Class 05: Data Visualization with GGPLOT

Jenny Zhou

Base R graphics vs ggplot2

There are many graphic systems available in R, including so-called "base" R graphics and the very popular **ggplot2** package.

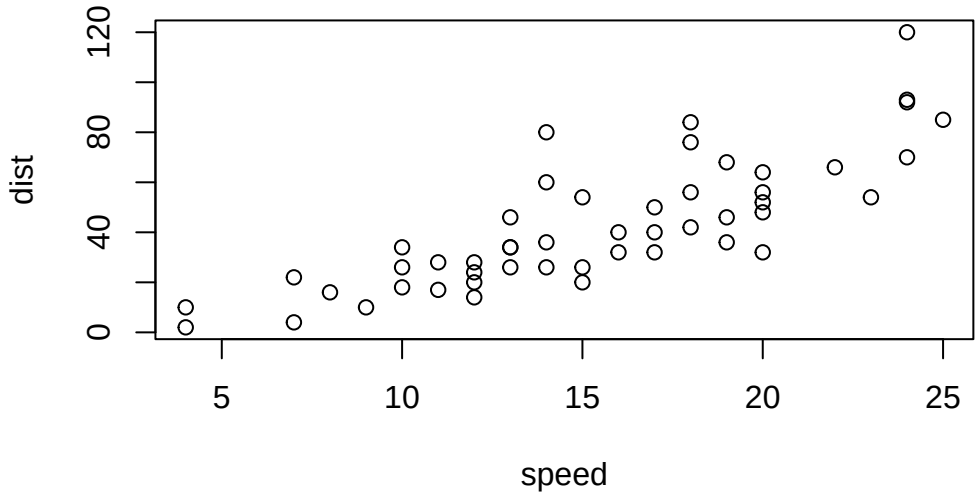
To compare these, lets play with the inbuilt `cars` data set.

```
head(cars)
```

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

To use "base" R I can simply call the `plot()` function:

```
plot(cars)
```



To use `ggplot2`, I first need to install it with the function `install.packages("ggplot2")`.

I will run this in my R console (i.e. the R brain) as I do not want to re-install it every time I render my report.

The main function in this package is called `ggplot()`

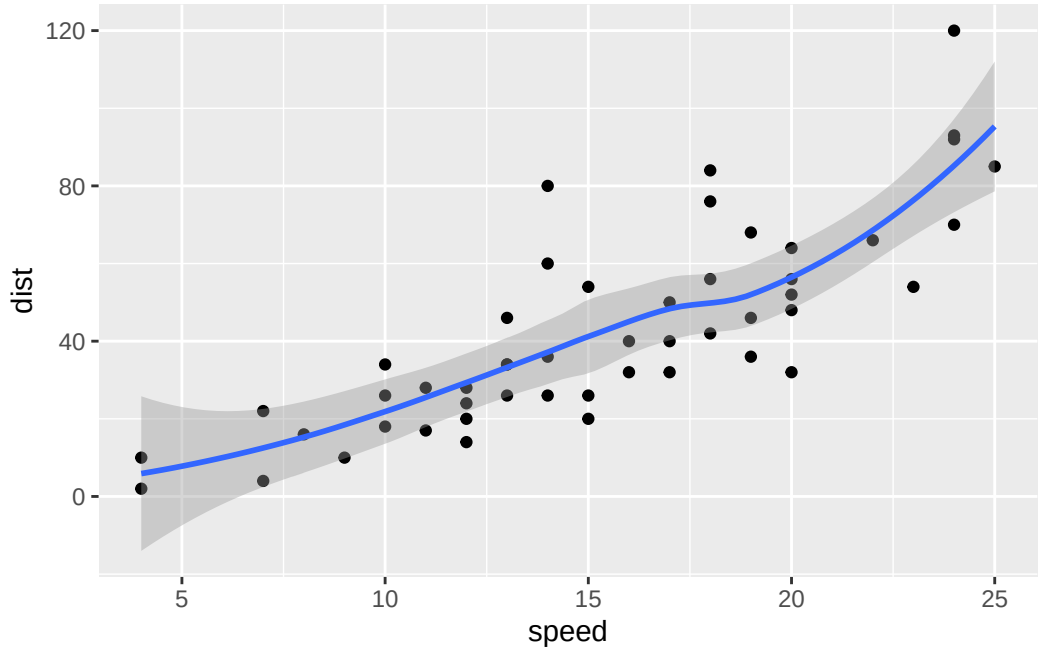
```
library(ggplot2)
ggplot()
```

To make a figure with ggplot, I need at least 3 things:

- **data** (i.e. what I want to plot)
- **aesthetics** (mapping of the data to the plot I want)
- **geometry** (How I want to plot the data)

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

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



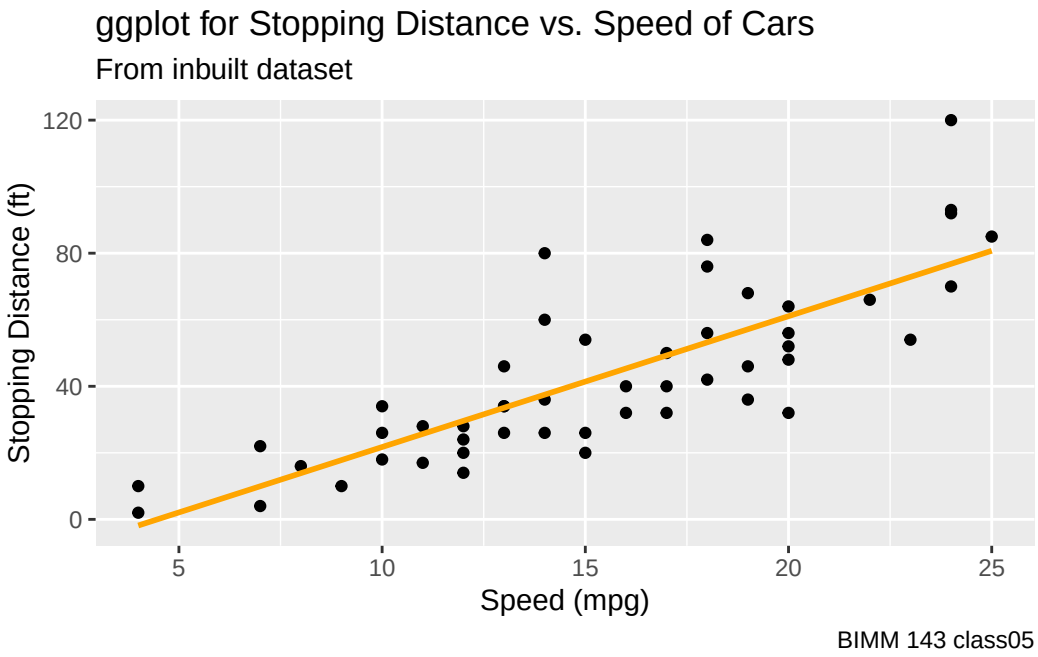
If want want to add more things I can just keep adding layers.

GGplot is much more verbose than base R plots but it has a consistent layer system that I can use to make just about any plot.

Make a plot with linear straight line:

```
ggplot(cars) +
  aes(speed,dist) +
  geom_point() +
  geom_smooth(method='lm', se=FALSE, color='orange') +
  labs(title = 'ggplot for Stopping Distance vs. Speed of Cars',
       subtitle = 'From inbuilt dataset',
       caption = 'BIMM 143 class05',
       x= 'Speed (mpg)',
       y='Stopping Distance (ft)') +
  theme_update()
```

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



Creating more complicated plot

Let's plot some gene expression data.

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes,10)
```

```
         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
7       ABCA7  3.4484220  3.8266509 unchanging
8   ABCA9-AS1 -3.6808610 -3.5921390 unchanging
9      ABCC11 -3.5288580 -1.8551732 unchanging
10      ABCC3  0.9305738  3.2603040         up
```

Q. How many genes are in the data set?

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

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

```
      down unchanging         up
       1.4       96.0        2.4
```

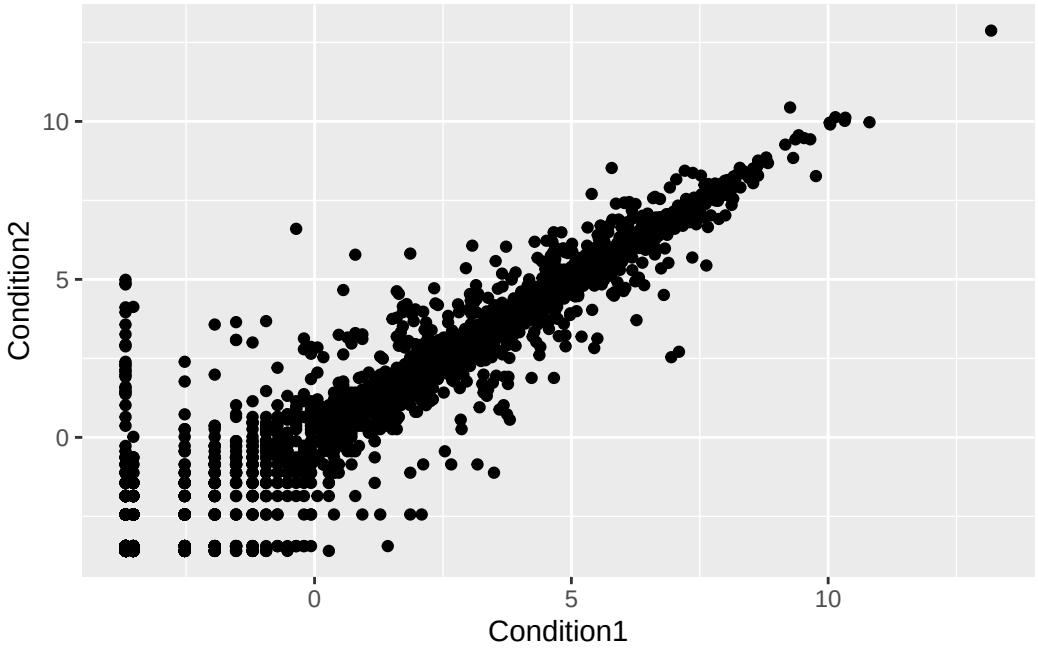
Q. How can we summarize that last column (State)?

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

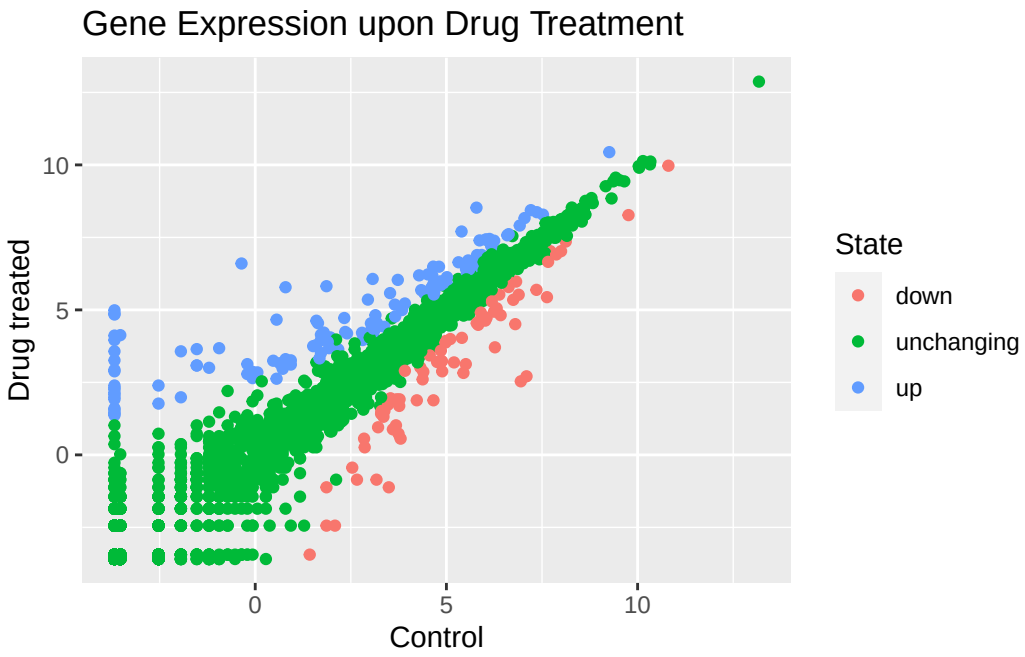
```
      down unchanging         up
        72       4997        127
```

Plot genes dataset

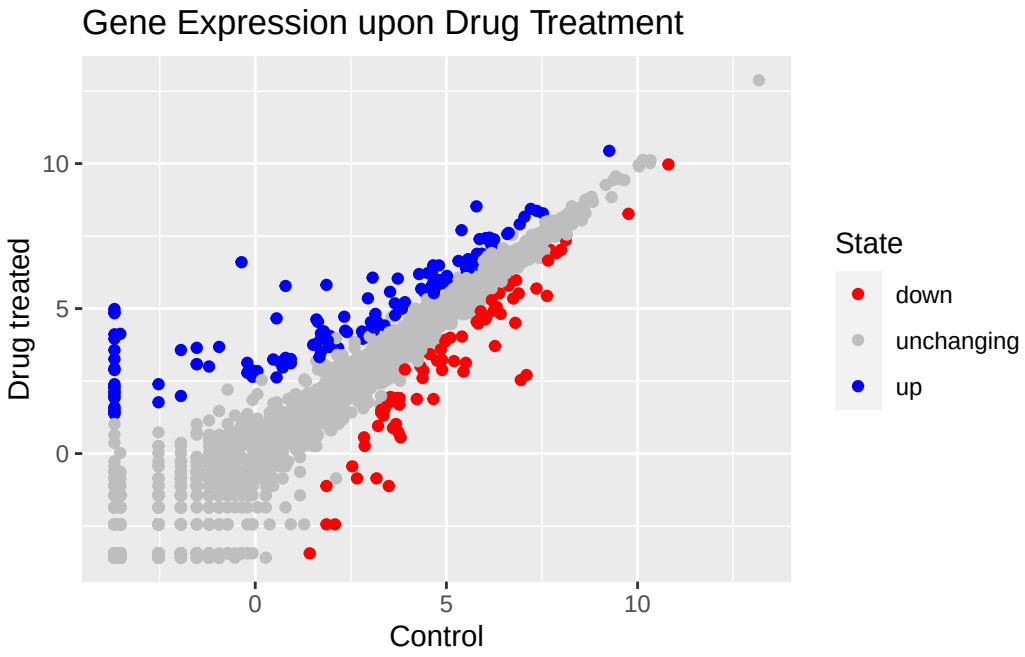
```
ggplot(genes) +
  aes(Condition1, Condition2) +
  geom_point()
```




```
p <- ggplot(genes) +
  aes (Condition1, Condition2, col=State) +
  geom_point() +
  labs(title = 'Gene Expression upon Drug Treatment',
       x='Control',
       y= 'Drug treated')

p
```




```
p + scale_color_manual(values = c("red", "grey", "blue"))
```



Going further

Here I read a slightly larger data set.

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

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

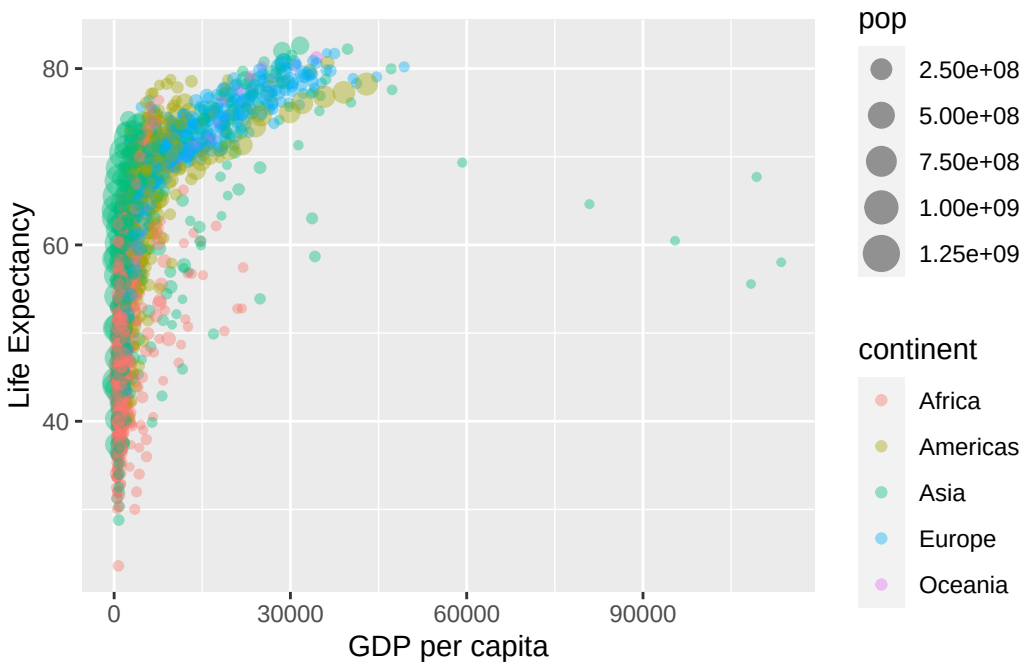
```
      country continent year lifeExp      pop gdpPercap
1 Afghanistan      Asia 1952  28.801  8425333  779.4453
2 Afghanistan      Asia 1957  30.332  9240934  820.8530
3 Afghanistan      Asia 1962  31.997 10267083  853.1007
4 Afghanistan      Asia 1967  34.020 11537966  836.1971
5 Afghanistan   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

```
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
```

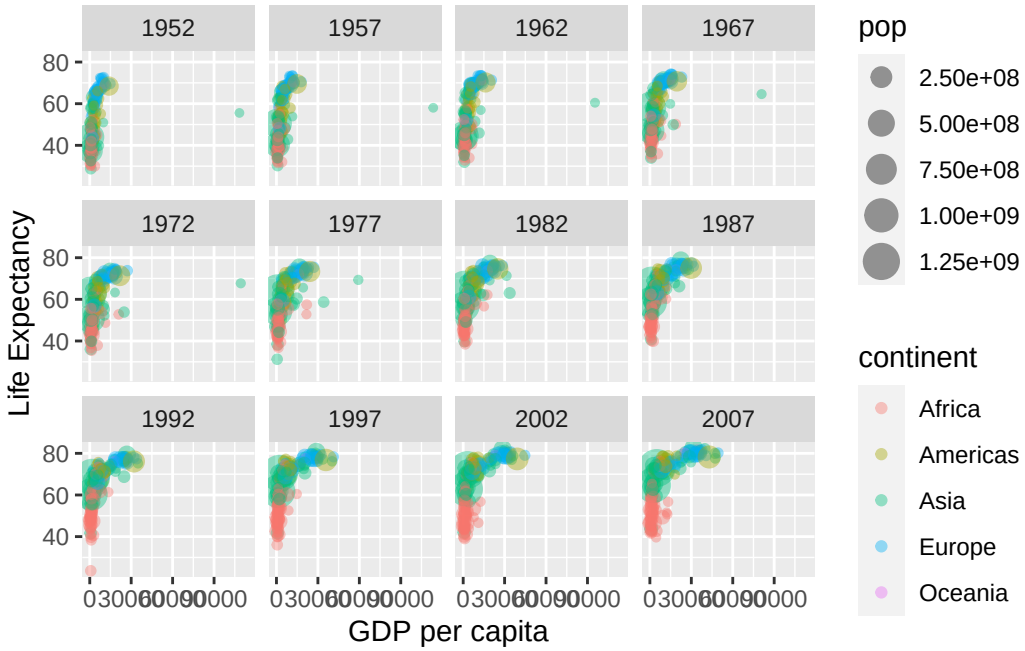
Make a basic scatter plot

```
ggplot(gapminder) +
  aes(gdpPercap, lifeExp, col=continent, size=pop) +
  geom_point(alpha = 0.4) +
  labs(x='GDP per capita',
      y='Life Expectancy')
```



A very useful layer to add sometimes is for 'facetting'.

```
ggplot(gapminder) +
  aes(gdpPercap, lifeExp, col=continent, size=pop) +
  geom_point(alpha = 0.4) +
  labs(x='GDP per capita',
       y='Life Expectancy') +
  facet_wrap(~year)
```



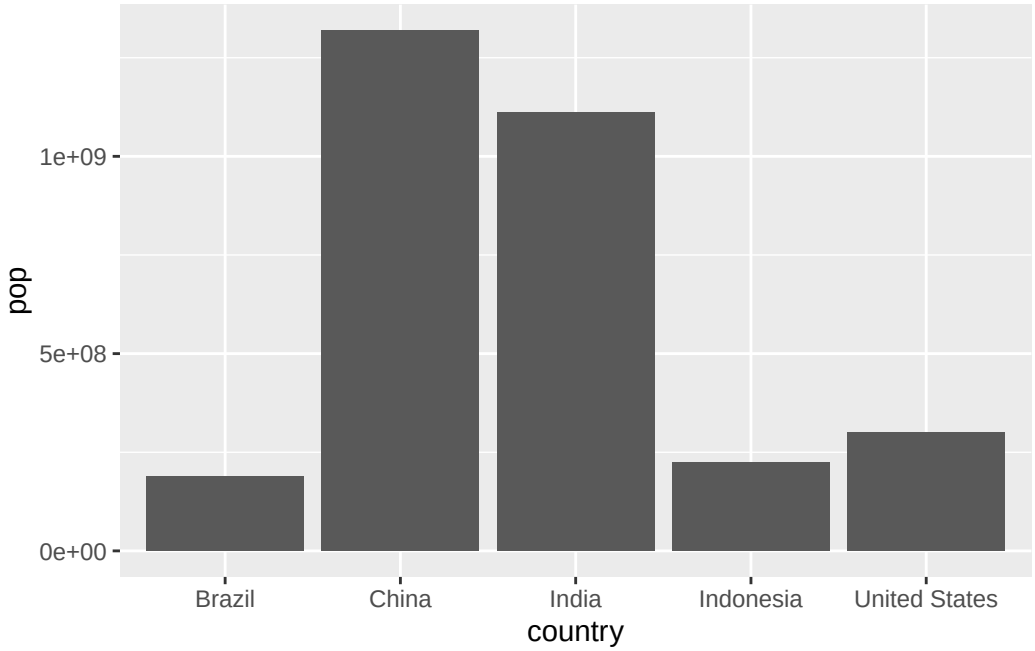
Bar charts

data:

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

gapminder_top5
```

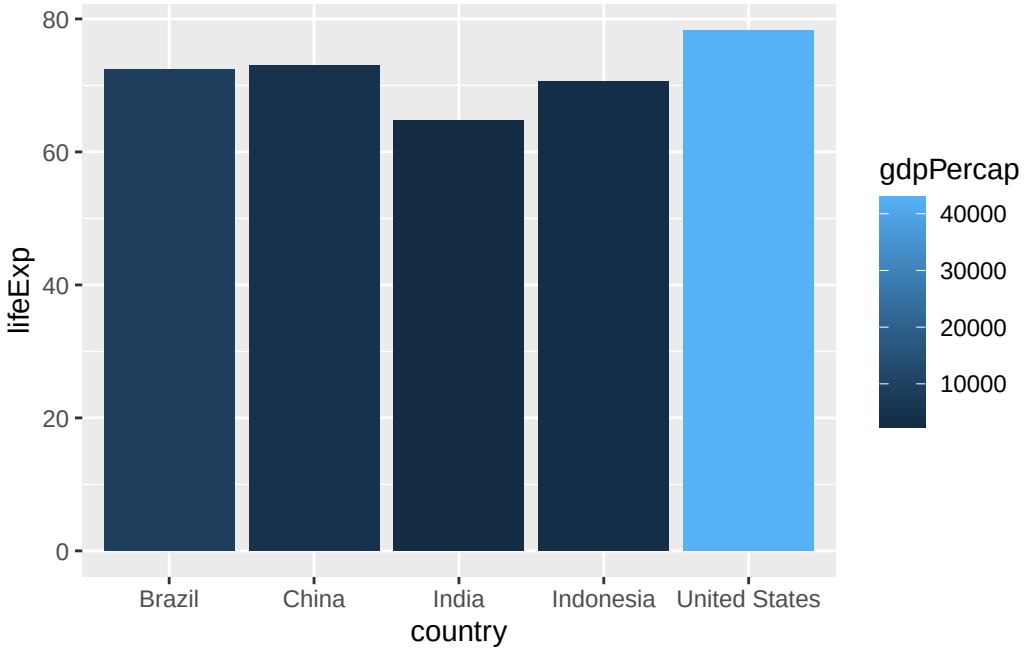
```
        country continent year lifeExp        pop gdpPercap
1         China      Asia 2007  72.961 1318683096  4959.115
2         India      Asia 2007  64.698 1110396331  2452.210
3 United States  Americas 2007  78.242  301139947 42951.653
4     Indonesia      Asia 2007  70.650  223547000  3540.652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

```
ggplot(gapminder_top5) + geom_col(aes(country,pop))
```



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

```
ggplot(gapminder_top5) +
  aes(country,lifeExp, fill = gdpPercap) +
  geom_col()
```


80
60
40
20
0
lifeExp
Brazil
China
India
Indonesia
United States
country
gdpPercap
40000
30000
20000
10000