Class 5: Data Vis with ggplot

Dylan Mullaney

Intro to ggplot

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

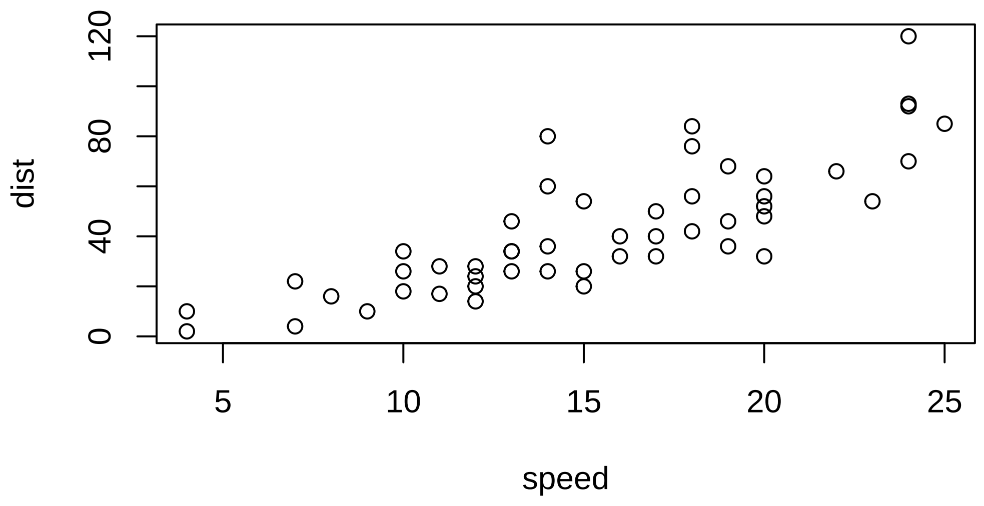
Let's start with a plot of a simple in-built dataset called 'cars'.

```
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
```

```
plot(cars)
```

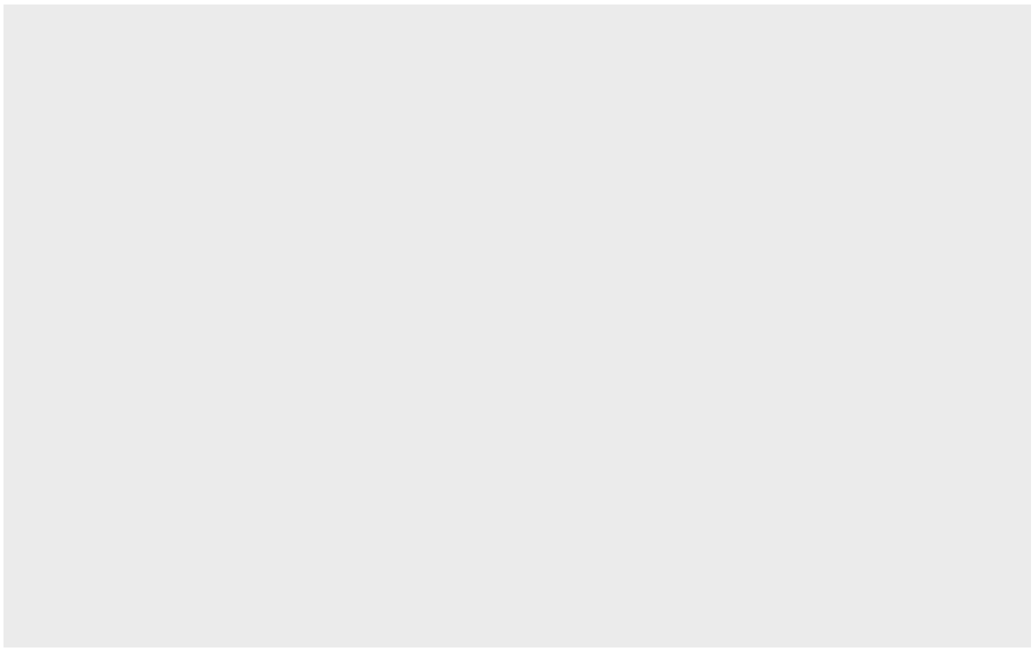


Let's see how we can make this figure using **ggplot**. First I need to install this package on my computer. To install any R package I use thfunction `install.packages()`

> I will run `install.packages()` im my R console, not this Quarto so I don't have to reinstall every time that I render.

Before I can use any functions from add-on packages I need to load the package from my "library()" with the "library(ggplot2)" call.

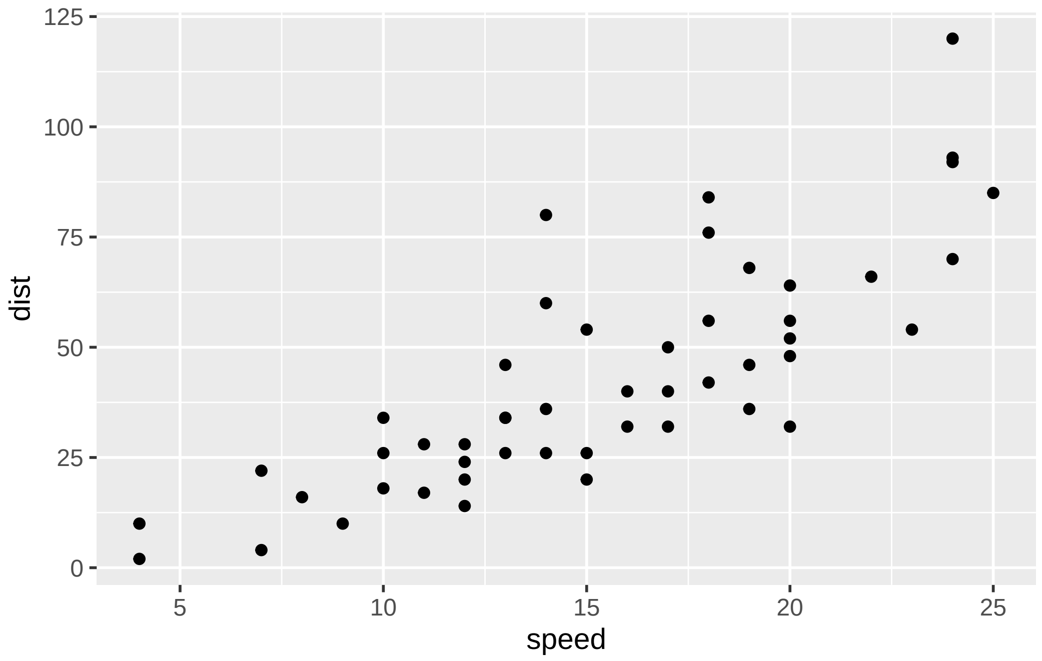
```r
library(ggplot2)
ggplot(cars)
```



All ggplot figures have at least 3 things (called layers). These include:

- **data** (the imput 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 from)

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

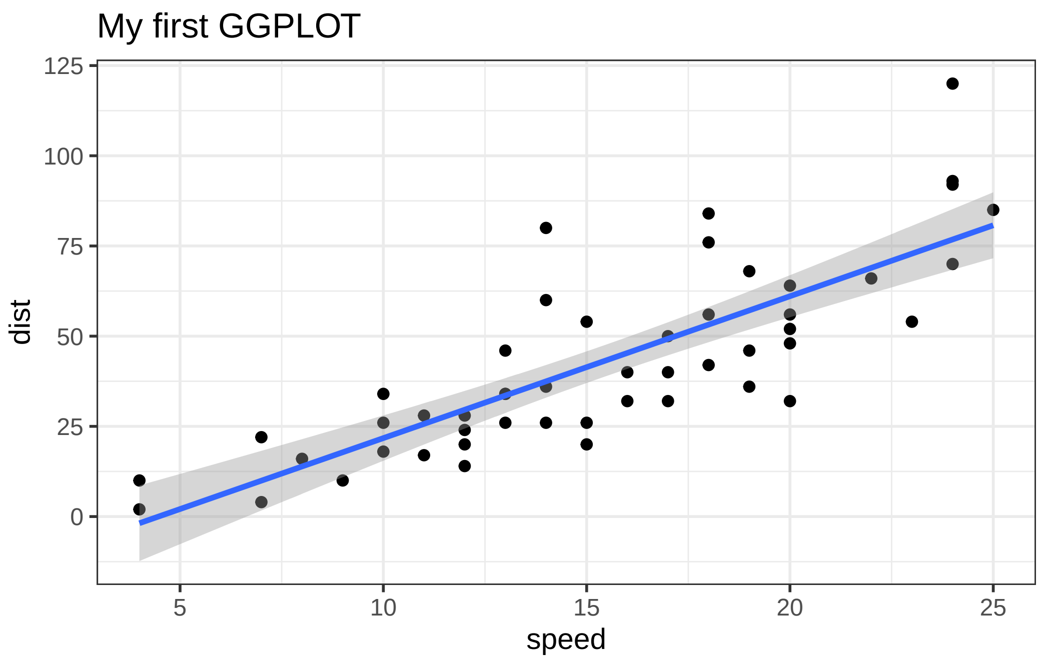


Use base R plots for data that you want to see but don't need a nice, polished graph. ggplot is more work, but more editable for nicer representations.

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

```
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'
```



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

geom_point()

Gene expression figure

The code to read the dataset, retrieving online from class 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?

```r
nrow(genes)
```

```
[1] 5196
```

A first plot of this dataset

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

colnames(genes)
```

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

```r
ncol(genes)
```

```
[1] 4
```

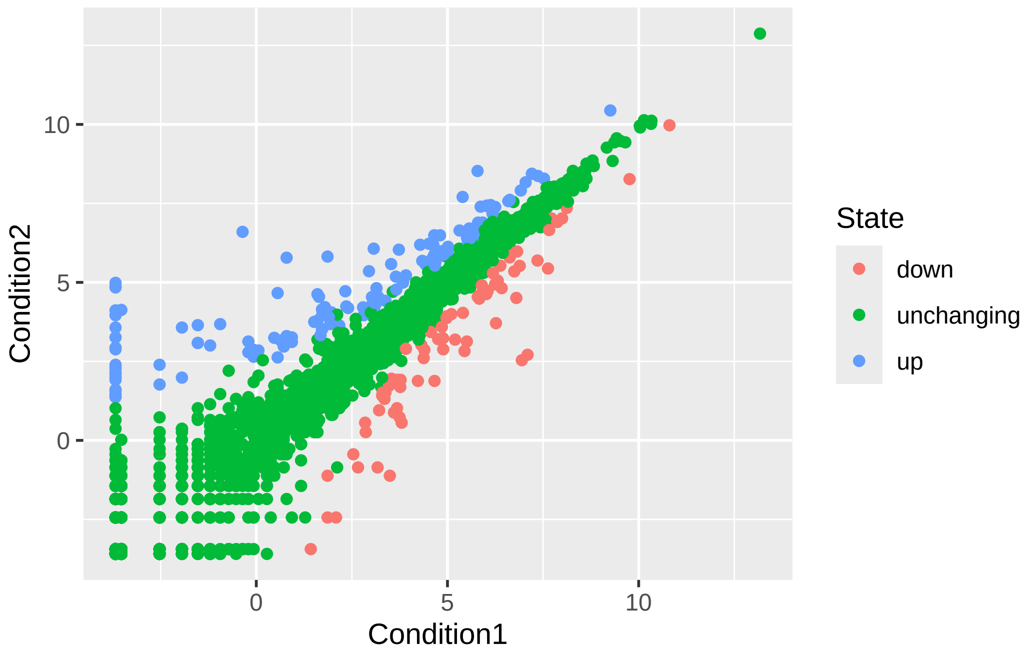
```r
## Graphical representation of the State values
table(genes$State)
```

```

      down unchanging         up
        72       4997        127
```

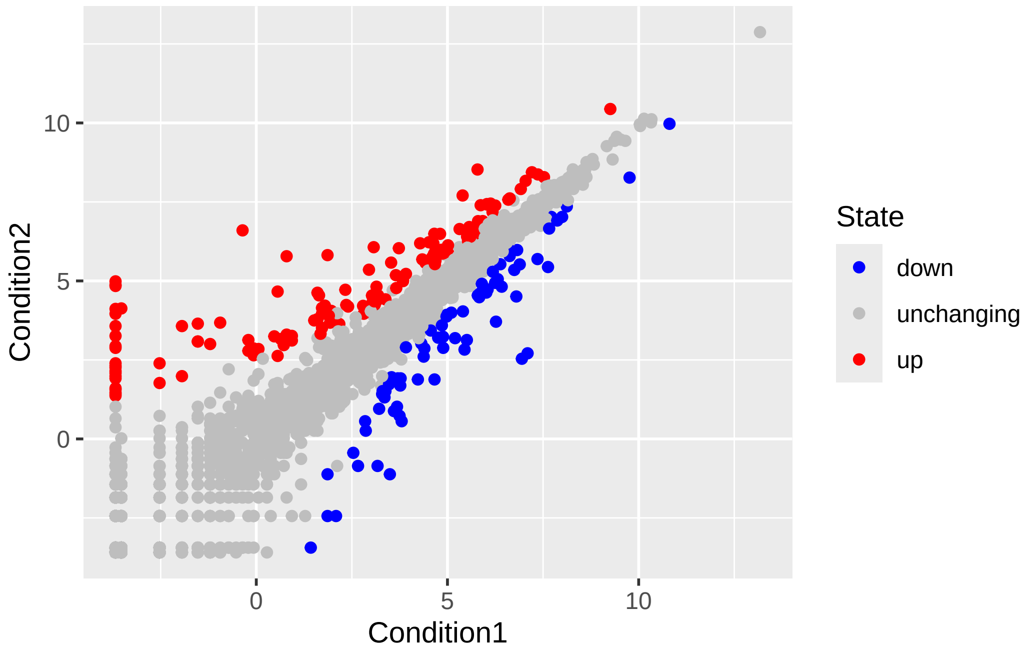
```r
## Graphical representation of the State values div by total # of genes
round(table(genes$State)/nrow(genes) *100, 2)
```

```

      down unchanging         up
      1.39      96.17       2.44
```

```r
## Default color scheme
p
```

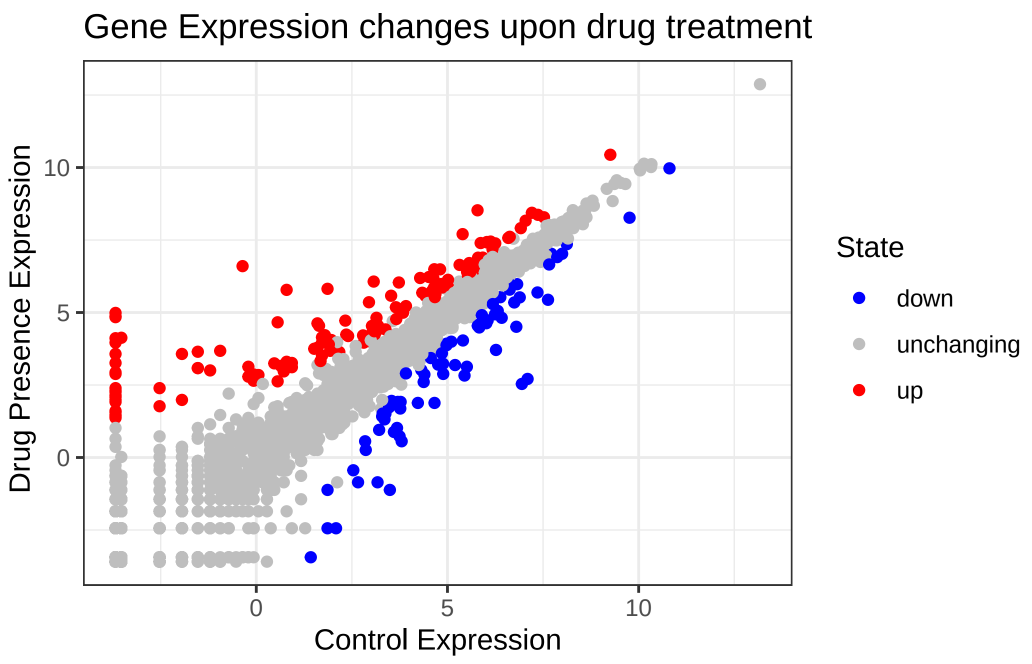


```
## Personalized color scheme
p + scale_color_manual(values= c("blue", "gray", "red"))
```



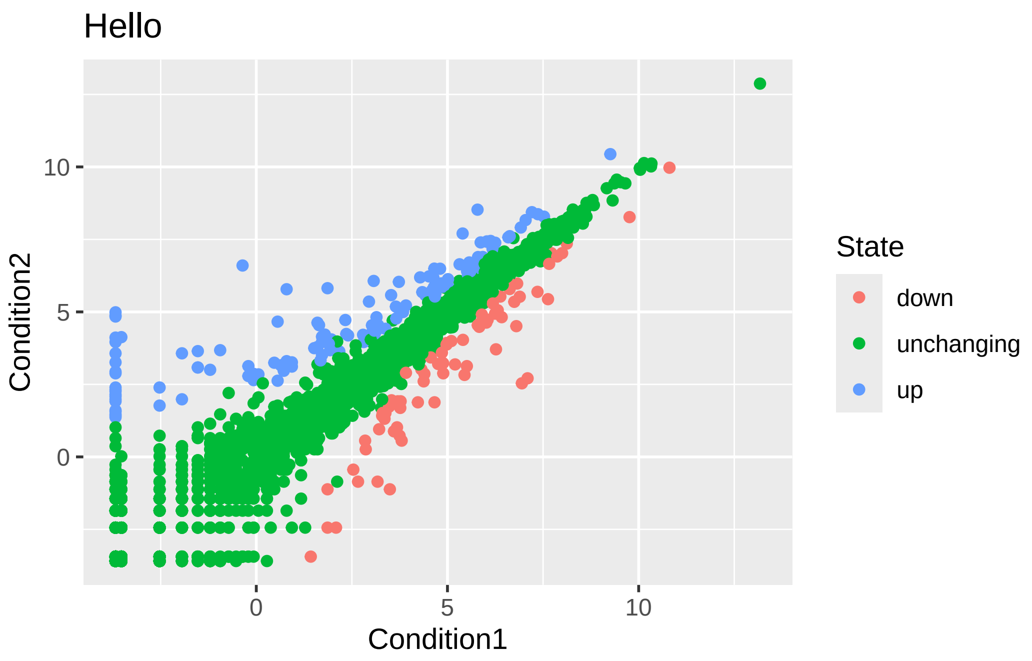
Lets add some labels and a title

```
ggplot(genes) +
  aes(x=Condition1, y=Condition2, col= State) +
  geom_point() +
  theme_bw() +
  labs(title= "Gene Expression changes upon drug treatment",
       x= "Control Expression",
       y= " Drug Presence Expression") +
  scale_color_manual(values= c("blue", "gray", "red"))
```



Defining the graph as an object makes it easier to edit or have multiple versions without having to rewrite the entire code again

```
p + labs(title="Hello")
```



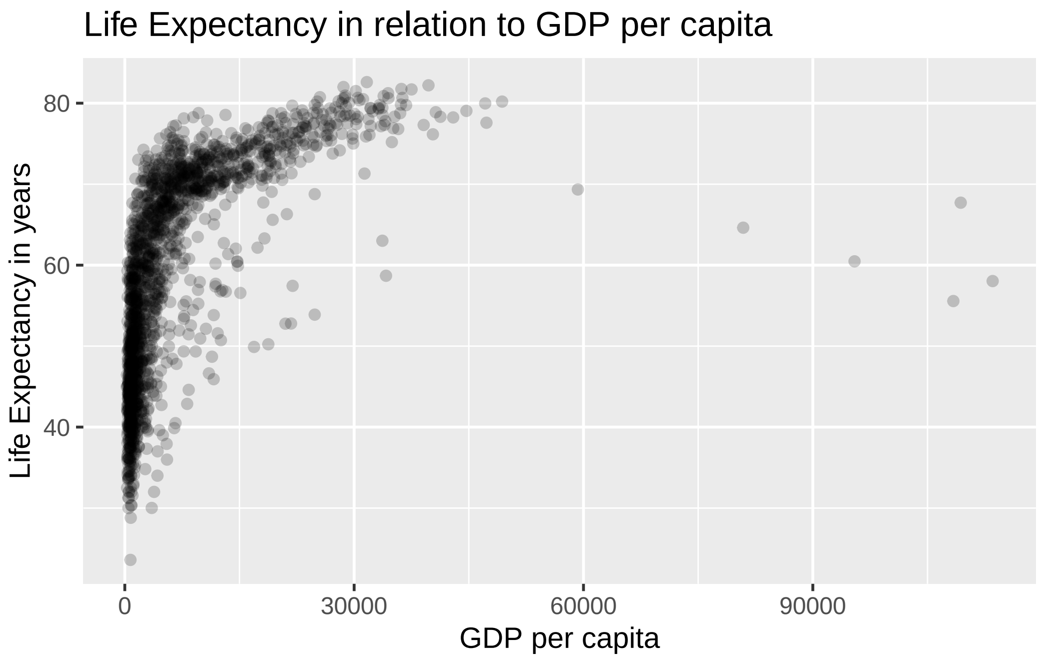
7. Going further

```
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.ts

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
```

```
#install.packages("dplyr")  ## un-comment to install if needed
#library(dplyr)

## alpha function edits transparency of points, on a scale for 0 - transparent, to 1 - full
ggplot (gapminder) +
```

```r
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha= 0.2) +
  labs (title= "Life Expectancy in relation to GDP per capita", x= "GDP per capita", y= "Lif
```



More specific, one year only

```r
## even though we installed dplyr in the brain below, we need to pull it from the library to
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
```

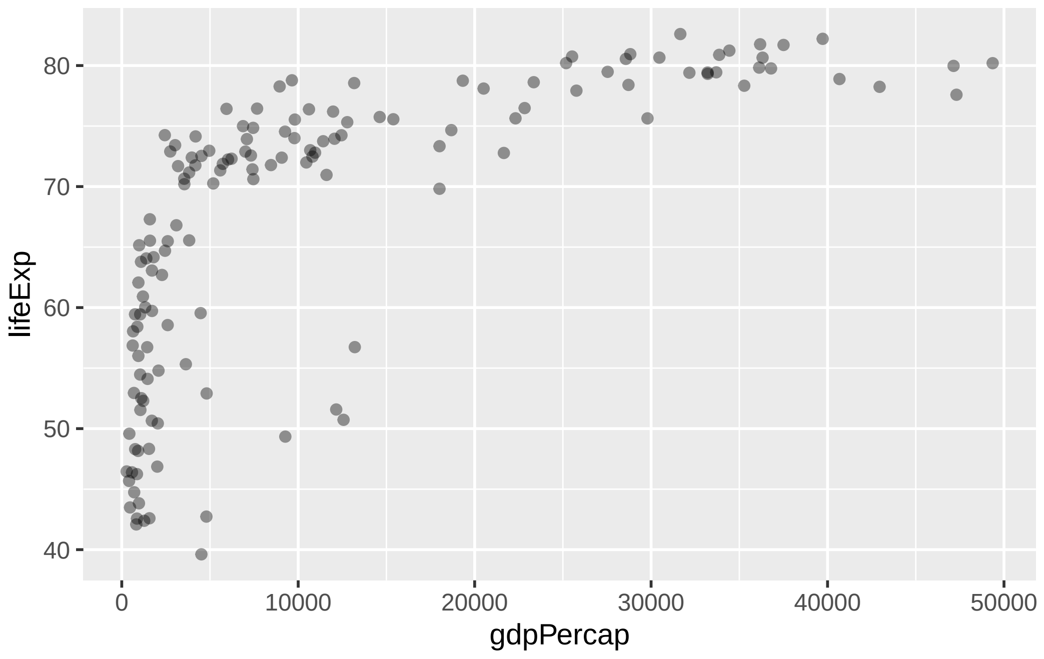
```
gapminder_2007 <- gapminder %>% filter(year==2007)

## There are a few new functions that have the same name as other ones in R, replacing it wi
```

Now we can see values only from 2007

```
head(gapminder_2007)
```

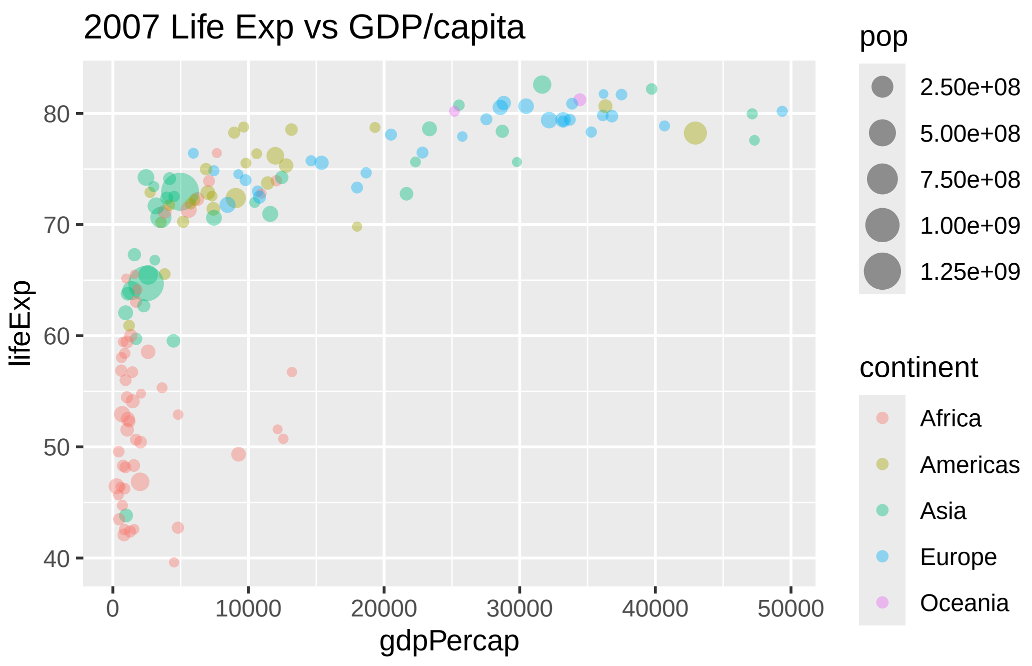
```
      country continent year lifeExp      pop  gdpPercap
1 Afghanistan      Asia 2007  43.828 31889923   974.5803
2     Albania    Europe 2007  76.423  3600523  5937.0295
3     Algeria    Africa 2007  72.301 33333216  6223.3675
4      Angola    Africa 2007  42.731 12420476  4797.2313
5   Argentina  Americas 2007  75.320 40301927 12779.3796
6   Australia   Oceania 2007  81.235 20434176 34435.3674
```

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



Adding more variables to aes()

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.4) +
  labs(title= "2007 Life Exp vs GDP/capita")
```


2007 Life Exp vs GDP/capita
lifeExp
gdpPercap
pop
2.50e+08
5.00e+08
7.50e+08
1.00e+09
1.25e+09
continent
Africa
Americas
Asia
Europe
Oceania