Class 5: Data Viz ggplot

Marriane A. (A16902759)

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()

Q6. Q. Use the nrow() function to find out how many genes are in this dataset. What is your answer?

5196

Q7. Use the colnames() function and the ncol() function on the genes data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find?

4

Q8. Use the nrow() function to find out how many genes are in this dataset. What is your answer?

5196

> Q9. Use the colnames() function and the ncol() function on the genes data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find?

4

> Q10. 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?

127

> Q11. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

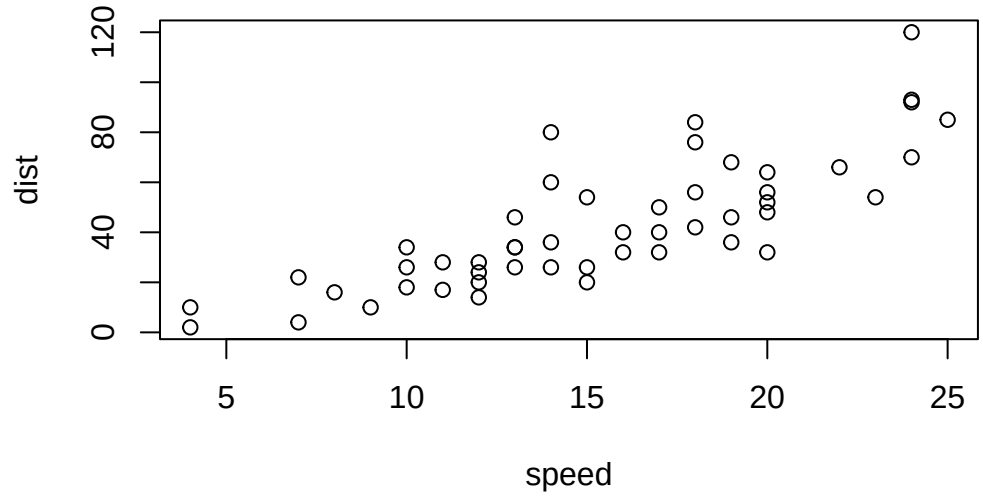
2.44

Plotting in R

R has lots of ways to make plots and figures. This includes so called **base** graphics and packages.

ggplot is the only way to create plots in R

```
plot(cars)
```



Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
head(cars)
```

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

Q. How would we plot this dataset with **ggplot2**?

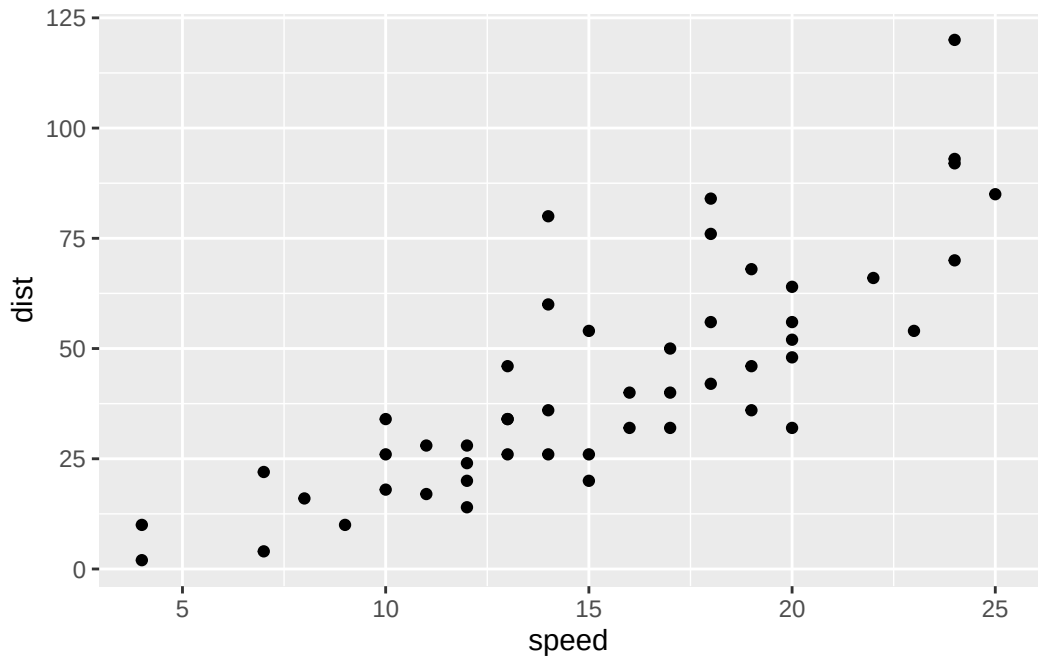
All ggplot figures have at least 3 layers:

-**data** -**aes** (how the data map to the plot) -**geoms** (how we draw the plot)

Before I use a new package. I need to download and install it with 'install.packages()' command. But don't install within my quarto document.

Once a package is installed I can load it up with the 'library()' function.

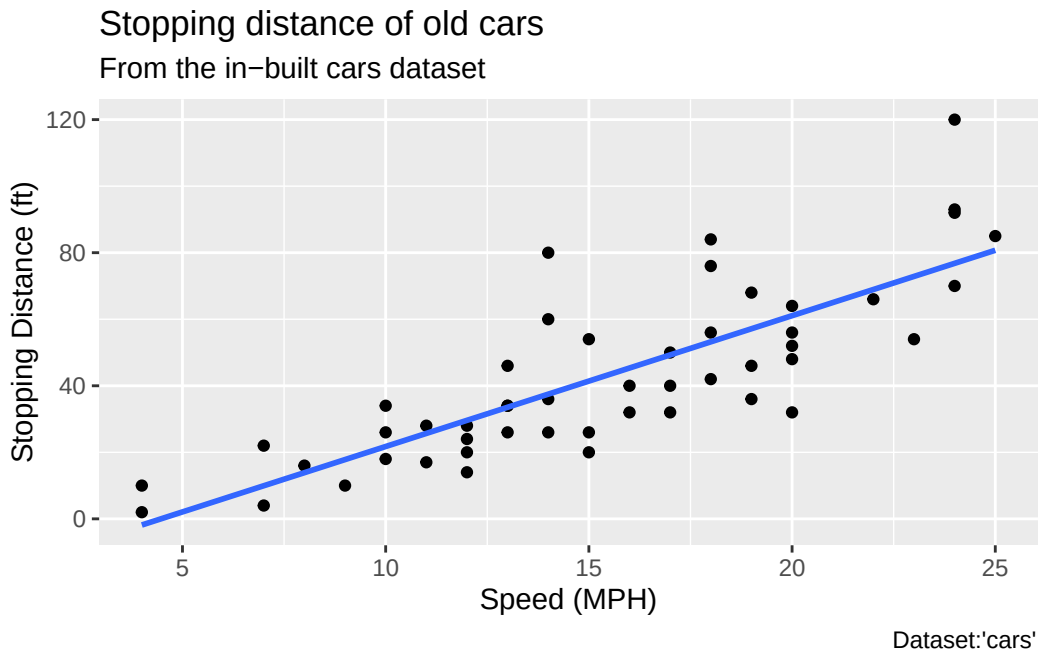
```
# install.packages("ggplot2")
library(ggplot2)
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



Key-point: For simple plots (like the one above) ggplot is more verbose (we need to do more typing) but as plots get more complicated ggplot starts to be more clear and simple than base R plot

```
p <- save
ggplot(cars) +
  aes(speed, dist) +
  geom_point() +
  geom_smooth(se=FALSE, method="lm") +
  labs(title="Stopping distance of old cars",
       subtitle = "From the in-built cars dataset",
       x="Speed (MPH)",
       y="Stopping Distance (ft)",
       caption="Dataset:'cars'")
```

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




```
  theme_bw()
```

```
List of 136
 $ line                            :List of 6
  ..$ colour       : chr "black"
  ..$ linewidth    : num 0.5
  ..$ linetype     : num 1
  ..$ lineend      : chr "butt"
  ..$ arrow        : logi FALSE
  ..$ inherit.blank: logi TRUE
  ..- attr(*, "class")= chr [1:2] "element_line" "element"
 $ rect                            :List of 5
  ..$ fill         : chr "white"
  ..$ colour       : chr "black"
  ..$ linewidth    : num 0.5
  ..$ linetype     : num 1
  ..$ inherit.blank: logi TRUE
  ..- attr(*, "class")= chr [1:2] "element_rect" "element"
 $ text                            :List of 11
  ..$ family       : chr ""
```

```
 ..$ face         : chr "plain"
 ..$ colour       : chr "black"
 ..$ size         : num 11
 ..$ hjust        : num 0.5
 ..$ vjust        : num 0.5
 ..$ angle        : num 0
 ..$ lineheight   : num 0.9
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : logi FALSE
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ title                          : NULL
$ aspect.ratio                   : NULL
$ axis.title                     : NULL
$ axis.title.x                   :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.x.top               :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 0
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
```

```
$ axis.title.x.bottom             : NULL
$ axis.title.y                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : num 90
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.y.left               : NULL
$ axis.title.y.right              :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : num -90
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text                       :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : chr "grey30"
 ..$ size         : 'rel' num 0.8
 ..$ hjust        : NULL
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : NULL
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
```

```
$ axis.text.x                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x.top                 :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 0
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x.bottom              : NULL
$ axis.text.y                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 1
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
```

```
$ axis.text.y.left                : NULL
$ axis.text.y.right               :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.theta                 : NULL
$ axis.text.r                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0.5
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 2.2points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.ticks                      :List of 6
 ..$ colour       : chr "grey20"
 ..$ linewidth    : NULL
 ..$ linetype     : NULL
 ..$ lineend      : NULL
 ..$ arrow        : logi FALSE
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_line" "element"
$ axis.ticks.x                    : NULL
$ axis.ticks.x.top                : NULL
$ axis.ticks.x.bottom             : NULL
$ axis.ticks.y                    : NULL
$ axis.ticks.y.left               : NULL
```

```
$ axis.ticks.y.right              : NULL
$ axis.ticks.theta                : NULL
$ axis.ticks.r                    : NULL
$ axis.minor.ticks.x.top          : NULL
$ axis.minor.ticks.x.bottom       : NULL
$ axis.minor.ticks.y.left         : NULL
$ axis.minor.ticks.y.right        : NULL
$ axis.minor.ticks.theta          : NULL
$ axis.minor.ticks.r              : NULL
$ axis.ticks.length               : 'simpleUnit' num 2.75points
 ..- attr(*, "unit")= int 8
$ axis.ticks.length.x             : NULL
$ axis.ticks.length.x.top         : NULL
$ axis.ticks.length.x.bottom      : NULL
$ axis.ticks.length.y             : NULL
$ axis.ticks.length.y.left        : NULL
$ axis.ticks.length.y.right       : NULL
$ axis.ticks.length.theta         : NULL
$ axis.ticks.length.r             : NULL
$ axis.minor.ticks.length         : 'rel' num 0.75
$ axis.minor.ticks.length.x       : NULL
$ axis.minor.ticks.length.x.top   : NULL
$ axis.minor.ticks.length.x.bottom: NULL
$ axis.minor.ticks.length.y       : NULL
$ axis.minor.ticks.length.y.left  : NULL
$ axis.minor.ticks.length.y.right : NULL
$ axis.minor.ticks.length.theta   : NULL
$ axis.minor.ticks.length.r       : NULL
$ axis.line                       : list()
 ..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ axis.line.x                     : NULL
$ axis.line.x.top                 : NULL
$ axis.line.x.bottom              : NULL
$ axis.line.y                     : NULL
$ axis.line.y.left                : NULL
$ axis.line.y.right               : NULL
$ axis.line.theta                 : NULL
$ axis.line.r                     : NULL
$ legend.background               :List of 5
 ..$ fill         : NULL
 ..$ colour       : logi NA
 ..$ linewidth    : NULL
 ..$ linetype     : NULL
```

```
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_rect" "element"
$ legend.margin                   : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5poi
 ..- attr(*, "unit")= int 8
$ legend.spacing                  : 'simpleUnit' num 11points
 ..- attr(*, "unit")= int 8
$ legend.spacing.x                : NULL
$ legend.spacing.y                : NULL
$ legend.key                      : NULL
$ legend.key.size                 : 'simpleUnit' num 1.2lines
 ..- attr(*, "unit")= int 3
$ legend.key.height               : NULL
$ legend.key.width                : NULL
$ legend.key.spacing              : 'simpleUnit' num 5.5points
 ..- attr(*, "unit")= int 8
$ legend.key.spacing.x            : NULL
$ legend.key.spacing.y            : NULL
$ legend.frame                    : NULL
$ legend.ticks                    : NULL
$ legend.ticks.length             : 'rel' num 0.2
$ legend.axis.line                : NULL
$ legend.text                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : 'rel' num 0.8
 ..$ hjust        : NULL
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : NULL
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ legend.text.position            : NULL
$ legend.title                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0
 ..$ vjust        : NULL
 ..$ angle        : NULL
```

```
  ..$ lineheight   : NULL
  ..$ margin       : NULL
  ..$ debug        : NULL
  ..$ inherit.blank: logi TRUE
  ..- attr(*, "class")= chr [1:2] "element_text" "element"
 $ legend.title.position           : NULL
 $ legend.position                 : chr "right"
 $ legend.position.inside          : NULL
 $ legend.direction                : NULL
 $ legend.byrow                    : NULL
 $ legend.justification            : chr "center"
 $ legend.justification.top        : NULL
 $ legend.justification.bottom     : NULL
 $ legend.justification.left       : NULL
 $ legend.justification.right      : NULL
 $ legend.justification.inside     : NULL
 $ legend.location                 : NULL
 $ legend.box                      : NULL
 $ legend.box.just                 : NULL
 $ legend.box.margin               : 'margin' num [1:4] 0cm 0cm 0cm 0cm
  ..- attr(*, "unit")= int 1
 $ legend.box.background           : list()
  ..- attr(*, "class")= chr [1:2] "element_blank" "element"
 $ legend.box.spacing              : 'simpleUnit' num 11points
  ..- attr(*, "unit")= int 8
  [list output truncated]
 - attr(*, "class")= chr [1:2] "theme" "gg"
 - attr(*, "complete")= logi TRUE
 - attr(*, "validate")= logi TRUE
```

```
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
```

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

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

```
ncol(genes)
```

```
[1] 4
```

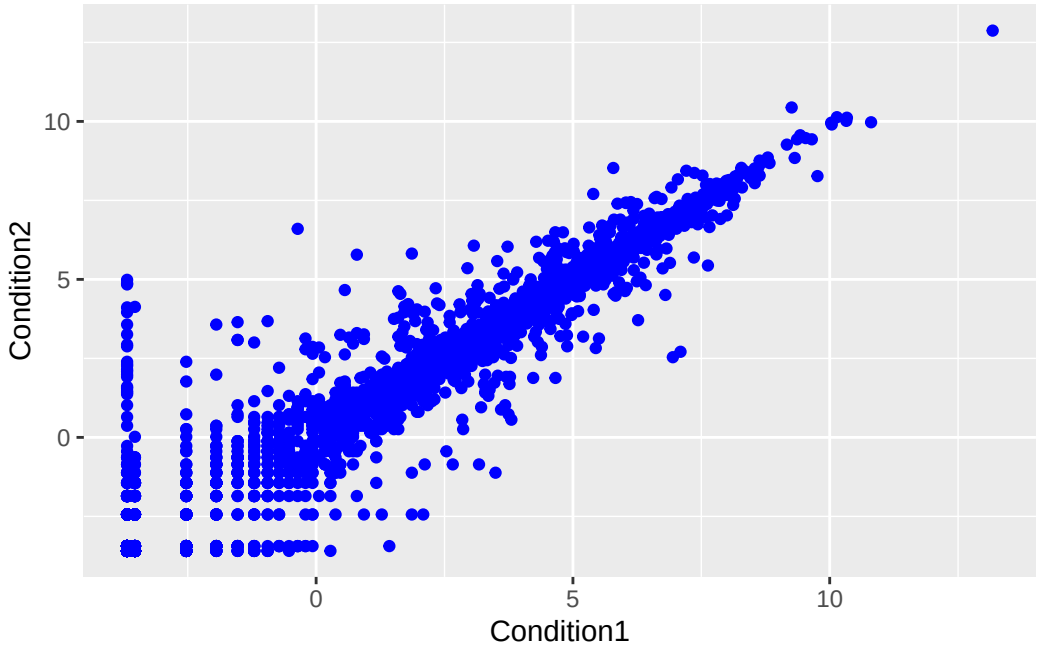
```
table(genes$State)
```

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

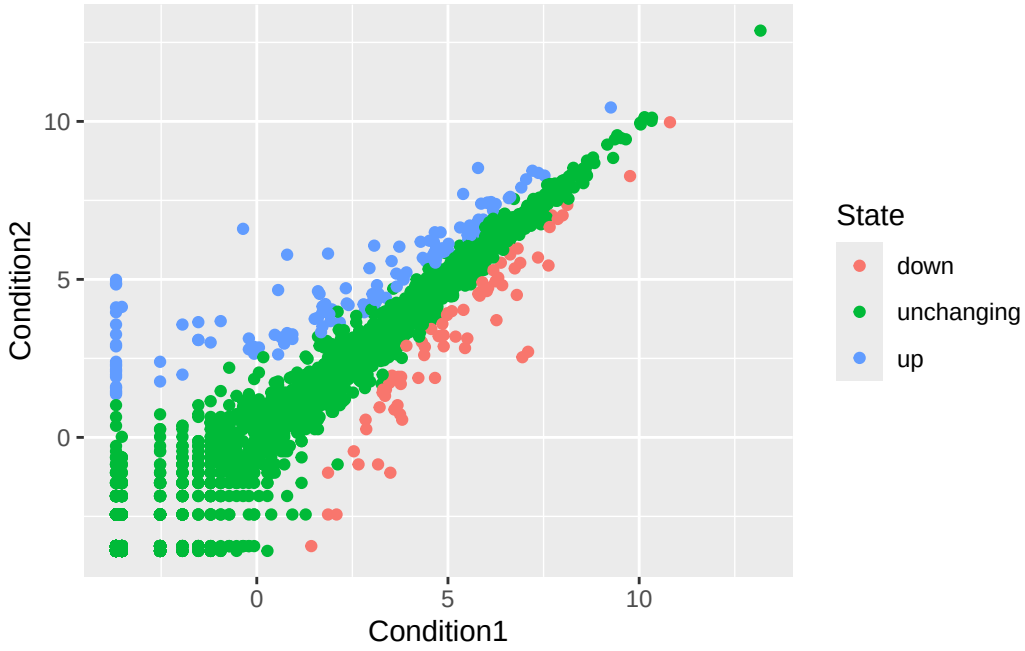
```
round (table(genes$State)/nrow(genes)*100, 2 )
```

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

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




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

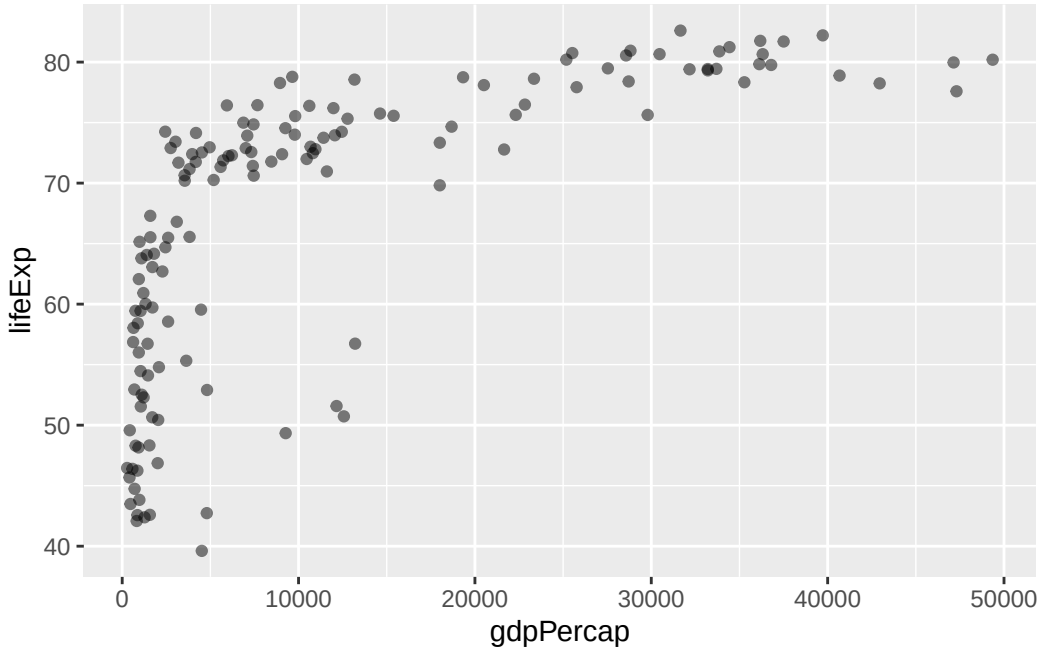



```
library(ggplot2)
# install.packages("dplyr")  ## un-comment to install if needed
library(dplyr)
```

```
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
```

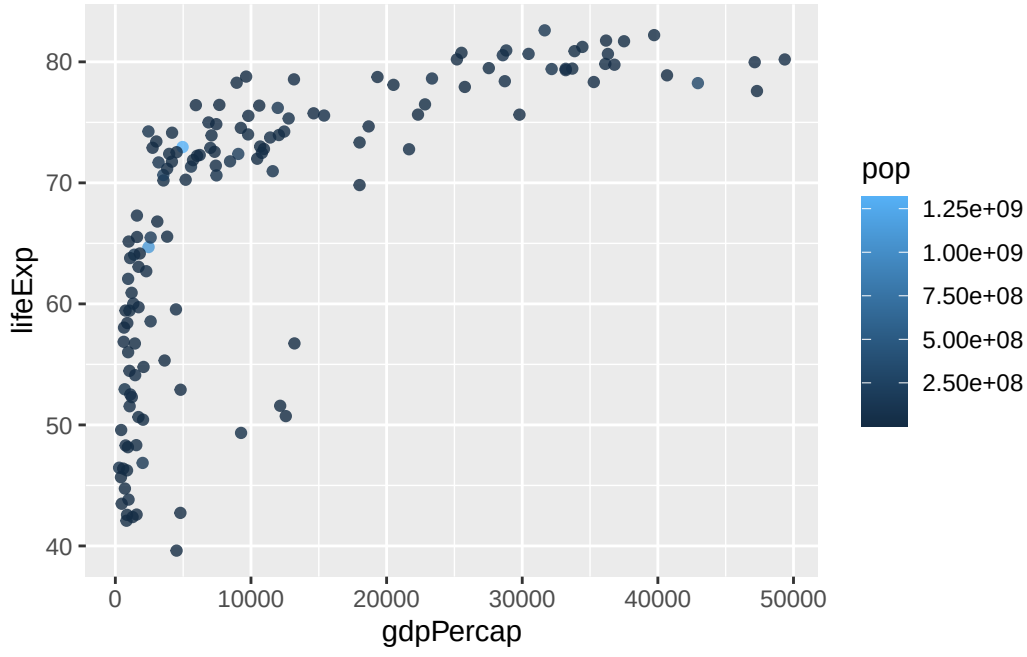
```
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.ts

gapminder <- read.delim(url)
gapminder_2007 <- gapminder %>% filter(year==2007)
```

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

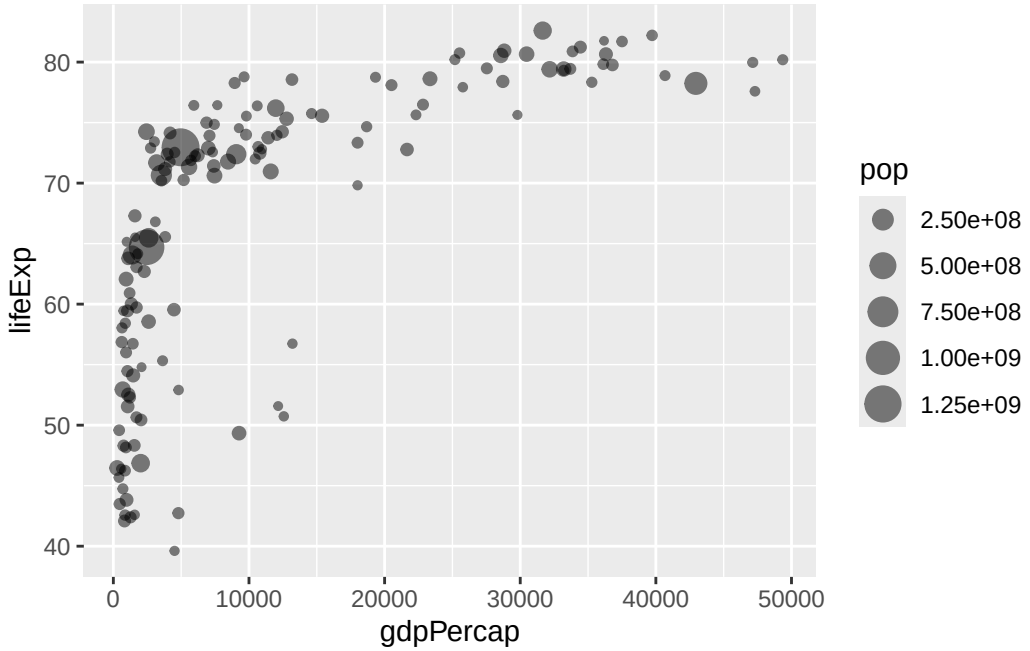



```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```


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

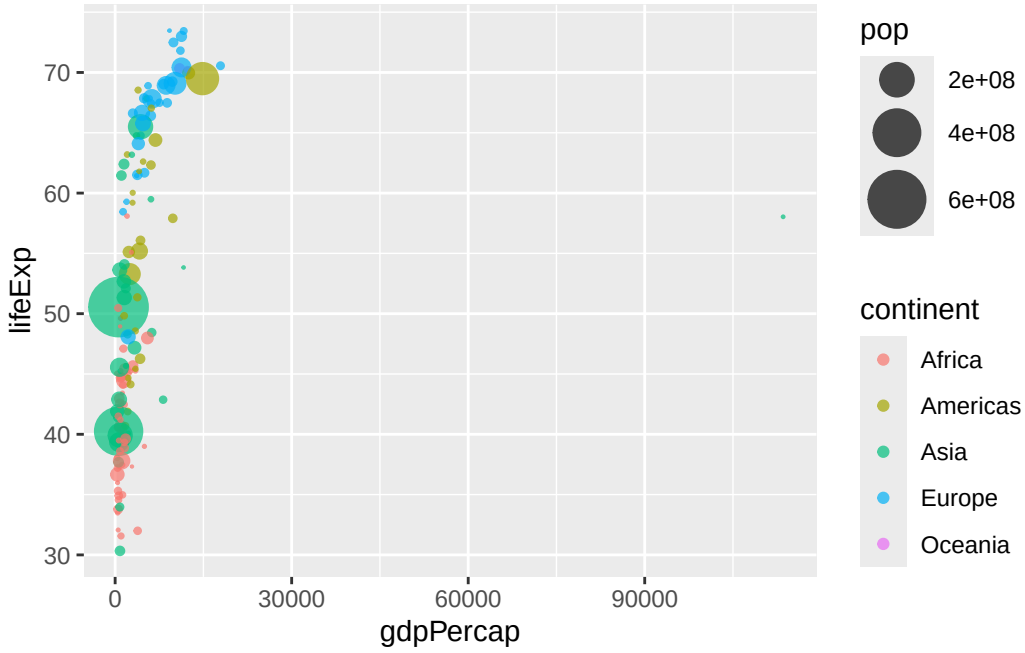


```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop) +
  geom_point(alpha=0.5)
```

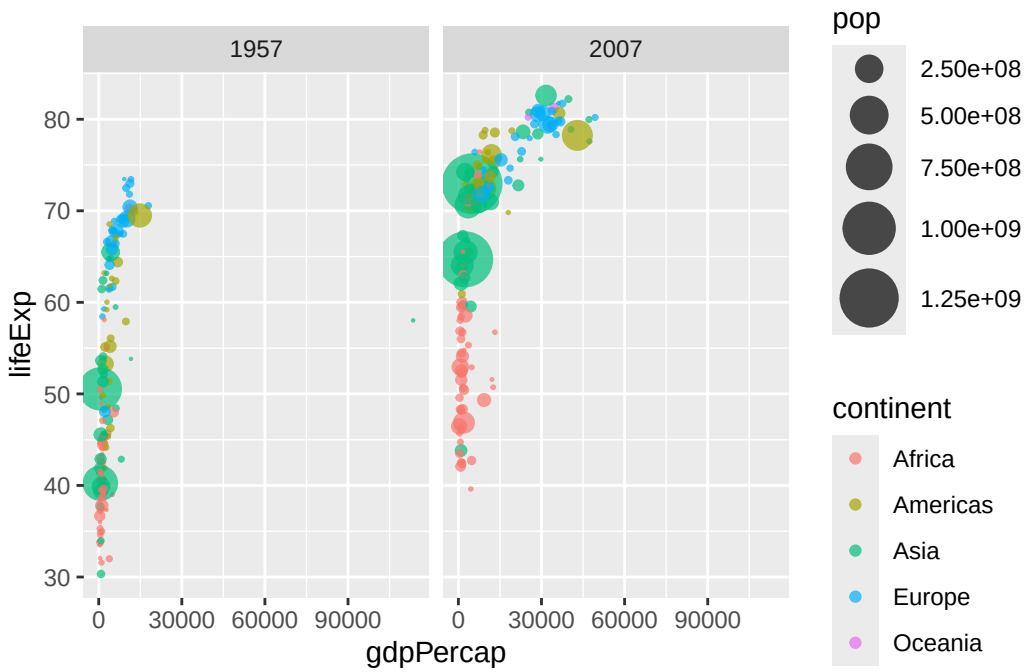


Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

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

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




```
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)
```

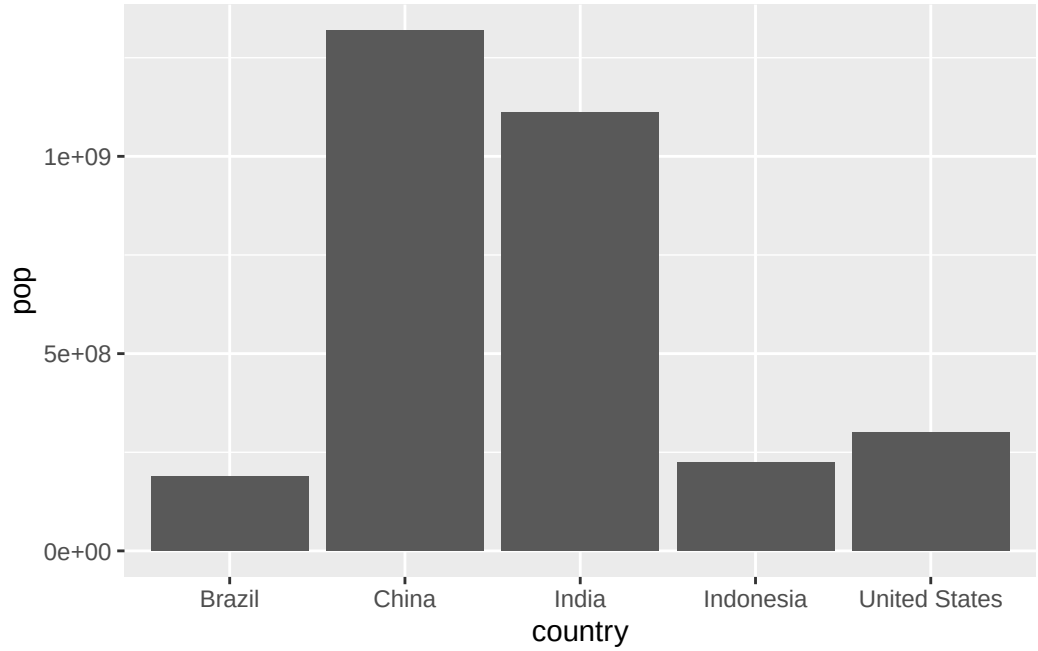


Bar Charts

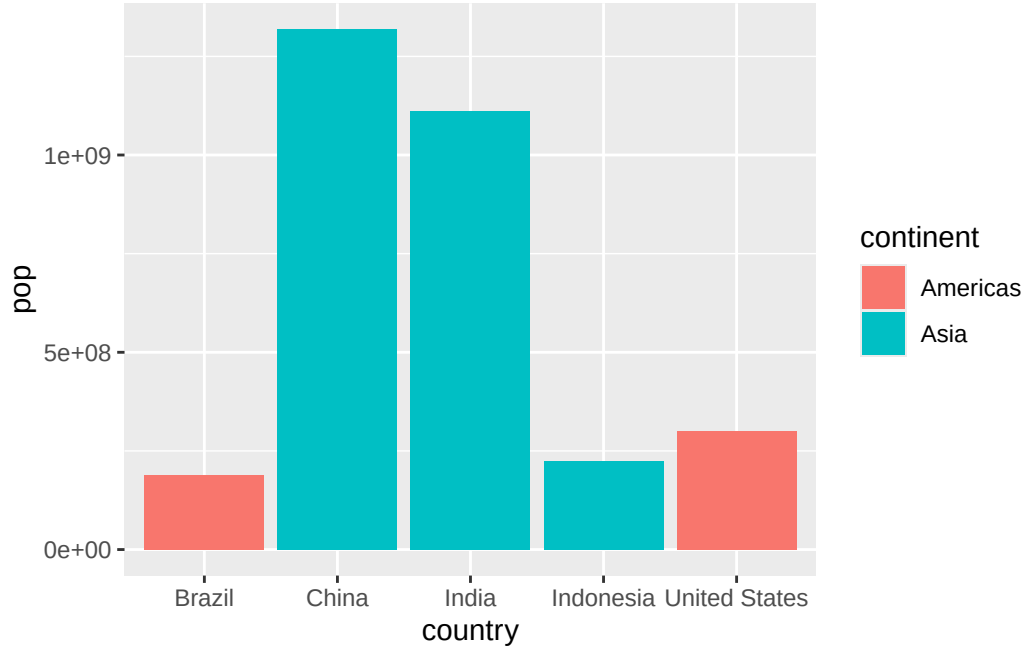
```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

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=pop))
```




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



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