# Class 05: Data Visualization with GGPLOT

Amanda Wilpitz (PID: A17463962)

## Background

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

## Intro to ggplot

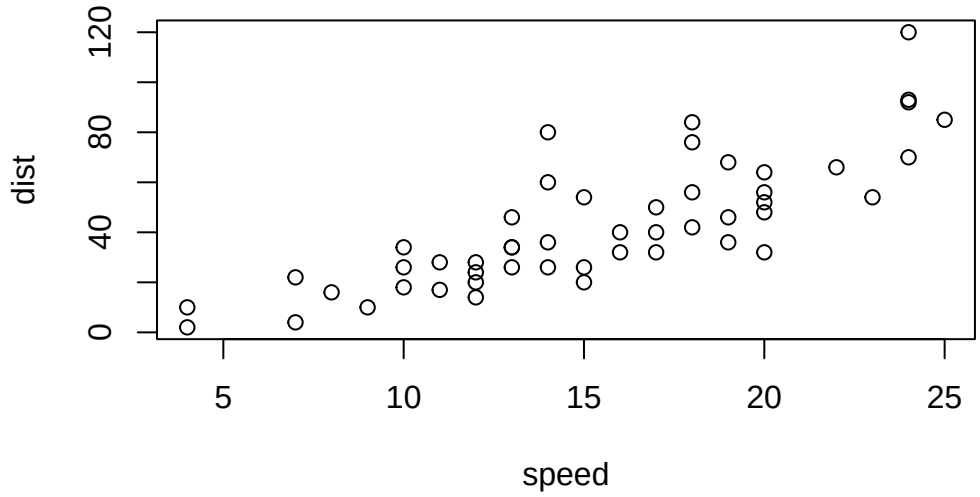
There are many graphics systems in R (ways to make plots and figures). These include "base" R plots. Today we will focus mostly on the **ggplot2** package.

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

```
head(cars)
```

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

```
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 used 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.

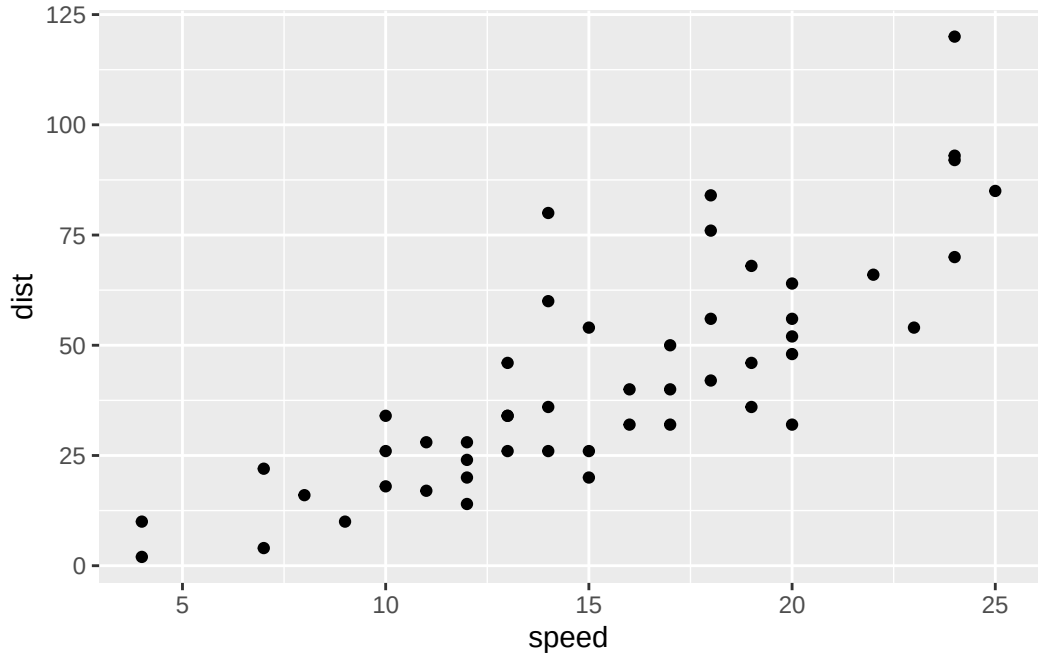
```
library(ggplot2)

ggplot(cars)
```

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

- **data** (the input dataset 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)

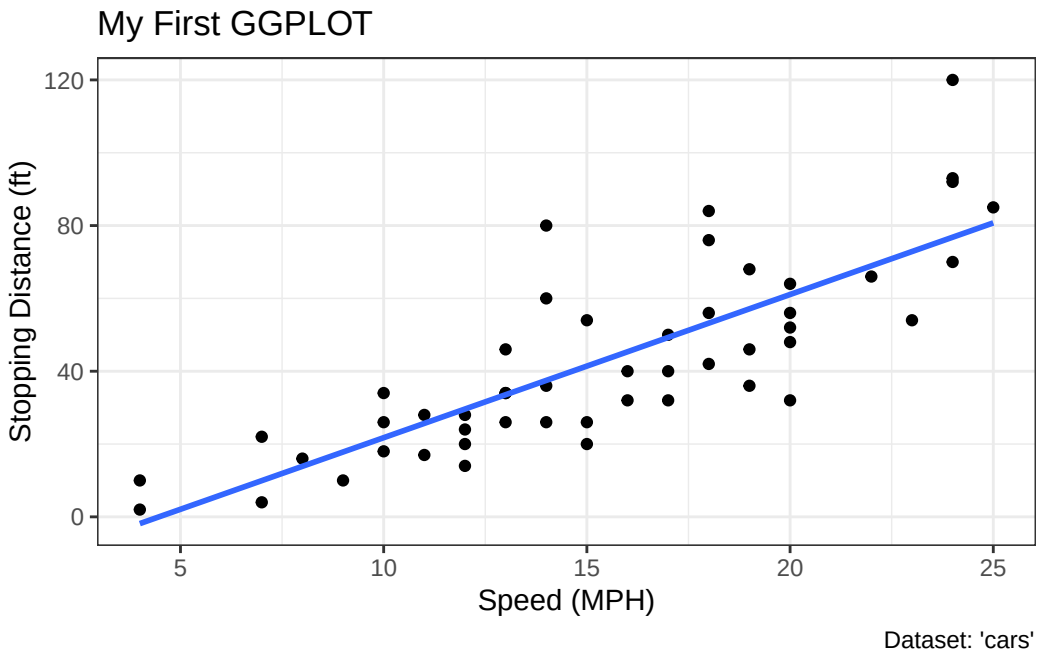
```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```




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

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  theme_bw() +
  labs(title = "My First GGPLOT", x = "Speed (MPH)", y = "Stopping Distance (ft)", caption =
```

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




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

geom_point()

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

### Gene Expression Figure

The code to read the dataset is

```
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.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?

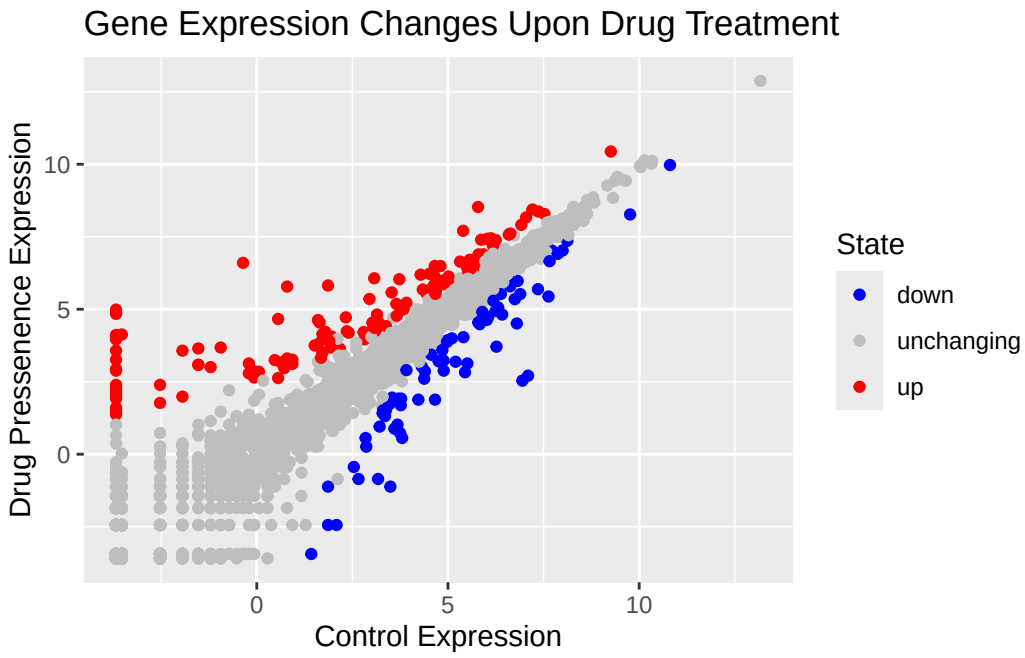
```
n.tot <- nrow(genes)
vals <- table(genes$State)

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

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

A first plot of this dataset

```
ggplot(genes) +
  aes(x = Condition1, y = Condition2, col = State) +
  geom_point() +
  scale_color_manual(values=c("blue", "grey", "red"))+
  labs(title = "Gene Expression Changes Upon Drug Treatment", x = "Control Expression", y = "
```




### Going Further

Take `gapminder` data frame and filter to contain only rows with `year` value of 2007>

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

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

```r
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
```

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

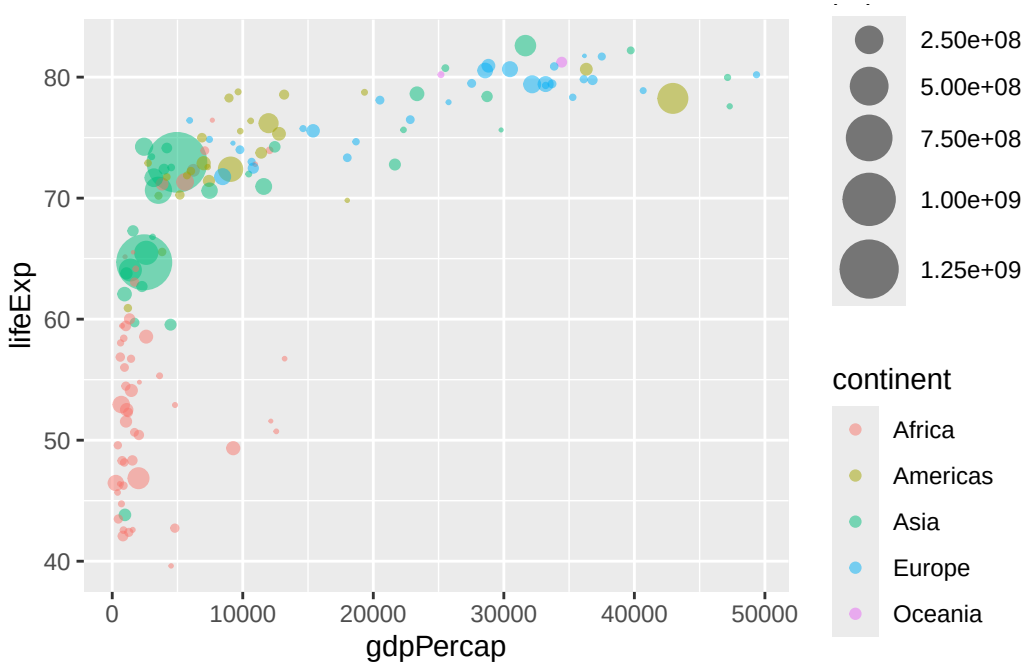
```r
head(gapminder_2007)
```

```
      country continent year lifeExp      pop  gdpPercap
1 Afghanistan      Asia 2007  43.828 31889923   974.5803
2     Albania    Europe 2007  76.423  3600523  5937.0295
3     Algeria    Africa 2007  72.301 33333216  6223.3675
4      Angola    Africa 2007  42.731 12420476  4797.2313
5   Argentina  Americas 2007  75.320 40301927 12779.3796
6   Australia   Oceania 2007  81.235 20434176 34435.3674
```

Q. Scatter plot of this gapminder_2007 dataset:

```r
plot_2007 <- ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, size = pop, col = continent) +
  geom_point(alpha=0.5) +
  scale_size_area(max_size = 10)

plot_2007
```

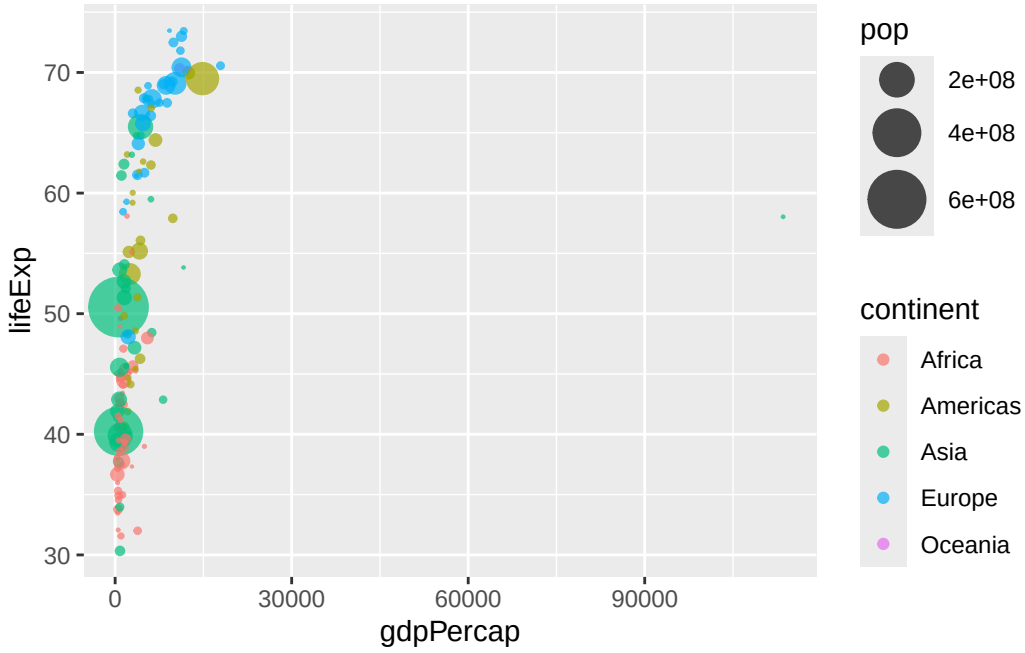



Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

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

```
plot_1957 <- ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, size = pop, col = continent) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 10)

plot_1957
```

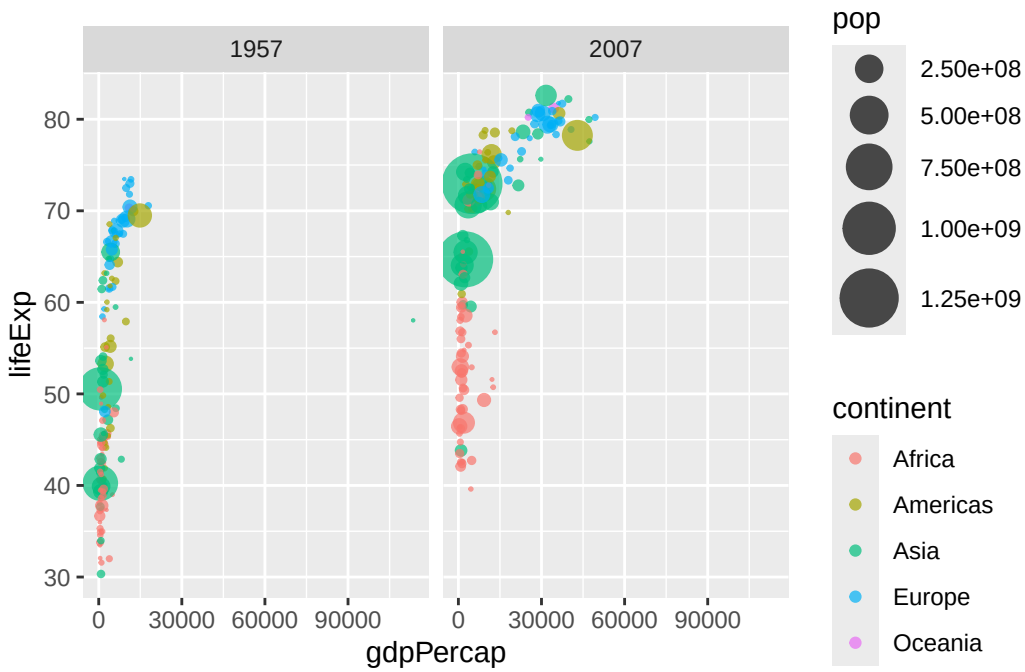



Q. Do the same steps above but include 1957 and 2007 in your input dataset for ggplot(). You should now include the layer facet_wrap(~year) to produce the following plot:

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

plot_1957_2007 <- ggplot(gapminder_1957_2007) +
  aes(x = gdpPercap, y = lifeExp, size = pop, col = continent) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)

plot_1957_2007
```


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