Class05: Data Visualization with GGPLOT

Aaron (PID:A17544470)

Intro to ggplot

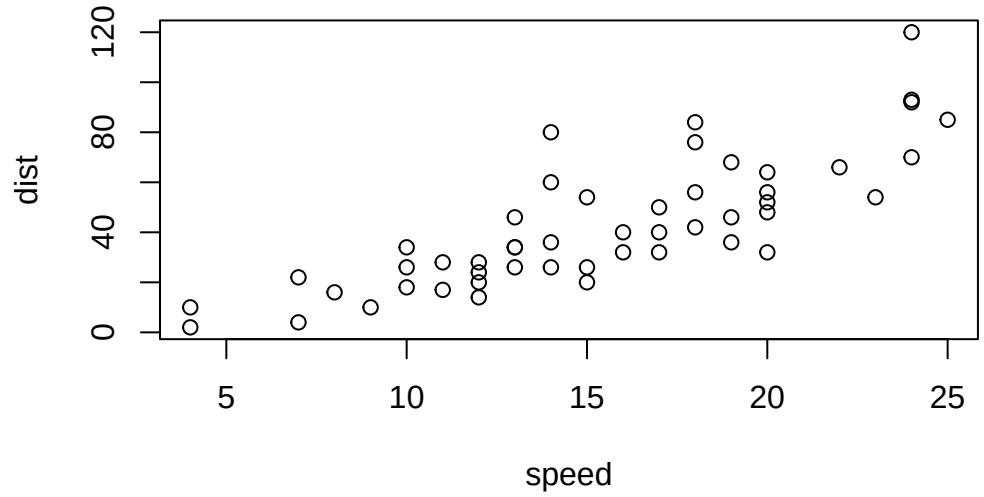
There are many graphic systems in R (ways to make plots and figures). These include "base" R plots. Today we will focus mostly on the **ggplot2** package.

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

```
head(cars)
```

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

```
plot(cars)
```



Let's see how we can make this figure using **ggplot2**. We need to install the **ggplot2** package first on this computer. For any R package use the function `install.package()`

> I will run `install.packages("ggplot2")` in my R console not this quarto document.

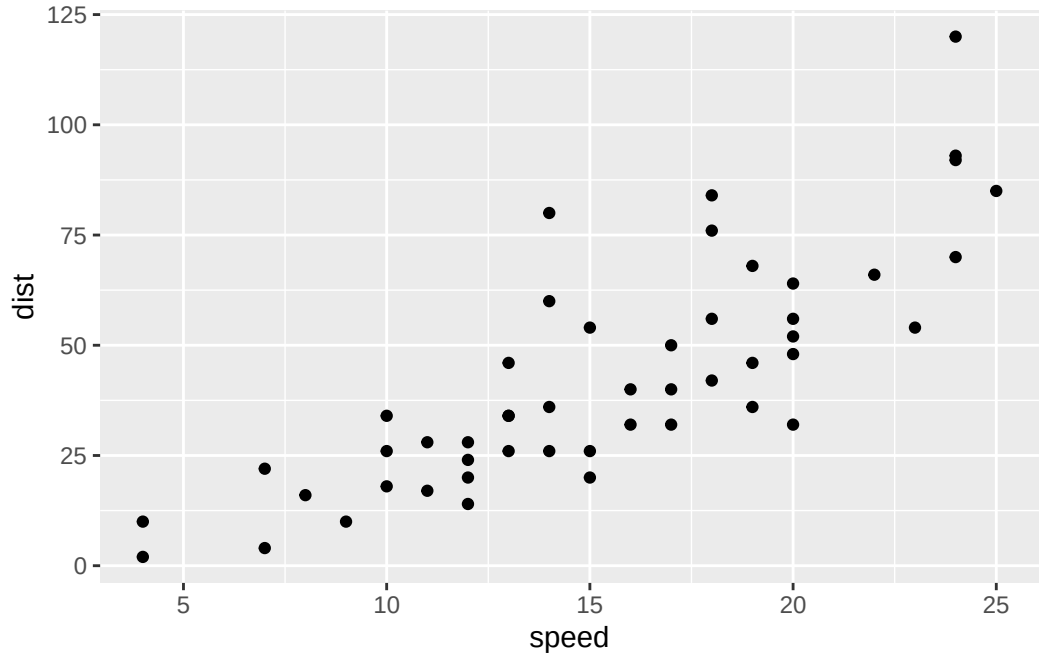
Before I can use any functions from add on package I need to load from my "library()" with the `library(ggplot2)` call.

```
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 in my plot),
- **geom** (the geom_plot(), geom_line() etc. that I want to draw).

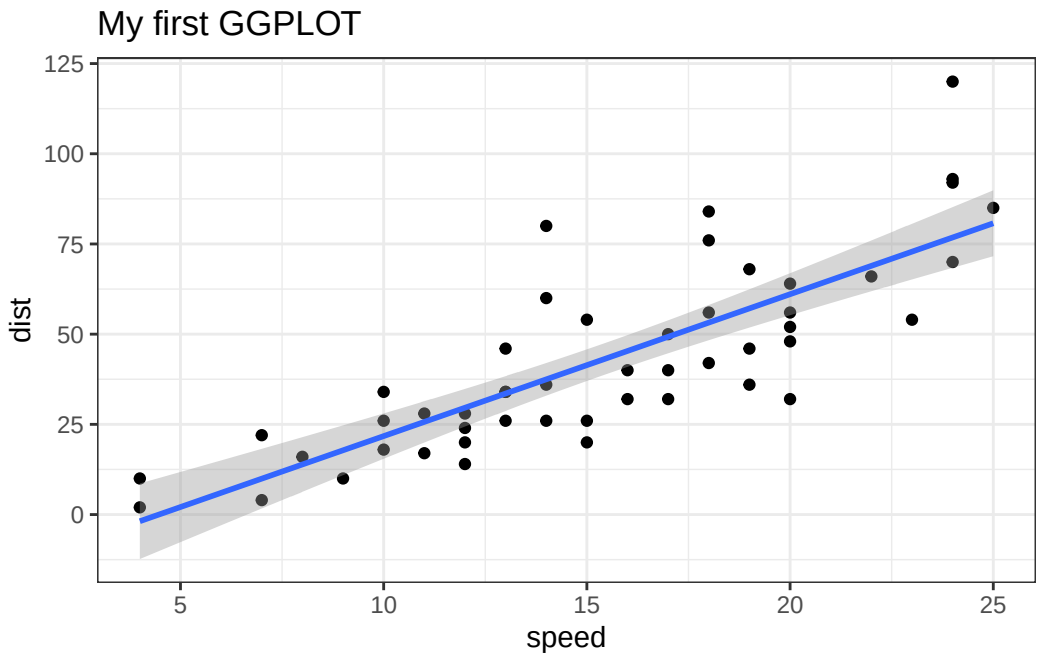
```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



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

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

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), 2)
```

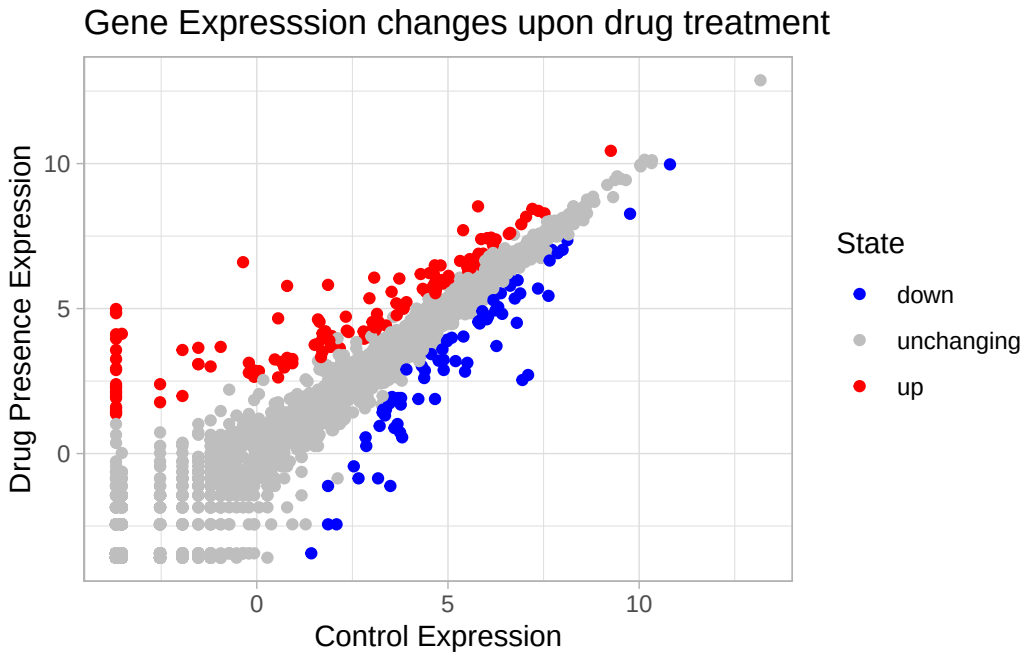
```
      down unchanging         up
      0.01       0.96       0.02
```

```
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
```

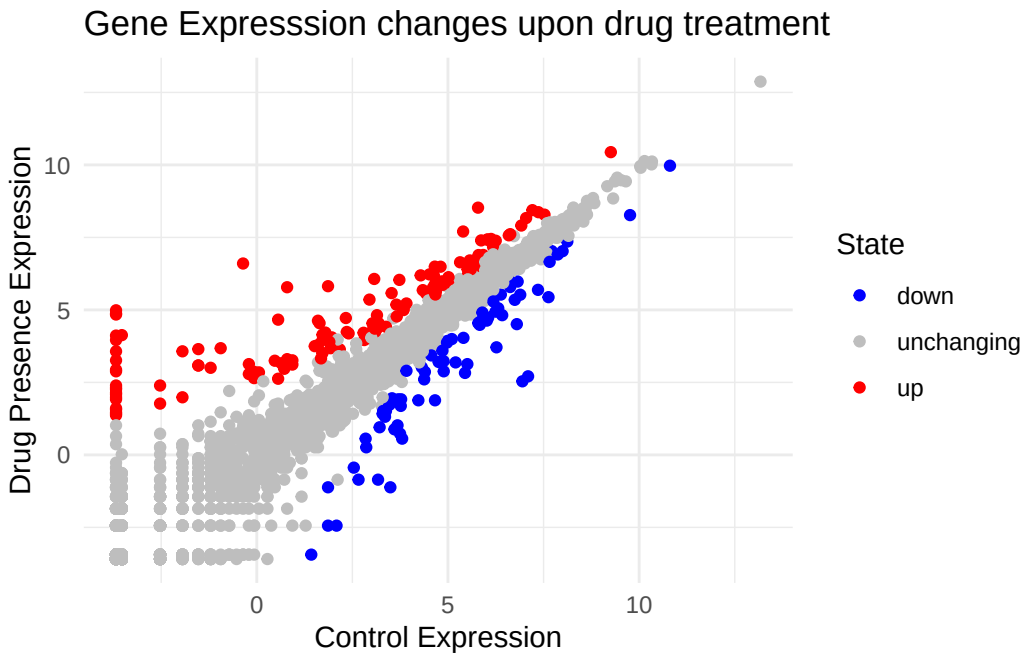
A first plot of this dataset. The plot is set as variable "p" to be easier to added to and then printed.

```
p <- ggplot(genes) +
  aes(x = Condition1, y = Condition2, colour = State) +
  geom_point() +
  theme_light() +
  labs(title="Gene Expresssion changes upon drug treatment",
       x = "Control Expression",
       y = "Drug Presence Expression") +
  scale_color_manual(values = c("blue","grey", "red"))

p
```



Adding on a minimal theme layer

```
p + theme_minimal()
```



##Gap Minder Section

The code to read in gapminder dataset

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

```
gapminder <- read.delim(url)
```

Before we make some plots we will use some **dplyr** code to focus in on a single year. You can install the **dplyr** package with the command install.packages("dplyr").

```
#install.packages("dplyr") and install.packages("ggplot2") ## un-comment to install if neede
library(dplyr)
```

```
Attaching package: 'dplyr'
```

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

```
    filter, lag
```

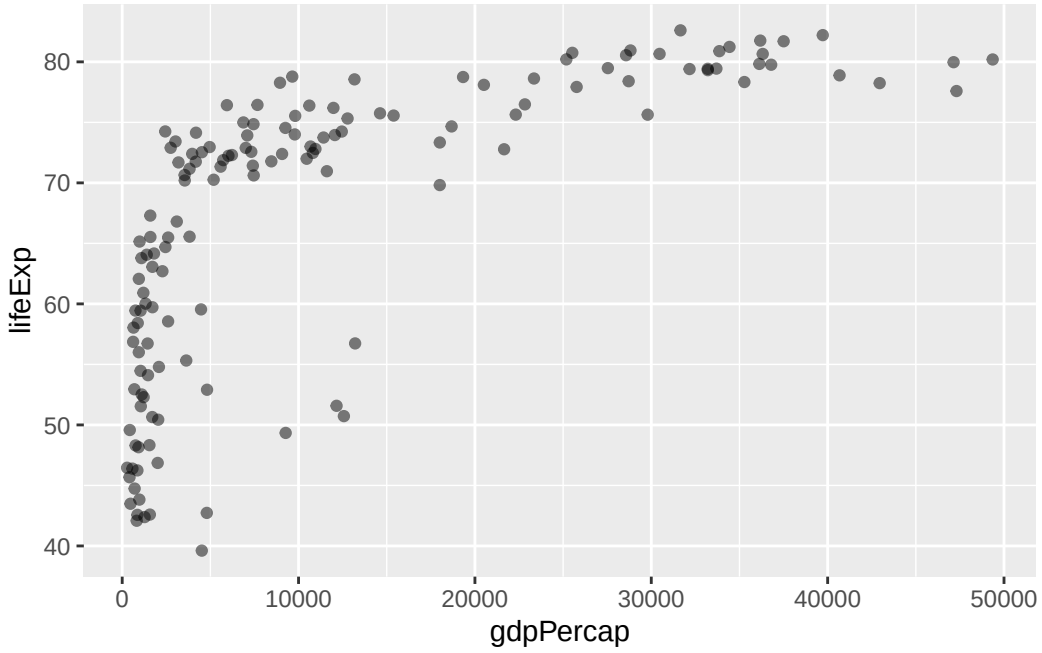
```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

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

Let's consider the gapminder_2007 dataset which contains the variables GDP per capita gdpPercap and life expectancy lifeExp for 142 countries in the year 2007.

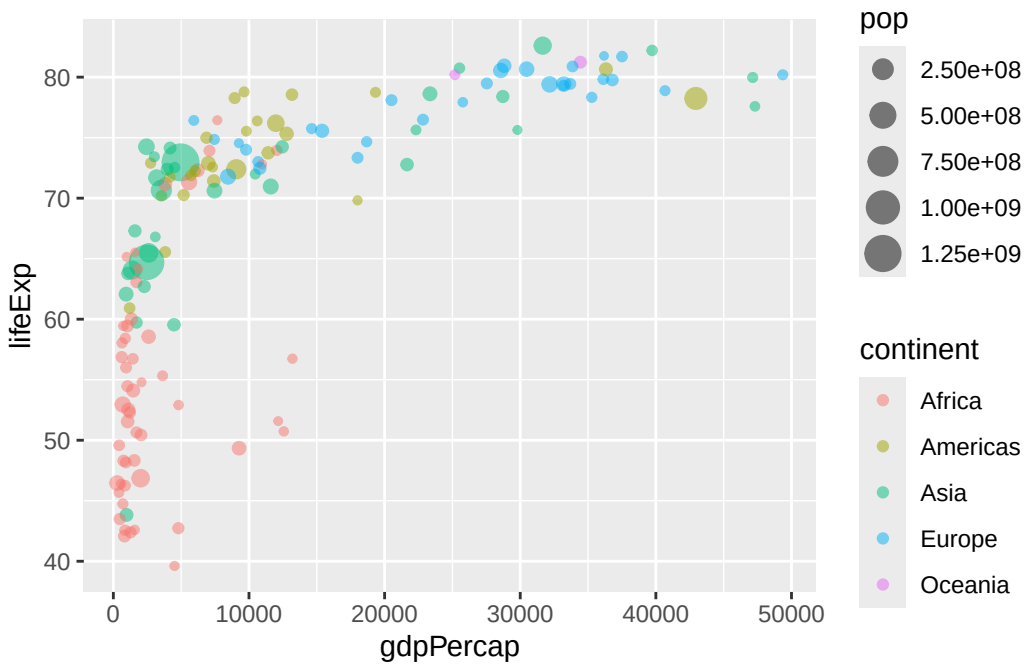
> Q Complete the code below to produce a first basic scatter plot of this gapminder_2007 dataset:

```
p1 <- ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp) +
  geom_point(alpha = 0.5)

p1
```



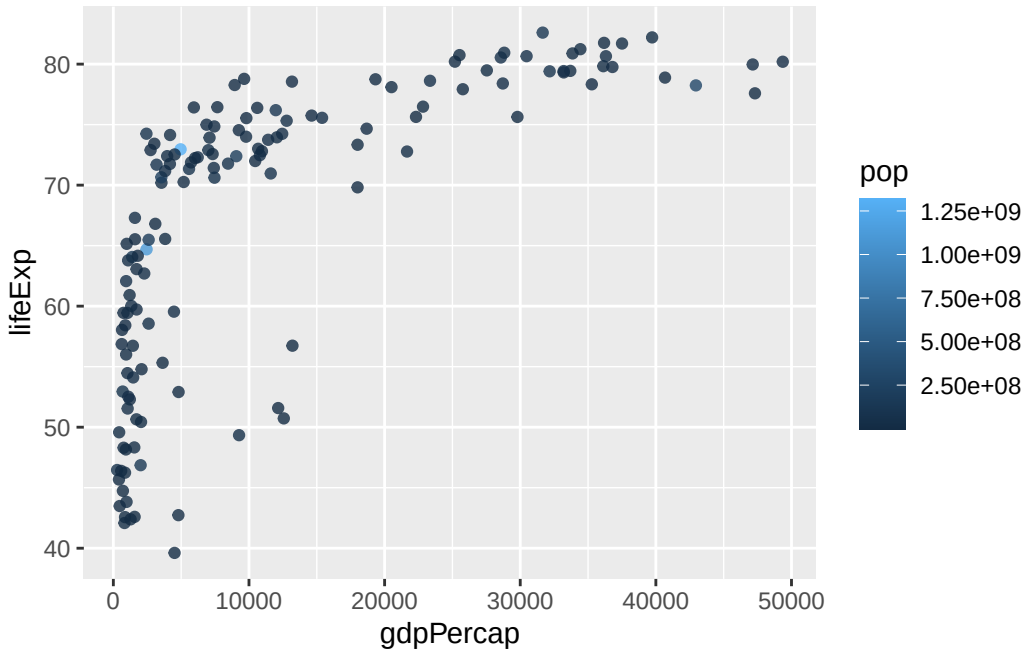
###Now we add more variables to aes()

```
p2 <- ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)

p2
```



This would be what the plot would look like if we color the points by the numeric variable population pop

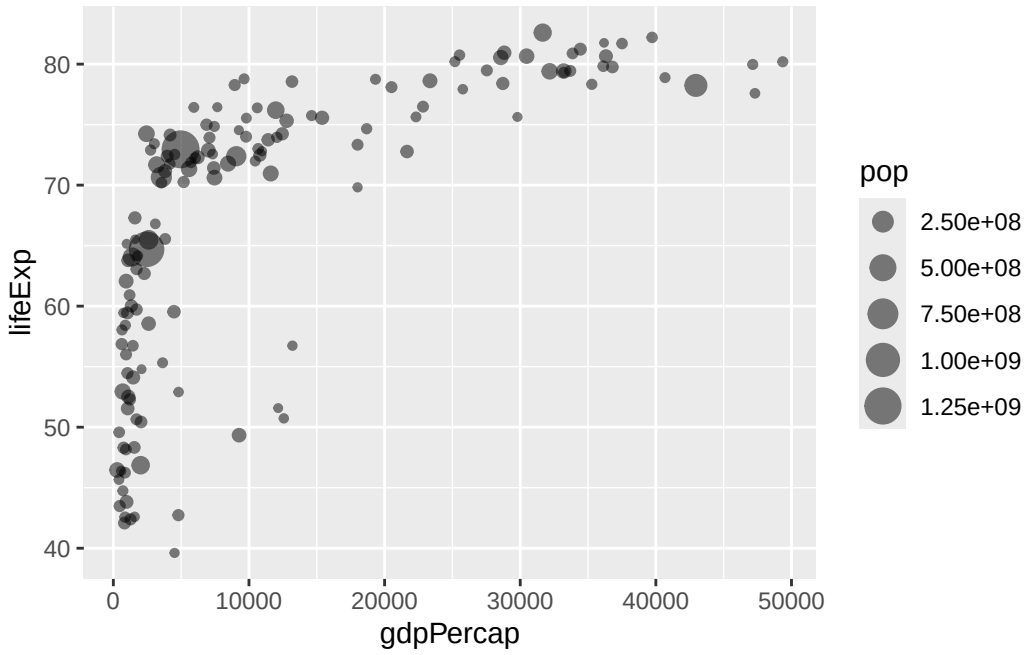
```
p3 <- ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)

p3
```



###Adjusting point size

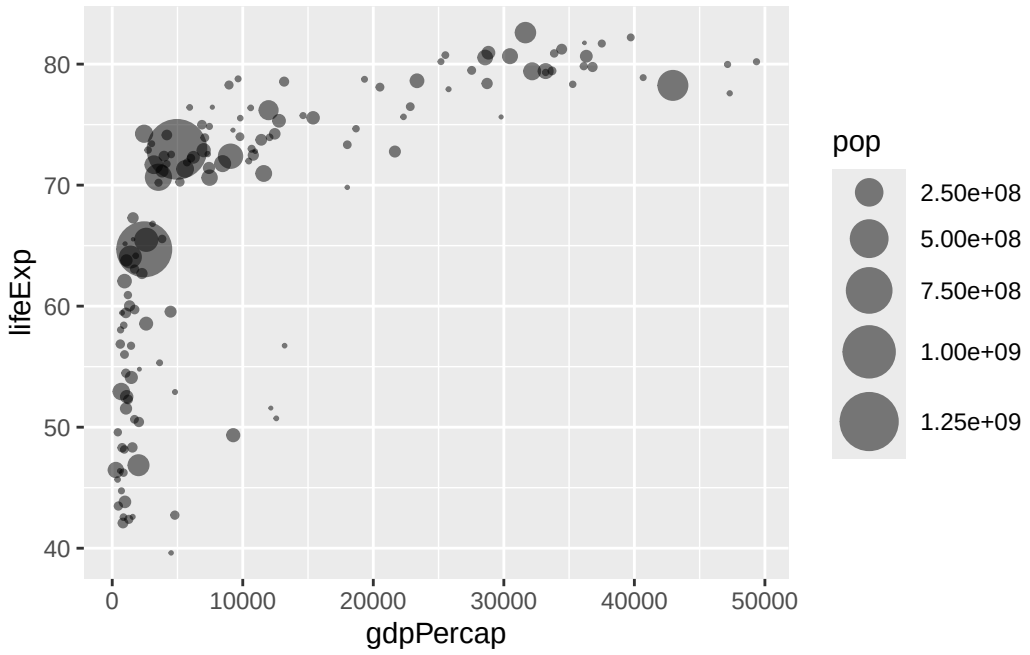
Setting the point size to be based on population.

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



The values are binned so instead we can use the scale_size_area() function instead.

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



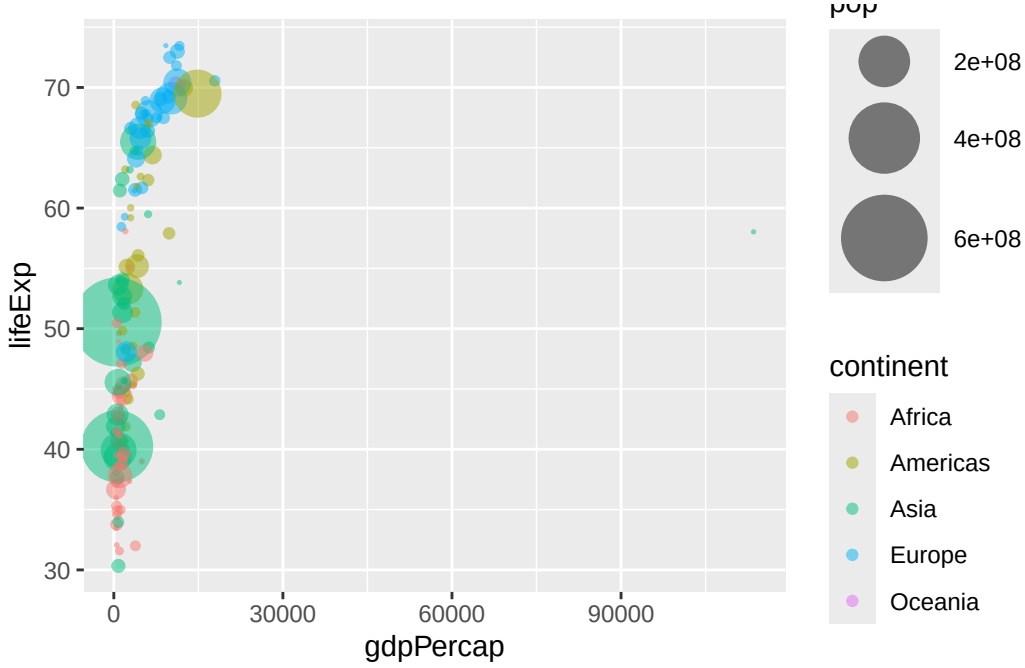
Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

- Use dplyr to filter the gapminder dataset

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

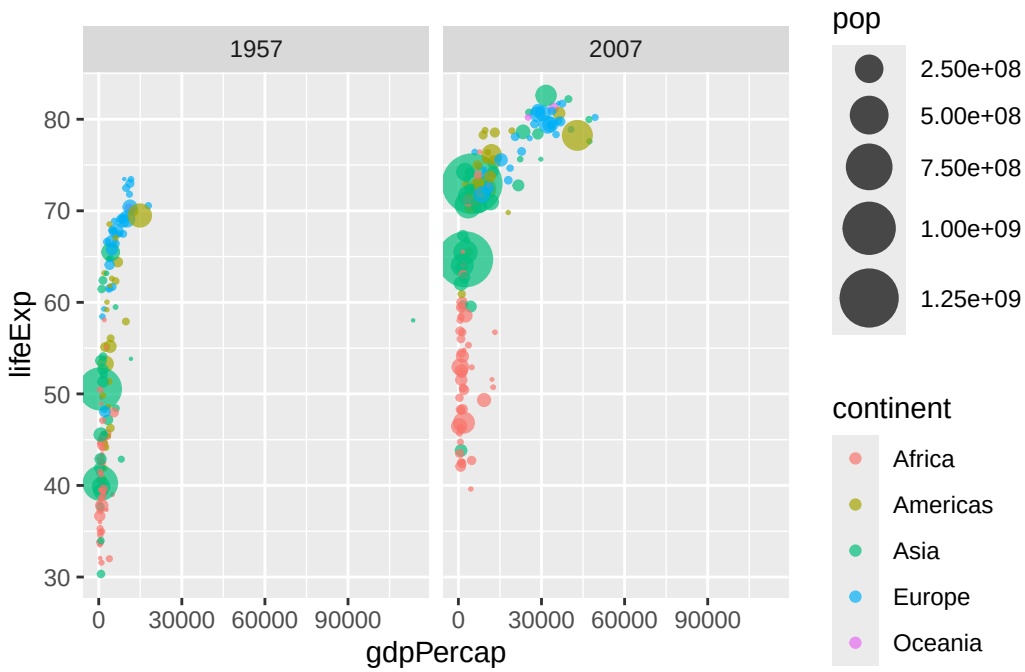
- Use the ggplot() function and specify the gapminder_1957 dataset as input
- Add a geom_point() layer to the plot and create a scatter plot showing the GDP per capita gdpPercap on the x-axis and the life expectancy lifeExp on the y-axis
- Use the color aesthetic to indicate each continent by a different color
- Use the size aesthetic to adjust the point size by the population pop
- Use scale_size_area() so that the point sizes reflect the actual population differences and set the max_size of each point to 15 -Set the opacity/transparency of each point to 70% using the alpha=0.7 parameter

```
p1957 <- ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, color = continent, size = pop) +
  geom_point( alpha = 0.5) +
  scale_size_area(max_size = 15)

p1957
```



Here is a comparison of the years 1957 and 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)
```



##Barplots

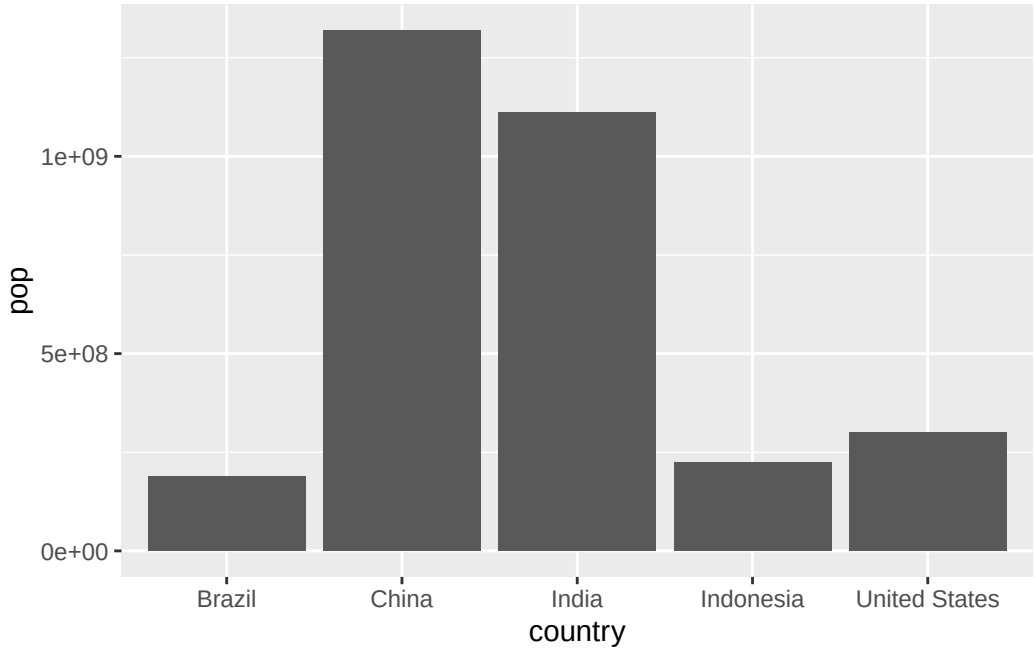
We generate the top 5 biggest countries from the gapminder dataset

```
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
```

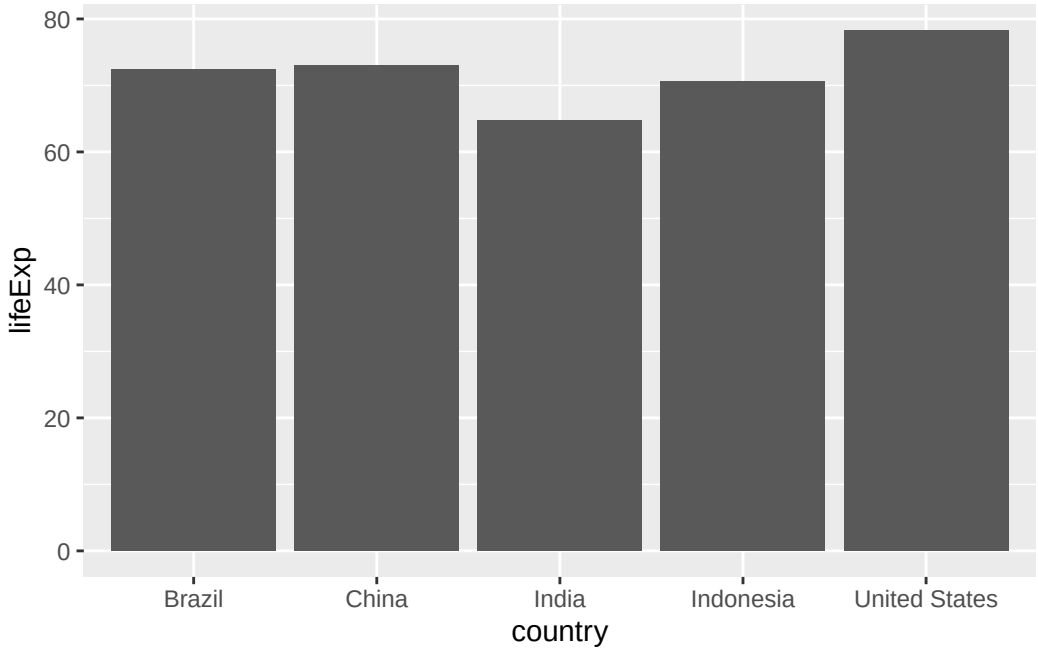
From there we can create a simple bar chart.

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



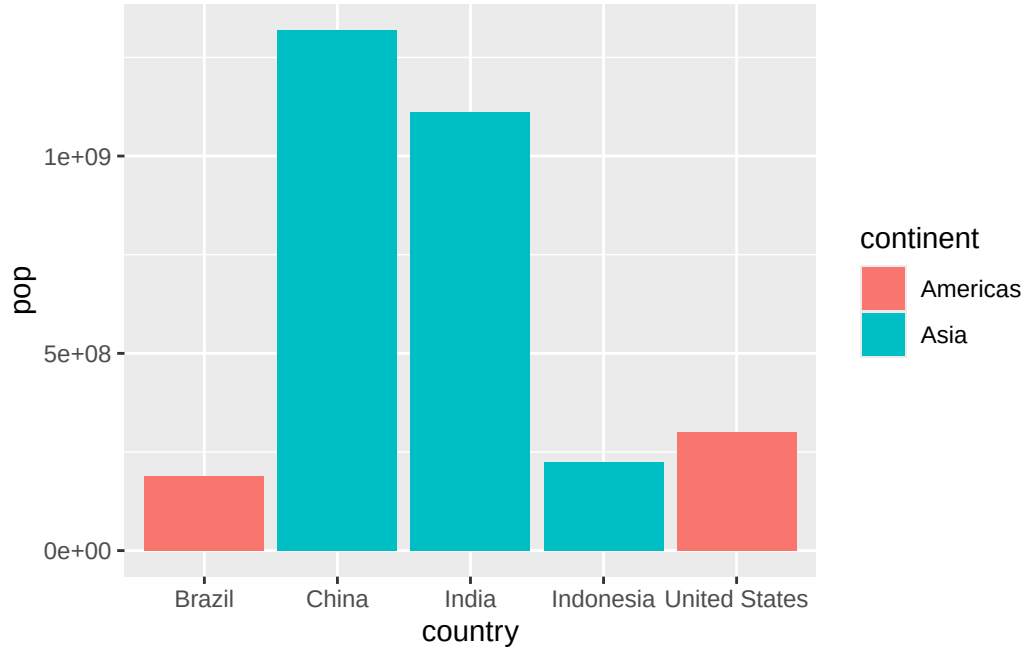
Q. Create a bar chart showing the life expectancy of the five biggest countries by population in 2007.

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



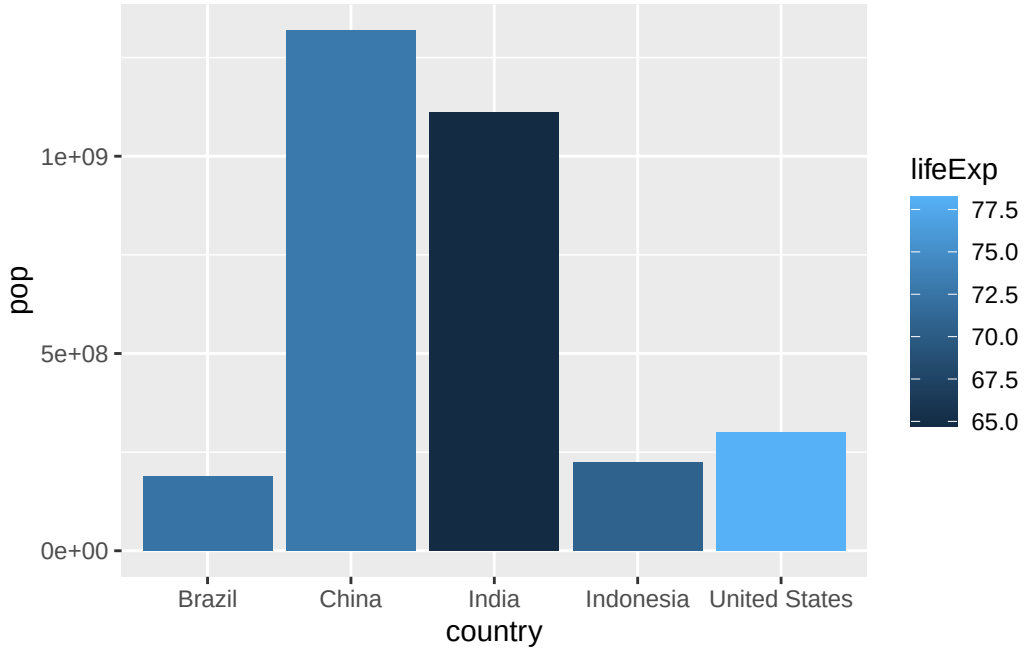
We can use the continent variable to color each bar.

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



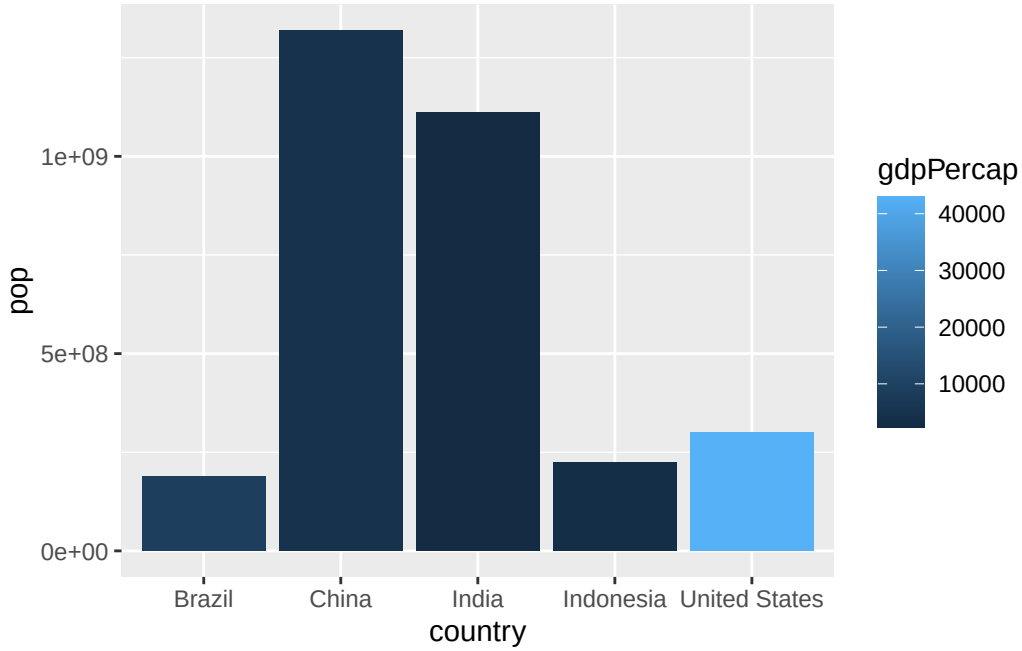
We can also use a numeric variable like lifeExp.

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



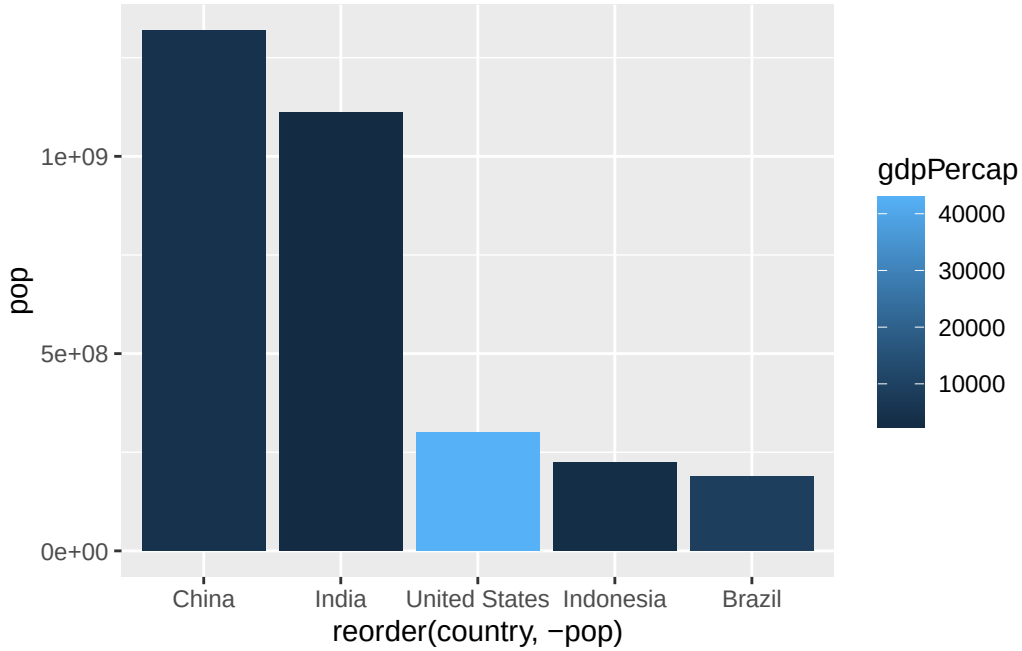
Q. Plot population size by country. Create a bar chart showing the population (in millions) of the five biggest countries by population in 2007.

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



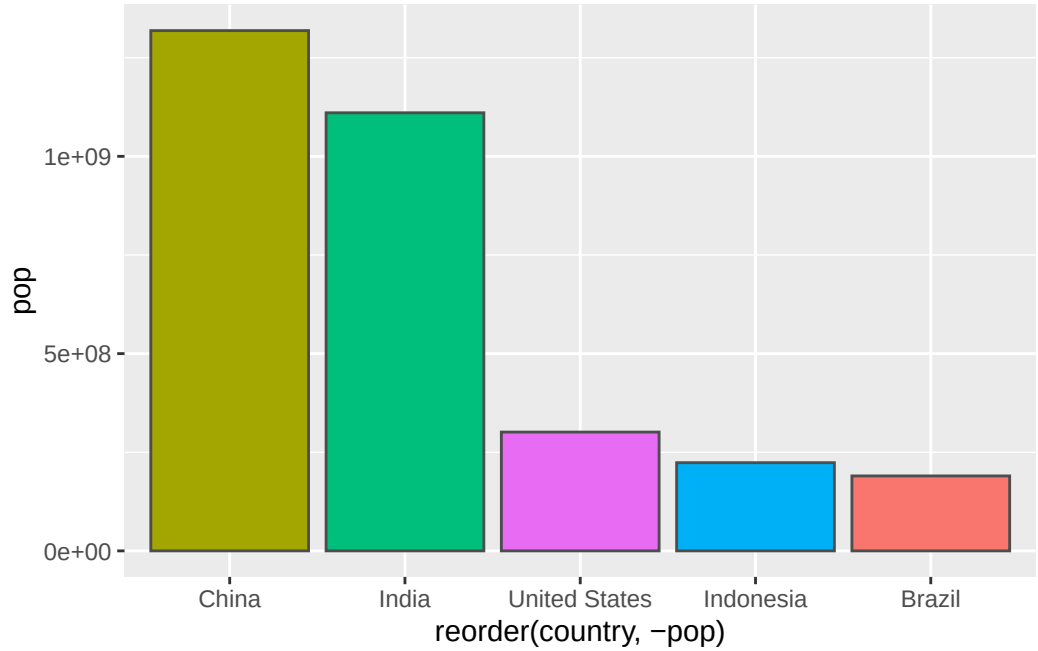
Ordering by country

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



Or filling by country

```
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=country) +
  geom_col(col="gray30") +
  guides(fill="none")
```

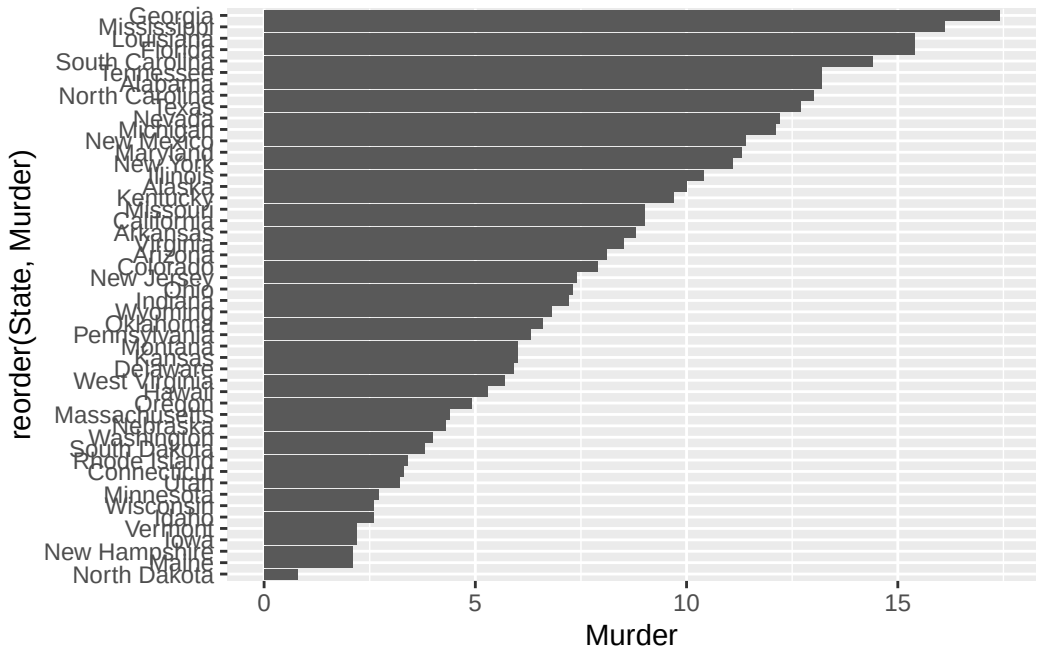


Looking at flipping bar charts

```
head(USArrests)
```

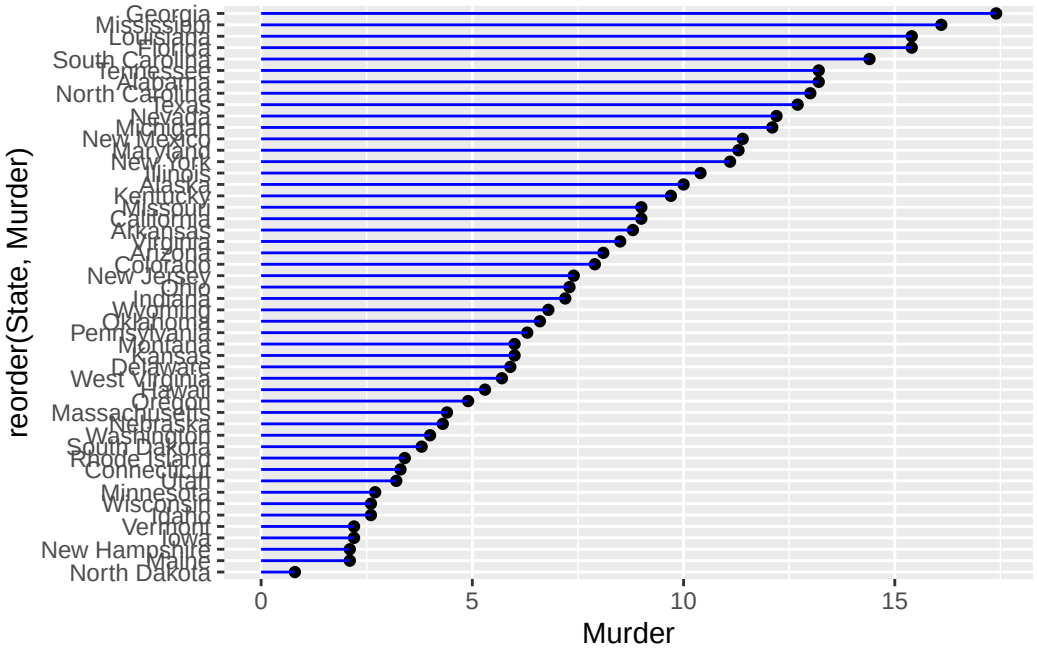
```
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
```

```
USArrests$State <- rownames(USArrests)
ggplot(USArrests) +
  aes(x=reorder(State,Murder), y=Murder) +
  geom_col() +
  coord_flip()
```



Or combining geom_point() and geom_segment()

```
ggplot(USArrests) +
  aes(x=reorder(State,Murder), y=Murder) +
  geom_point() +
  geom_segment(aes(x=State,
                   xend=State,
                   y=0,
                   yend=Murder), color="blue") +
  coord_flip()
```


reorder(State, Murder)
Georgia
Mississippi
Louisiana
Florida
South Carolina
Tennessee
Alabama
North Carolina
Texas
Nevada
Michigan
New Mexico
Maryland
New York
Illinois
Alaska
Kentucky
Missouri
California
Arkansas
Virginia
Arizona
Colorado
New Jersey
Ohio
Indiana
Wyoming
Oklahoma
Pennsylvania
Montana
Kansas
Delaware
West Virginia
Hawaii
Oregon
Massachusetts
Nebraska
Washington
South Dakota
Rhode Island
Connecticut
Utah
Minnesota
Wisconsin
Idaho
Vermont
Iowa
New Hampshire
Maine
North Dakota
0
5
10
15
Murder