

# Class05:Data Visualization with GGLOT

Shivani Lakkaraju

## Table of contents

RNAseq plot with more aes() values . . . . .	4
combining plots . . . . .	8

#Graphs and plots in R

R has tons of different graphics systems. These include “**base R**” (e.g. the ‘plot()’ function) and add on packages like **ggplot2**.

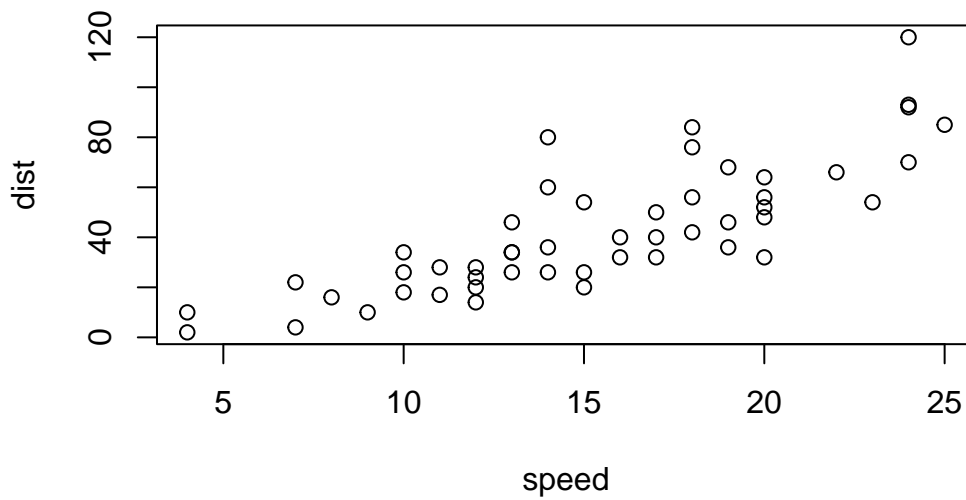
Let’s start with plotting a simple dataset in “base R” and then ggplot2 to see how they differ.

```
head(cars)
```

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

To plot this in base R I just use ‘plot()’

```
plot(cars)
```

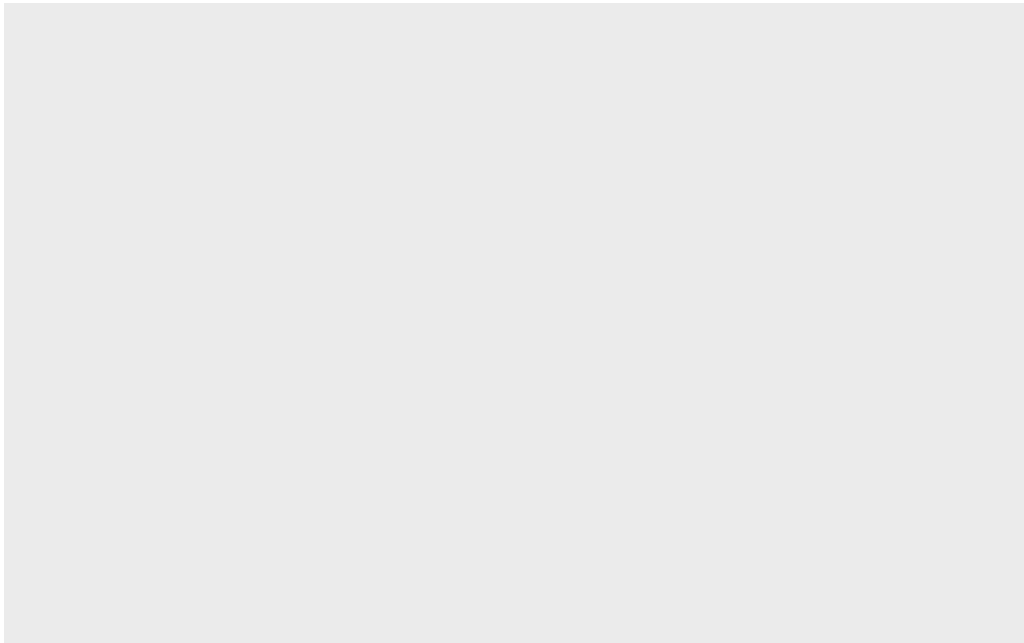


First to use ggplot2 I need to install the package. For this I use the ‘install.packages()’ function.

I will not run ‘install.packages()’ in my quarto document as this would reinstall the package every time I render the document.

The main function in the ggplot2 package is ‘ggplot()’. Before I can use it this function I need to load the package with a ‘library()’ call.

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



There are at least 3 things every ggplot needs:

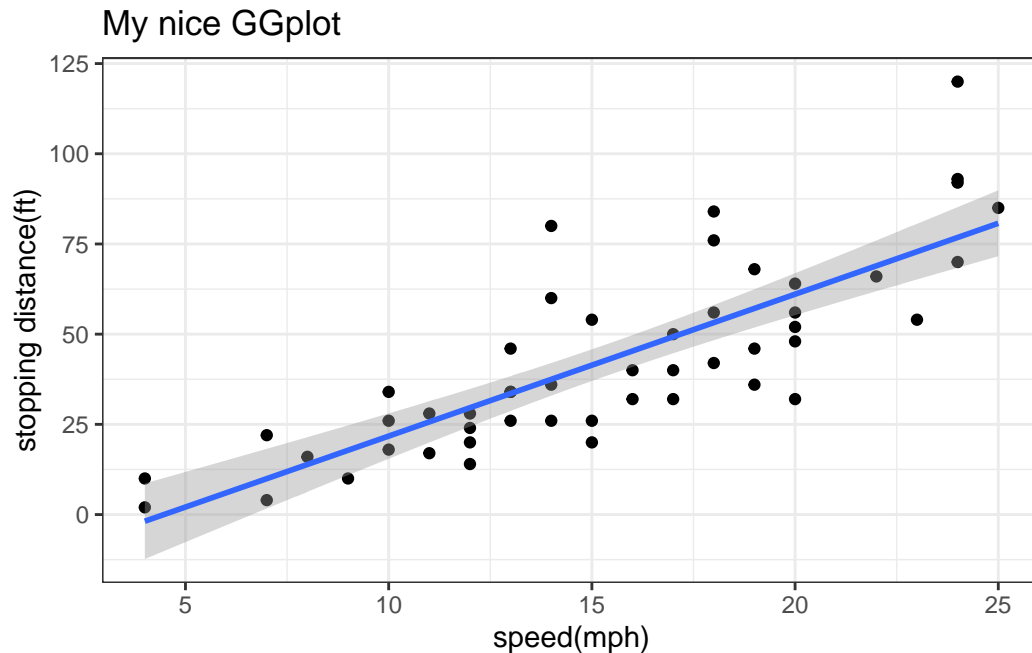
- dataset i want to plot: **data**
- aesthetic mapping: **aes**
- geometries to style: **geoms**

adding layers makes more complicated plots:

```
c <- ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  geom_smooth(method = "lm") +  
  labs()
```

```
c + labs(title="My nice GGplot", x="speed(mph)", y="stopping distance(ft)") +  
  theme_bw()
```

`geom\_smooth()` using formula = 'y ~ x'



### RNAseq plot with more aes() values

```
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. Use the nrow() function to find out how many genes are in this dataset. What is your answer? 5196

```
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? 4

```
ncol(genes)
```

```
[1] 4
```

```
colnames(genes)
```

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

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? 127

```
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) * 100, 2 )
```

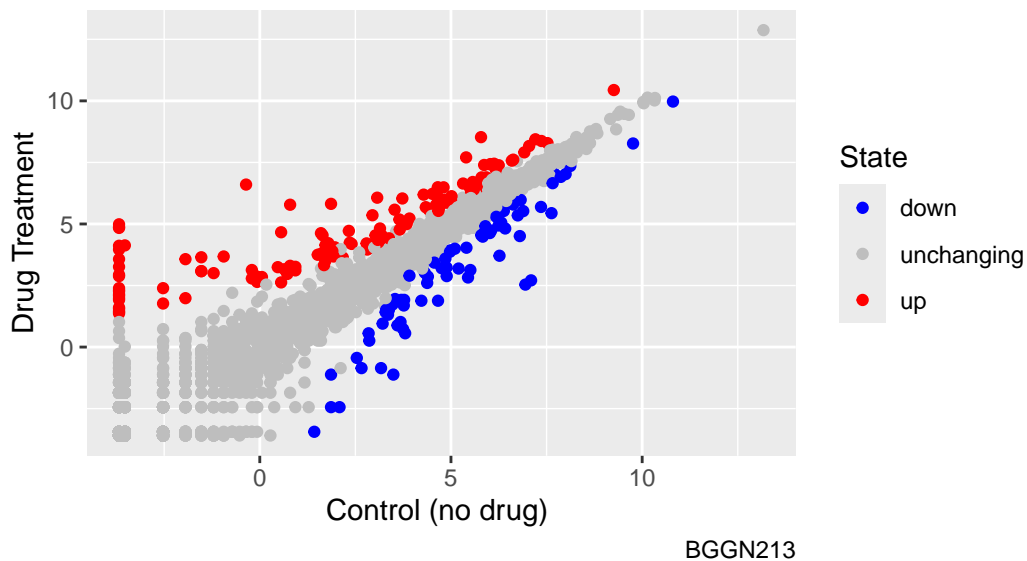
down	unchanging	up
1.39	96.17	2.44

Plot this dataset

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

## Gene Expression Changes Upon Drug Treatment

RNAseq data



##Gapminder dataset

We can get exposure to setting more 'aes()' parameters with datasets that include more columns of useful data. For example the **gapminder** dataset on GPD and life expectancy for different countries over time.

```
library(gapminder)
nrow(gapminder)
```

```
[1] 1704
```

```
head(gapminder)
```

```
# A tibble: 6 x 6
  country    continent  year lifeExp    pop gdpPercap
  <fct>      <fct>    <int>  <dbl>   <int>   <dbl>
1 Afghanistan Asia      1952   28.8  8425333    779.
2 Afghanistan Asia      1957   30.3  9240934    821.
3 Afghanistan Asia      1962   32.0 10267083    853.
4 Afghanistan Asia      1967   34.0 11537966    836.
5 Afghanistan Asia      1972   36.1 13079460    740.
6 Afghanistan Asia      1977   38.4 14880372    786.
```

how many unique countries?

```
length(unique(gapminder$country))
```

```
[1] 142
```

what years does the dataset cover?

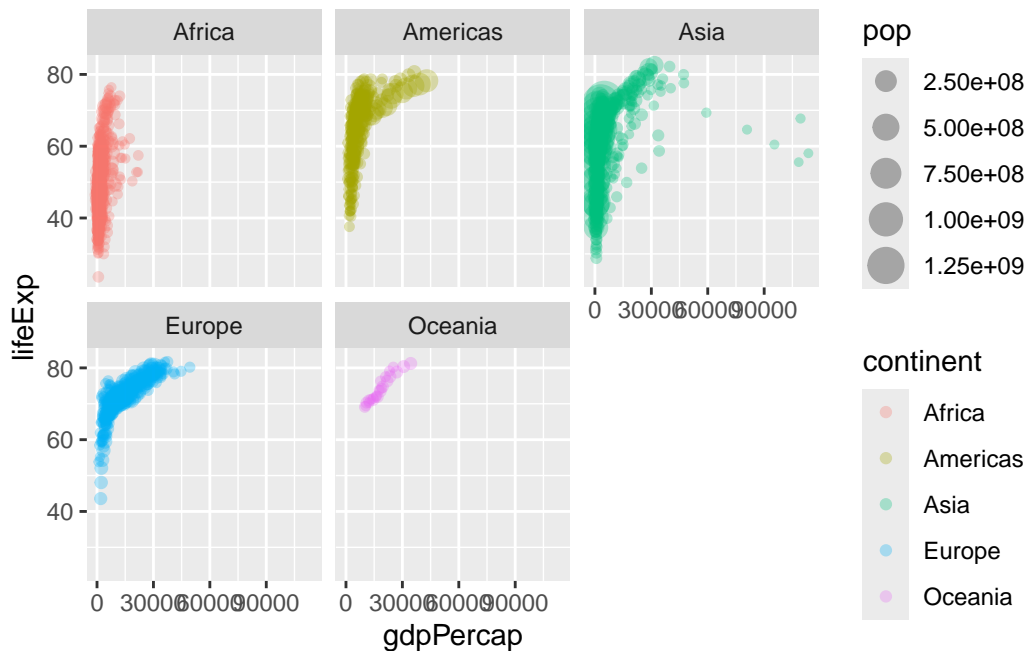
```
tail(gapminder)
```

```
# A tibble: 6 x 6
  country continent year lifeExp      pop gdpPercap
  <fct>      <fct>    <int>   <dbl>   <int>    <dbl>
1 Zimbabwe Africa     1982    60.4  7636524    789.
2 Zimbabwe Africa     1987    62.4  9216418    706.
3 Zimbabwe Africa     1992    60.4 10704340    693.
4 Zimbabwe Africa     1997    46.8 11404948    792.
5 Zimbabwe Africa     2002    40.0 11926563    672.
6 Zimbabwe Africa     2007    43.5 12311143    470.
```

this dataset covers 1952 - 2007

useful key functions: - nrow() - ncol() - length() - unique() - table()

```
ggplot(gapminder, aes(gdpPercap, lifeExp, col=continent, size = pop)) + geom_point(alpha=0.3)
  facet_wrap(~continent)
```



## combining plots

```
library(patchwork)

# Setup some example plots
p1 <- ggplot(mtcars) + geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) + geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) + geom_smooth(aes(dis, qsec))
p4 <- ggplot(mtcars) + geom_bar(aes(carb))

# Use patchwork to combine them here:
(p1 | c ) /
  (p4 | p3)
```

```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



