Class 05: Data Visualization with GGPLOT

Libby Gilmore, pid: A69047570

Table of contents

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

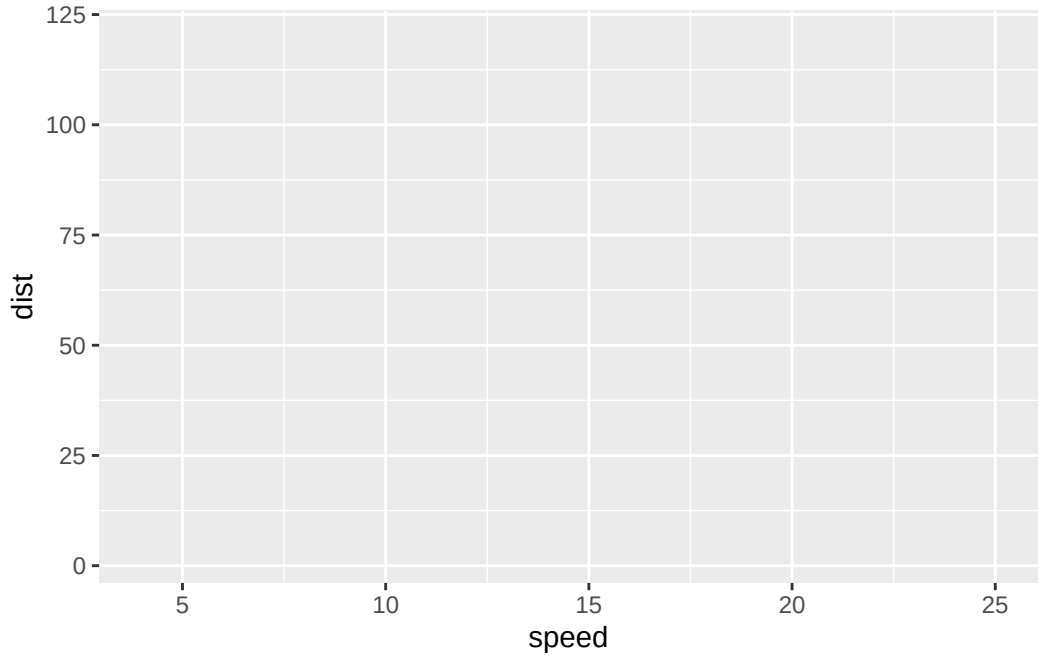
Which statement about data visualization with ggplot2 is incorrect? - ggplot2 is the only way to create plots in R

creating scatter plots

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

ggplot(cars) # this just plots an empty grey screen
```

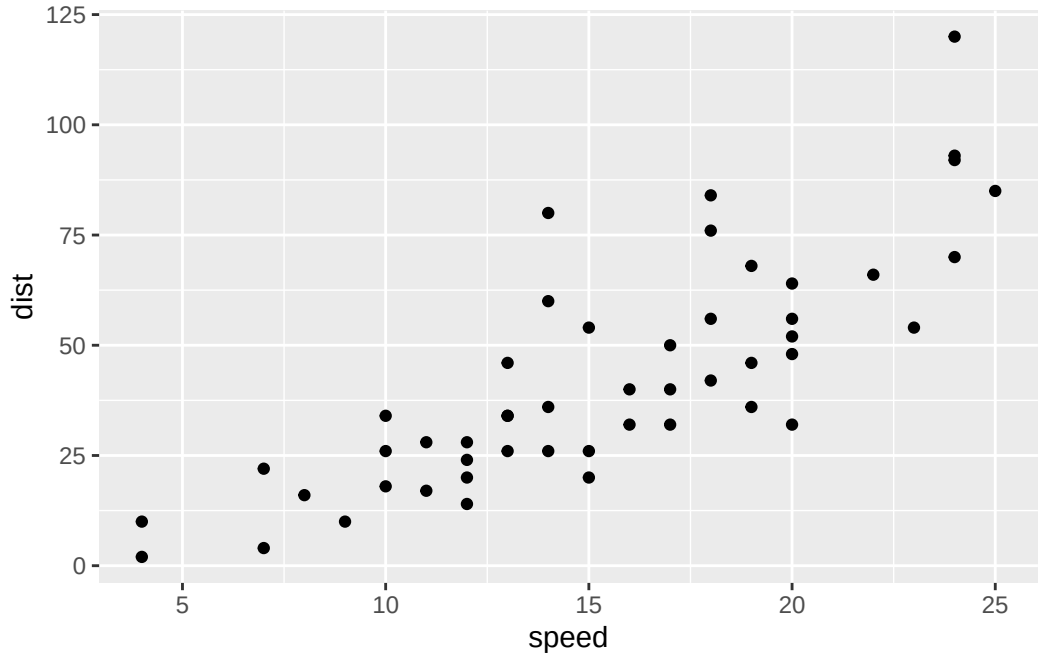


```
# specific aesthetic mappings with aes()
ggplot(cars) +
  aes(x=speed, y=dist)
```



Specify a geom layer with `geom_point()`

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



q5

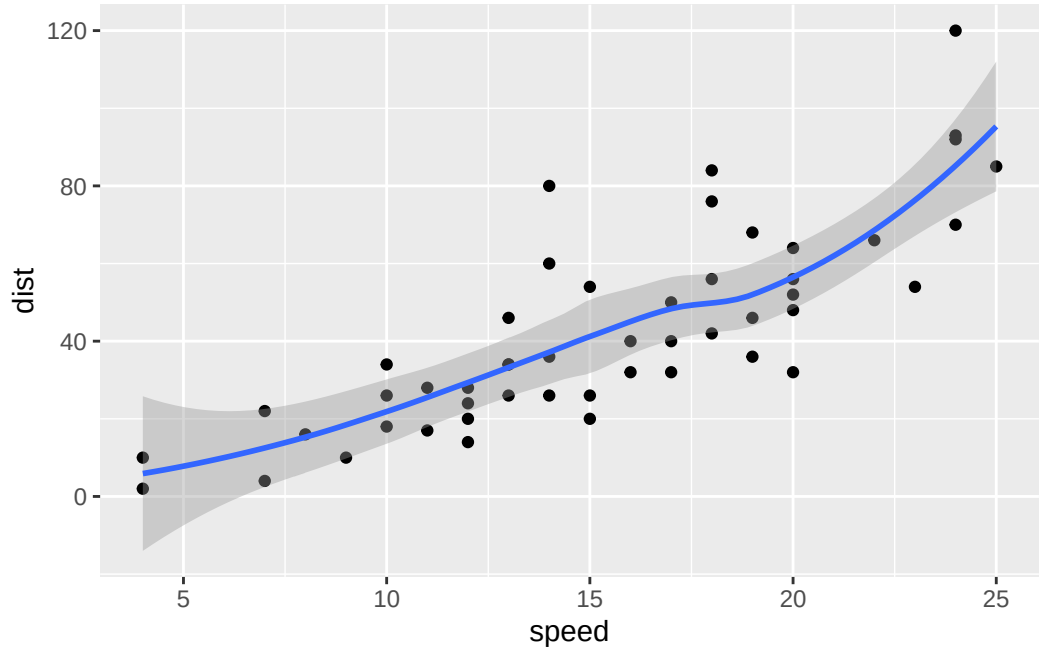
Which geometric layer should be used to create scatter plots in ggplot2? - geom_point()

q6

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'
```

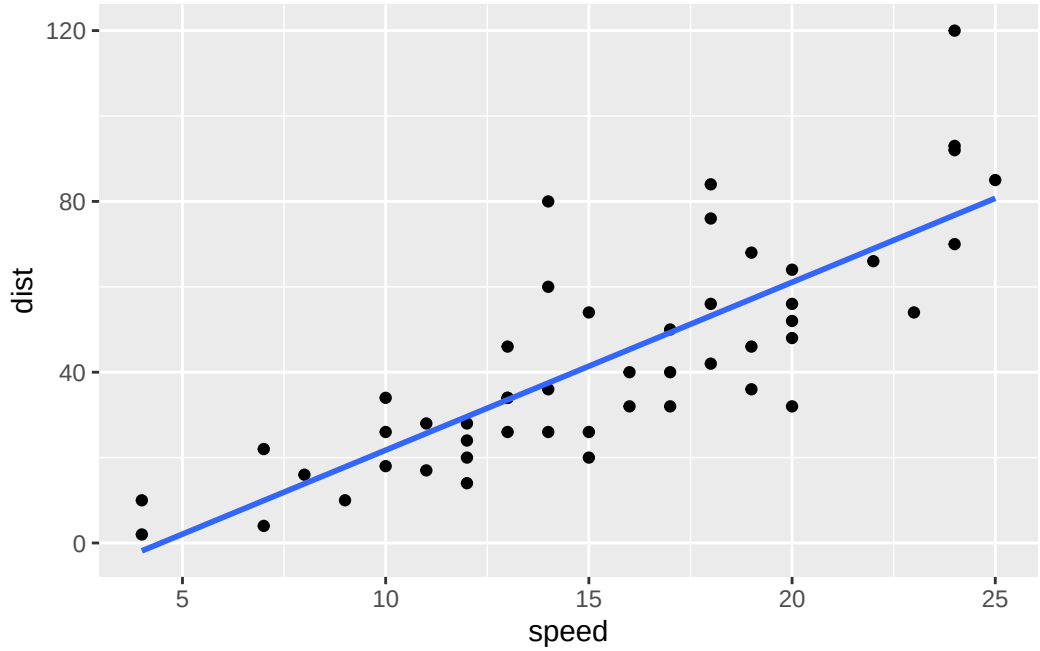


q7

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

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

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

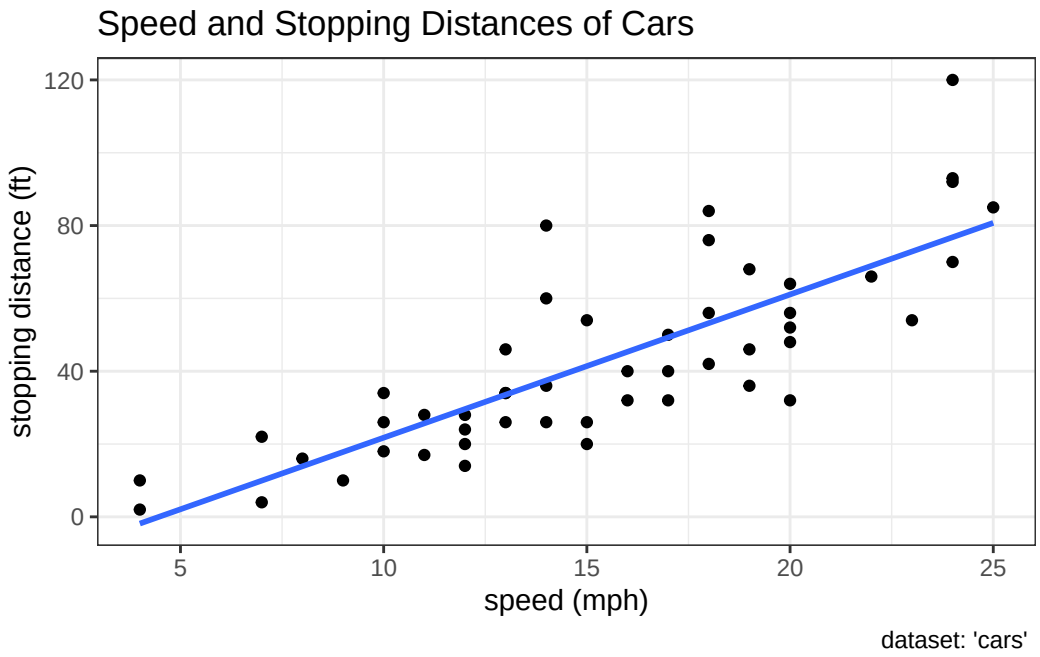


q8

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:

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

```
#Q. Use the nrow() function to find out how many genes are in this dataset. What is your answ
nrow(genes)
```

```
[1] 5196
```

```
#Q. Use the colnames() function and the ncol() function on the genes data frame to find out w
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

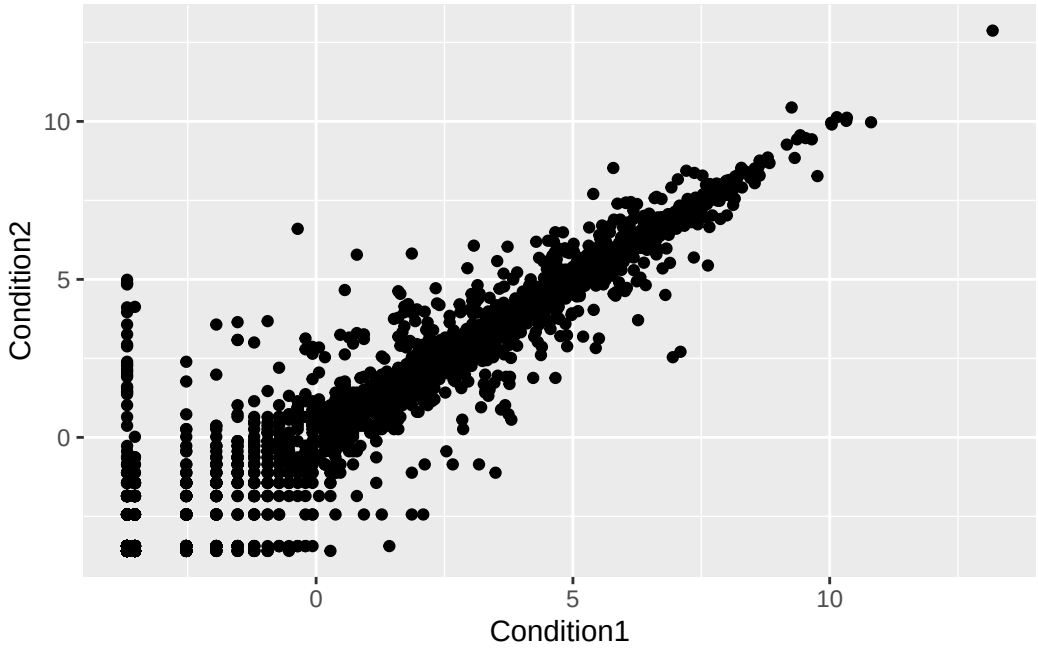
```
#Q. Use the table() function on the State column of this data.frame to find out how many 'up
table(genes$State)
```

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

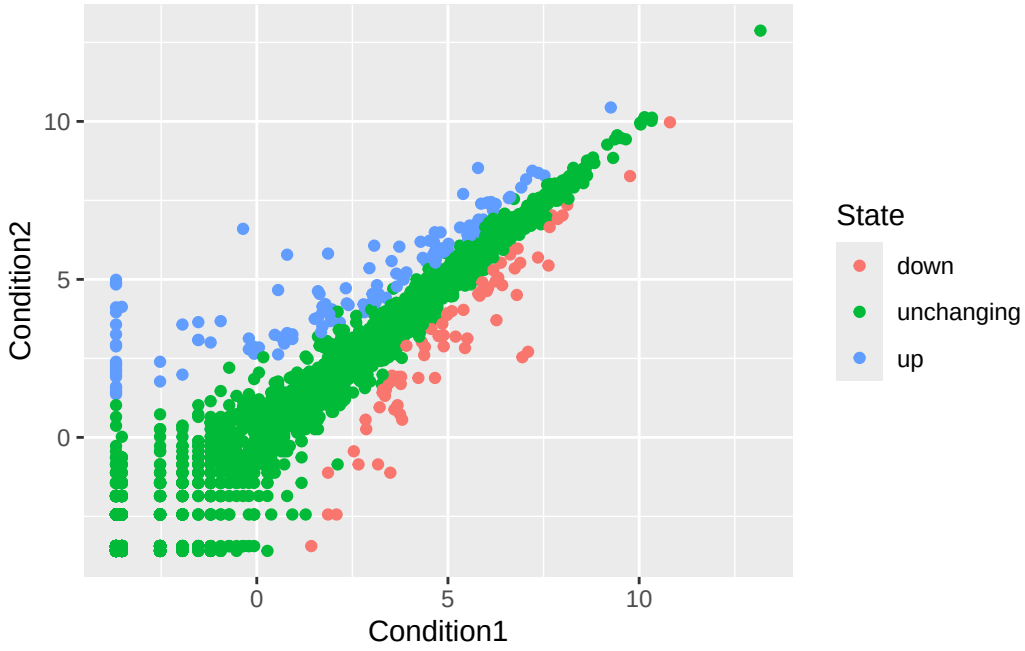
```
#Q. Using your values above and 2 significant figures. What fraction of total genes is up-reg
round( table(genes$State)/nrow(genes) * 100, 2 )
```

```
      down unchanging         up
      1.39      96.17       2.44
```

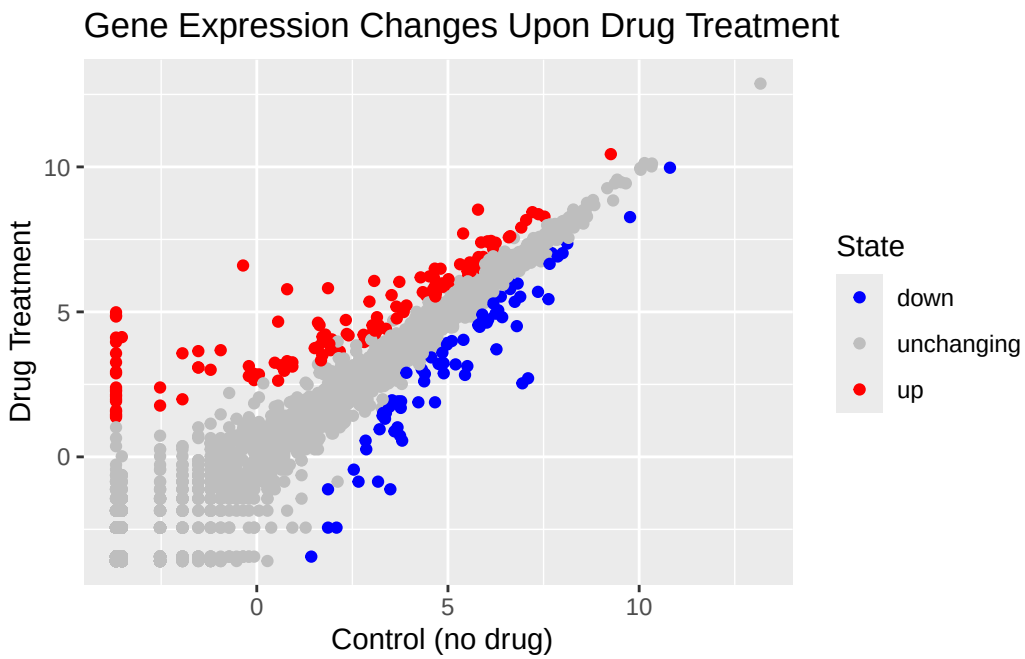
```
# Q. Complete the code below to produce the following plot

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




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




```
p +
  scale_colour_manual( values=c("blue","gray","red") ) +
  labs(title = "Gene Expression Changes Upon Drug Treatment",
       x = " Control (no drug)",
       y= "Drug Treatment")
```



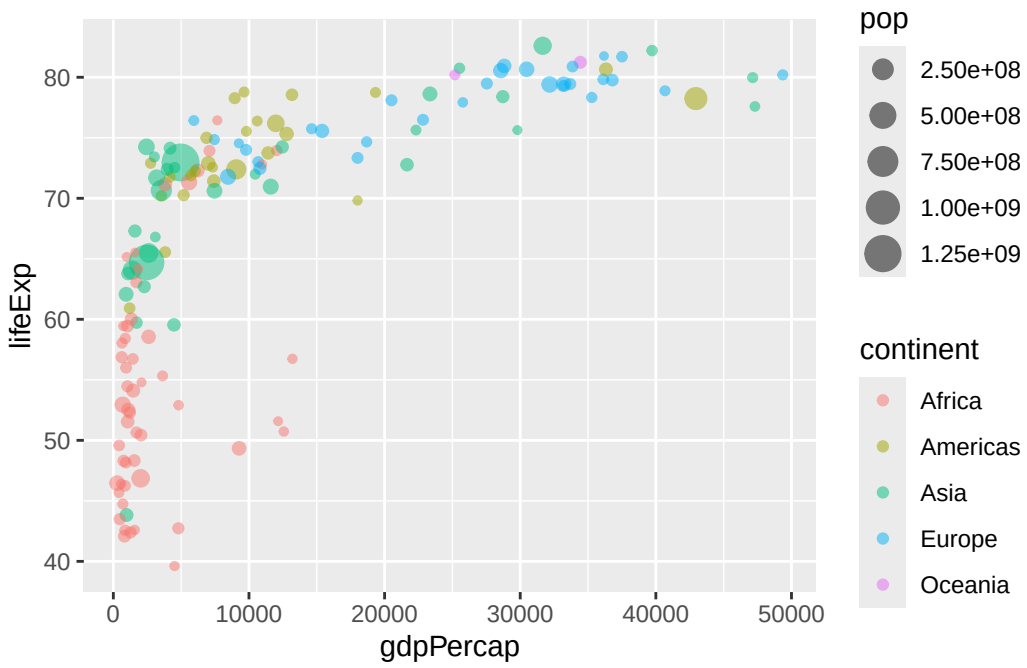
7. Going Further

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

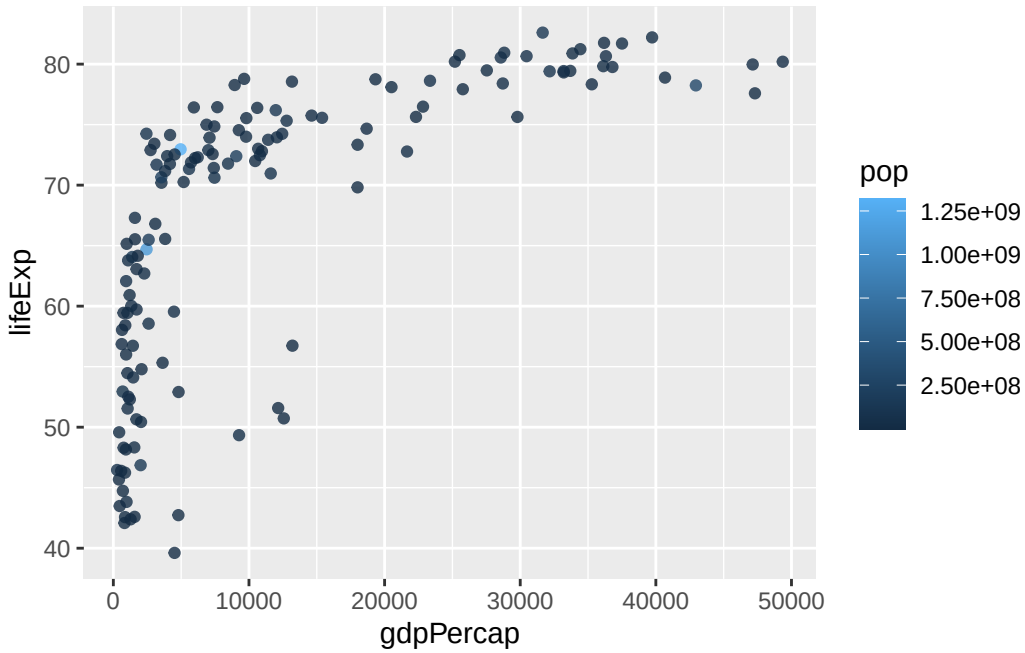
```
Attaching package: 'dplyr'

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)
```

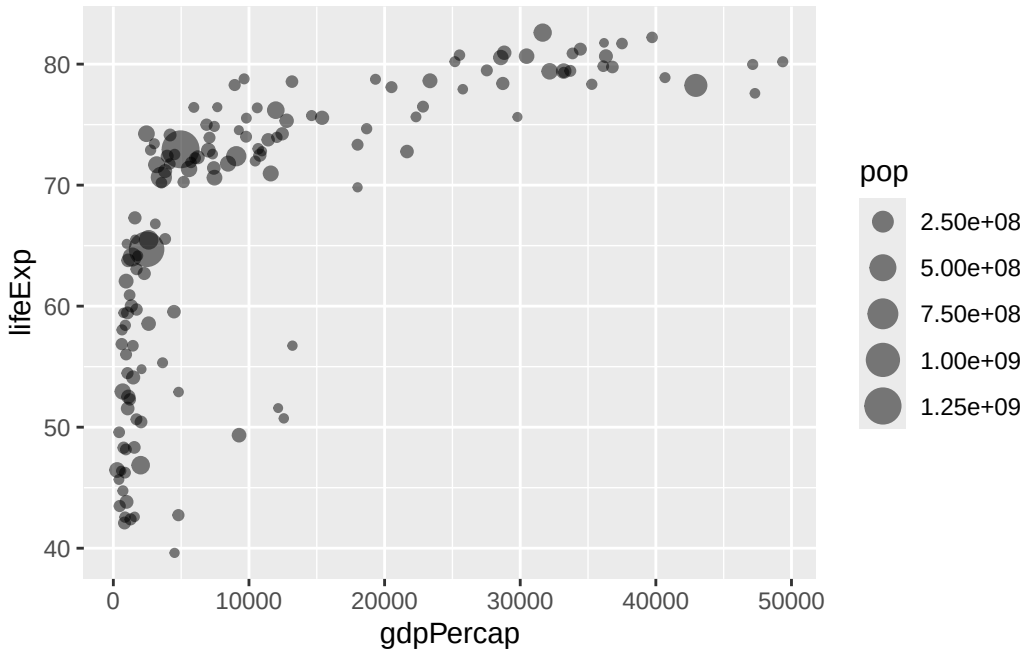
```
# Q. Complete the code below to produce a first basic scatter plot of this gapminder_2007 da
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```




```
# color the points by the numeric variable population pop
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```




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

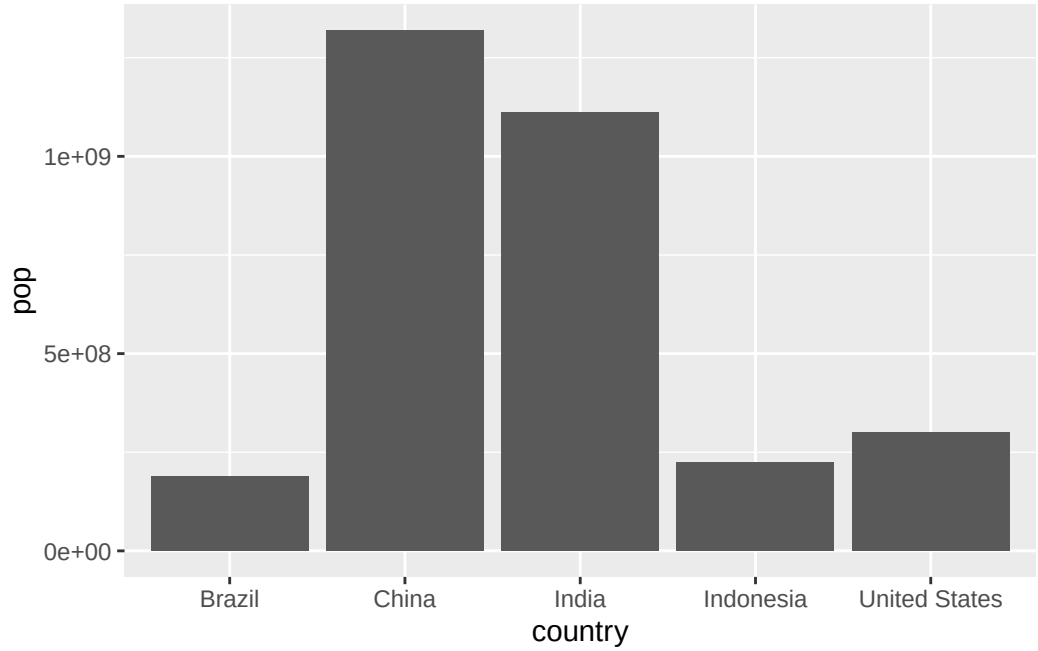


8. Bar Charts

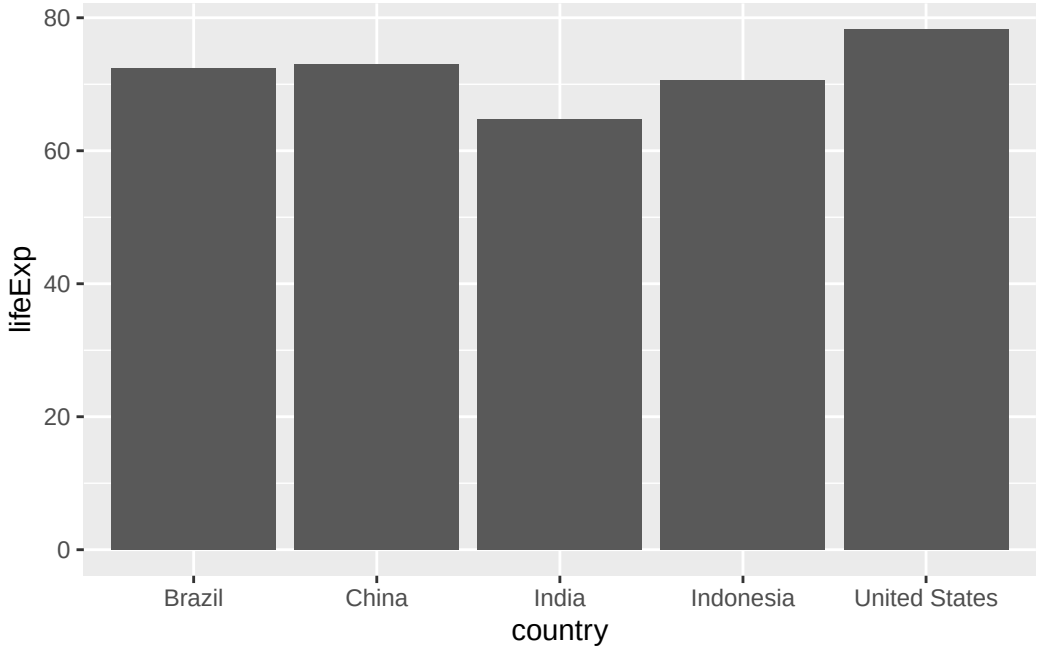
```
gapminder_top5 <- gapminder |>
  filter(year==2007) |>
  arrange(desc(pop)) |>
  top_n(5, pop)

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.
```

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




```
# Create a bar chart showing the life expectancy of the five biggest countries by population
ggplot(gapminder_top5) +
  geom_col(aes(x = country, y = lifeExp))
```



```r
# Plot population size by country. Create a bar chart showing the population (in millions) o

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