Class 5: Data Visualization with GGPlot

Elena

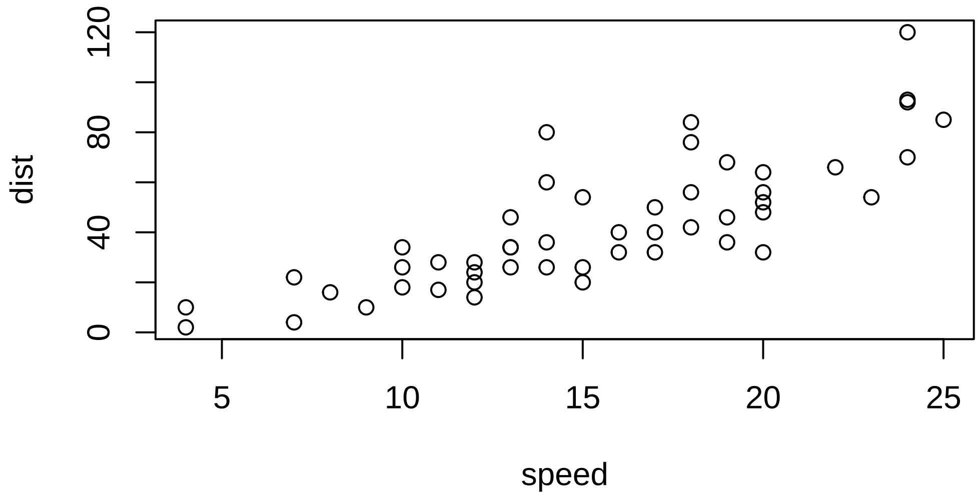
Table of contents

Our First Plot

R has base graphics.

Note: cmd+opt+i to create R code

```
plot(cars)
```



Plotting with ggplot

Note: before using package, need to load using 'library()'.

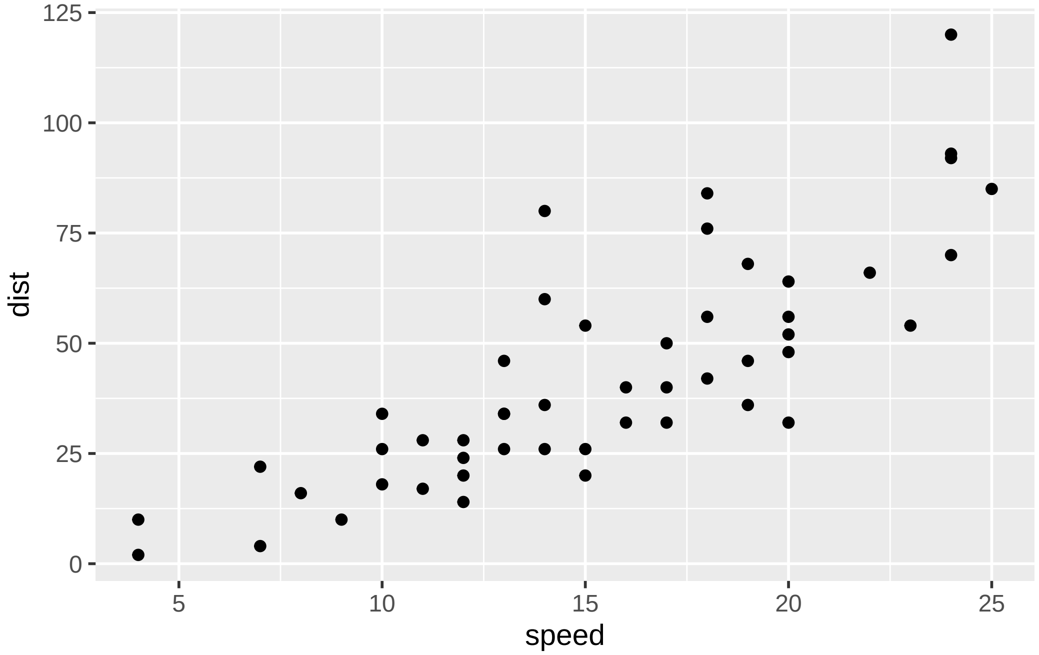
Every ggplot needs 3 layers:

-**Data** (i.e. the dataframe)

-**Aes** (the aesthetic mapping of our data to what we want to plot)

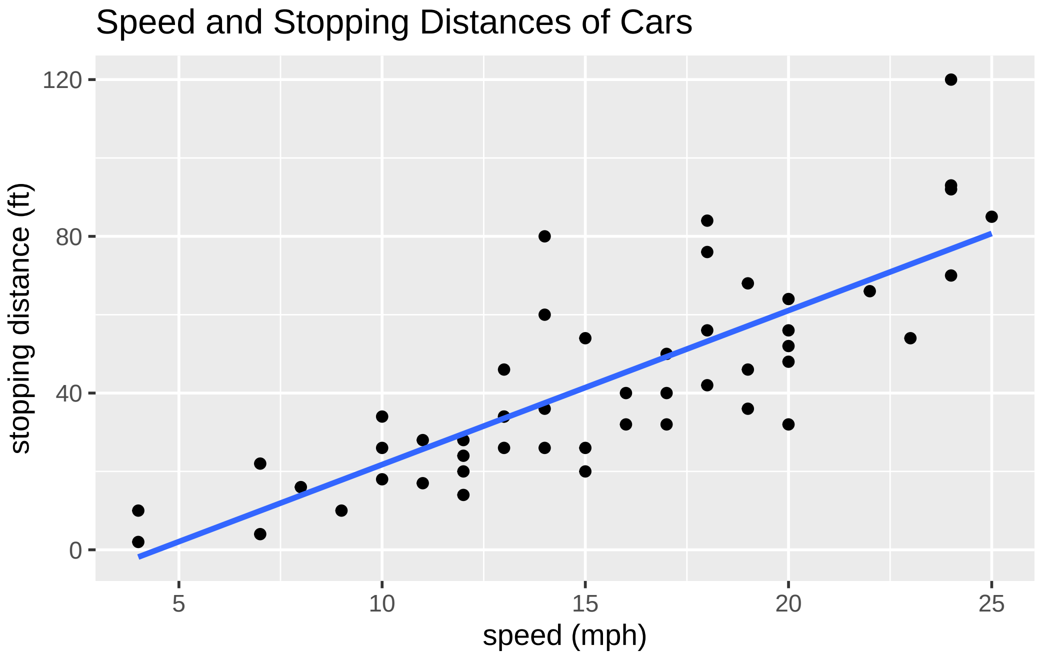
-**Geoms** (how we want to plot)

```
#install.packages("ggplot2")
library(ggplot2)
ggplot(cars) +
  aes(x=speed,
      y=dist) +
  geom_point()
```



```
ggplot(cars) +
  aes(x=speed,
      y=dist) +
  geom_point() +
  geom_smooth(method=lm,se=F) +
  labs(title="Speed and Stopping Distances of Cars",
       x="speed (mph)",
       y="stopping distance (ft)")
```

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



Lab 5 Questions

Q1 For which phases is data visualization important in our scientific workflows?

All of the above

Q2 True or False? The ggplot2 package comes already installed with R?

False

Q3 Which plot types are typically NOT used to compare distributions of numeric variables?

Network graphs

Q4 Which statement about data visualization with ggplot2 is incorrect?

ggplot2 is the only way to create plots in R

Q5 Which geometric layer should be used to create scatter plots in ggplot2?

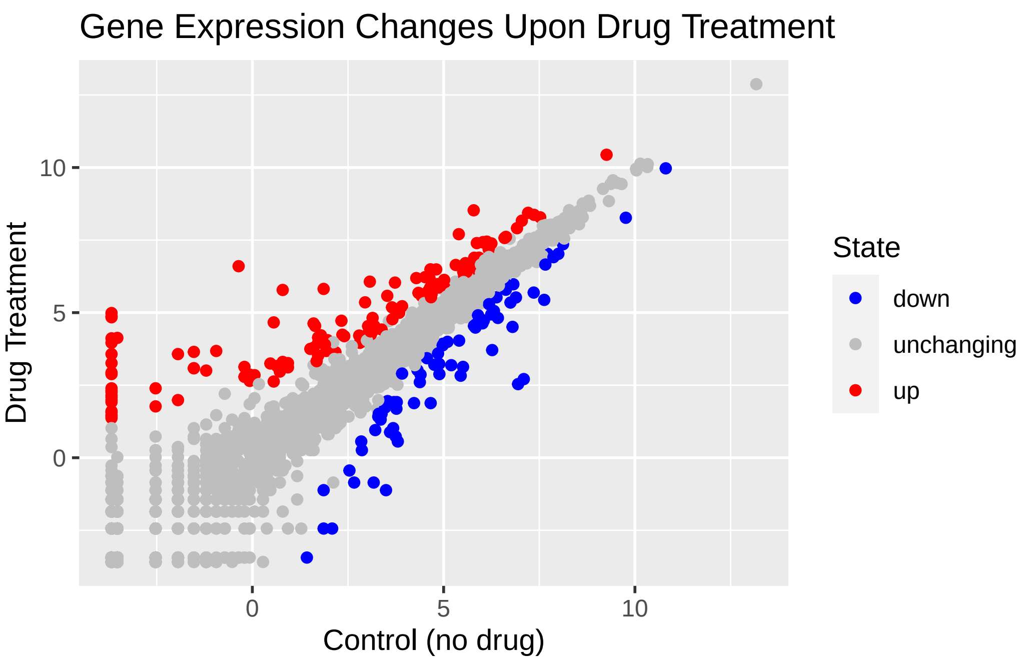
geom_point()

Differential Gene Expression 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
```

```
p <- ggplot(genes) +
  aes(x=Condition1,
      y=Condition2,
      color=State) +
  geom_point()
p + scale_colour_manual(values=c("blue","grey","red")) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
       x="Control (no drug)",
       y="Drug Treatment")
```



```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

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

```

      down unchanging         up 
        72       4997        127 
```

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

```

      down unchanging         up 
      1.39      96.17       2.44 
```

Q1 How many genes

There are 5196 genes in this data set.

Q2 Column names & how many columns

Gene, Condition1, Condition2, State

There are 4 columns.

Q3 How many genes are up

127.

Q4 Fraction of total genes that are up

2.44% genes are up.

Optional Extension

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

gapminder <- read.delim(url)

head(gapminder)
```

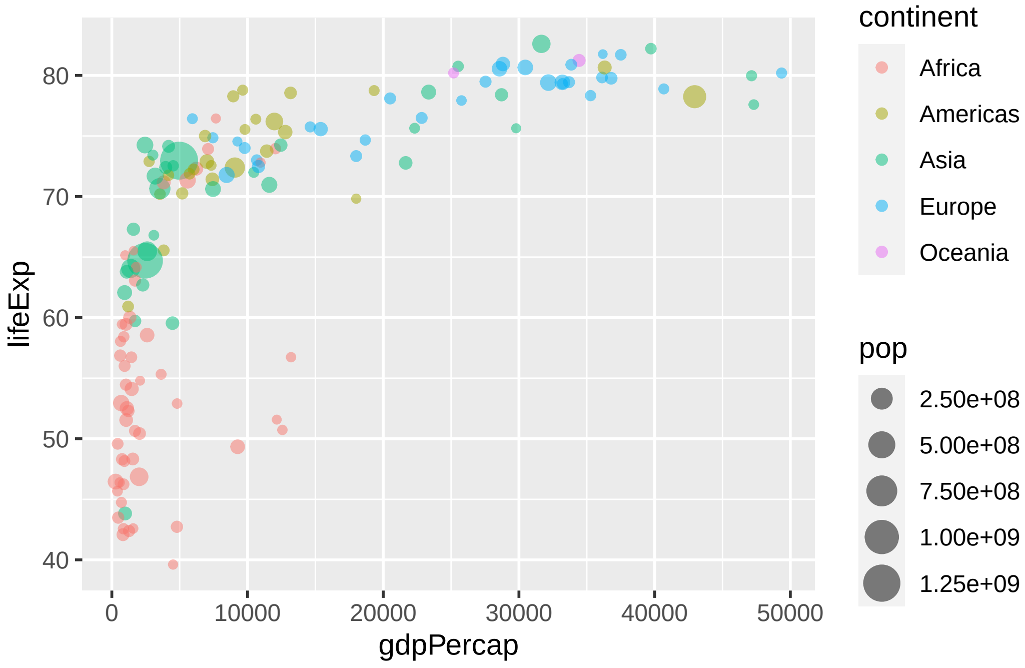
```
      country continent year lifeExp      pop gdpPercap
1 Afghanistan      Asia 1952  28.801  8425333  779.4453
2 Afghanistan      Asia 1957  30.332  9240934  820.8530
3 Afghanistan      Asia 1962  31.997 10267083  853.1007
4 Afghanistan      Asia 1967  34.020 11537966  836.1971
5 Afghanistan      Asia 1972  36.088 13079460  739.9811
6 Afghanistan      Asia 1977  38.438 14880372  786.1134
```

```r
##install.packages("dplyr")
library(dplyr)
```

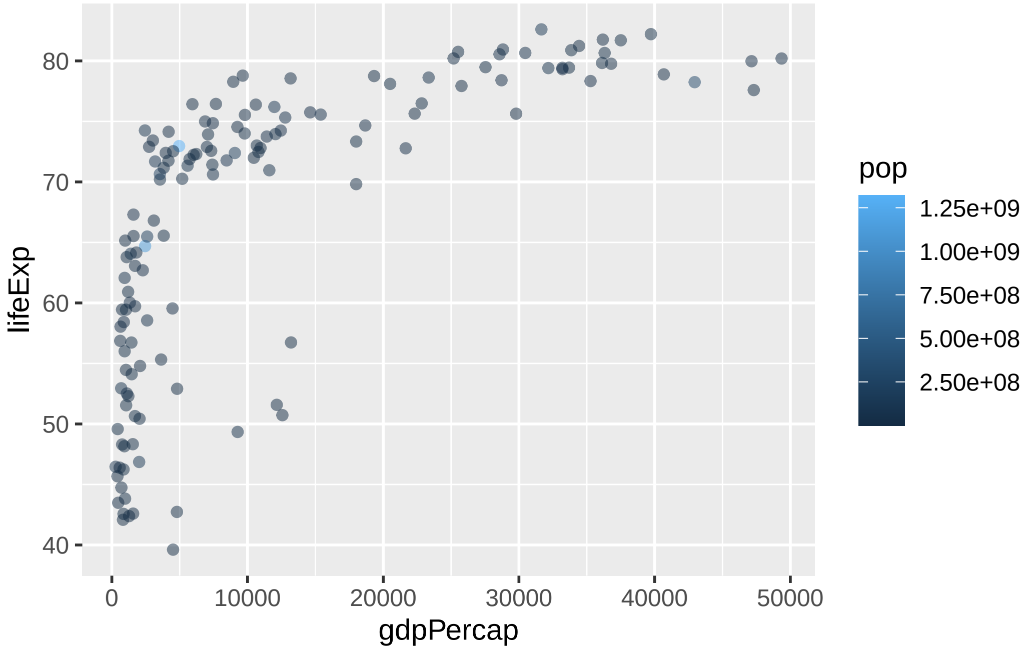
```
Attaching package: 'dplyr'

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

    filter, lag

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

    intersect, setdiff, setequal, union
```

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

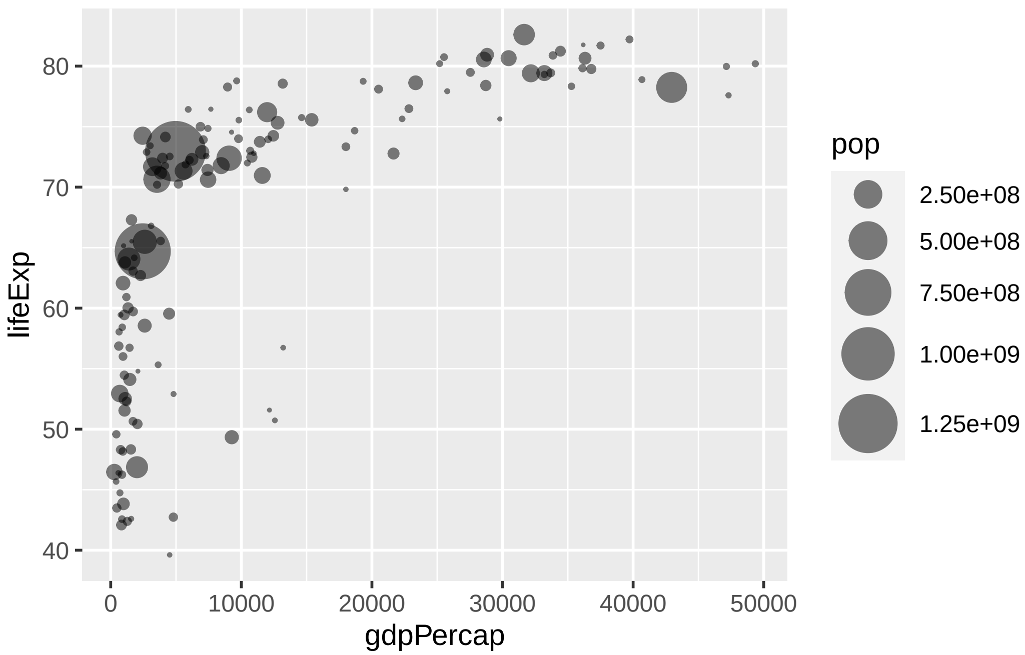
```r
ggplot(gapminder_2007) +
  aes(x=gdpPercap,
      y=lifeExp,
      colour=continent,
      size=pop) +
  geom_point(alpha=0.5)
```



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



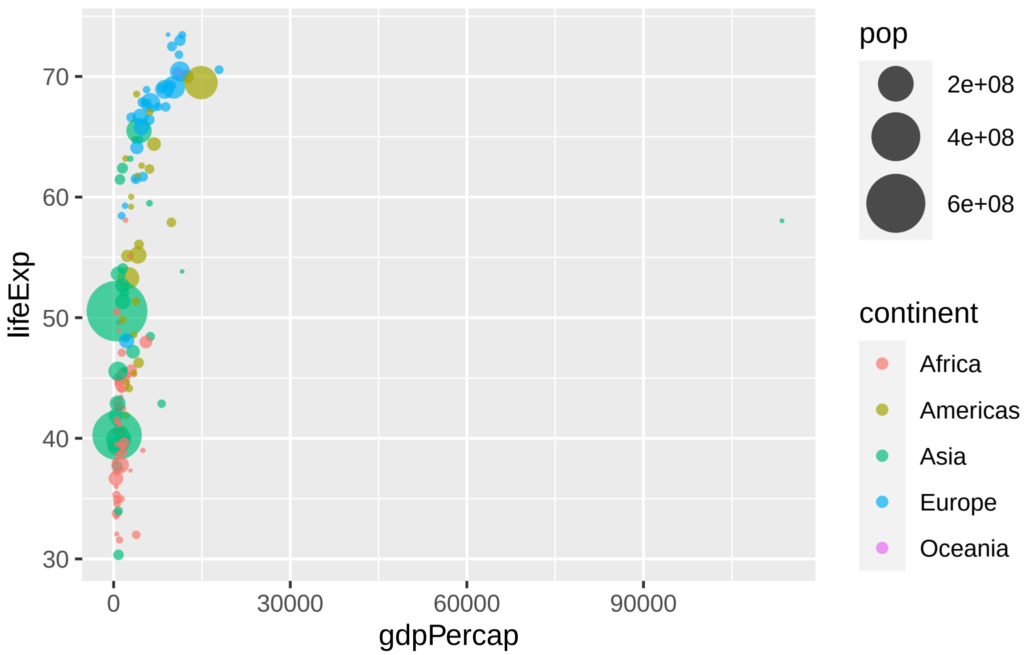
```
ggplot(gapminder_2007) +
  aes(x=gdpPercap,
      y=lifeExp,
      size=pop) +
  geom_point(alpha=0.5) +
  scale_size_area(max_size = 10)
```



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

```
      country continent year lifeExp      pop gdpPercap
1 Afghanistan      Asia 1957  30.332  9240934   820.853
2     Albania    Europe 1957  59.280  1476505  1942.284
3     Algeria    Africa 1957  45.685 10270856  3013.976
4      Angola    Africa 1957  31.999  4561361  3827.940
5   Argentina  Americas 1957  64.399 19610538  6856.856
6   Australia   Oceania 1957  70.330  9712569 10949.650
```

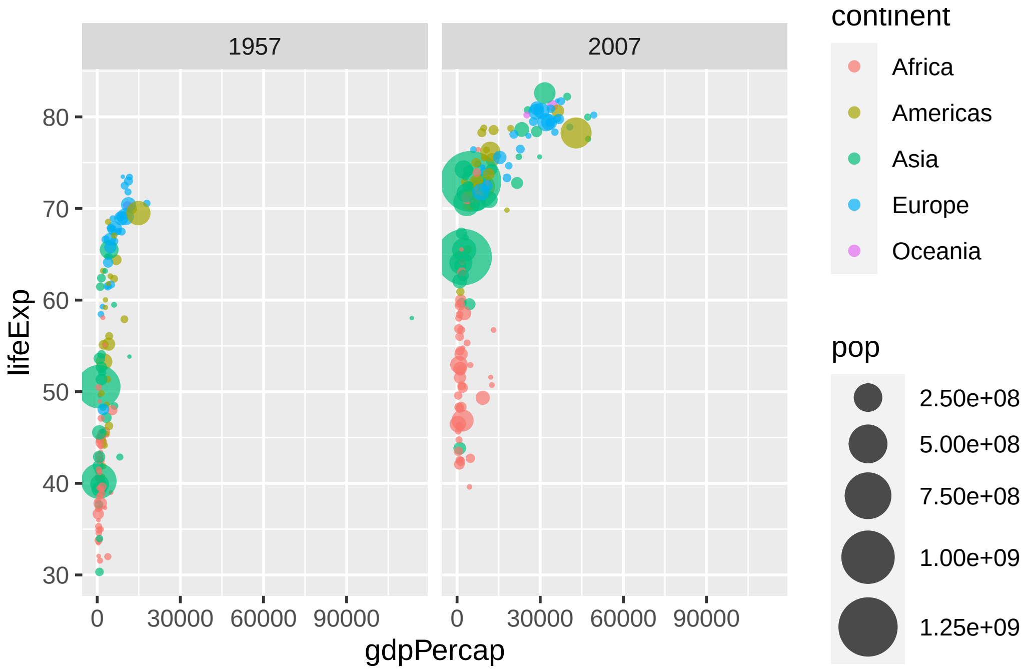
```
ggplot(gapminder_1957) +
  aes(x=gdpPercap,
      y=lifeExp,
      colour=continent,
      size=pop) +
  geom_point(alpha=0.7) +
  scale_size_area(max_size = 10)
```



```
gapminder_1957_2007 <- gapminder %>% filter(year==1957 | year==2007)
head(gapminder_1957_2007)
```

```
      country continent year lifeExp      pop gdpPercap
1 Afghanistan      Asia 1957  30.332  9240934  820.8530
2 Afghanistan      Asia 2007  43.828 31889923  974.5803
3     Albania    Europe 1957  59.280  1476505 1942.2842
4     Albania    Europe 2007  76.423  3600523 5937.0295
5     Algeria    Africa 1957  45.685 10270856 3013.9760
6     Algeria    Africa 2007  72.301 33333216 6223.3675
```

```
ggplot(gapminder_1957_2007) +
  aes(x=gdpPercap,
      y=lifeExp,
      colour=continent,
      size=pop) +
  geom_point(alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```

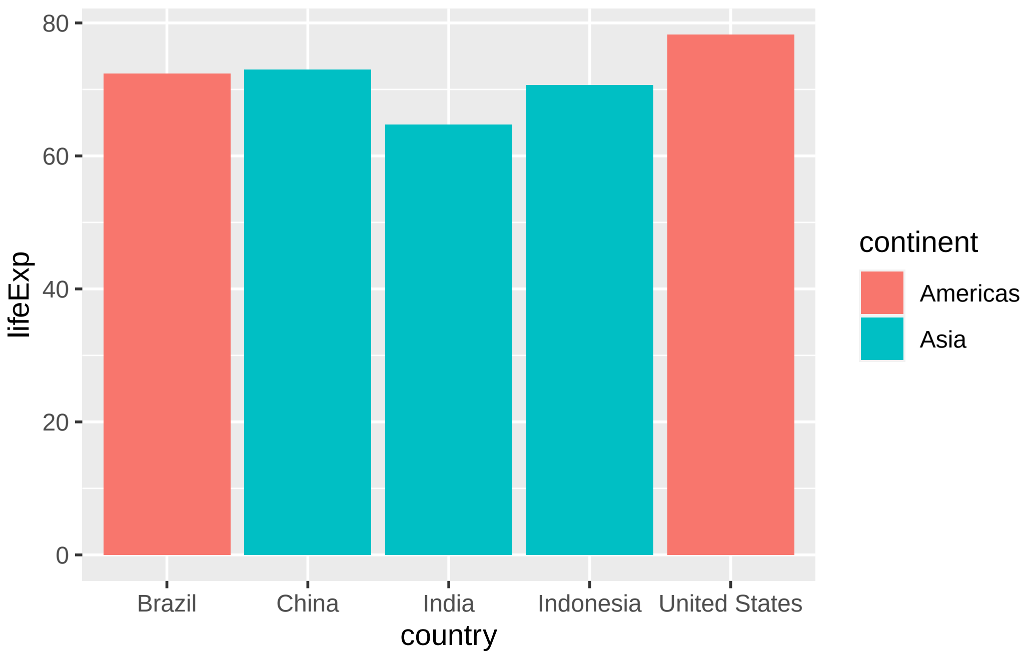


Bar Plots

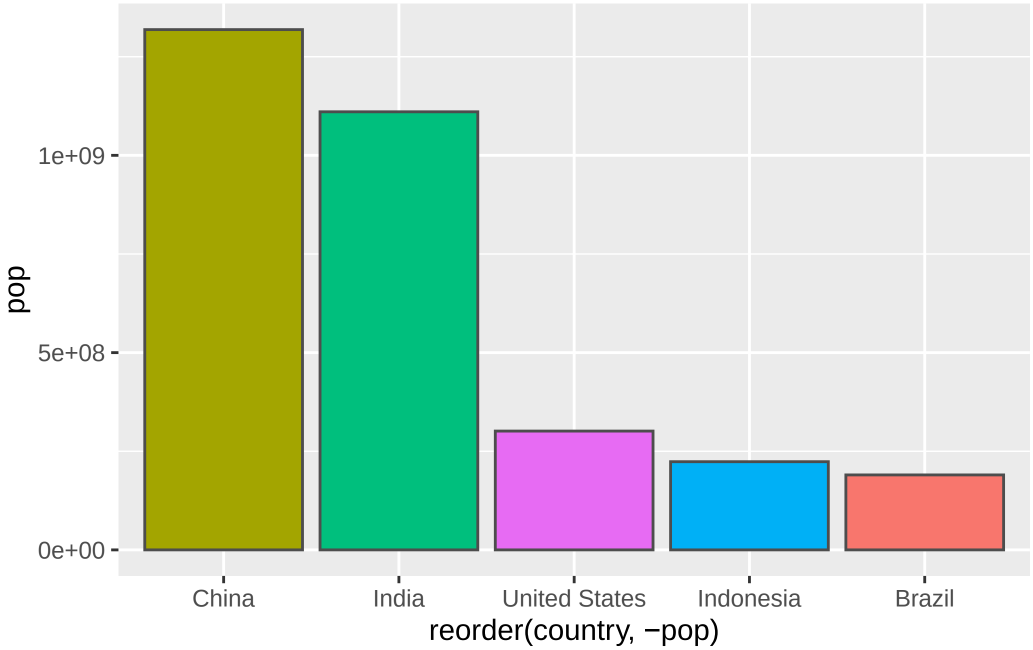
```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

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
```

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



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



Playing with Plotly

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

#library(plotly)

#example dataset

#set.seed(100)
#d <- diamonds[sample(nrow(diamonds), 1000), ]

#p <- ggplot(data = d, aes(x = carat, y = price)) +
#  geom_point(aes(text = paste("Clarity:", clarity)), size = 4) +
#  geom_smooth(aes(colour = cut, fill = cut)) + facet_wrap(~ cut)

#ggplotly(p)
```