

Class 5: Data Vis with ggplot

Farnam Tavakoli (A17628539)

Graphics system in R

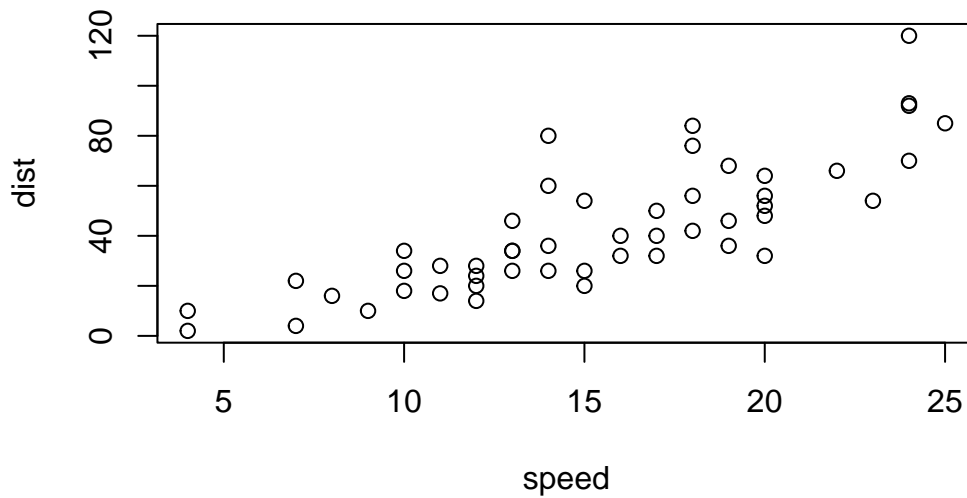
There are many graphics systems in R for making plots and figures.

we have already played a little with “**base R**”is graphics and the `plot()` function.

Today we will start learning about a popular graphics package called `ggplot2()`.

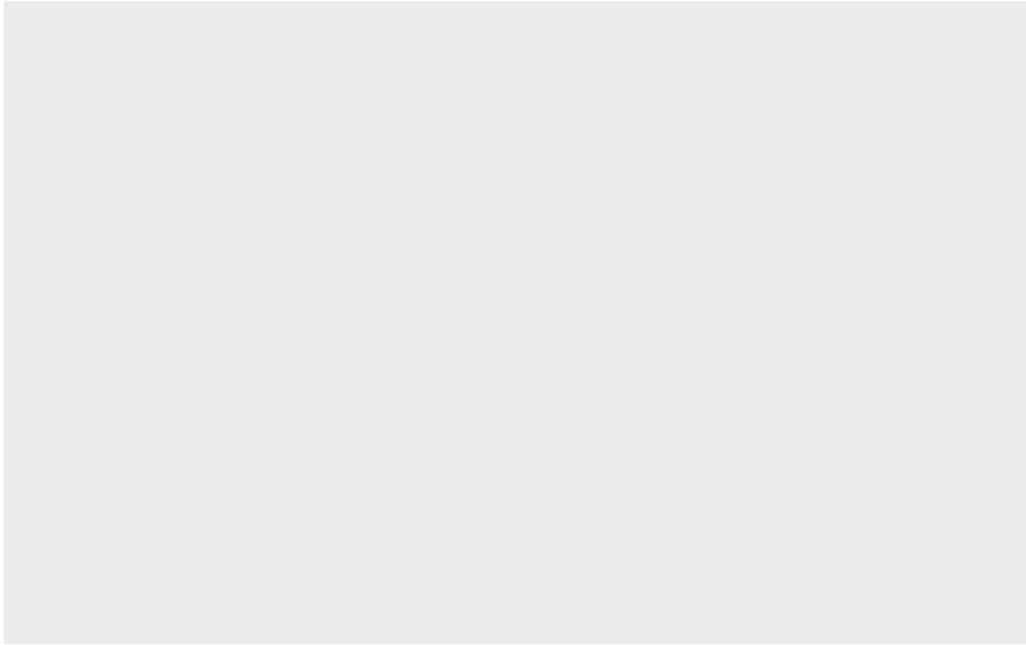
This is an add on package - i.e. we need to install it. I install it (like I install any package) with the `install.packages()` function.

```
plot(cars)
```



Before I can use the functions from a package I have to load up the package from my “library”. We use the `library(ggplot2)` command to load it up.

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



Every ggplot is made of at least 3 things: - data (the number etc. that will go into your plot) - aesthetics (how the columns of data map to the plot aesthetics) - geoms (how the plot actually looks, points, bars, lines etc.)

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



for simple plots ggplot is more verbose - it takes more code than base R plot.

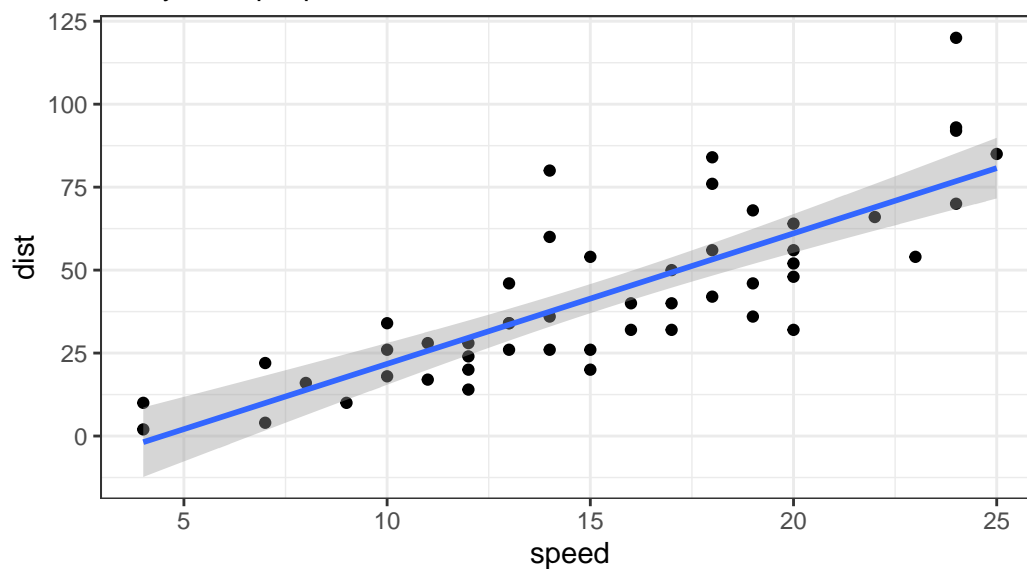
add some more layers to our ggplot:

```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  geom_smooth(method="lm") +  
  labs(title="stopping distance of old cars",  
        subtitle = "A silly example plot") +  
  theme_bw()
```

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

stopping distance of old cars

A silly example plot



```
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?

```
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

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

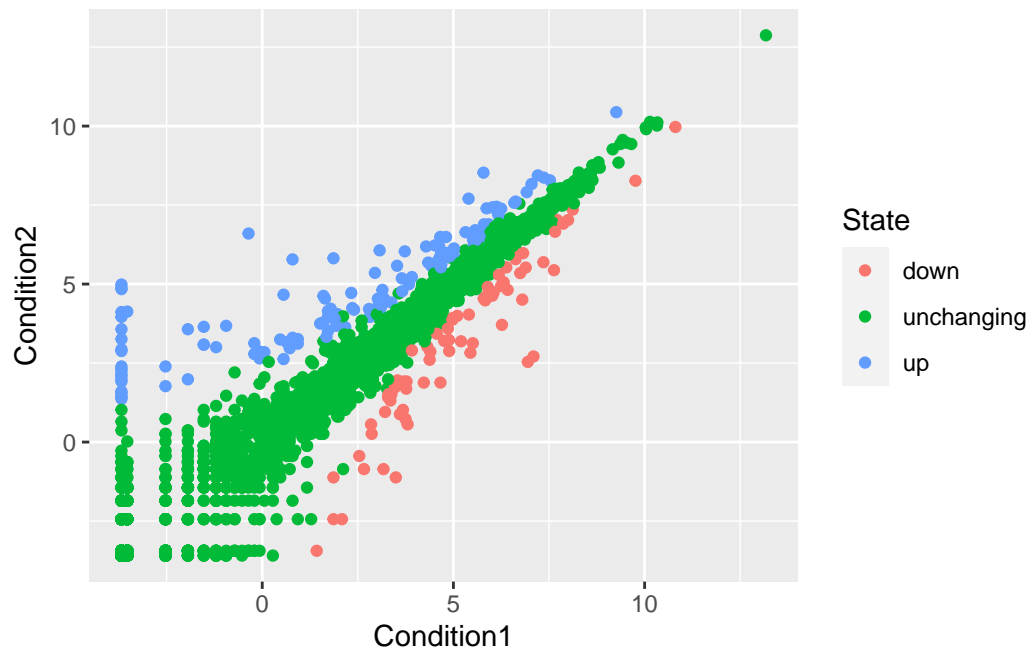
```
table(genes$State) / nrow(genes)
```

down	unchanging	up
0.01385681	0.96170131	0.02444188

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

down	unchanging	up
1.39	96.17	2.44

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



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

gapminder <- read.delim(url)

# install.packages("dplyr") ## un-comment to install if needed
library(dplyr)
```

Attaching package: 'dplyr'

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

filter, lag

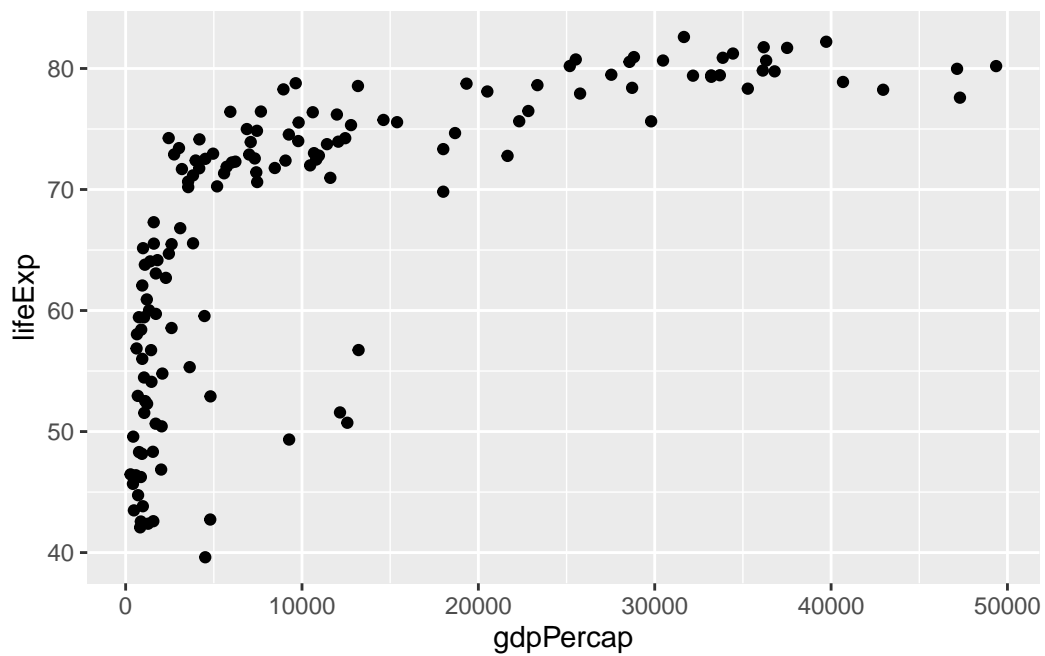
The following objects are masked from 'package:base':

```
intersect, setdiff, setequal, union
```

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

Q. Complete the code below to produce a first basic scatter plot of this gapminder_2007 dataset:

```
ggplot(gapminder_2007) +  
  aes(x=gdpPerCap, y=lifeExp) +  
  geom_point()
```



```
ggplot(gapminder_2007) +  
  aes(x=gdpPerCap, y=lifeExp) +  
  geom_point(alpha=0.5)
```

