Class 05: Data Visualization with GGPLOT

Joshua Mac

Overview/Background (Sections 1-5)

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

A. All of the above

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

A. FALSE

Q3. Which plot types are typically NOT used to compare distributions of numeric variables?

A. Network graphs

Q4. Which statement about data visualization with ggplot2 is incorrect?

A. ggplot is the only way to create plots in R

Intro to ggplot (Section 6)

There are many graphics systems in R (ways to make plots and figures). These include "base" R plots. Today we will focus mostly on the **ggplot2** package.

Let's start with a plot of a simple in-built dataset called `cars`.

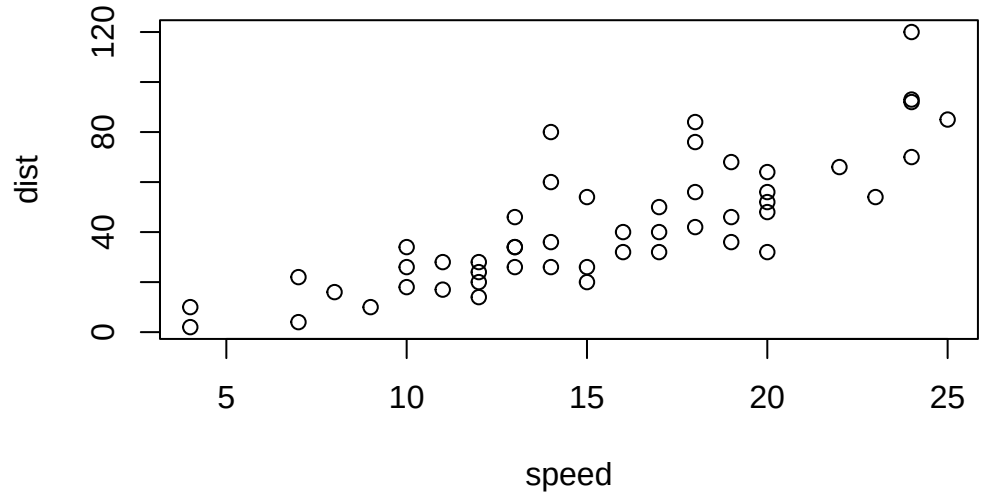
```
cars
```

```
   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)
```



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

> I will run 'install.packages("ggplot2") in my R console, not this Quarto document :)

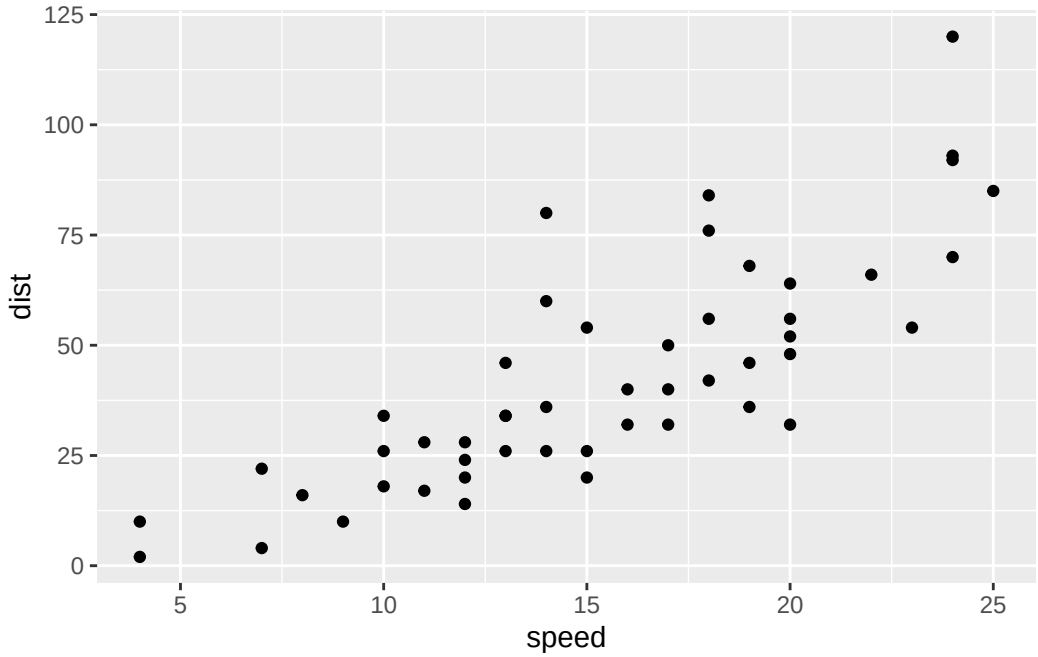
Before I can use any functions from add-on packages, I need to load the package from my "library()" with the `library(ggplot2)` call.

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

All ggplot figures have at least 3 things (called layers). These include:

- **data** (the input data set I want to plot from),
- **aes** (the aesthetic mapping of the data to my plot),
- **geoms** (the geom_point(), geom_line(), etc. that I want to draw)

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



Q5. Which geometric layer should be used to create scatter plots in ggplot2?

A. **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?

Q7. Argue with geom_smooth() to add a straight line from a linear model without the shaded standard error region?

A. Use **method="lm"** to argue with the geom_smooth function and make the line linear.

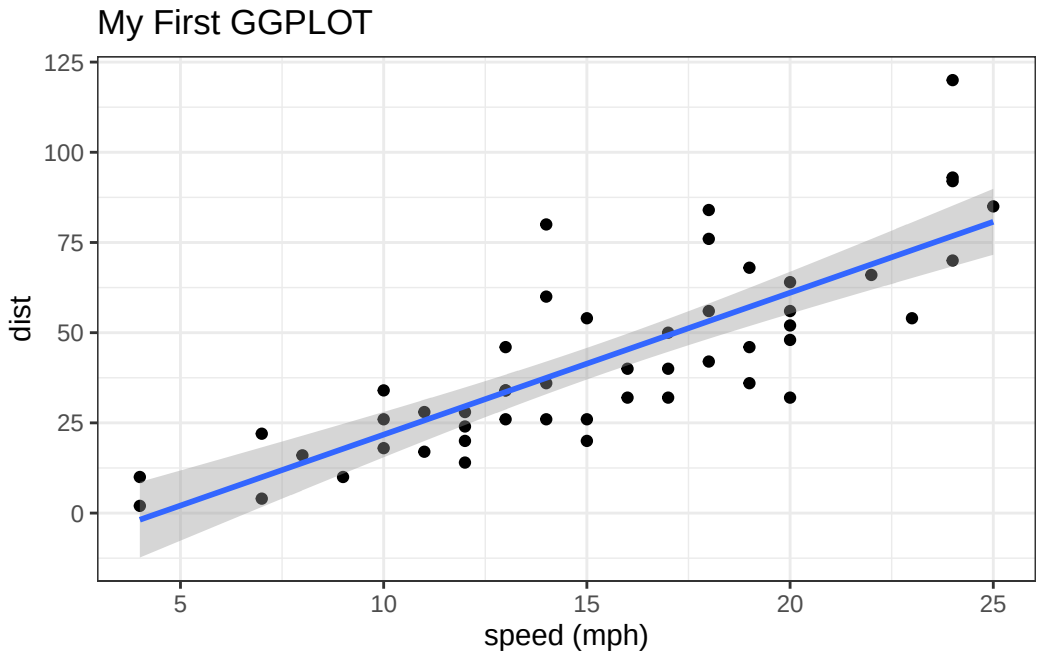
Q8. Can you finish this plot by adding various label annotations with the labs() function and changing the plot look to a more conservative "black & white" theme by adding the theme_bw() function:

Let's add a line-of-best fit to show the relationship here:

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm") +
  theme_bw() +
  labs(title="My First GGPLOT") +
```

```
  xlab("speed (mph)") +
  ylab("dist")
```

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



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)
```

```
        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
```

How many genes are in this dataset? Q9. Use the nrow() function to find out how many genes are in this dataset. What is your answer?

```
nrow(genes)
```

```
[1] 5196
```

A. There are **5196** genes in this dataset.

Q10. 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
```

A. I found **4** columns.

Q11. 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
```

A. There are **127** up-regulated genes.

Q12. 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), 3)
```

```
      down unchanging         up
     0.014      0.962      0.024
```

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

vals/n.tot
```

```
      down unchanging         up
0.01385681 0.96170131 0.02444188
```

```
round(vals/n.tot,2)
```

```
      down unchanging         up
      0.01       0.96       0.02
```

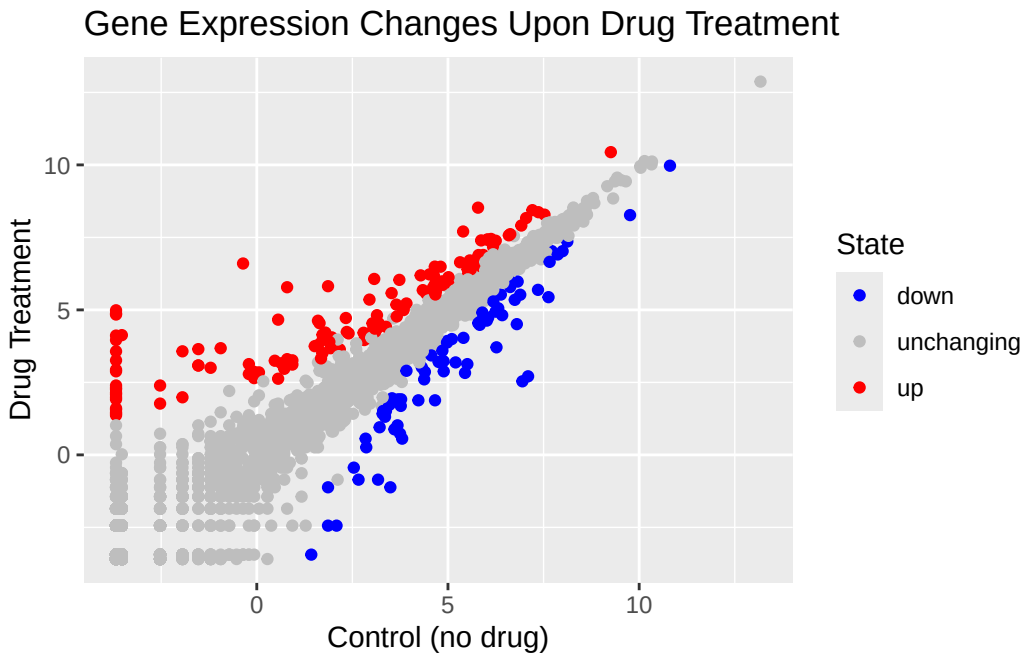
```
vals.percent<-vals/n.tot*100
round(vals.percent, 2)
```

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

A. Same, same. 2nd method a bit easier to read. Answer is **2.44%** either way (which is technically 3 sig figs).

Q. Complete the code below to produce the following plot

```
p<-ggplot(genes) +
  aes(x=Condition1, y=Condition2, col=State)+
  geom_point()+
  labs(title= "Gene Expression Changes Upon Drug Treatment")+
  xlab("Control (no drug)")+
  ylab("Drug Treatment")+
  scale_color_manual(values=c("blue", "gray", "red"))
p
```



Q13. Nice, now add some plot annotations to the p object with the labs() function so your plot looks like the following:

A. See above plot, already done >:)

Going Further (Section 7)

Working with a new dataset, use `install.packages()` with 'gapminder' in `install.packages("gapminder)` to install and then call with `library(gapminder)`.

```
library(gapminder)
```

Also install and call the dplyr package to specify part of the data and for next class in the same way.

```
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
```

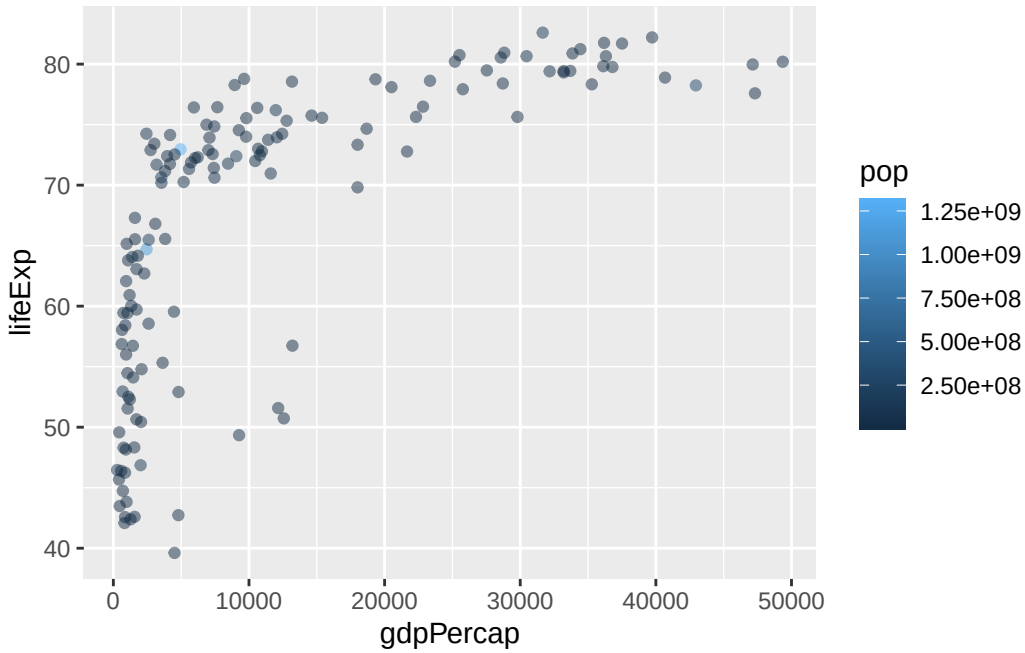
Using dplyr %>%, assign to gapminder_2007 the data from 2007.

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

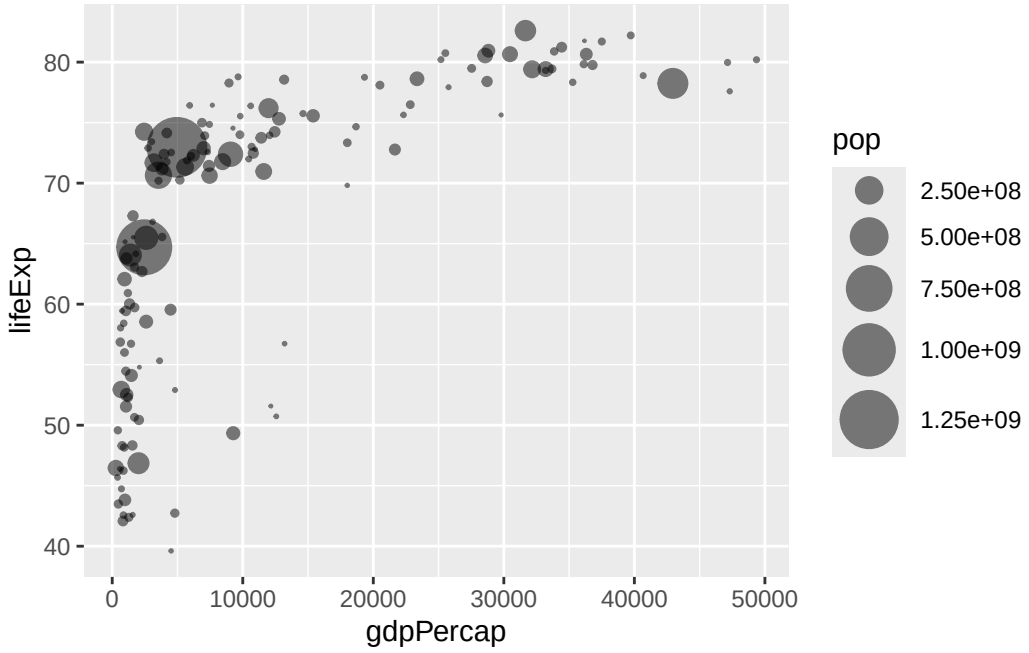
```
# A tibble: 142 x 6
   country     continent  year lifeExp       pop gdpPercap
   <fct>       <fct>     <int>   <dbl>     <int>     <dbl>
 1 Afghanistan Asia       2007    43.8  31889923      975.
 2 Albania     Europe     2007    76.4   3600523     5937.
 3 Algeria     Africa     2007    72.3  33333216     6223.
 4 Angola      Africa     2007    42.7  12420476     4797.
 5 Argentina   Americas   2007    75.3  40301927    12779.
 6 Australia   Oceania    2007    81.2  20434176    34435.
 7 Austria     Europe     2007    79.8   8199783    36126.
 8 Bahrain     Asia       2007    75.6    708573    29796.
 9 Bangladesh  Asia       2007    64.1 150448339     1391.
10 Belgium     Europe     2007    79.4  10392226    33693.
# i 132 more rows
```

Q14. Complete the code below to produce a first basic scatter plot of this gapminder_2007 dataset (comparing GDP per capita and Life Expectancy)

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




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



Q15. 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?

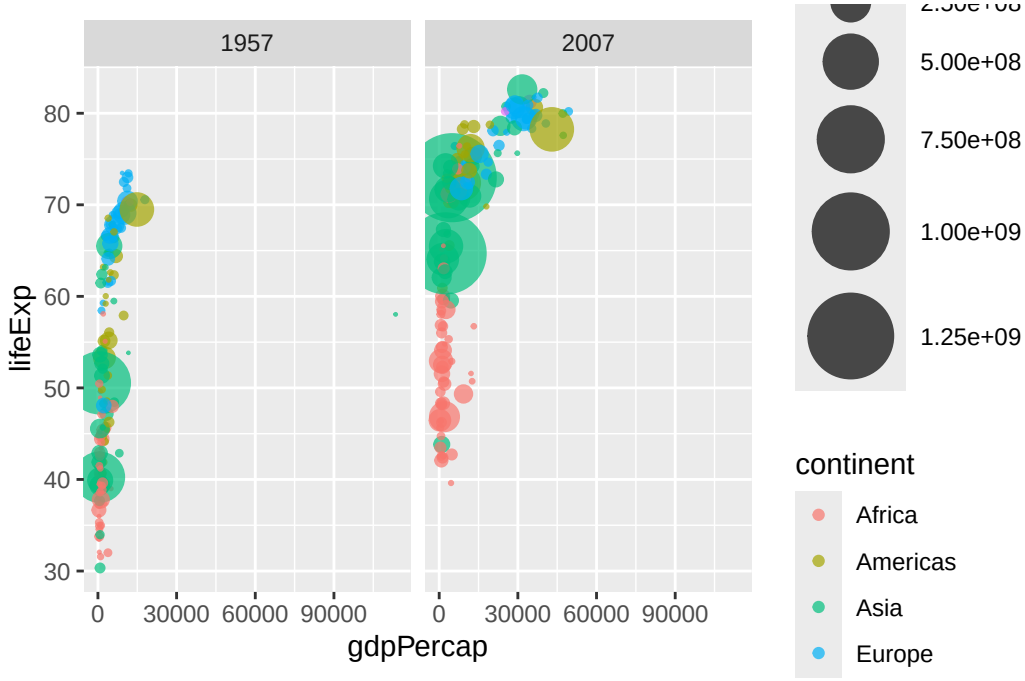
```
gapminder_1957<-gapminder %>% filter(year==1957)

p1957<-ggplot(gapminder_1957) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop)+
  geom_point(alpha=0.7)+
  scale_size_area(max_size=15)
```

Q16. 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_1957n2007<-gapminder %>% filter(year==1957| year== 2007)

p1957n2007<-ggplot(gapminder_1957n2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop)+
  geom_point(alpha=0.7)+
  scale_size_area(max_size=15)+
  facet_wrap(~year)
p1957n2007
```


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