# Class 5: Data Viz with ggplot

Bayah Essayem (A17303992)

## Intro to ggplot

There are many graphics system in R (ways to make plots and figures). These include "base" R plots. Today we will focus mostly on **ggplot2** package.

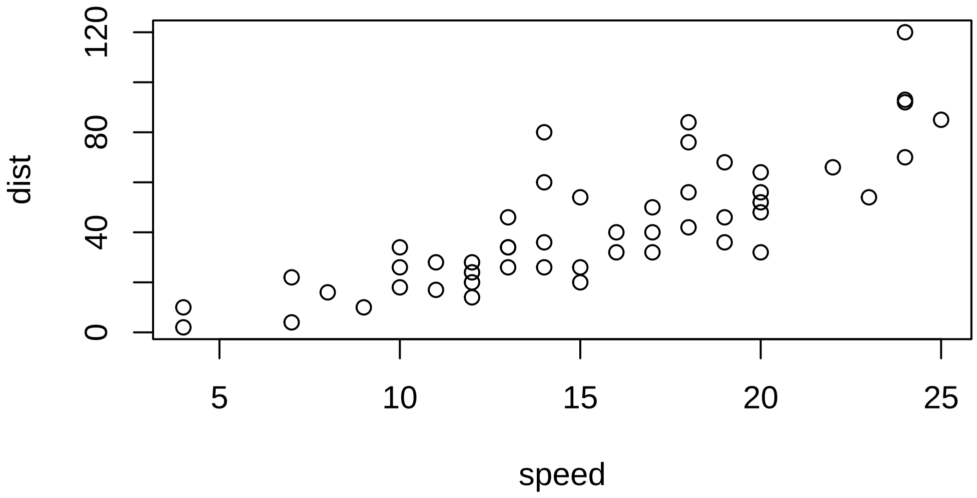
Let's start with a plot of a simple built in dataset called 'cars'.

```
cars
```

```
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
11    11   28
12    12   14
13    12   20
14    12   24
15    12   28
16    13   26
17    13   34
18    13   34
19    13   46
20    14   26
21    14   36
22    14   60
```

| | | |
|---|---|---|
| 23 | 14 | 80 |
| 24 | 15 | 20 |
| 25 | 15 | 26 |
| 26 | 15 | 54 |
| 27 | 16 | 32 |
| 28 | 16 | 40 |
| 29 | 17 | 32 |
| 30 | 17 | 40 |
| 31 | 17 | 50 |
| 32 | 18 | 42 |
| 33 | 18 | 56 |
| 34 | 18 | 76 |
| 35 | 18 | 84 |
| 36 | 19 | 36 |
| 37 | 19 | 46 |
| 38 | 19 | 68 |
| 39 | 20 | 32 |
| 40 | 20 | 48 |
| 41 | 20 | 52 |
| 42 | 20 | 56 |
| 43 | 20 | 64 |
| 44 | 22 | 66 |
| 45 | 23 | 54 |
| 46 | 24 | 70 |
| 47 | 24 | 92 |
| 48 | 24 | 93 |
| 49 | 24 | 120 |
| 50 | 25 | 85 |

```r
plot(cars)
```



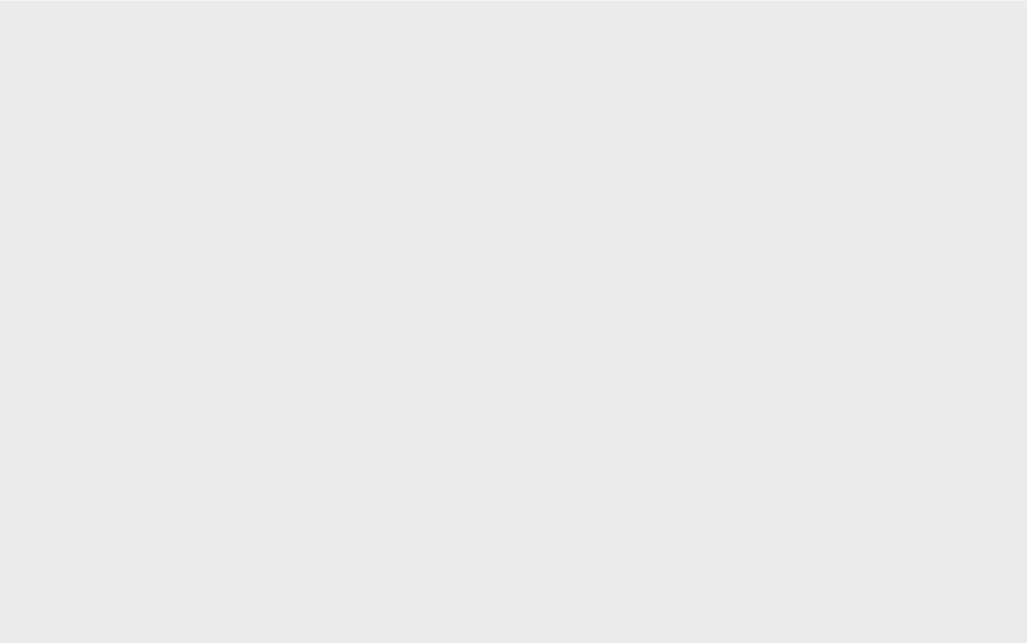
Let’s see how we can make this figure using **ggplot**. First I need to install this package on my computer. To install any R package I use the function ‘install.packages()’

> I will run ’install.packages(“ggplot2”) in my R console not this quarto document.

```
#ggplot(cars)
```

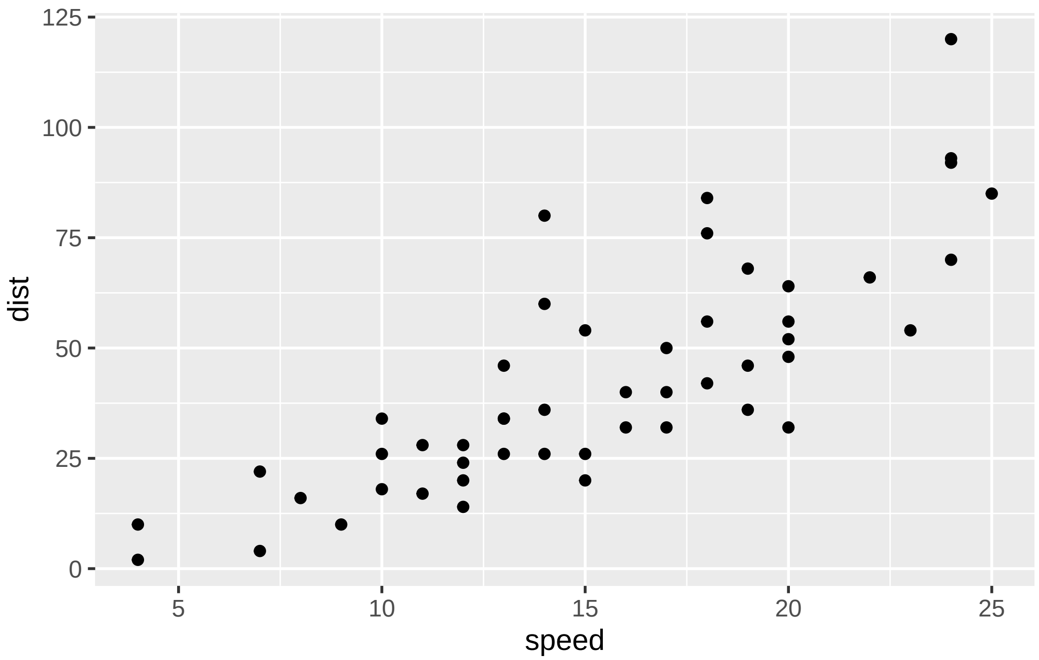
This creates an error because we don’t have the R package installed yet. Now that it is installed, you still have to “call” the add on package or load it up from “library()” with the ‘library(ggplot2)’ so it can work (this applies to any coding language).

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



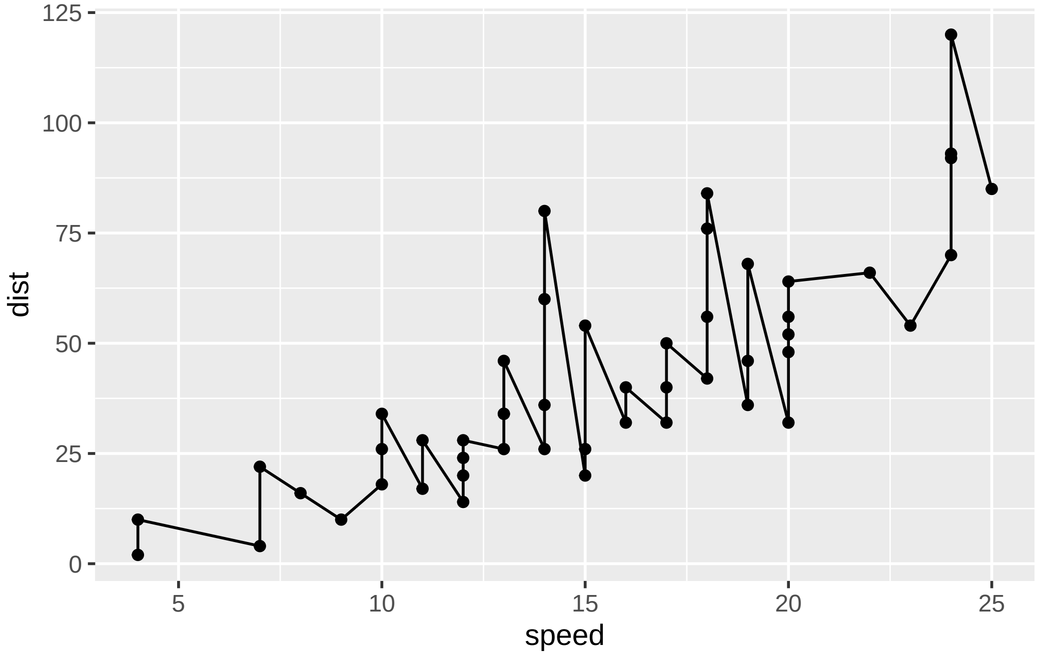
All ggplot figures have at least 3 things (called layers). These include: **data** (the input dataset I want to plot from), **aes** (the aesthetic mapping of the data to my plot), **geoms** (the geom_point(), geom_line(), etc. that I want to draw).

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



Let's add a line to show the relationship here:

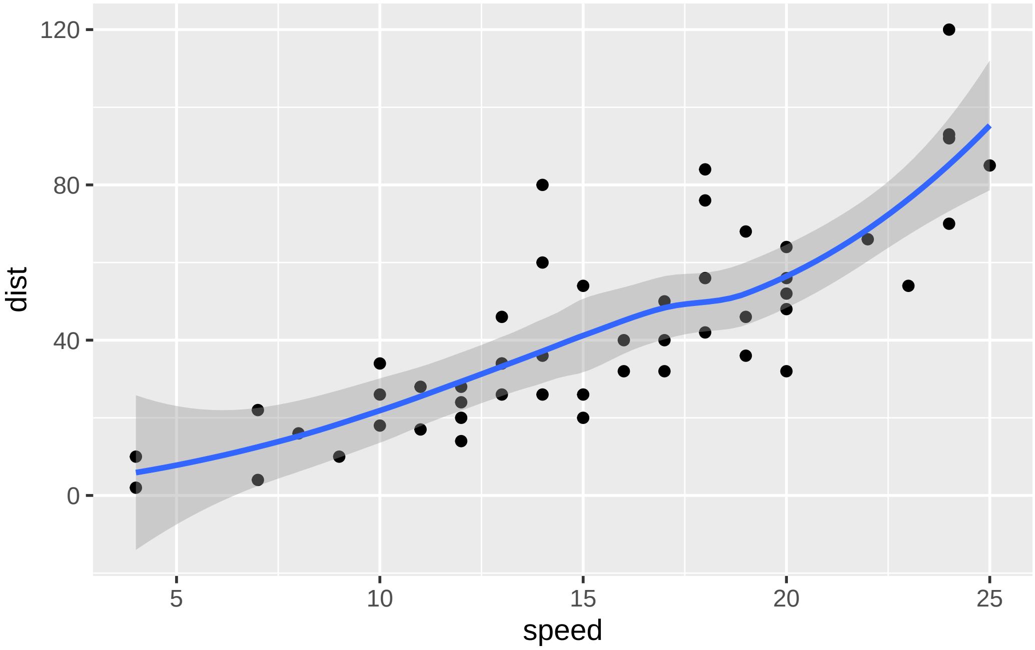
```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_line()
```



Let’s add a shadow thingy, that is referred to as smooth:

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

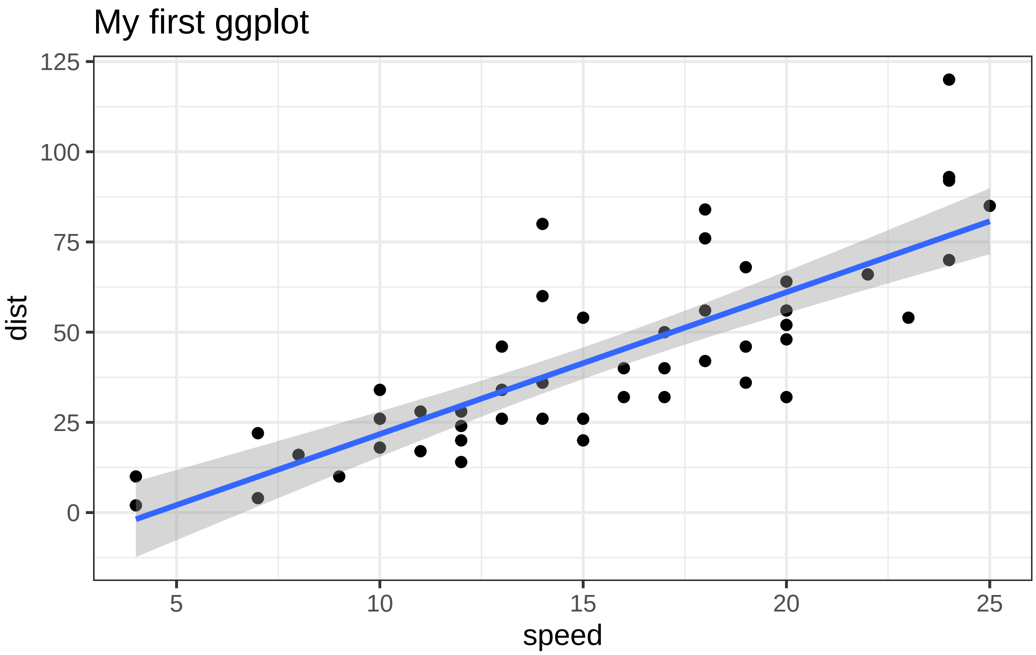
```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Make it more linear:

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm") +
  theme_bw() +
  labs(title="My first ggplot")
```

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



Q1 Which geometric layer should be used to create scatter plots in ggplot2?

geom_point()

### Gene expression figure

The code to read the dataset

```
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
```

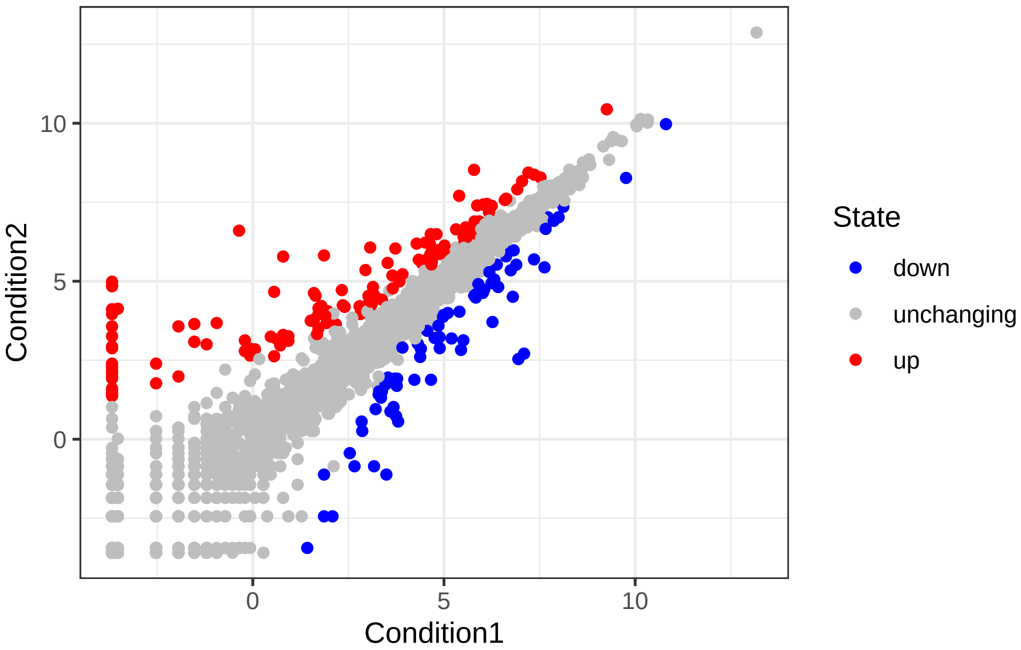
How many genes are in this dataset?

```
nrow(genes)
```

```
[1] 5196
```

A first plot of this dataset

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



```
  labs(title="Gene Expression Changes Upon Drug Treatment",
         x="Control Expression ",
         y="Drug Presence Expression")
```

```
$x
[1] "Control Expression "

$y
[1] "Drug Presence Expression"

$title
[1] "Gene Expresion Changes Upon Drug Treatment"

attr(,"class")
[1] "labels"
```

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

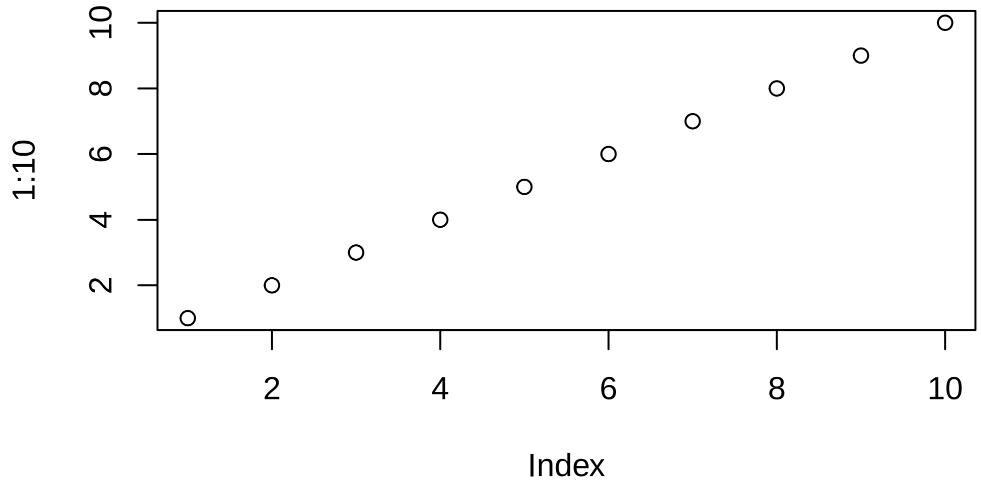
```r
round( table(genes$State)/nrow(genes),4)
```

```
      down unchanging         up 
    0.0139     0.9617     0.0244 
```

```r
n.tot <- nrow(genes)
vals <- table(genes$State)
vals.percent <- vals/n.tot *100
round(vals.percent,2)
```

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

```r
plot(1:10)
```



Here library that I already installed in my console is being called using “library”, I repeated this for both gapminder package and dplyr

```
library(gapminder)
```

# install.packages(“dplyr”)

# un-comment to install if needed

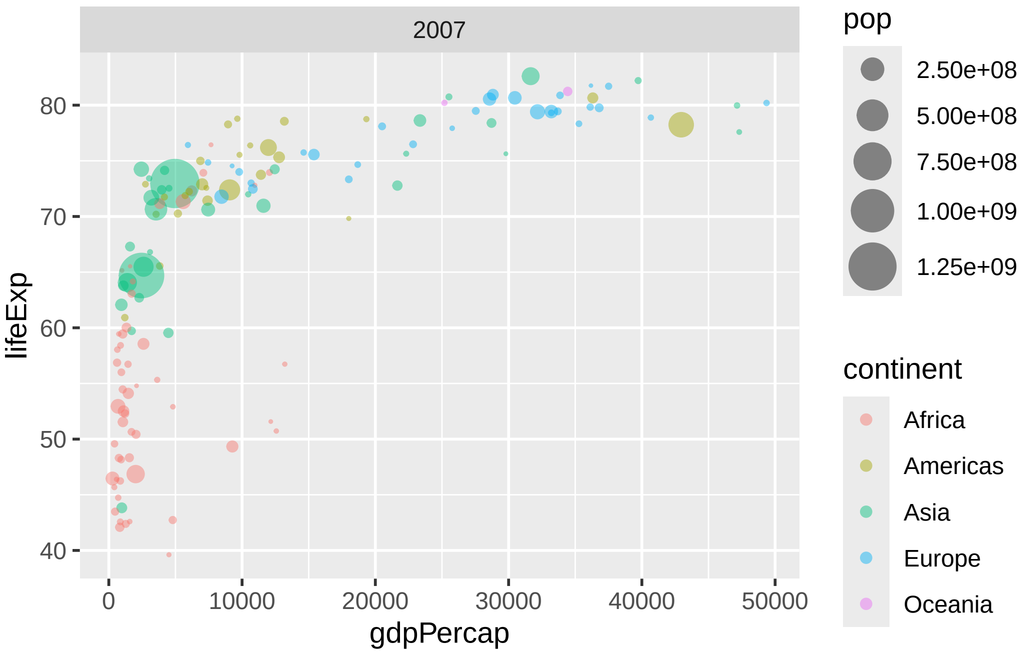
```
library(dplyr)
```

```

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
```

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

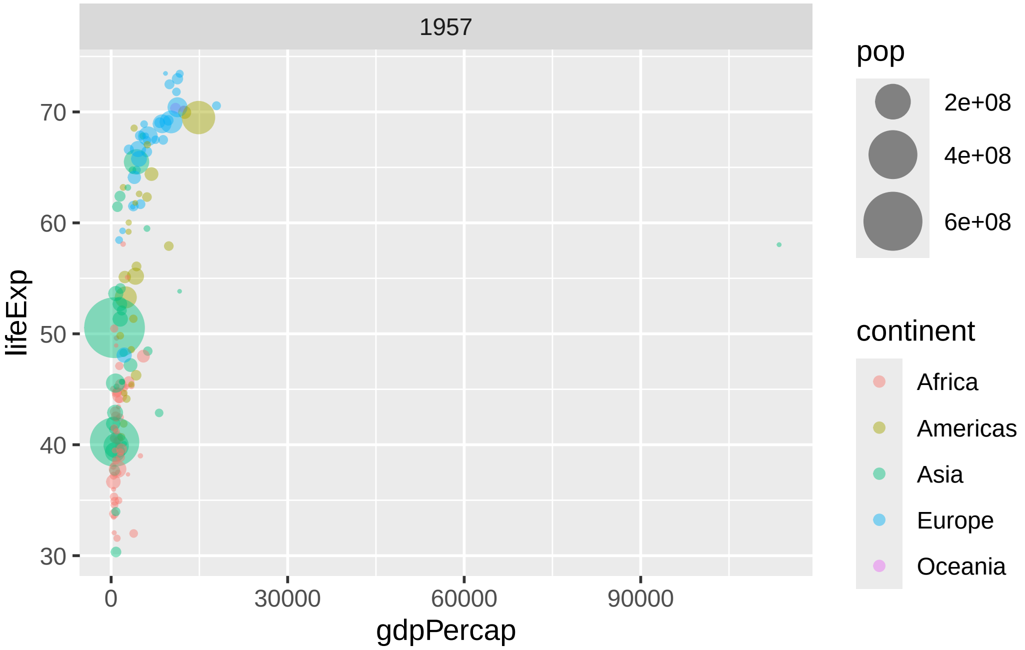
This is for creating a graph representing population by continent in 2007

```r
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.45) +
  scale_size_area(max_size = 8) +
facet_wrap(~year)
```



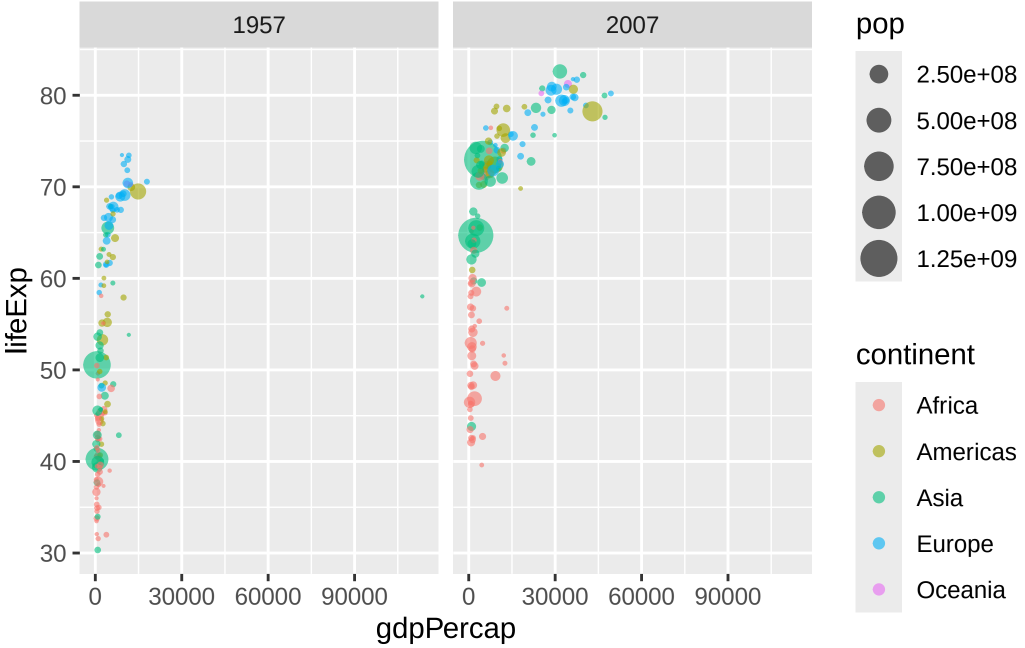
Here I am adding another graph this time representing the 1957 population

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

ggplot(gapminder_1957) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.45) +
  scale_size_area(max_size = 10) +
facet_wrap(~year)
```



Placing them side by side

```
gapminder_1957 <- gapminder %>% filter(year==1957 | year==2007)
ggplot(gapminder_1957) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.6) +
  scale_size_area(max_size = 6) +
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