your answer?

```
nrow(genes)
```

```
[1] 5196
```

Q. Use the colnames() function and the ncol() function on the genes data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find?

```
ncol(genes)
```

```
[1] 4
```

Q. Use the table() function on the State column of this data.frame to find out how many 'up' regulated genes there are. What is your answer?

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

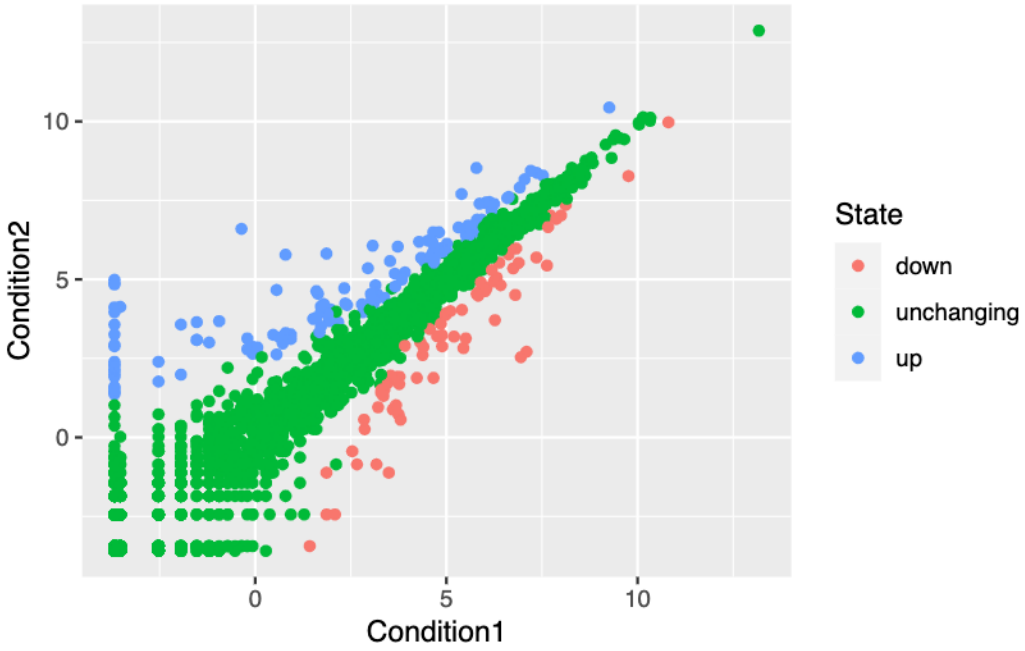
```
      down unchanging         up
        72       4997        127
```

Q. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

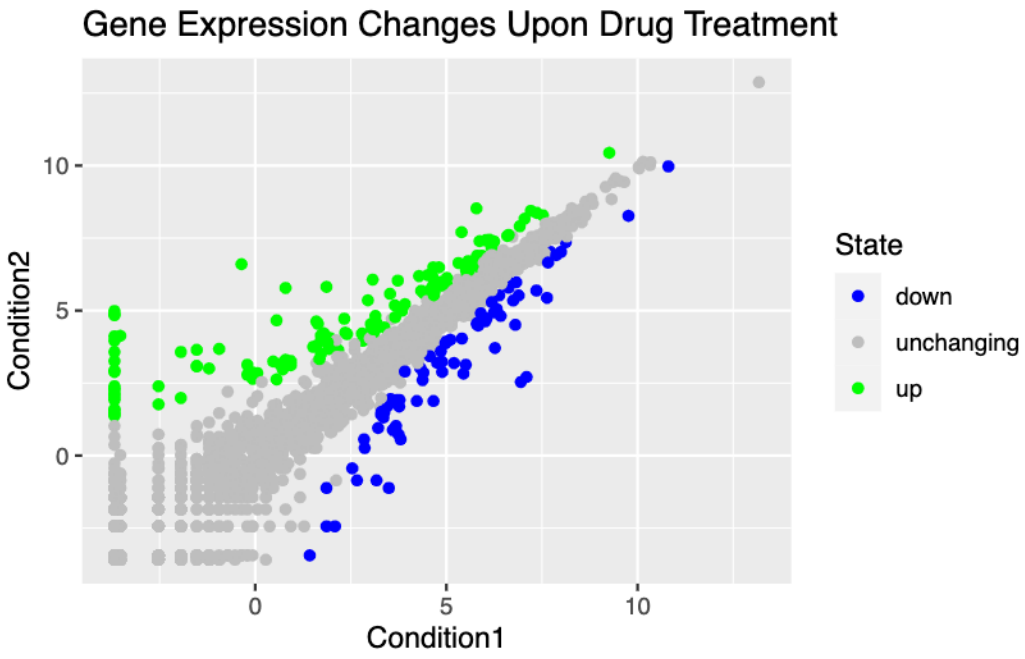
```
round(table(genes$State) / nrow(genes) * 100, 2)
```

```
      down unchanging         up
      1.39      96.17       2.44
```

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




```
p + scale_colour_manual( values=c("blue","grey","green") )+ labs(title="Gene Expression Ch
```



Step 7 Going Further!

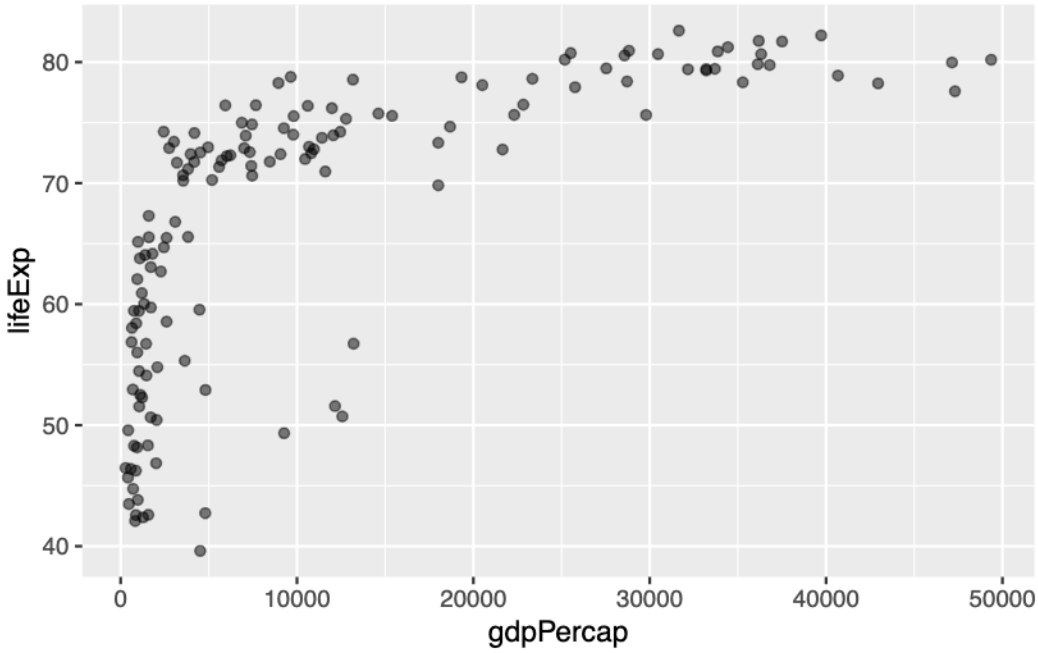
```
url2 <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder
library(dplyr)
```

```
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(url2)
gapminder_2007 <- gapminder %>% filter(year==2007)
```

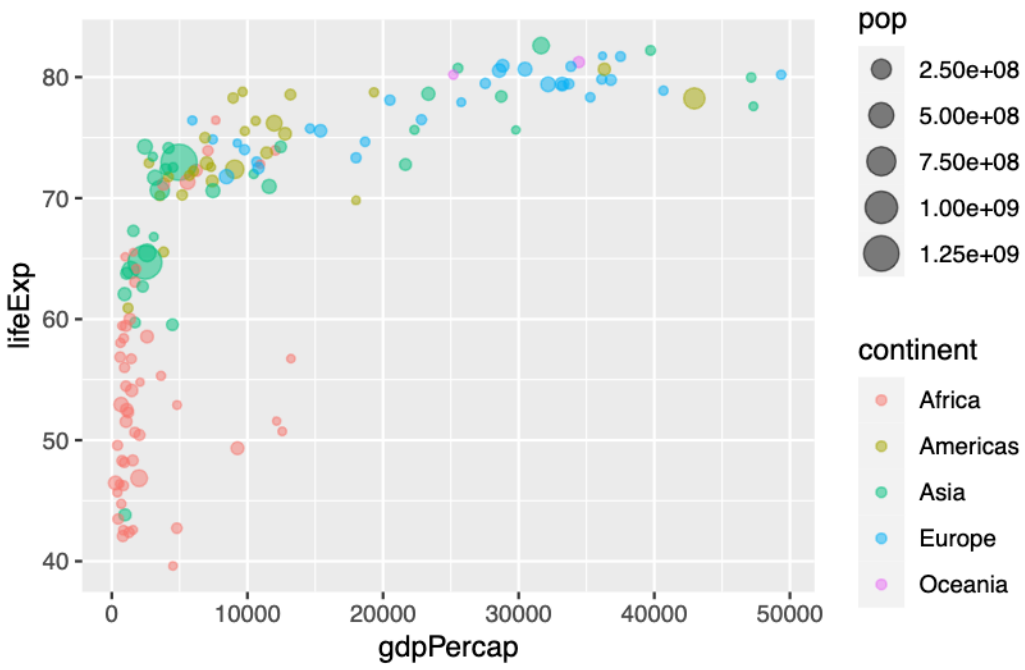
Now we are going to make a scatterplot.

```
pl <- ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
pl
```



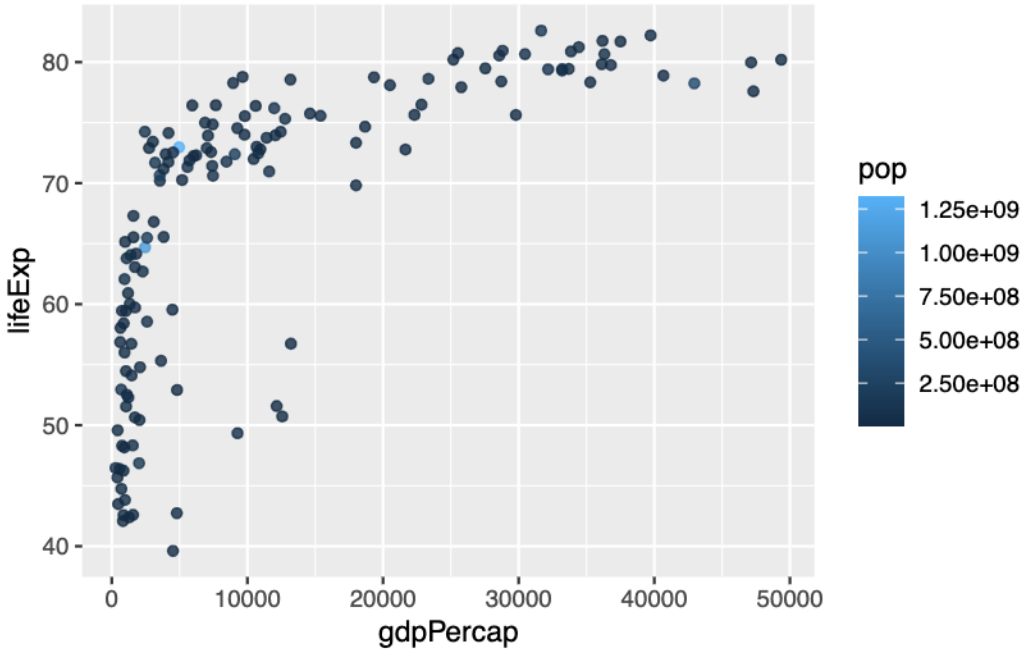
Now we are adjusting the size based on population and the color based on the continent:

```
pl + aes(color=continent, size=pop)
```



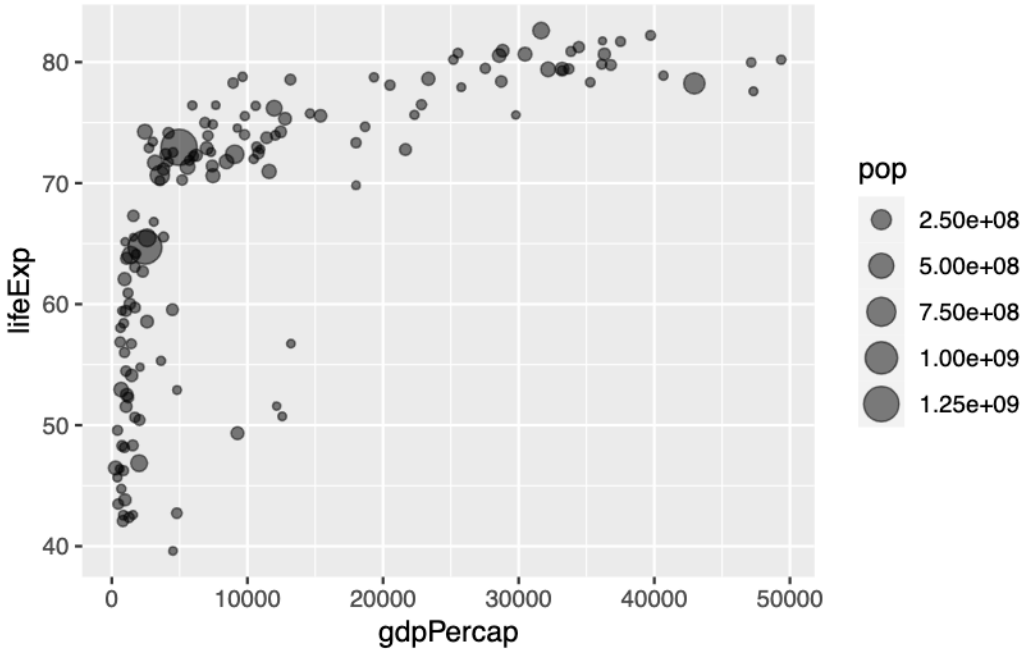
Now we are adjusting the color of the dot based on population:

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



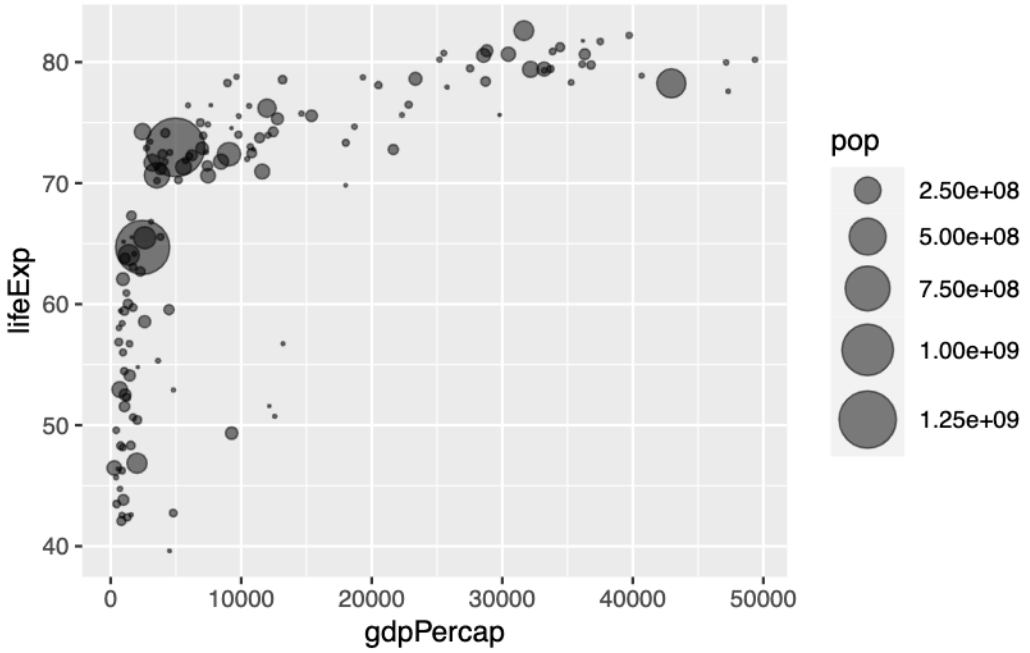
Now we are adjusting the size of the point based on the size of the population:

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



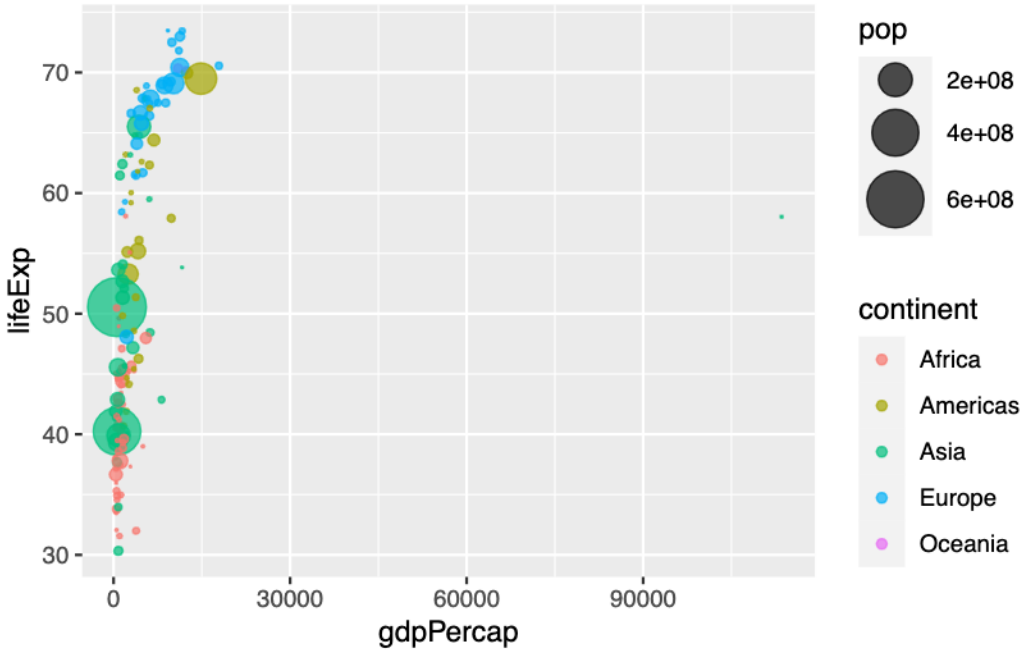
However, the scaling is all wrong, we are going to add a code to change the scale:

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



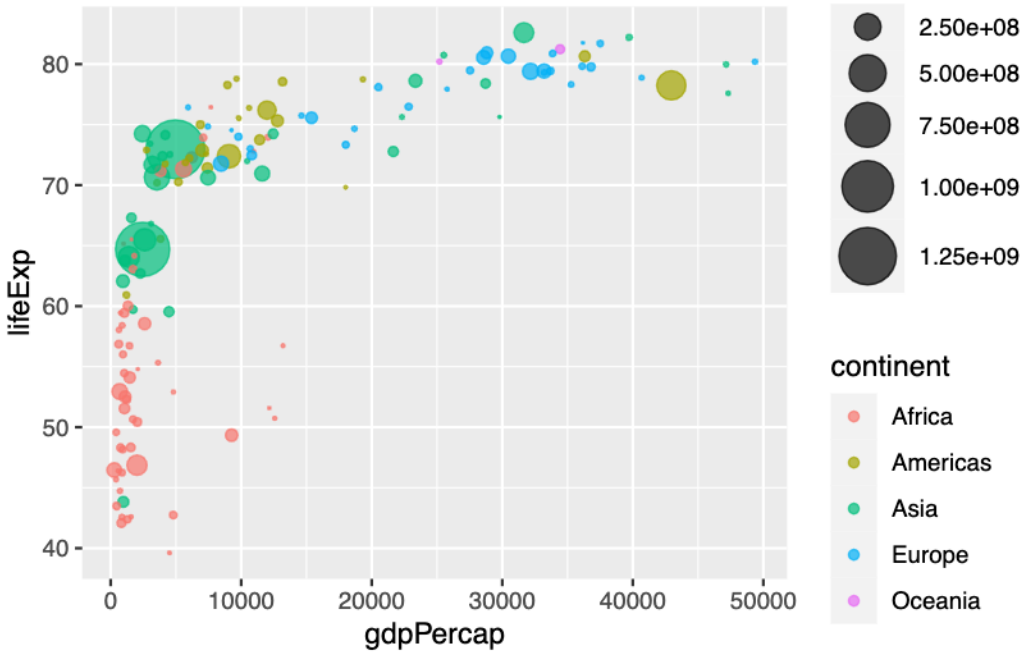
Now we are going to do the same but for the yaer 1957! We are going to filter for the year 1957 and 2007. Then we are going to use the same codes in order to compare the two different years:

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



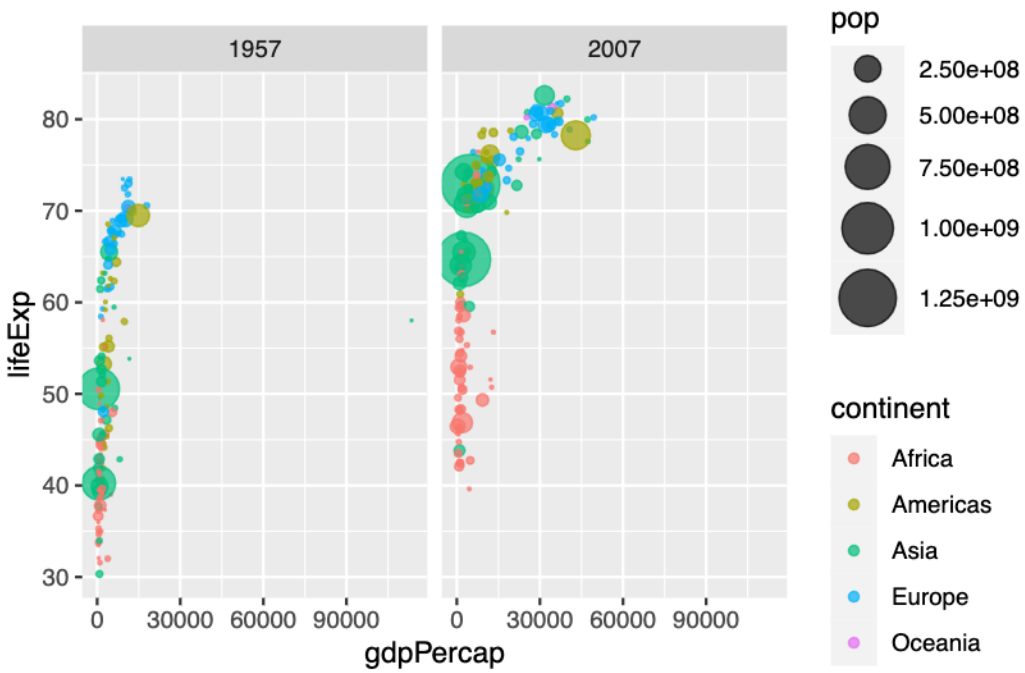
Above is the plot for 1957. We are going to add the plot for 2007 and then compare:

```
ggplot(gapminder_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp,
                 size = pop, color=continent), alpha=0.7) +
  scale_size_area(max_size = 10)
```



Now we are going to compare the two different years.

```
gapminder_both <- gapminder %>% filter(year==1957 | year==2007)

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


1957
2007
lifeExp
80
70
60
50
40
30
0
30000
60000
90000
gdpPercap
pop
2.50e+08
5.00e+08
7.50e+08
1.00e+09
1.25e+09
continent
Africa
Americas
Asia
Europe
Oceania