

# Class 05: Data Visualization with GGLOT

Isabel Mejia

## Table of contents

Our first plot	1
Drug Gene Regulation Table	5
Going further..	8
Bar Charts	14

## Our first plot

R has base graphics

```
head(cars)
```

```
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
```

```
plot(cars)
```



How would I plot this with `ggplot2`?

We need to install and load `ggplot2` To install any package in R we use the `install.packages()` function.

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

Before I can use this package, I need to load it with a `library()` function

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



Every ggplot needs at least 3 layers:

- **Data** (i.e. the data.frame we have),
- **Aes** (aesthetic mapping of our data to what we want to plot),
- **Geoms** (how we want to plot this stuff)

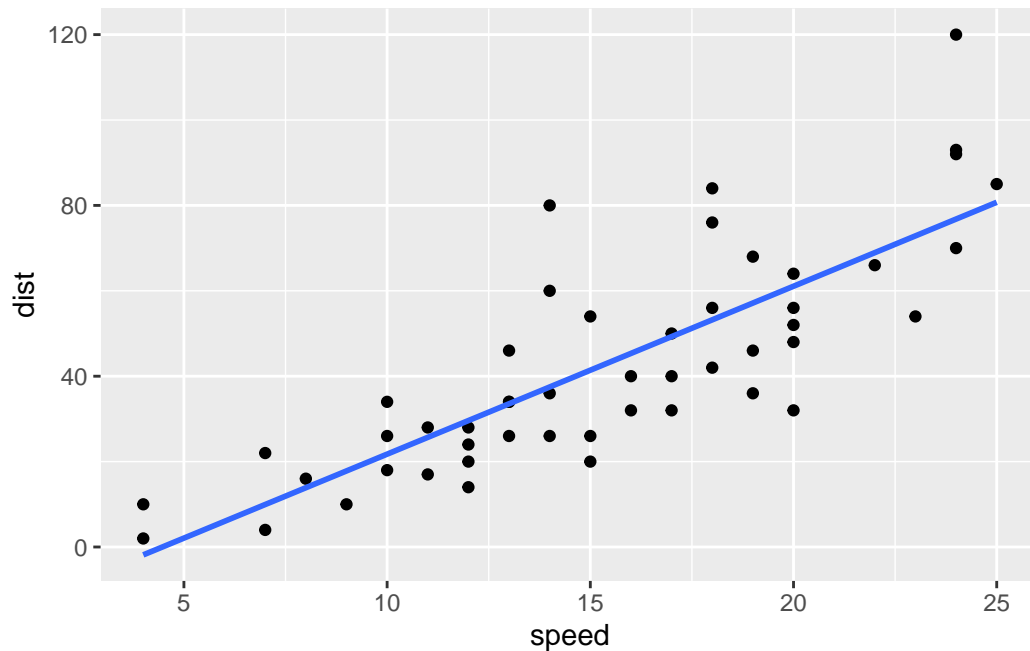
```
ggplot(data=cars) +  
  aes(x=speed, y=dist) +  
  geom_point()
```



Add another geom

```
ggplot(data=cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE)
```

`geom\_smooth()` using formula 'y ~ x'



## Drug Gene Regulation Table

```
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
```

There are 5196 genes in this dataset.

How many columns are there and what are the names?

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

There are 4 columns in this dataset.

The names of the columns are: Gene, Condition1, Condition2, State

How many genes are upregulated?

```
state_tabulation <- data.frame(table(genes$State))
state_tabulation
```

```
      Var1 Freq
1      down   72
2 unchanging 4997
3         up  127
```

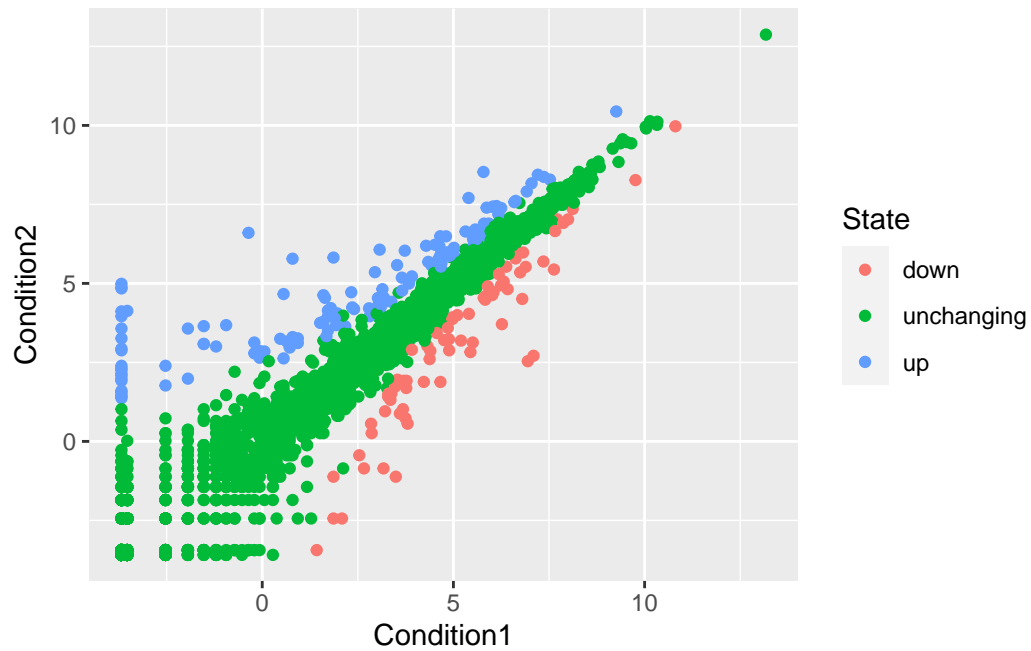
```
state_tabulation[3,2]
```

```
[1] 127
```

There are 127 genes that go up.

Plotting the genes with colors

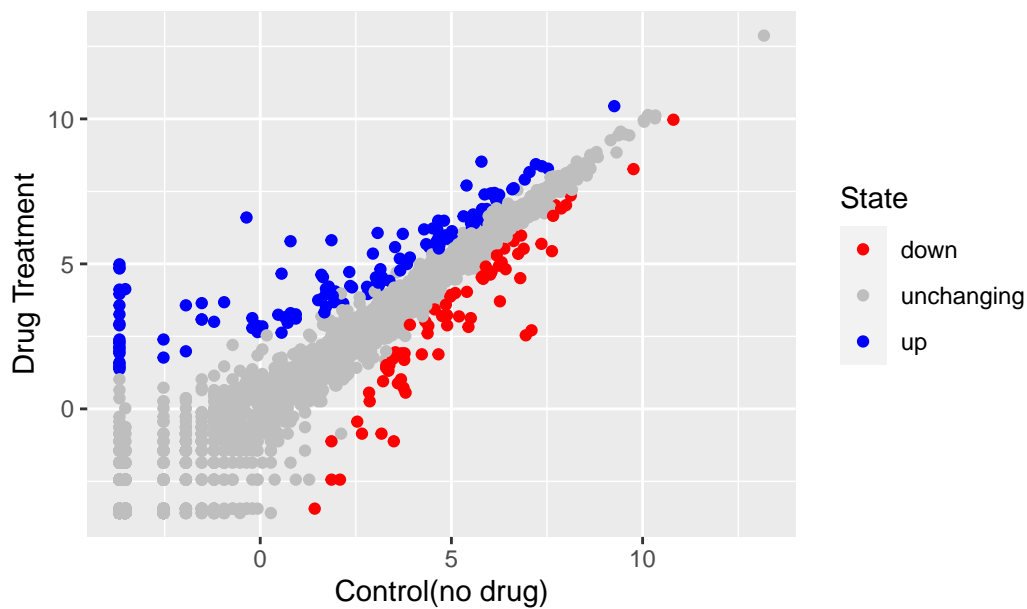
```
ggplot(data= genes)+
  aes(x = Condition1, y = Condition2, col=State)+
  geom_point()
```



Changing the genes colors to custom+adding labels

```
gene_plot <- ggplot(data= genes)+
  aes(x = Condition1, y = Condition2, col=State)+
  geom_point()+
  labs(title = "Gene Expression Changes Upon Drug Treatment",
        x = "Control(no drug)", y = "Drug Treatment")
gene_plot + scale_color_manual(values = c("red", "gray", "blue"))
```

## Gene Expression Changes Upon Drug Treatment



## Going further..

Download gapminder dataset.

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

Install the dplyr package and filter data to contain only 2007

```
#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':



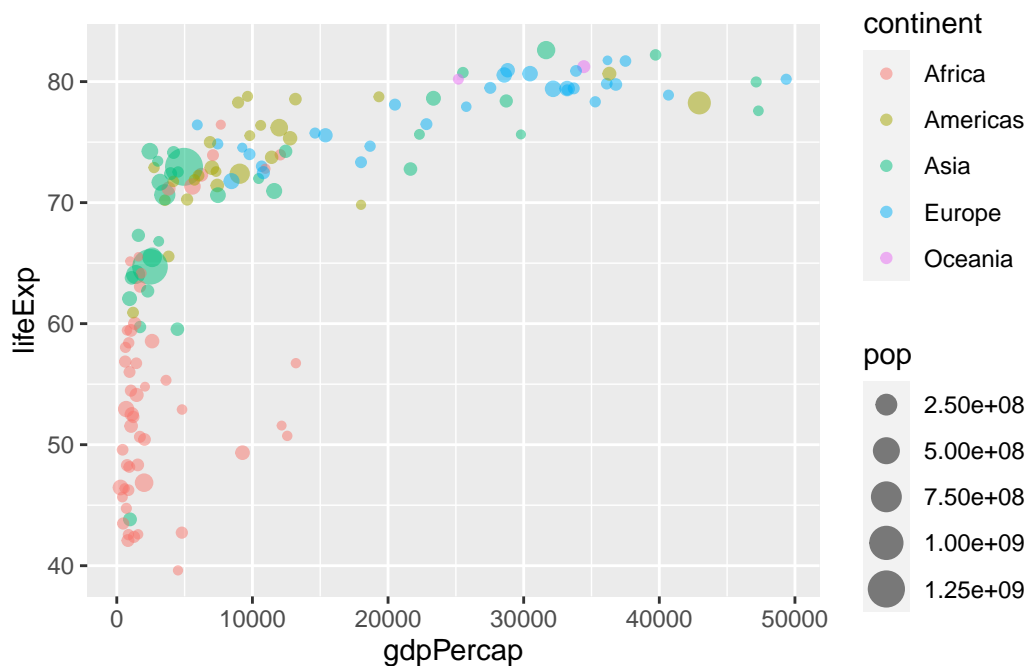
```
intersect, setdiff, setequal, union
```

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

Adding colors to match continent and change points to reflect sizes

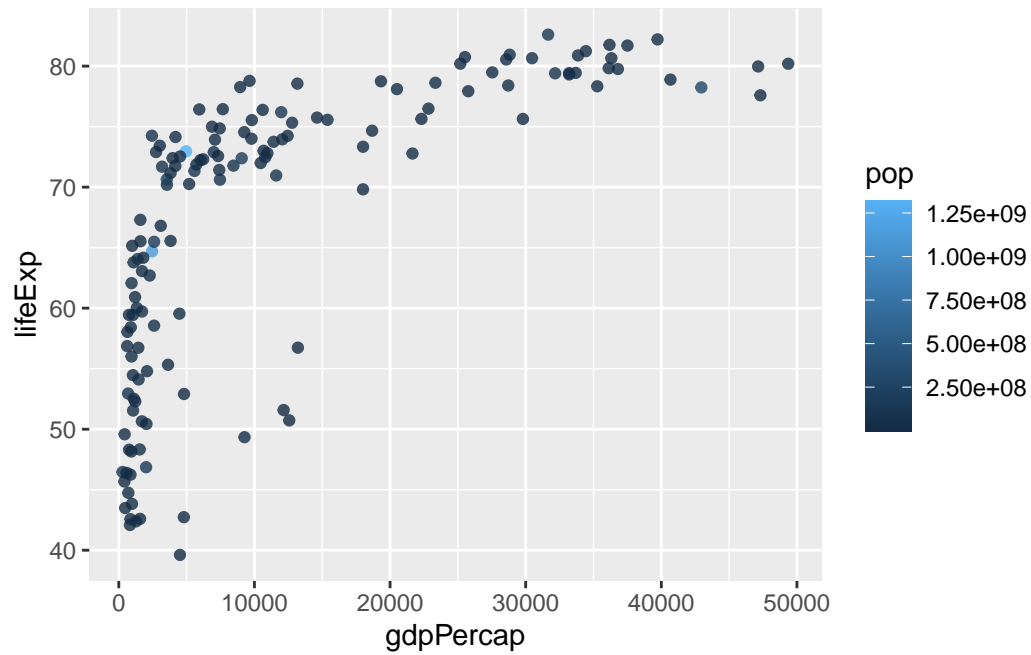
```
library(ggplot2)
```

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



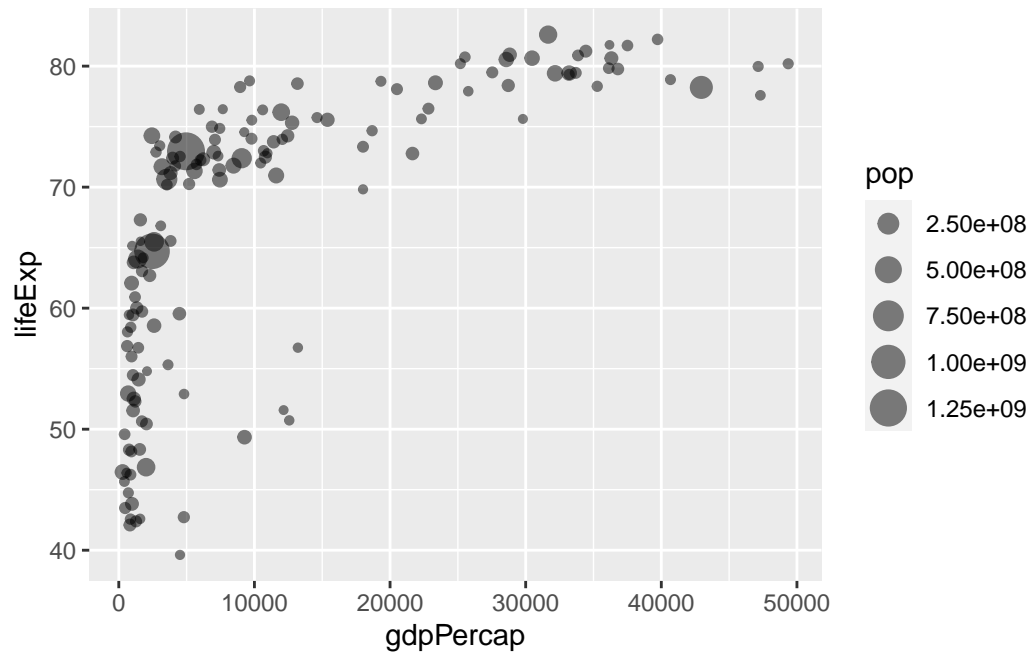
Changing color to reflect population

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



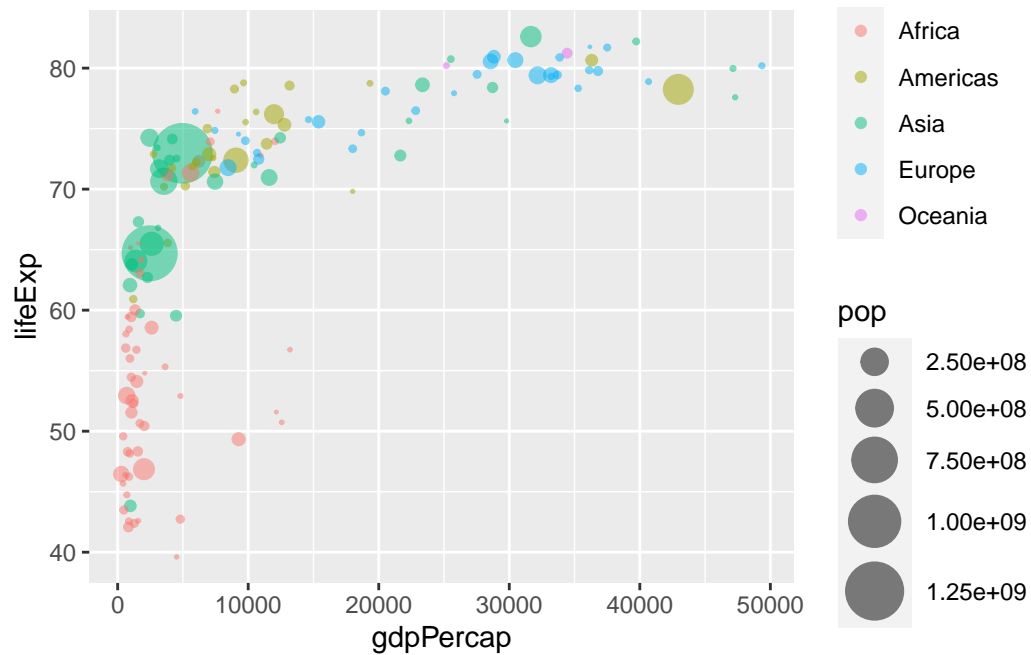
Changing size to reflect population

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



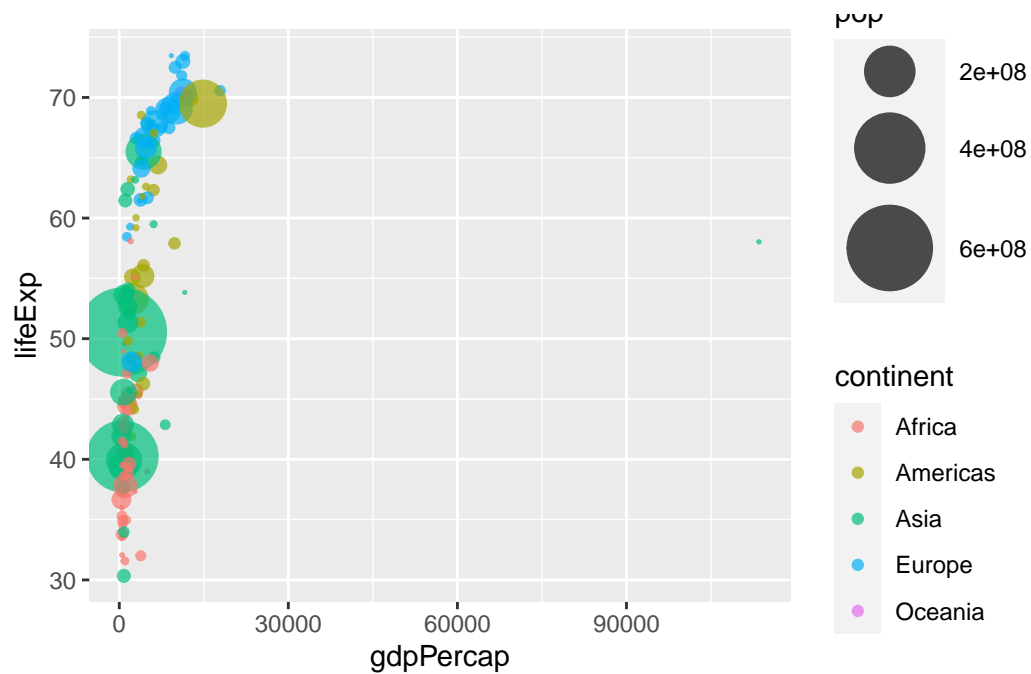
Changing size to reflect actual population size and rendering with color matching continents

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



Graphing the 1957 data

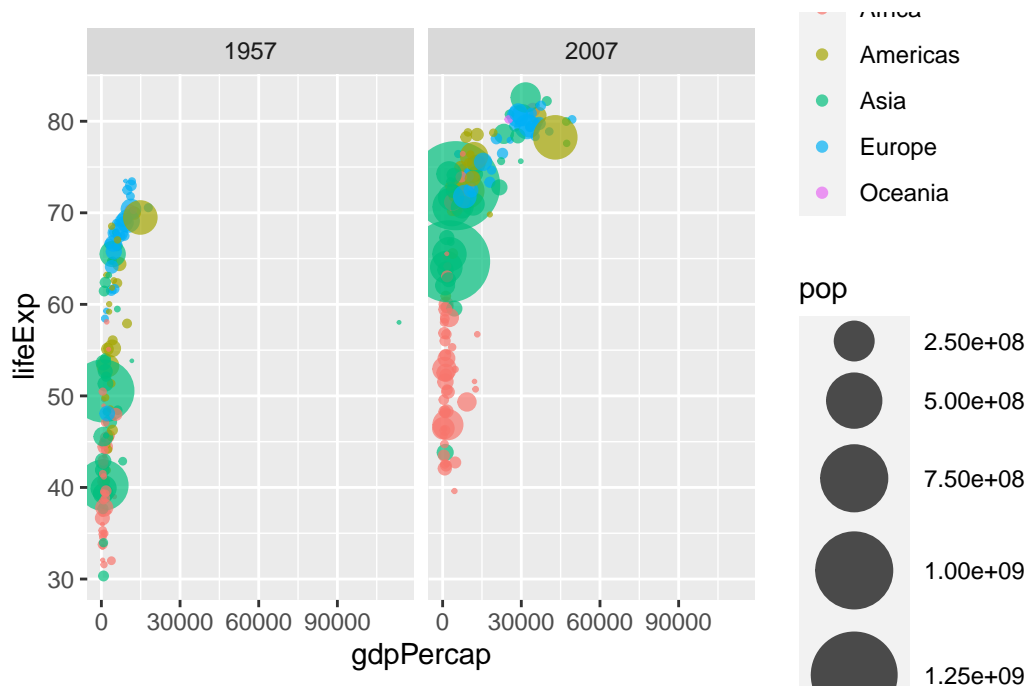
```
library(dplyr)
gapminder_1957 <- gapminder %>% filter (year==1957)
library(ggplot2)
ggplot(data=gapminder_1957)+
  aes(x = gdpPercap, y = lifeExp, color=continent, size=pop)+
  geom_point(alpha=0.7)+
  scale_size_area(max_size = 15)
```



Combining the 1957 and 2007 data into one plot

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

library(ggplot2)
ggplot(data=gapminder_data)+
  aes(x = gdpPercap, y = lifeExp, color=continent, size=pop)+
  geom_point(alpha=0.7)+
  scale_size_area(max_size = 15)+
  facet_wrap(~year)
```



## Bar Charts

Looking at data for the 5 biggest countries

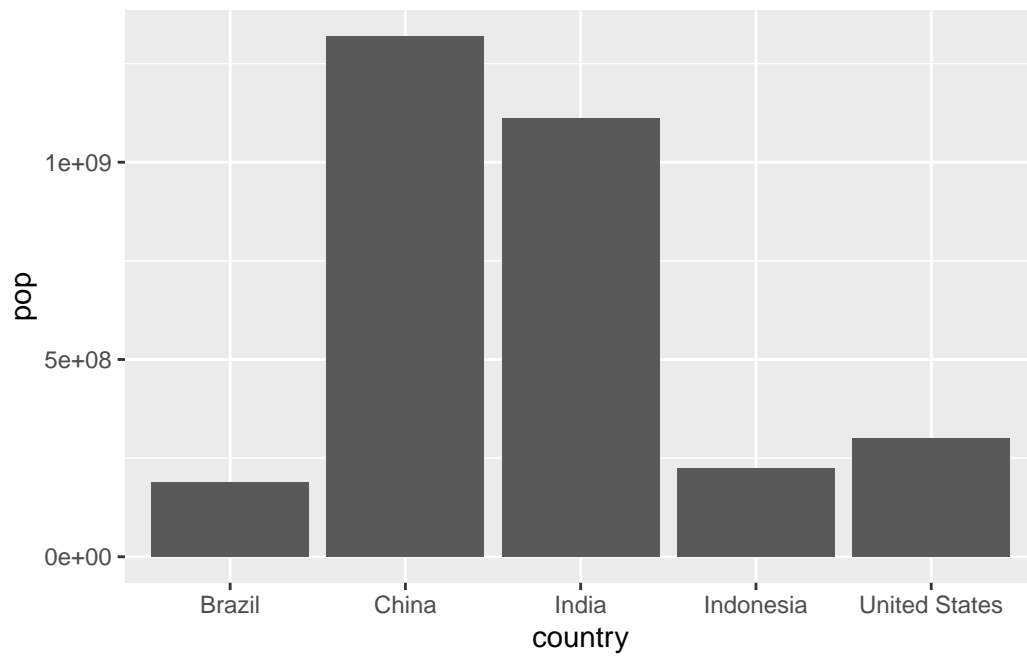
```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

gapminder_top5
```

	country	continent	year	lifeExp	pop	gdpPercap
1	China	Asia	2007	72.961	1318683096	4959.115
2	India	Asia	2007	64.698	1110396331	2452.210
3	United States	Americas	2007	78.242	301139947	42951.653
4	Indonesia	Asia	2007	70.650	223547000	3540.652
5	Brazil	Americas	2007	72.390	190010647	9065.801

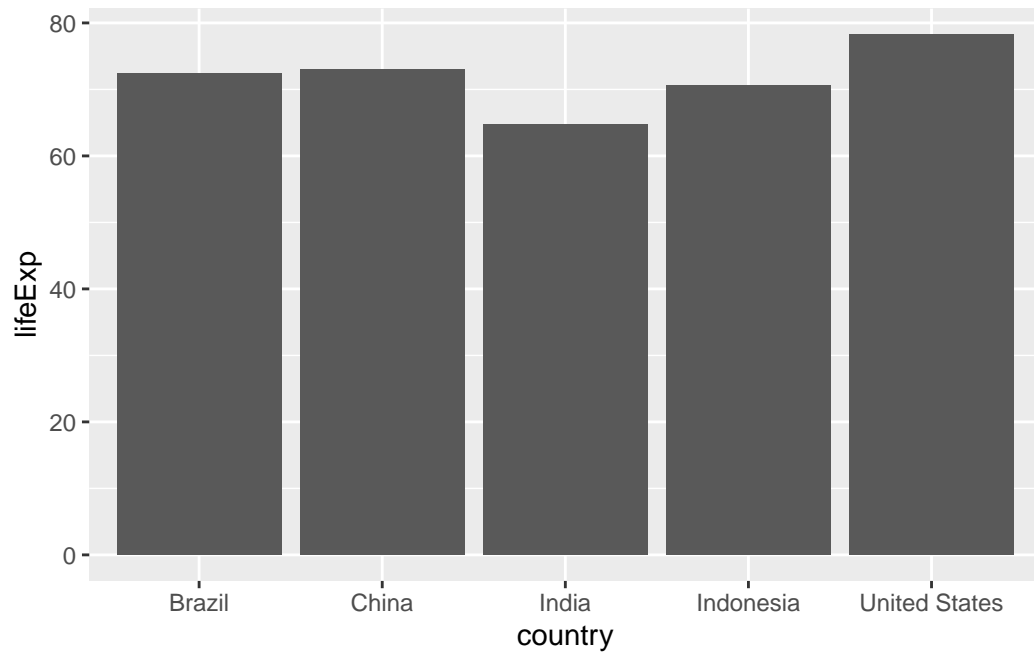
Creating a simple bar chart of the previous data

```
ggplot(gapminder_top5)+  
  geom_col(aes(x= country, y= pop))
```



Creating bar chart for life expectancy in the 5 biggest countries

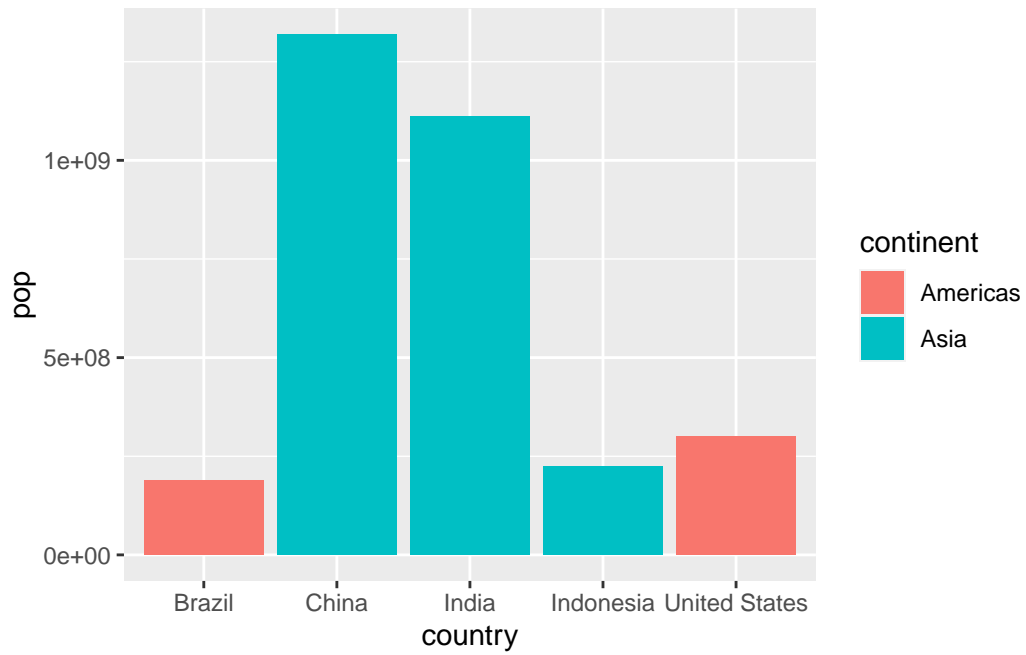
```
ggplot(gapminder_top5)+  
  geom_col(aes(x= country, y= lifeExp))
```



Playing with colors :)

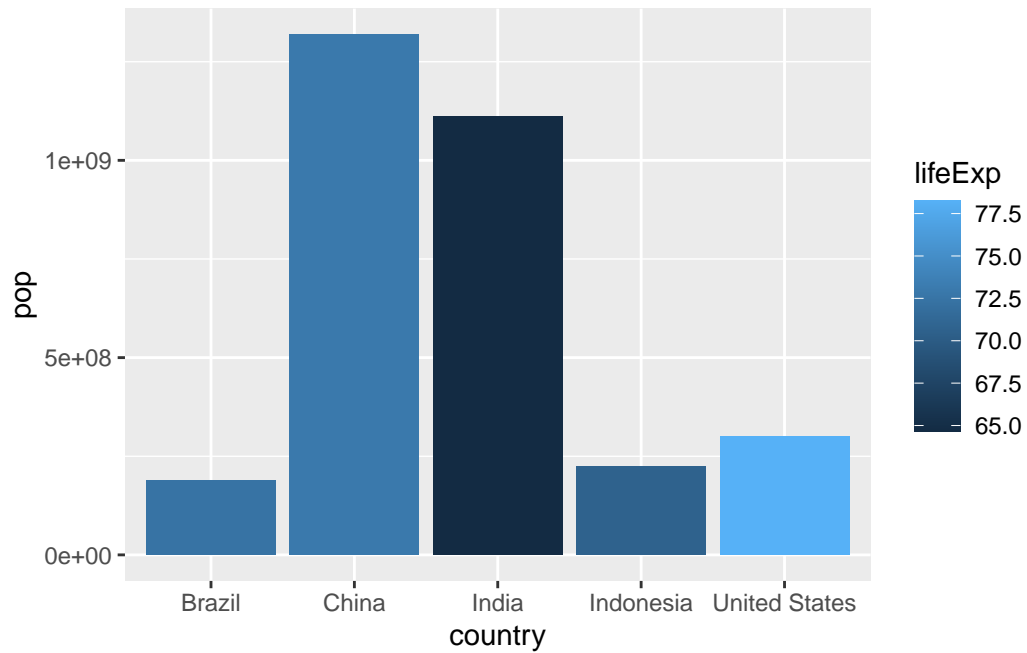
```
library(ggplot2)
ggplot(gapminder_top5)+ geom_col(aes(x= country, y = pop, fill = continent))
```





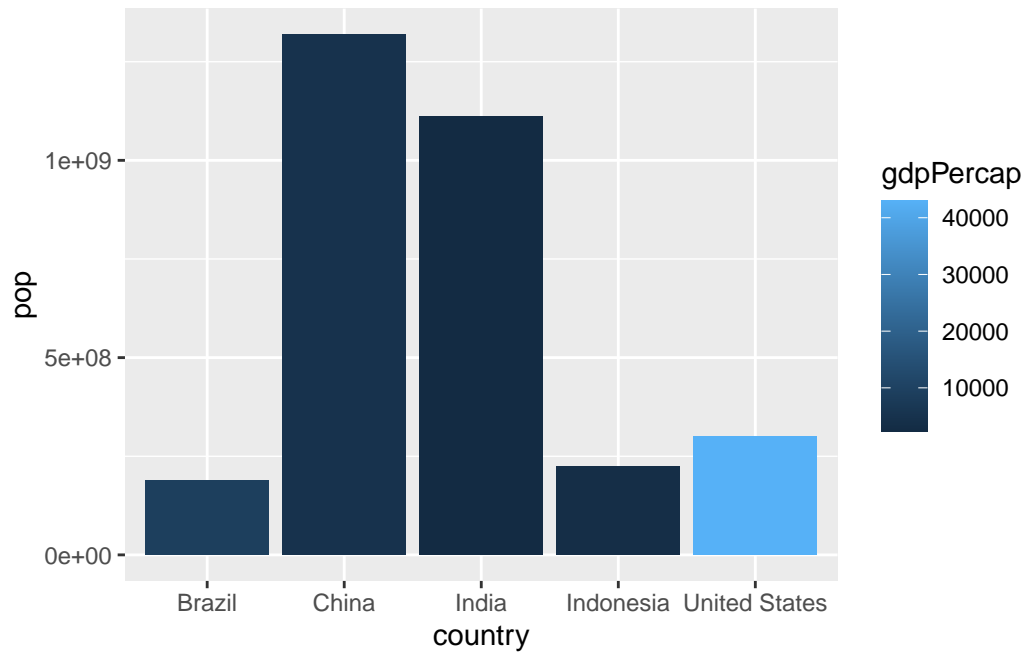
Using a numeric value to color

```
ggplot(gapminder_top5)+ geom_col(aes(x= country, y = pop, fill = lifeExp))
```



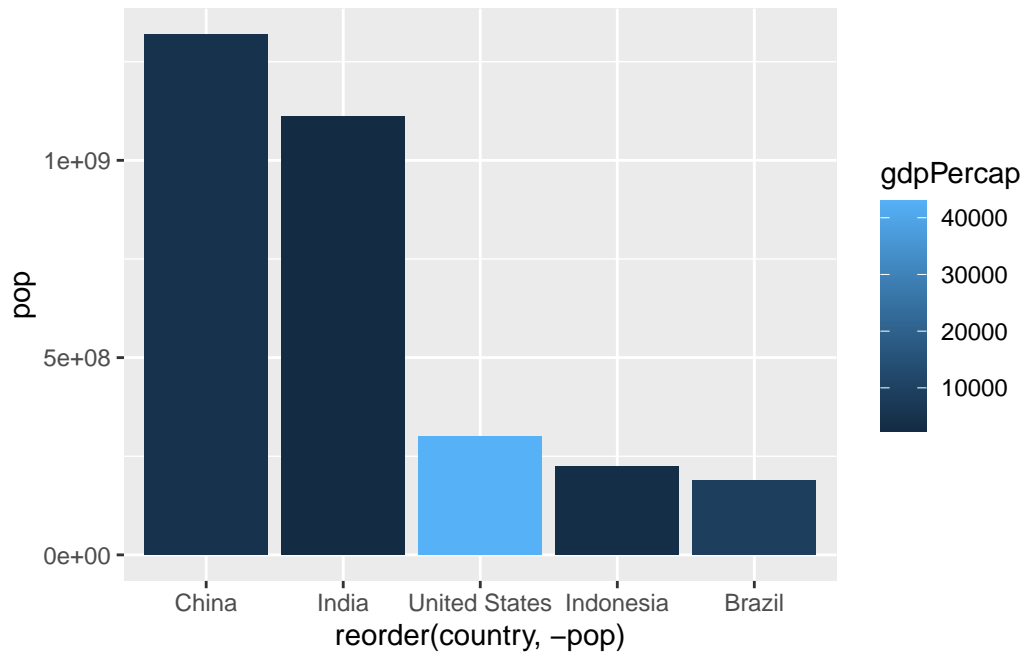
Plotting the population size by country

```
ggplot(gapminder_top5)+geom_col(aes(x= country, y=pop, fill=gdpPercap))
```



Reordering the bars

```
ggplot(gapminder_top5)+  
  aes(x=reorder(country, -pop), y=pop, fill=gdpPercap)+  
  geom_col()
```



Fill by country

```
ggplot(gapminder_top5)+  
  aes(x=reorder(country, -pop), y=pop, fill=country)+  
  geom_col()
```

