

# Class 5: Data Visualization with ggplot

Snehita Vallumchetla (PID A16853399)

## Background

Q1. Which plot types are typically NOT used to compare distributions of numeric variables?

Network graph

Q2. Which statement about data visualization with ggplot2 is incorrect?

ggplot is the only way to create plots in R

## Intro to ggplot

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

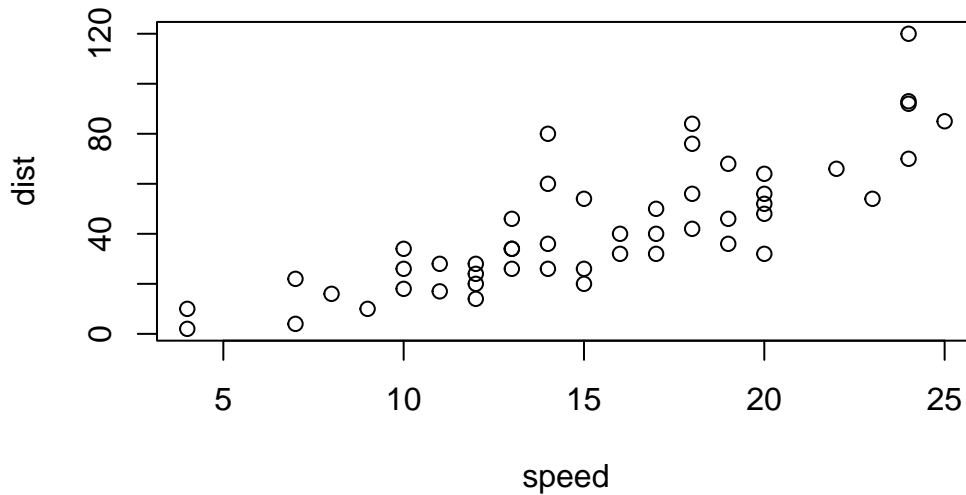
Let’s start with a plot of a simple in-built dataset called **cars**.

```
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)
```

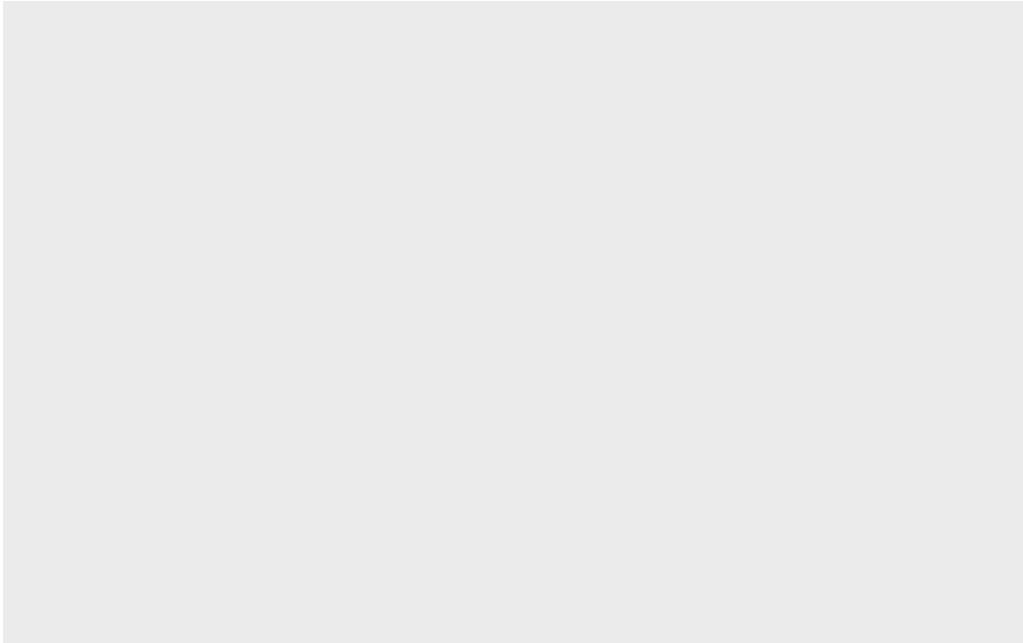


Let's see how we can make this figure using **ggplot2**. First I need to install this package on my computer. To install any R package I use the function `install.packages()`

I will run `install.packages("ggplot2")` in my R console not this quatro 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.

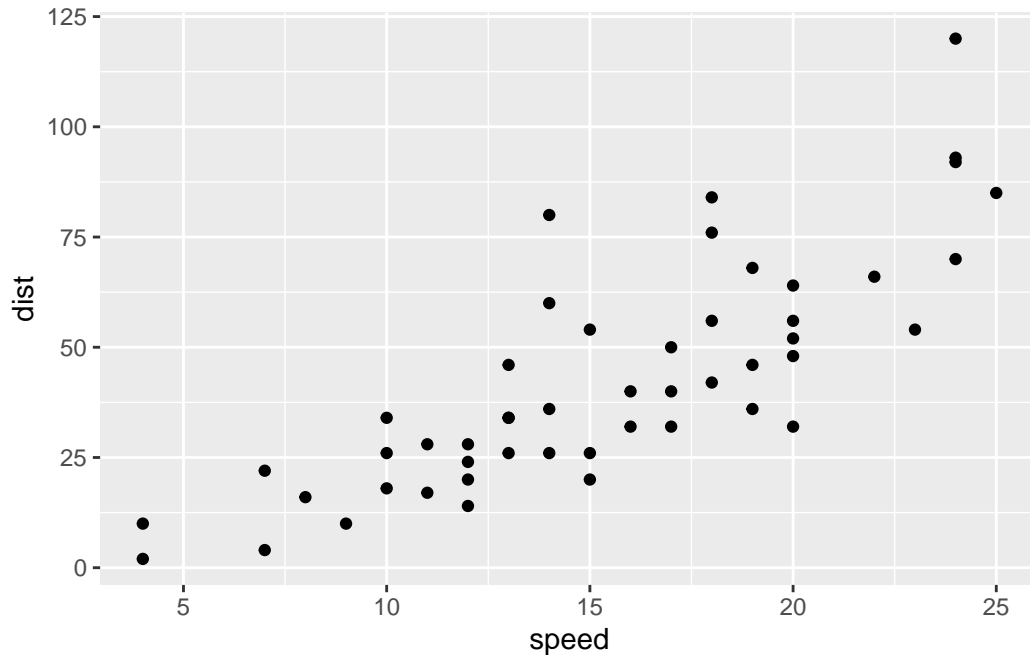
```
library(ggplot2)
ggplot(data = cars)
```



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)
- **geom** (the `geom_point()`, `geom_line()`, etc)

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



Q3. Which geometric layer should be used to create scatter plots in ggplot2?

`geom_point()`

Q4. 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?

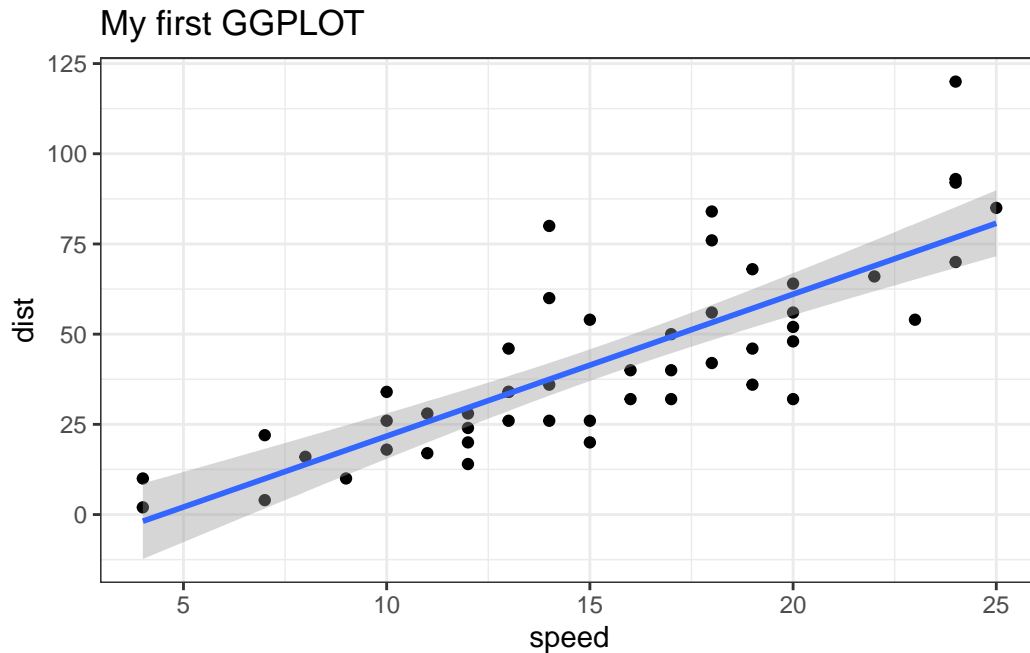
Q5. Argue with `geom_smooth()` to add a straight line from a linear model without the shaded standard error region?

Yes, using `geom_smooth(method = "lm")`

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

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

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



Q6. Which geometric layer should be used to create scatter plots in ggplot2?

`geom_point()`

### Gene expression figure

the code to read the data set

```
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

Q7. How many genes are in this dataset?

```
nrow(genes)
```

```
[1] 5196
```

There are 5196 genes in this dataset.

Q8. 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
```

There are **4 columns** in this dataset with the names `Gene`, `Condition1`, `Condition2`, and `State`

Q9. 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

There are **127** upregulated genes in this dataset

Q10. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

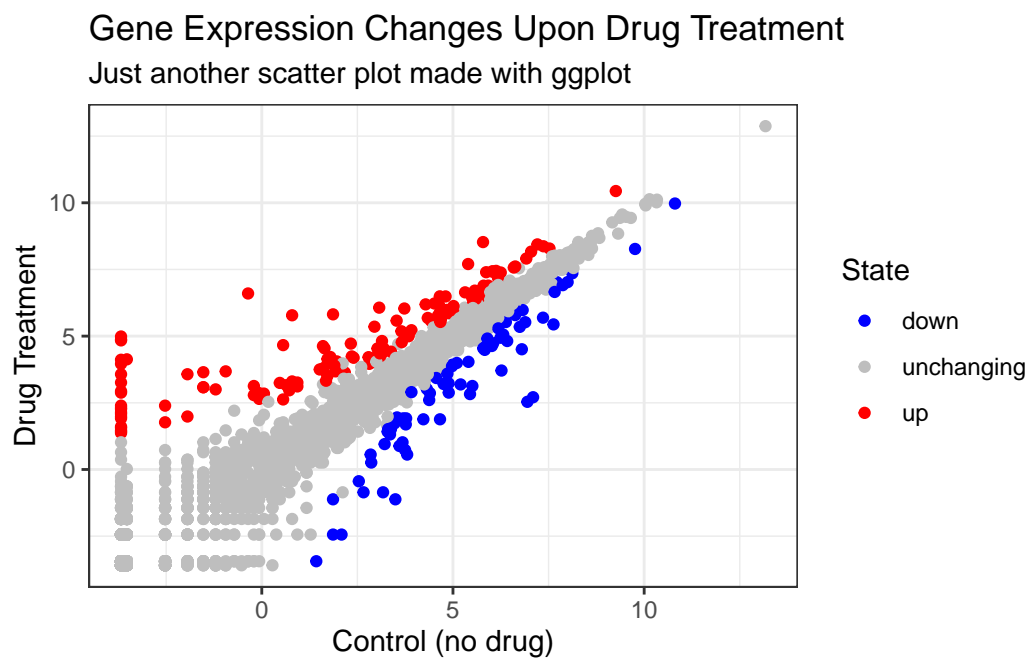
```
round( table(genes$State)/nrow(genes) * 100, 2 )
```

down	unchanging	up
1.39	96.17	2.44

The fraction of total genes that is upregulated is **2.44**.

A final plot with all of the elements (layers):

```
ggplot(genes) +
  aes(x = Condition1, y = Condition2, col = State) +
  geom_point() +
  theme_bw() +
  labs(title = "Gene Expression Changes Upon Drug Treatment",
       x = "Control (no drug)",
       y = "Drug Treatment",
       subtitle = "Just another scatter plot made with ggplot") +
  scale_colour_manual(values = c("blue", "grey", "red"))
```



Gapminder Data Set



```
#read in the file of interest
```

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

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

```
# install necessary packages
```

```
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

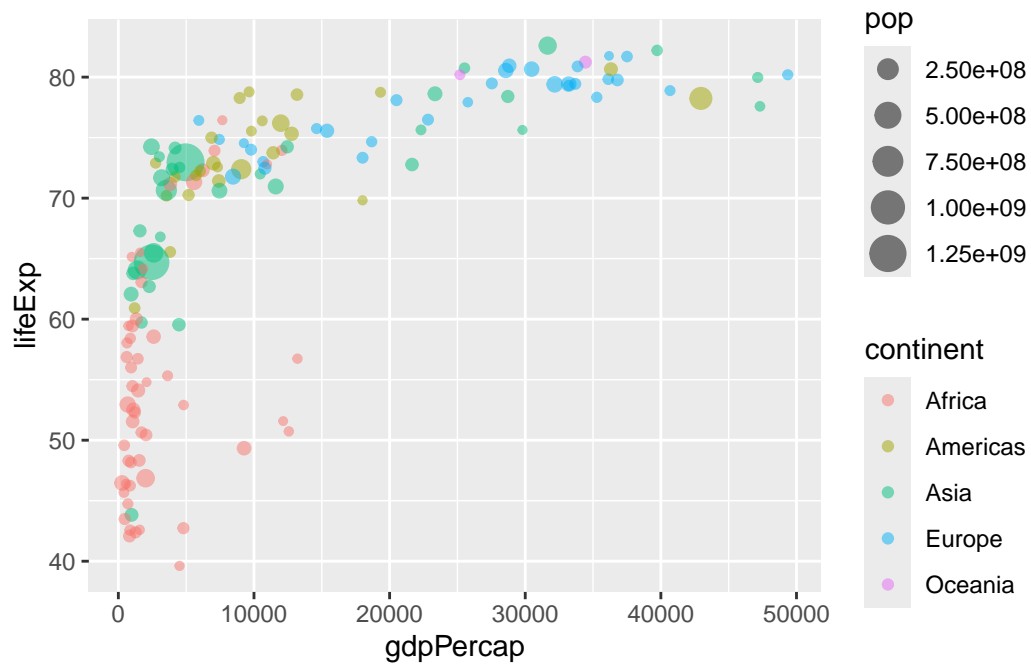
intersect, setdiff, setequal, union

Create a new dataset which filters data for the year 2007

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

Create a plot of the 2007 data

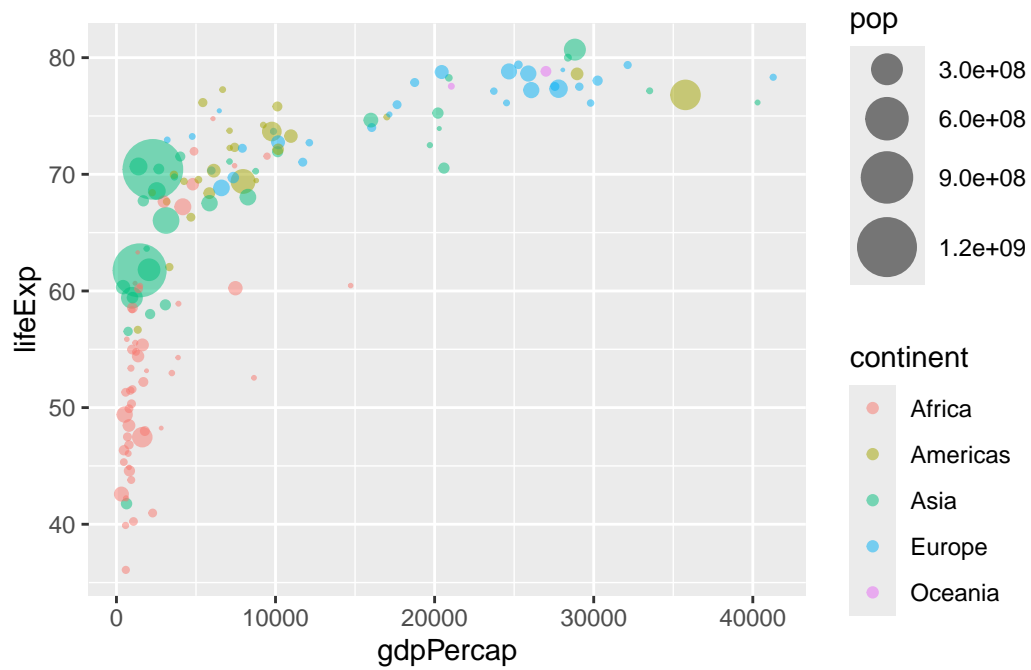
```
ggplot(gapminder_2007) +  
  aes(x = gdpPercap, y = lifeExp, color = continent, size = pop) +  
  geom_point(alpha = 0.5)
```



Side by side figure

```
gapminder_1997 <- gapminder %>% filter(year == 1997)

ggplot(gapminder_1997) +
  aes(x = gdpPercap, y = lifeExp, color=continent, size = pop) +
  geom_point(alpha = 0.5) +
  scale_size_area(max_size = 10)
```



```
gapminder_1997_2007 <- gapminder %>% filter(year==1997 | year==2007)

ggplot(gapminder_1997_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```

