Class 5: Data Viz with ggplot

Carly Chang (PID A16843962)

Intro to ggplot

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

Communication of results, exploratory data analysis (EDA), detection of outliers

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

FALSE

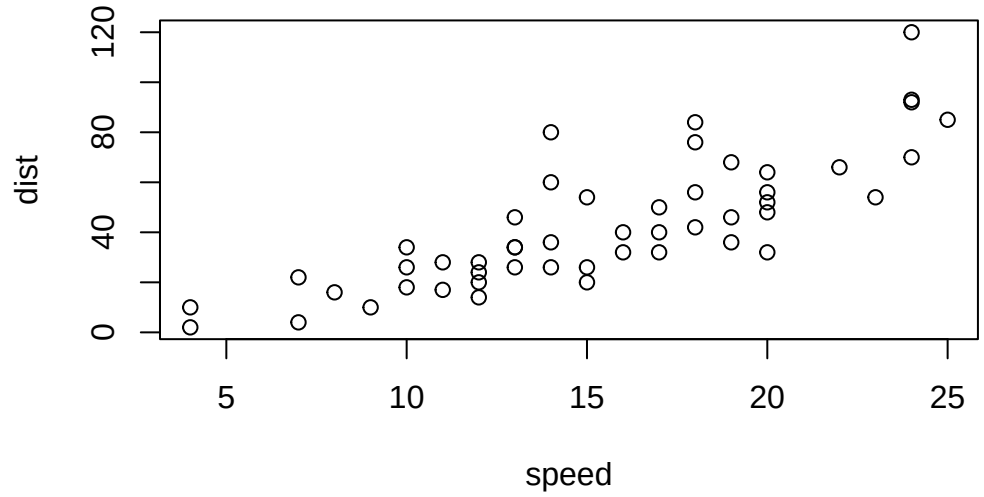
There are many graphics systems in R (ways to make plots and figures). These include "base" R plots. Today, we will focuse mostly of theh **ggplot2** package. Let's start with a plot of a simple in-built dataset called `cars`.

```
cars
```

```
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
11    11   28
12    12   14
13    12   20
14    12   24
15    12   28
16    13   26
```

```
17    13   34
18    13   34
19    13   46
20    14   26
21    14   36
22    14   60
23    14   80
24    15   20
25    15   26
26    15   54
27    16   32
28    16   40
29    17   32
30    17   40
31    17   50
32    18   42
33    18   56
34    18   76
35    18   84
36    19   36
37    19   46
38    19   68
39    20   32
40    20   48
41    20   52
42    20   56
43    20   64
44    22   66
45    23   54
46    24   70
47    24   92
48    24   93
49    24  120
50    25   85
```

```
plot(cars)
```



> Q. Which plot types are typically NOT used to compare distributions of numeric variables?

Network graphs

> Q. Which statement about data visualization with ggplot2 is incorrect?

ggplot2 is the only way to create plots in R

Let's see how we can make this figure using **ggplot**. First we need to install this package on our computer. To install any R package, we use the function `install.packages()`.

> I will run `install.packages("ggplot2")` in my R console, not this quarto document to avoid installing it every time I run the document.

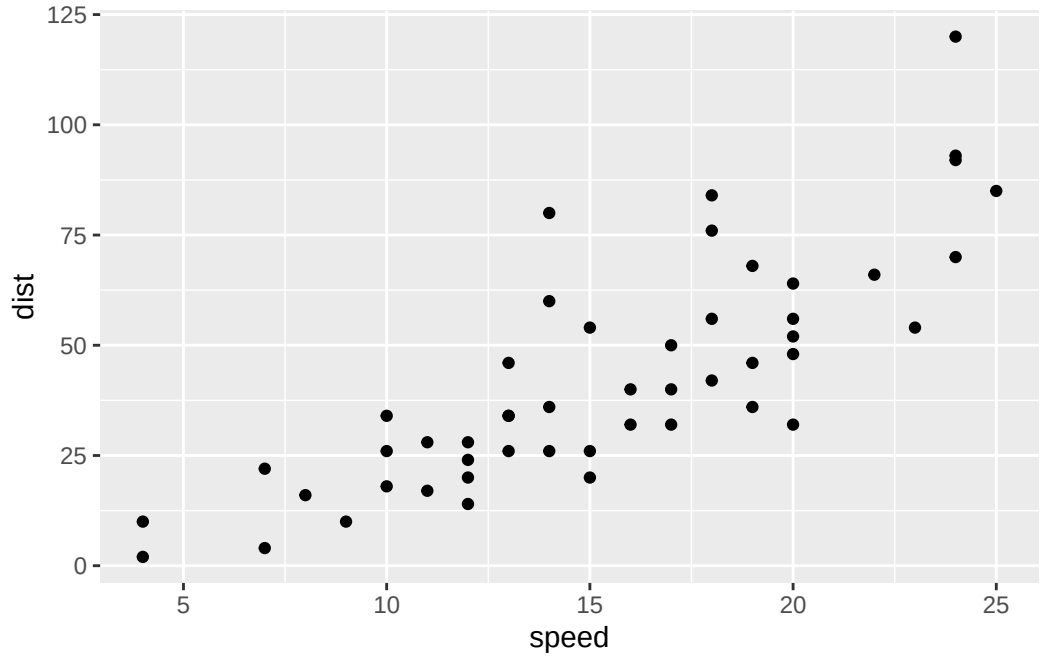
Before we can use any functions from add on packages, we need to load the package from our "library()" with the `libary(ggplot2)` function

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

All ggplot figures have 3 layers:

- **data** (the input dataset I want to plot from)
- **aes** (the aesthetic mapping of the data to my plot, such as x-y axis, line type, etc.)
- **geometry** (the type of plot, such as geom_point(), geom_line(), etc.)

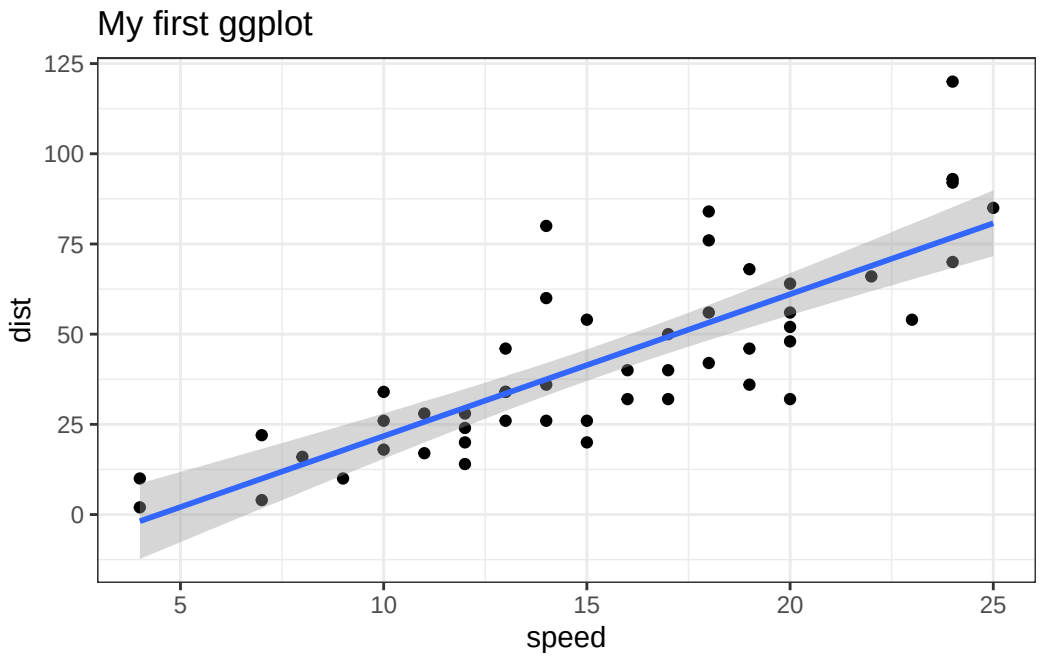
```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



Let's add a fitted line to show the relationship here:

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method = "lm") + #makes a linear fitted line
  theme_bw() + #makes figure black and white
  labs(title="My first ggplot") #makes a title
```

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



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

geom_point()

Gene expression figure

The code to read the dataset:

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes) #only shows the first 6 rows
```

```
        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. How many genes are in this dataset?

```
nrow(genes)
```

```
[1] 5196
```

Q. 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?

```
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' regulated genes there are. What is your answer?

```
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-regulated in this dataset?

```
round(table(genes$State)/nrow(genes), 4)
```

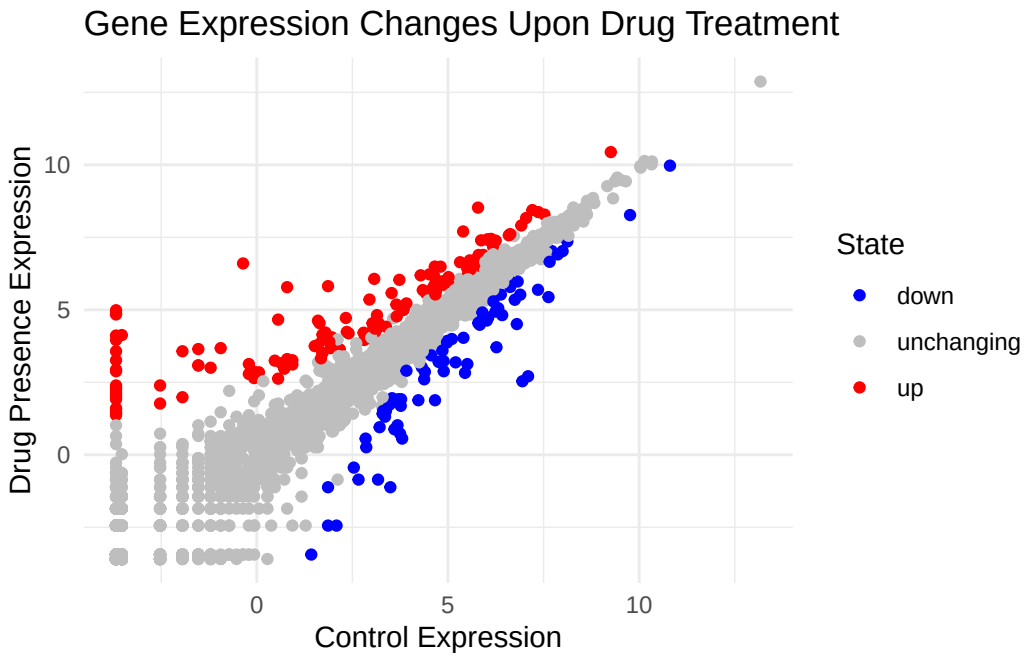
```
      down unchanging         up
    0.0139     0.9617     0.0244
```

```
n.tot <- nrow(genes)
vals <- table(genes$State)

vals.percent <- vals/n.tot * 100 #put in percentage
round(vals.percent, 2) #round to 2 sig figs
```

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

A first plot of this dataset

```
p1 <- ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State) + #color by state
  geom_point() +
 theme_bw() +
  labs(title="Gene Expression Changes Upon Drug Treatment",
       x="Control Expression",
       y="Drug Presence Expression") +
  scale_colour_manual(values=c("blue","gray","red"))
p1 + theme_minimal()
```



Points on the diagonal = no difference in condition before and after drug treatment

Going Further

```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.tsv

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

Make gapminder plot for year 2007:

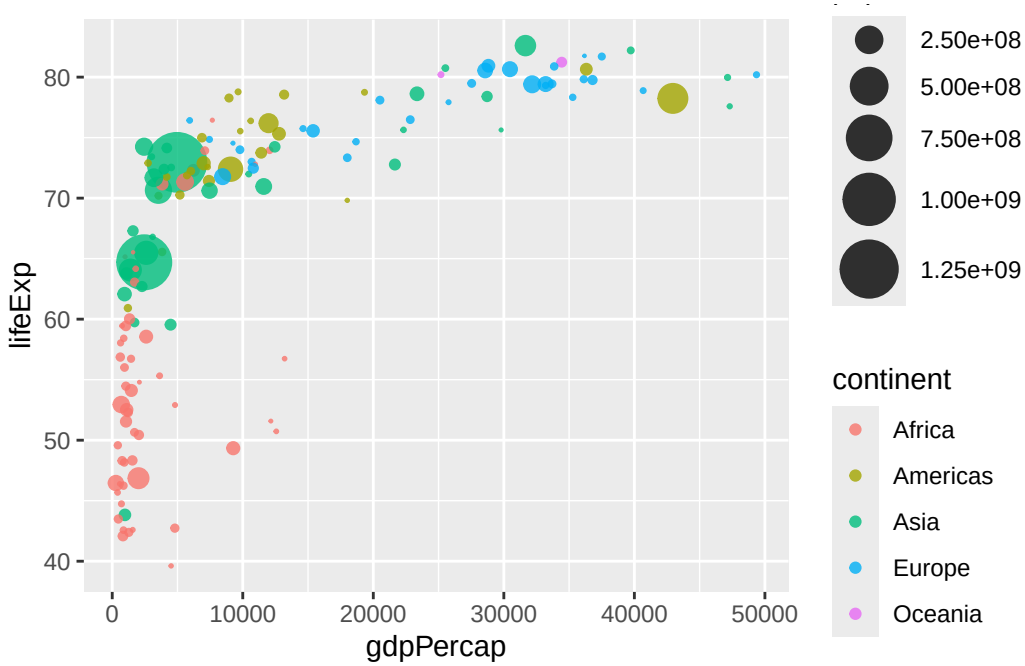
```
library(gapminder)
```

```
Attaching package: 'gapminder'

The following object is masked _by_ '.GlobalEnv':

    gapminder
```

```
library(dplyr)
```

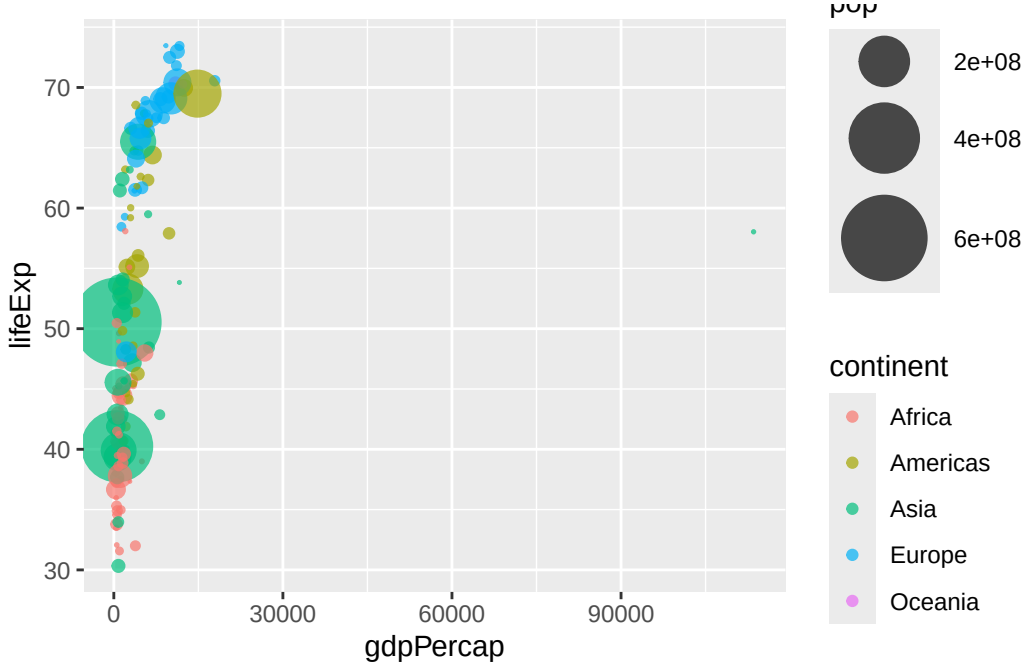
```
Attaching package: 'dplyr'

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)
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color = continent, size=pop) + #color by continent and make siz
  geom_point(alpha=0.8) + #makes point less bold
  scale_size_area(max_size = 10) #scales size more so that it is easier to see a difference
```



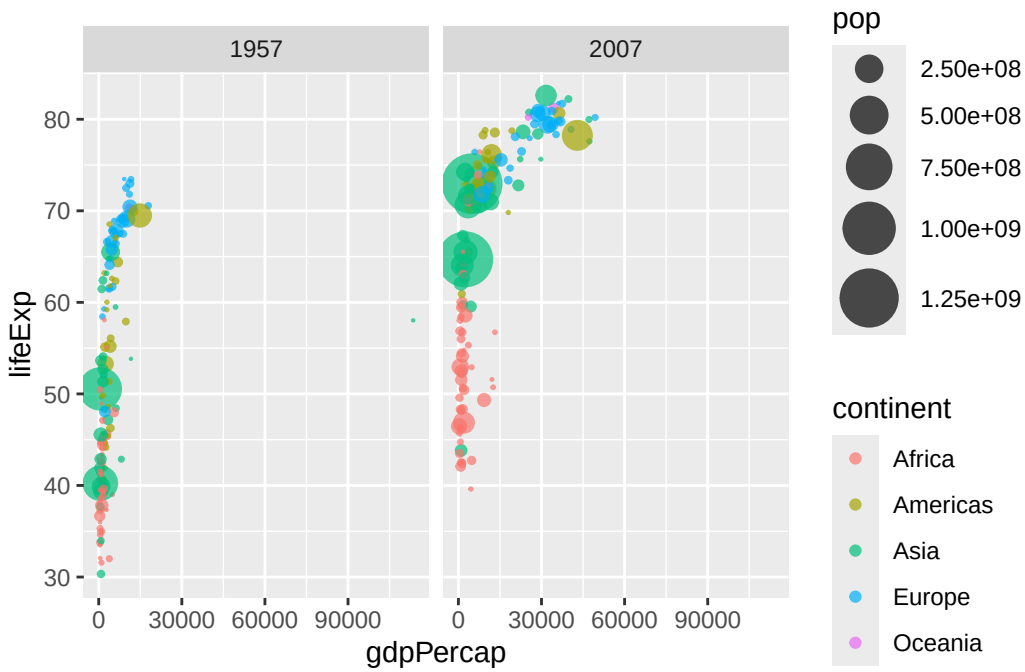
Make plot for year 1957:

```
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 = 15)
```



Make plot for both 1957 and 2007:

```
gapminder_both <- gapminder %>% filter(year==1957 | year==2007)
  ggplot(gapminder_both) +
  geom_point(aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year) #puts graphs next to each other
```


1957
2007
lifeExp
80
70
60
50
40
30
0
30000
60000
90000
gdpPercap
pop
2.50e+08
5.00e+08
7.50e+08
1.00e+09
1.25e+09
continent
Africa
Americas
Asia
Europe
Oceania