

Class 5: Data visualization with ggplot2

Xiaoxuan Teng (PID: A69028742)

2024-01-28

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

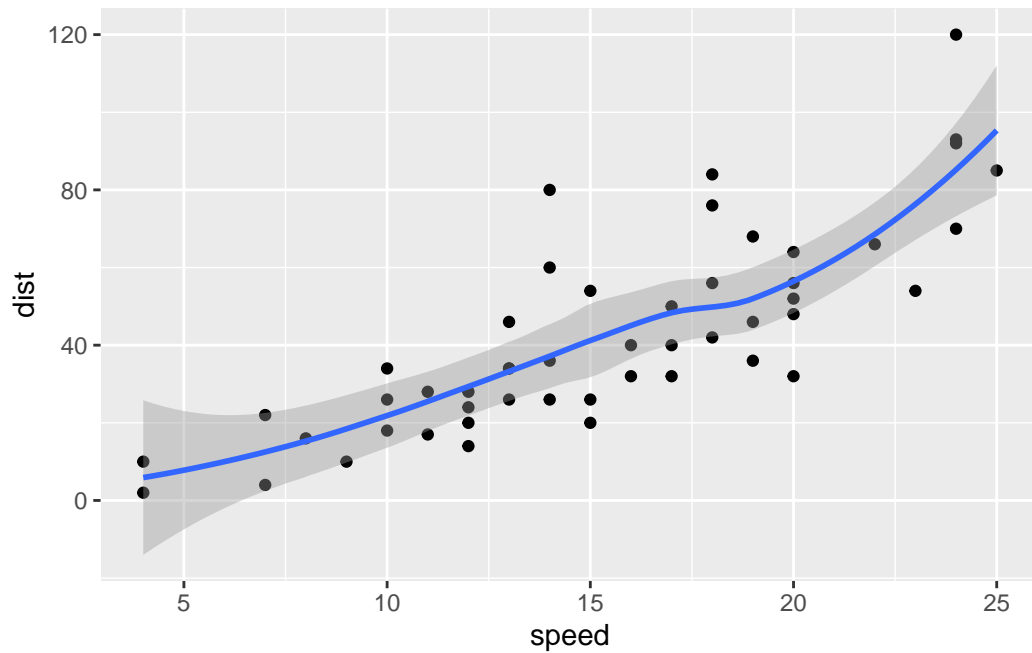
Section 6. Creating Scatter Plots

Specifying a geom layer with geom_point()

Q. In your own RStudio can you add a trend line layer to help show the relationship between the plot variables with the geom_smooth() function?

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

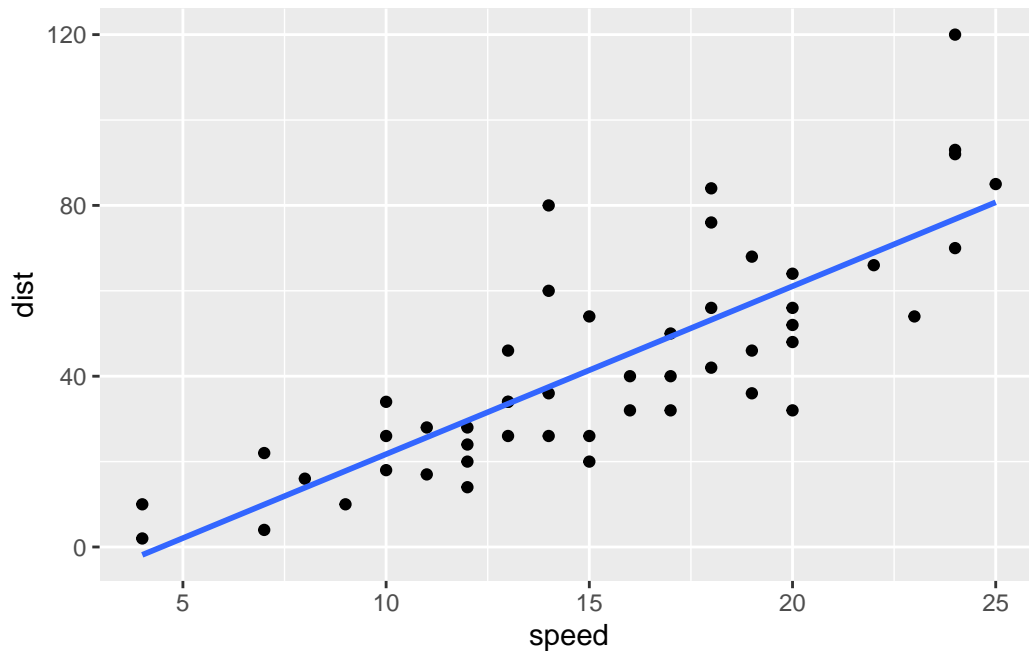
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'



Q. Argue with `geom_smooth()` to add a straight line from a linear model without the shaded standard error region?

```
# set the method to "linear model", and don't show the confidence interval (se = FALSE)
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

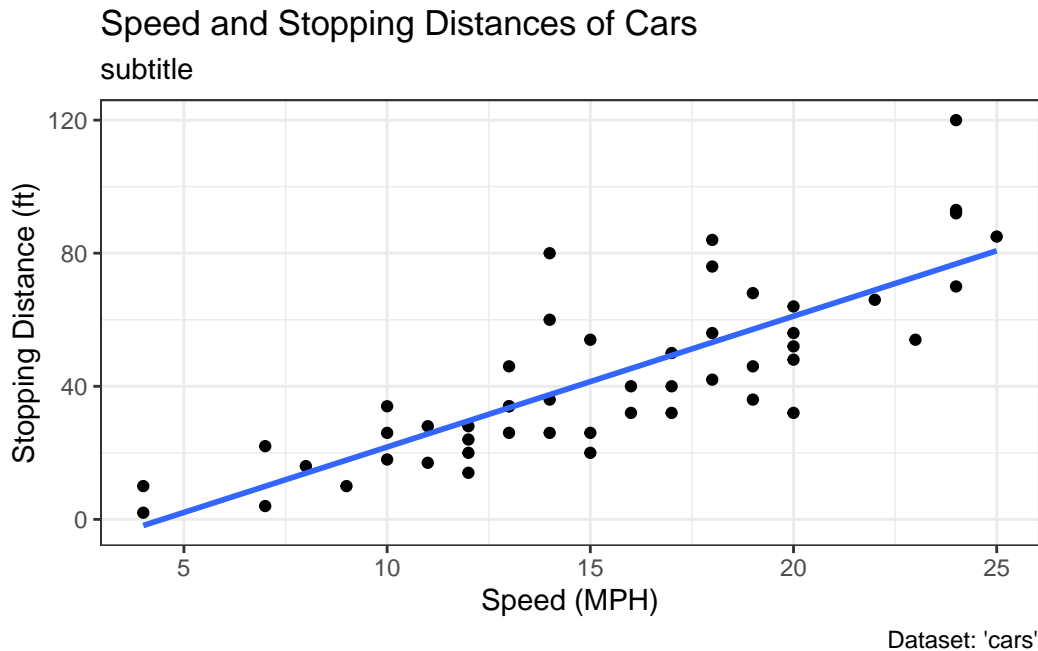
`geom_smooth()` using formula = 'y ~ x'



Q. Can you finish this plot by adding various label annotations with the `labs()` function and changing the plot look to a more conservative “black & white” theme by adding the `theme_bw()` function:

```
# Add label annotations by labs().
# theme_bw(): build-in theme, white background and thin grey grid lines
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)",
       subtitle = "subtitle",
       caption="Dataset: 'cars'") +
  theme_bw()
```

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



Adding more plot aesthetics through aes()

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
# read.delim() function is used to read delimited text files
genes <- read.delim(url)
# head() function is used to display the first n rows present in the input data frame.
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

```
# colnames() function returns or sets the names of the columns in a data frame.
colnames(genes)
```

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

```
# nrow()/ncol() function returns the number of rows/columns in a data frame.  
nrow(genes)
```

```
[1] 5196
```

```
ncol(genes)
```

```
[1] 4
```

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

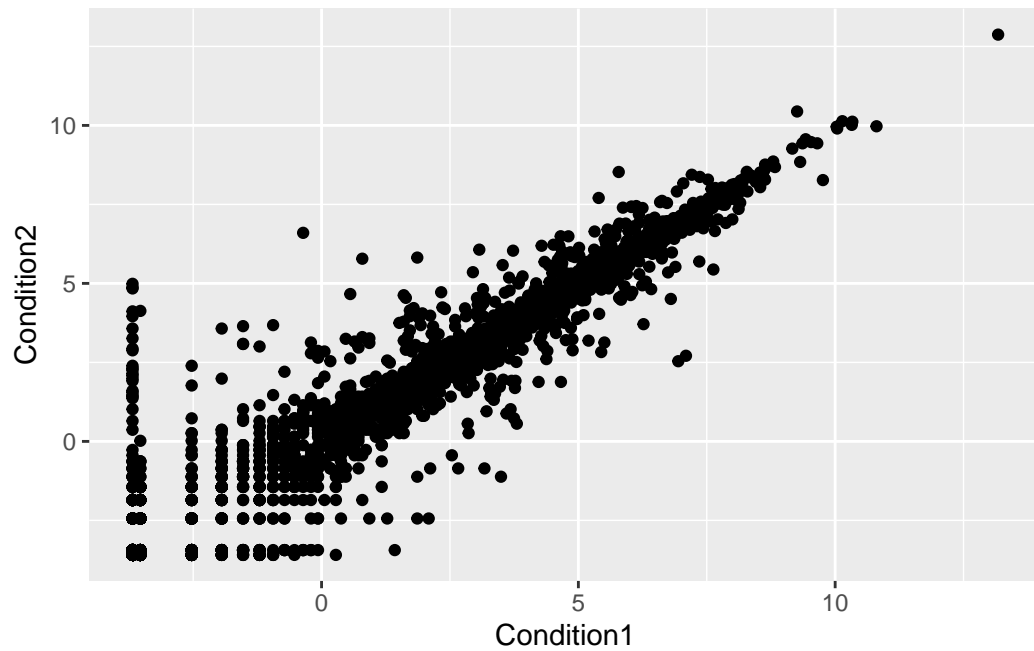
down	unchanging	up
72	4997	127

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

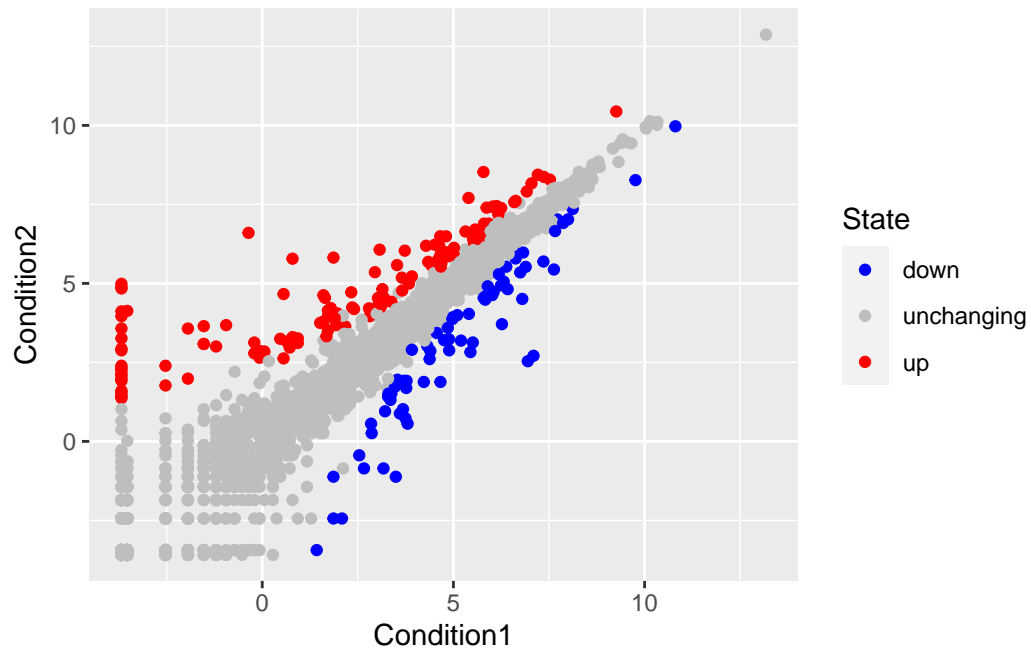
down	unchanging	up
1.39	96.17	2.44

Q. Complete the code below to produce the following plot

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



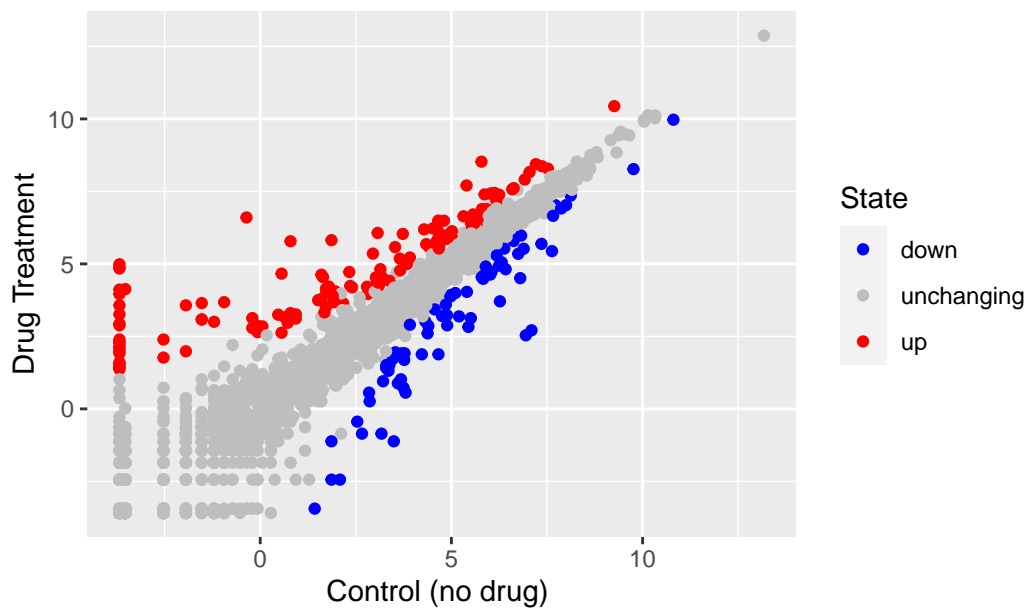
```
p <- ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) +  
  geom_point()  
# scale_colour_manual(): Create your own discrete scale  
p + scale_colour_manual( values=c("blue","gray","red") )
```



Q. Nice, now add some plot annotations to the `p` object with the `labs()` function so your plot looks like the following:

```
# Add label annotations by labs().
p2 <- p + scale_colour_manual(values=c("blue","gray","red")) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
       x="Control (no drug) ",
       y="Drug Treatment")
p2
```

Gene Expression Changes Upon Drug Treatment



```
# plotly library makes interactive graphs
library(plotly)
# ggplotly(p2)
```

Section 7. Going Further

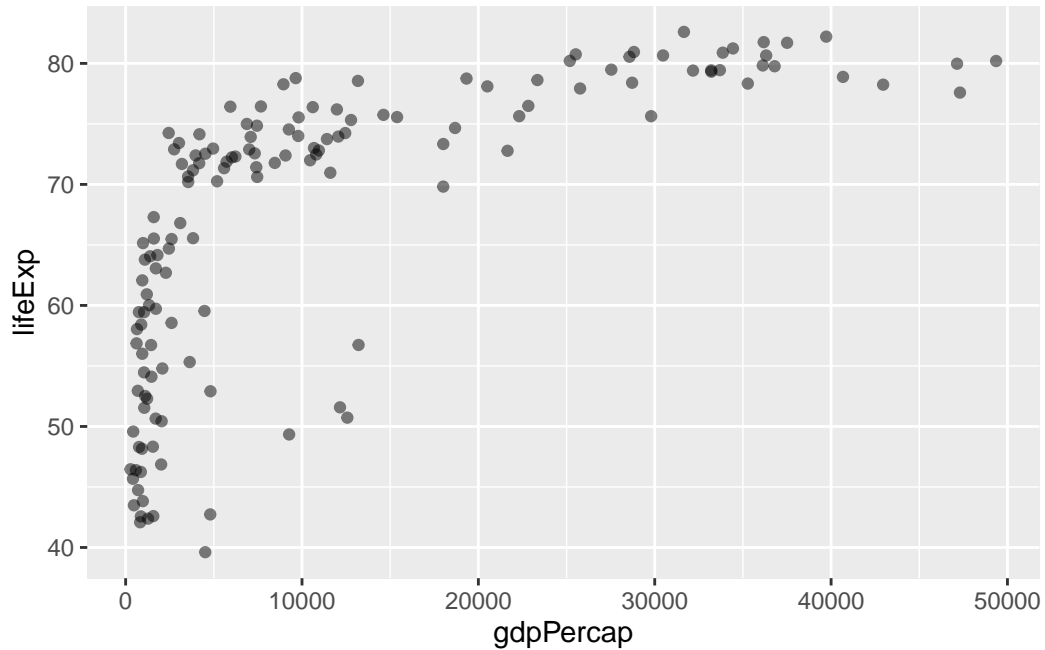
```
# install.packages("gapminder")
# install.packages("dplyr")
library(gapminder)
library(dplyr)

gapminder_2007 <- gapminder %>% filter(year==2007)
head(gapminder_2007, 2)
```

```
# A tibble: 2 x 6
  country    continent  year lifeExp      pop gdpPercap
  <fct>      <fct>    <int> <dbl>    <int>    <dbl>
1 Afghanistan Asia      2007  43.8 31889923    975.
2 Albania    Europe    2007  76.4  3600523   5937.
```

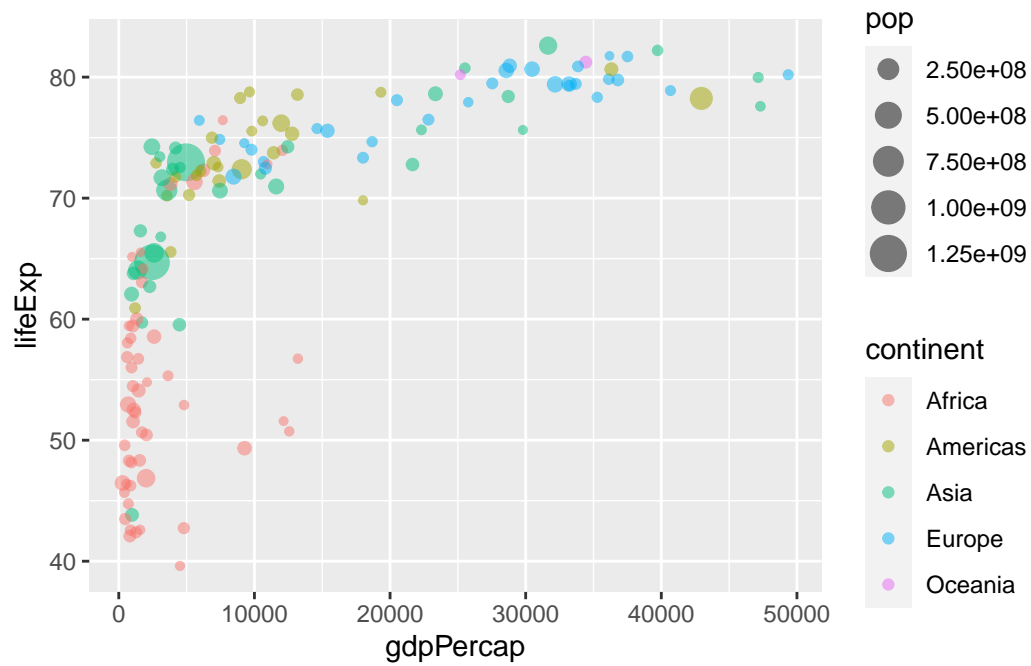

Q. Complete the code below to produce a first basic scatter plot of this gapminder_2007 dataset:

```
ggplot(gapminder_2007) +  
  aes(x=gdpPerCap, y=lifeExp) +  
  geom_point(alpha=0.5)
```



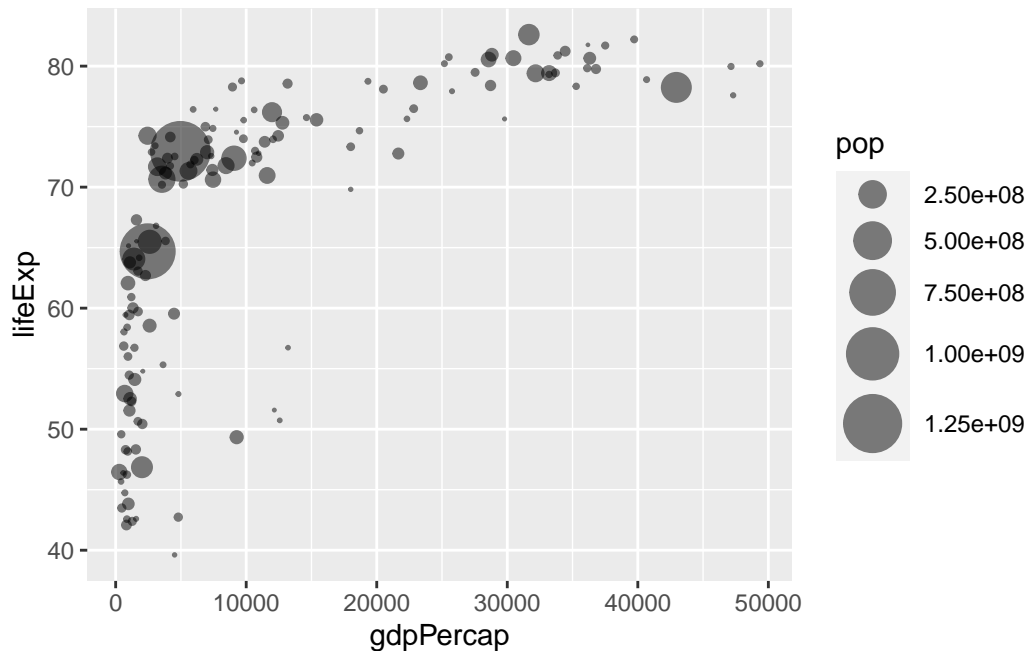
Adding more variables to aes()

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



Adjusting point size

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

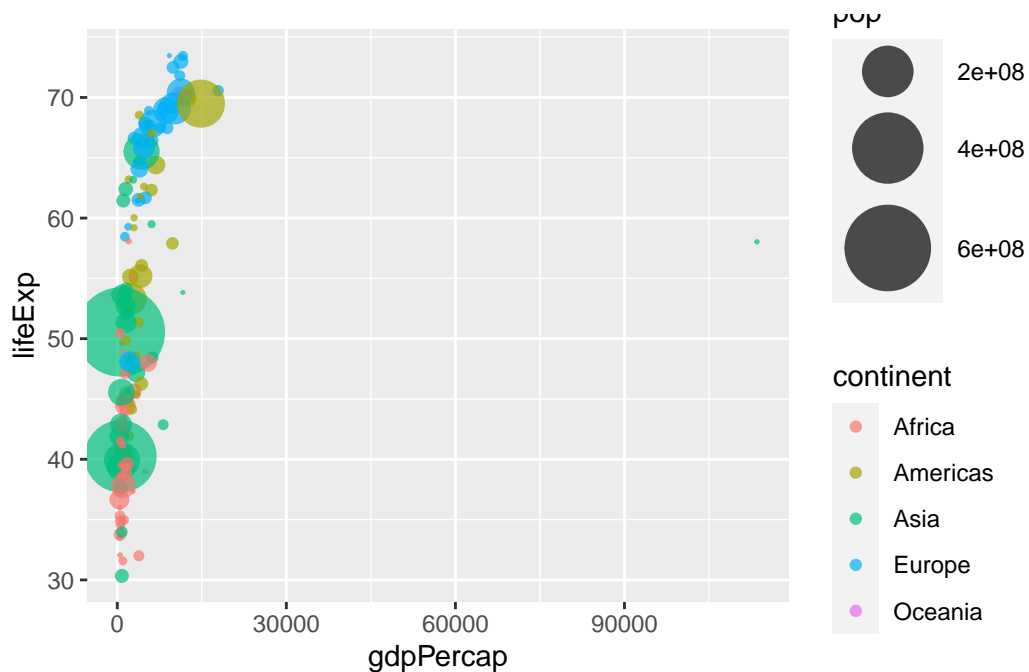


Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

```
gapminder_1957 <- gapminder %>% filter(year==1957)
head(gapminder_1957, 2)
```

```
# A tibble: 2 x 6
  country    continent  year lifeExp    pop gdpPercap
<fct>      <fct>    <int> <dbl>   <int>   <dbl>
1 Afghanistan Asia      1957  30.3  9240934    821.
2 Albania    Europe    1957  59.3  1476505   1942.
```

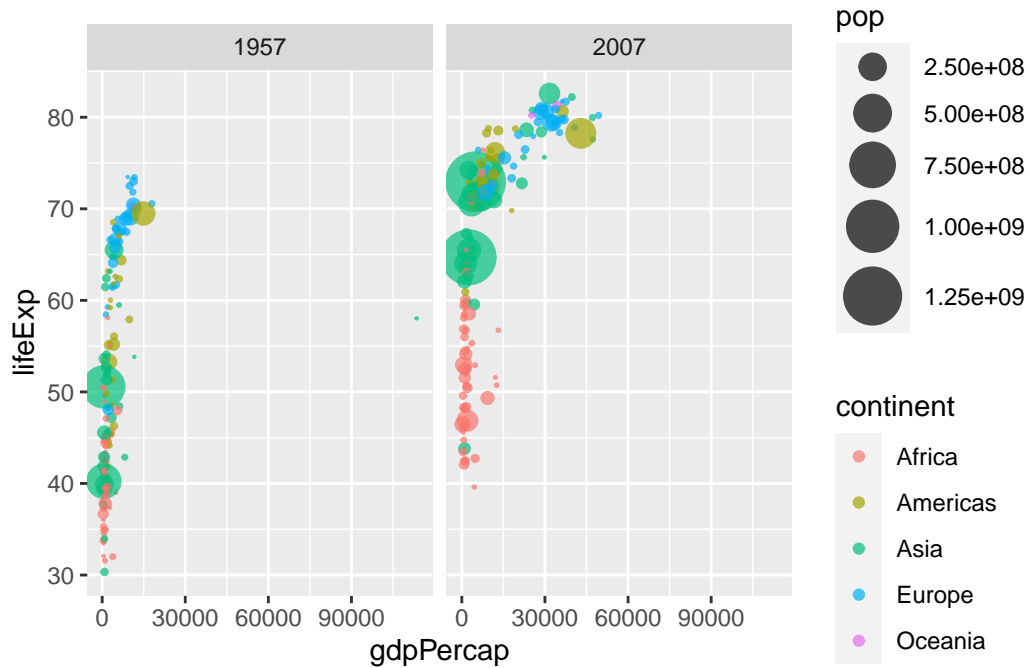
```
ggplot(gapminder_1957, aes(gdpPercap, lifeExp, color = continent, size = pop)) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 15)
```



Q. Do the same steps above but include 1957 and 2007 in your input dataset for `ggplot()`. You should now include the layer `facet_wrap(~year)` to produce the following plot:

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

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



8. OPTIONAL: Bar Charts

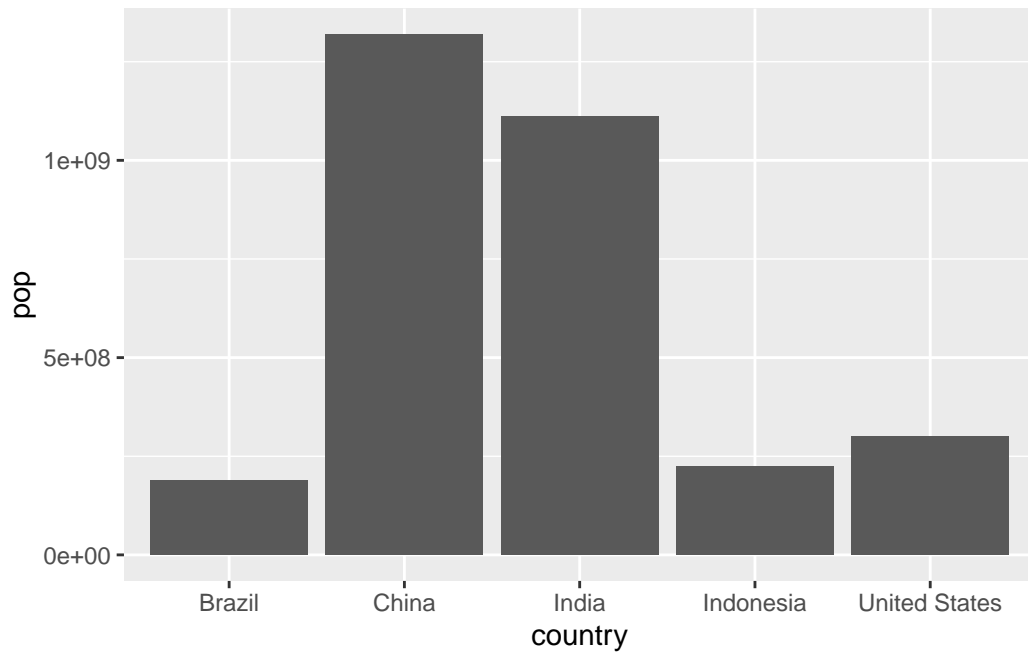
```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)
```

```
gapminder_top5
```

A tibble: 5 x 6

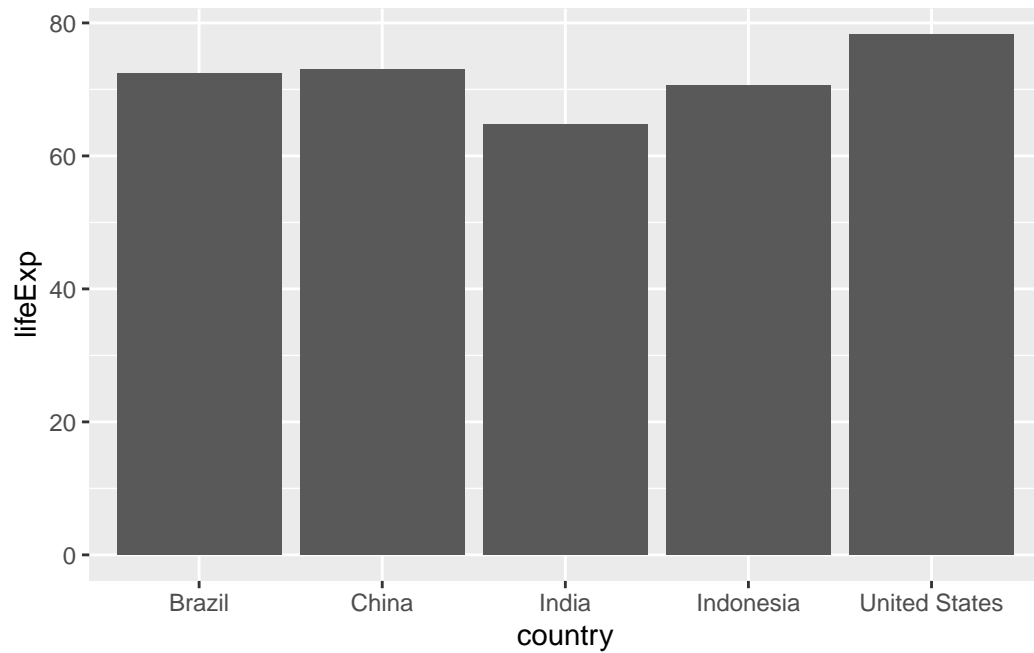
	country	continent	year	lifeExp	pop	gdpPercap
	<fct>	<fct>	<int>	<dbl>	<int>	<dbl>
1	China	Asia	2007	73.0	1318683096	4959.
2	India	Asia	2007	64.7	1110396331	2452.
3	United States	Americas	2007	78.2	301139947	42952.
4	Indonesia	Asia	2007	70.6	223547000	3541.
5	Brazil	Americas	2007	72.4	190010647	9066.

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



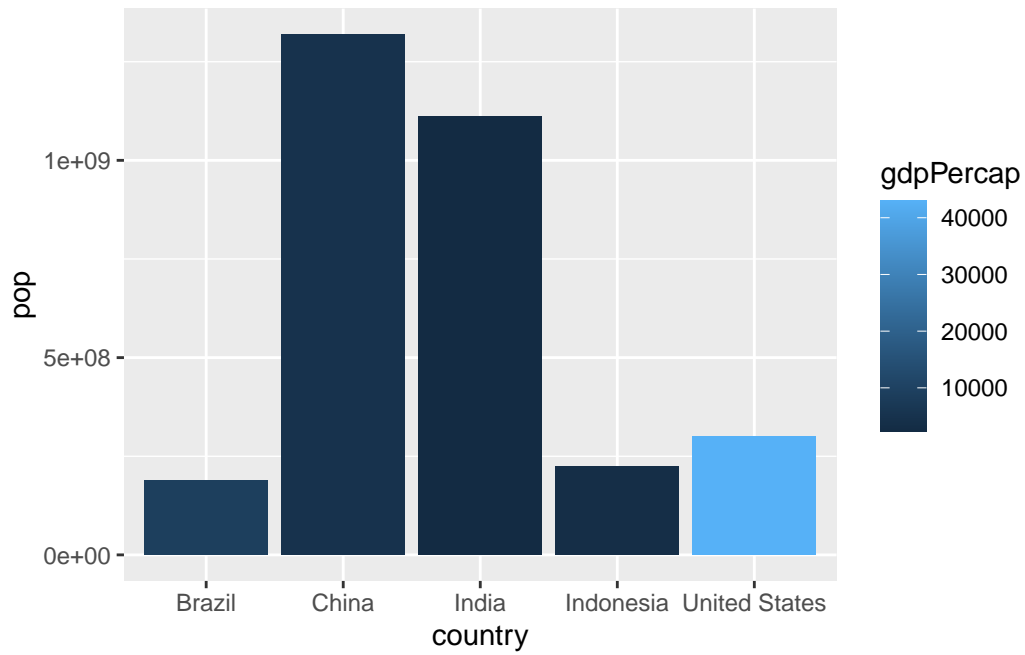
Q Create a bar chart showing the life expectancy of the five biggest countries by population in 2007.

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

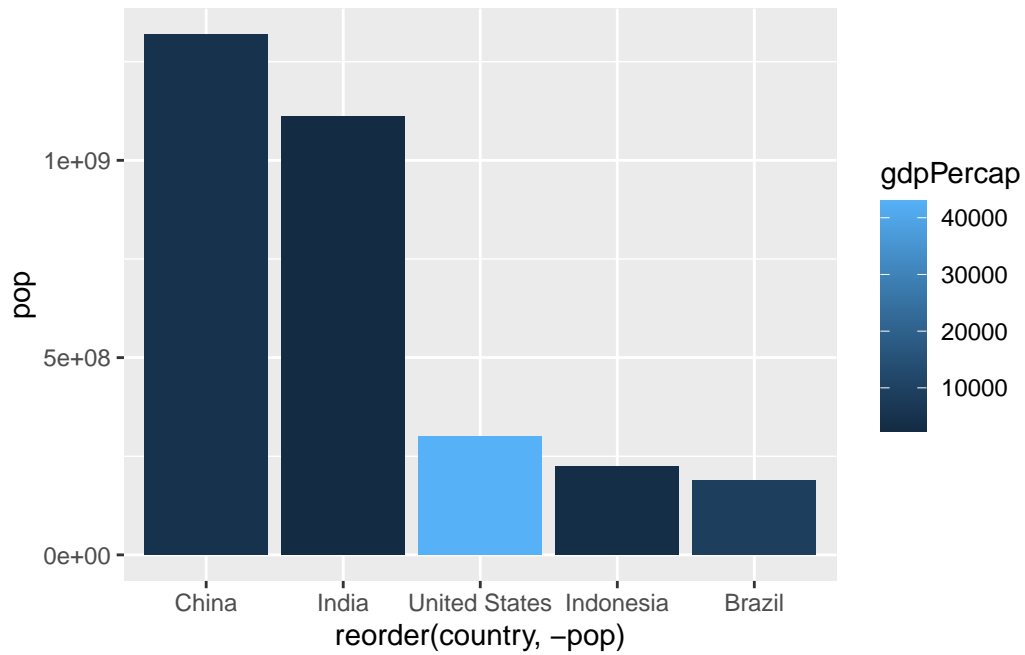


Q. Plot population size by country. Create a bar chart showing the population (in millions) of the five biggest countries by population in 2007.

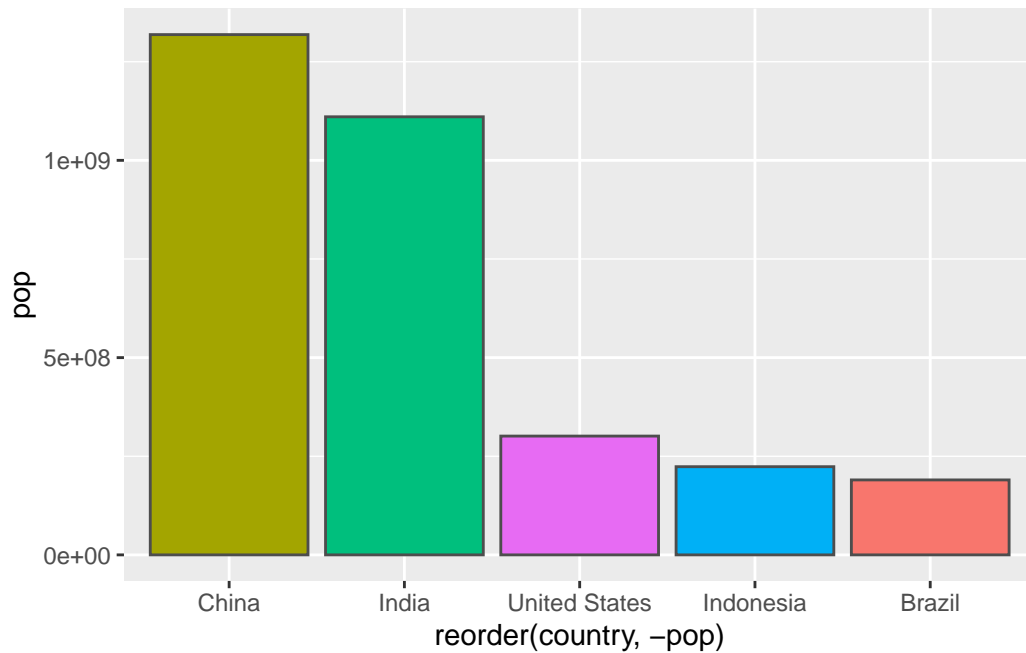
```
ggplot(gapminder_top5) +  
  aes(x=country, y=pop, fill=gdpPercap) +  
  geom_col()
```



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

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

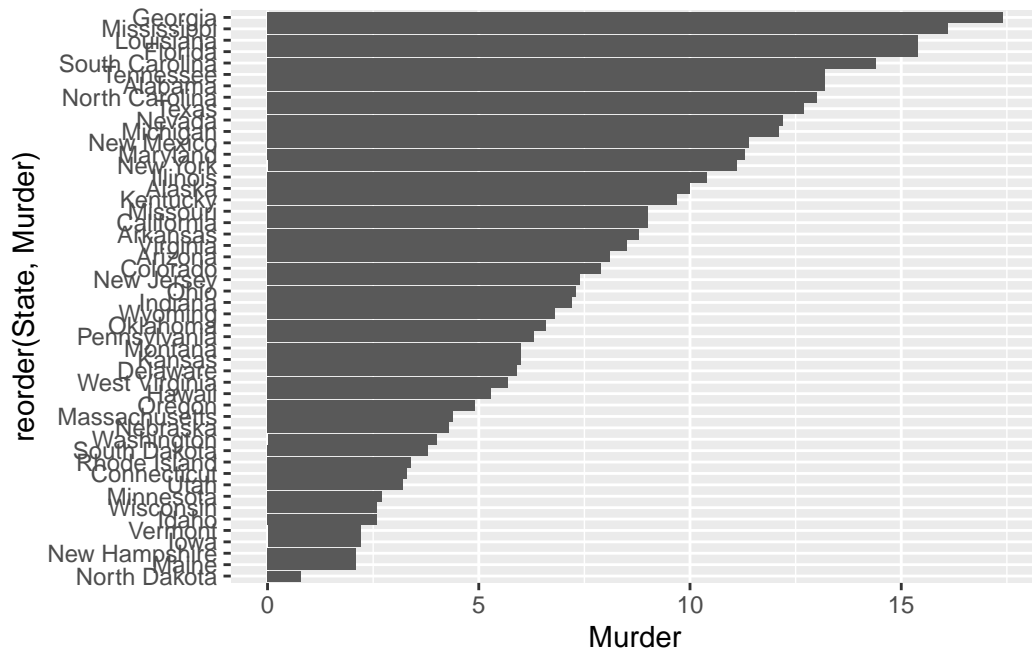


Flipping bar charts

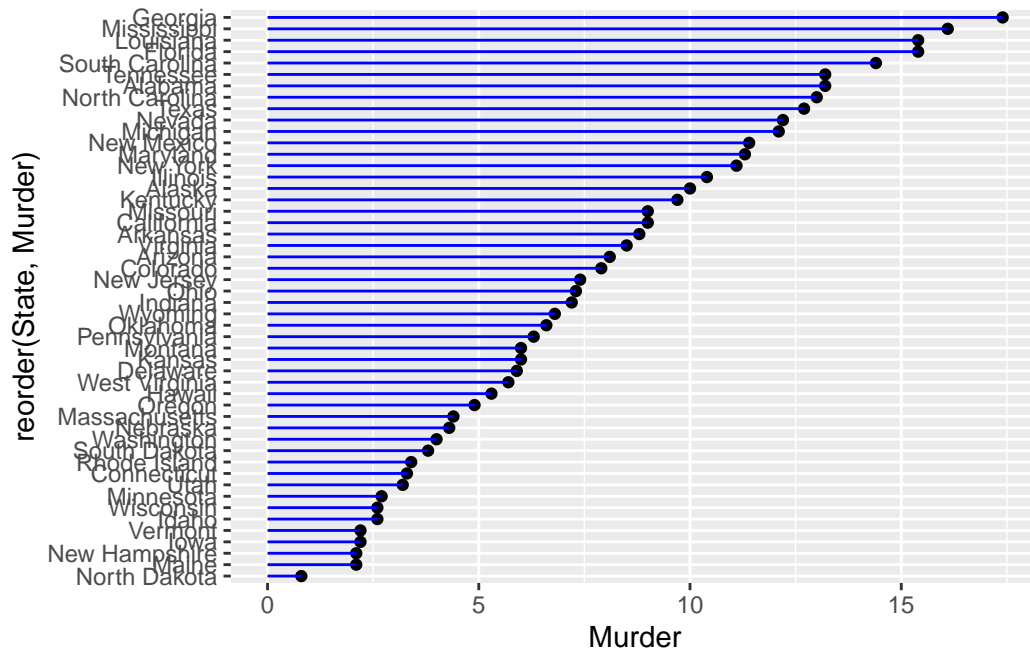
```
head(USArrests)
```

	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()
```



```
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()
```



Section 9. Extensions: Animation

```
#install.packages("gifski")
#install.packages("gganimate")
library(gapminder)
library(gganimate)

# Setup nice regular ggplot of the gapminder data
#ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
# geom_point(alpha = 0.7, show.legend = FALSE) +
# scale_colour_manual(values = country_colors) +
# scale_size(range = c(2, 12)) +
# scale_x_log10() +
# Facet by continent
# facet_wrap(~continent) +
# Here comes the gganimate specific bits
# labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +
# transition_time(year) +
# shadow_wake(wake_length = 0.1, alpha = FALSE)
```