Updated!

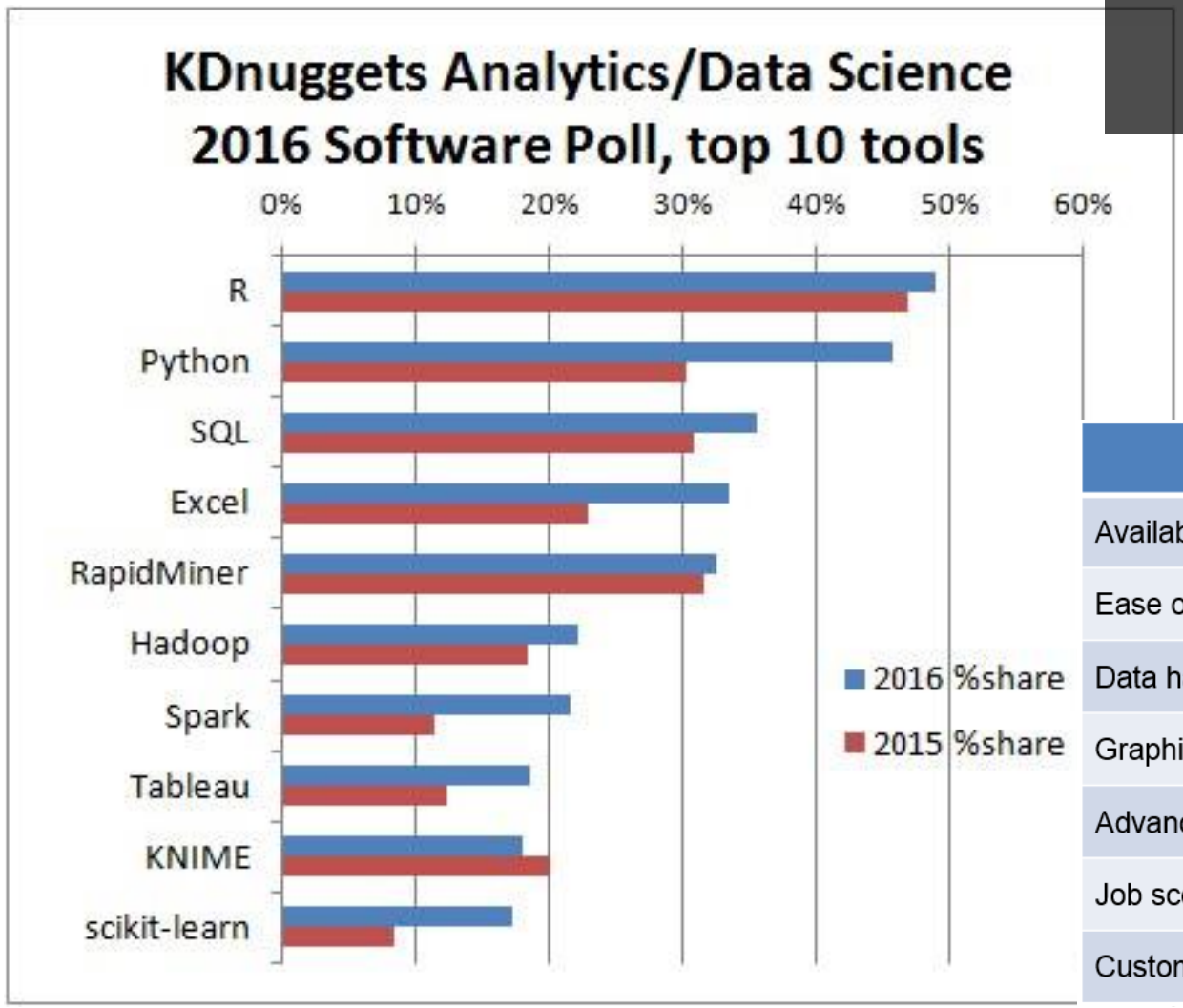
R Basics for Starters

24 JUNE 2017 www.facebook.com/datarockie


Data
is the sword of the 21st century, and those who wield it the samurai.
Jonathan Rosenberg

Why use R?

Free, Powerful, Strong community

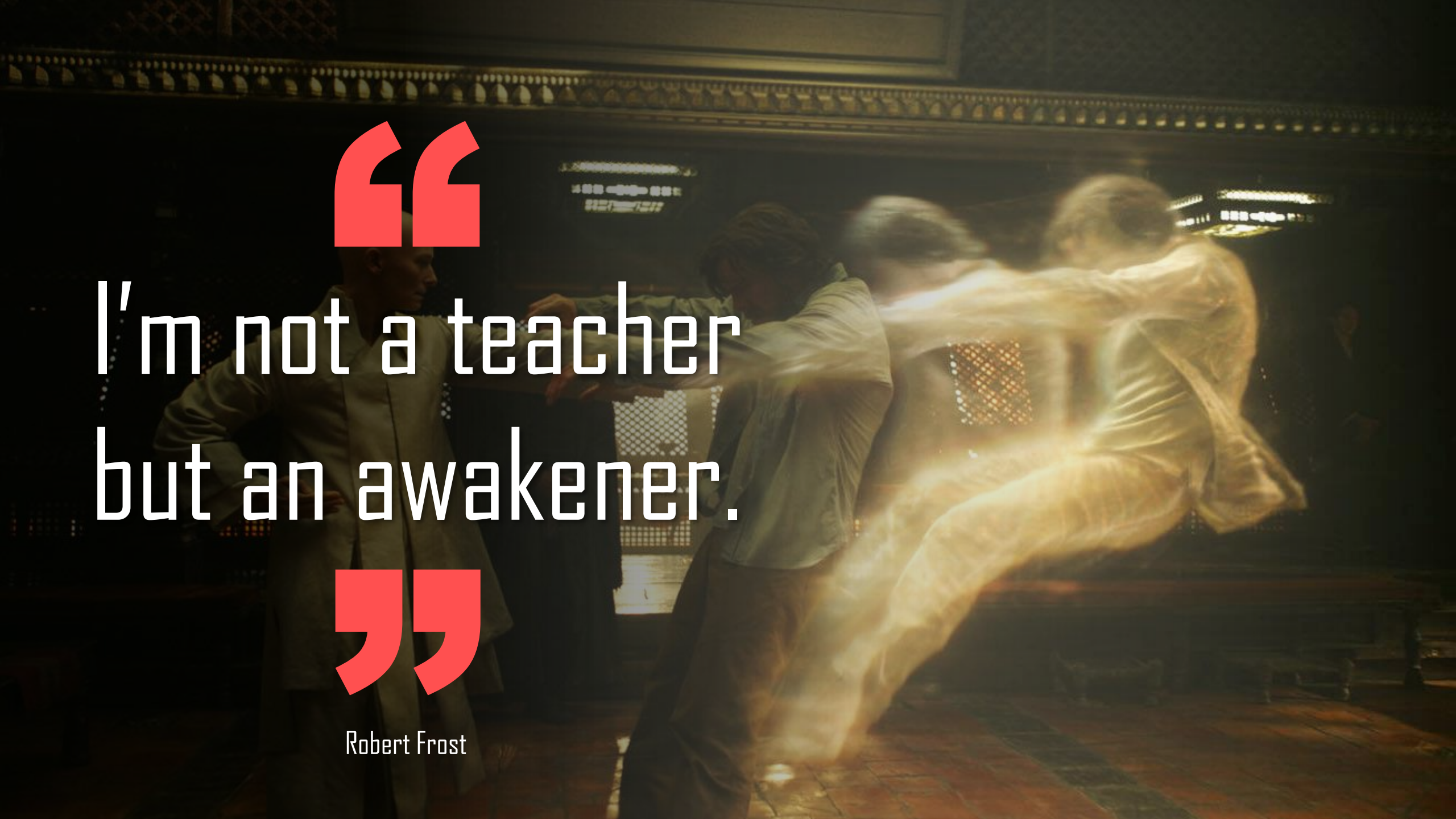


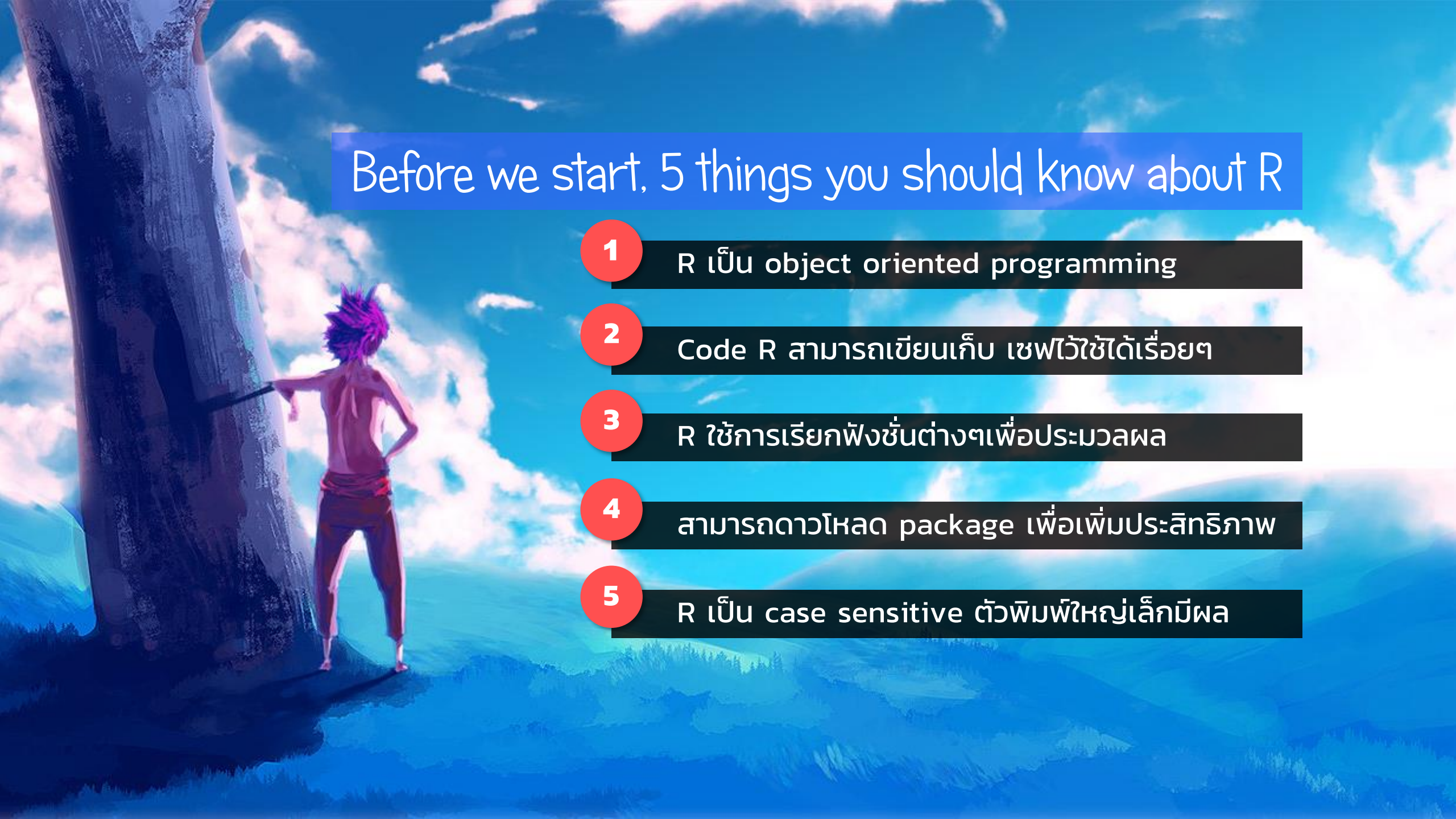
Parameter	SAS	R	Python
Availability / Cost	2	5	5
Ease of learning	4.5	2.5	3.5
Data handling capabilities	4	4	4
Graphical capabilities	3	4.5	4
Advancements in tool	4	4.5	4
Job scenario	4.5	3.5	2.5
Customer service support and Community	4	3.5	3

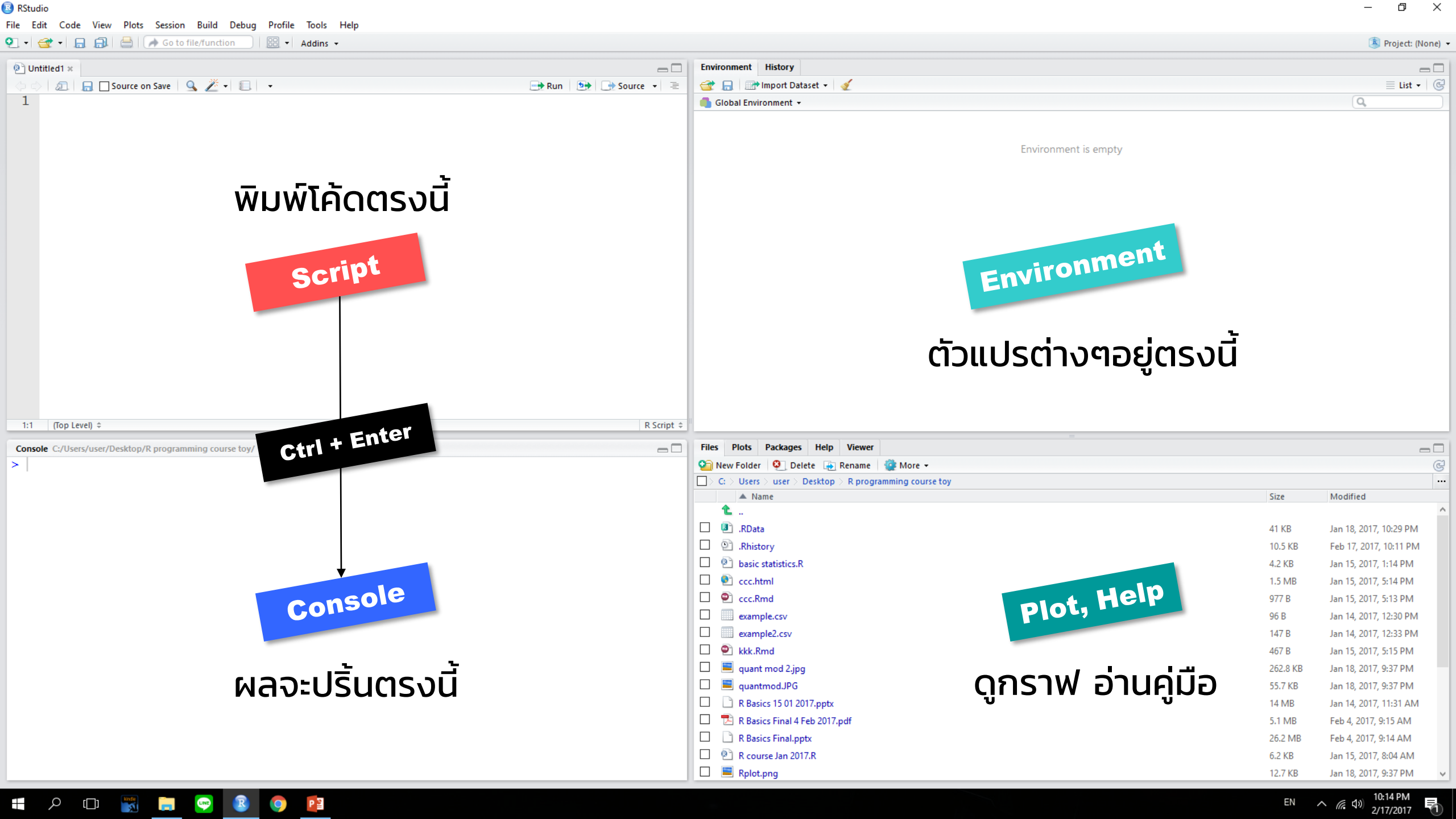
http://www.kdnuggets.com/2016/06/r-python-top-analytics-data-mining-data-science-software.html

https://www.analyticsvidhya.com/blog/2014/03/sas-vs-vs-python-tool-learn/


Fear Not
Insights and persistence are what counts
Pedro Domingos

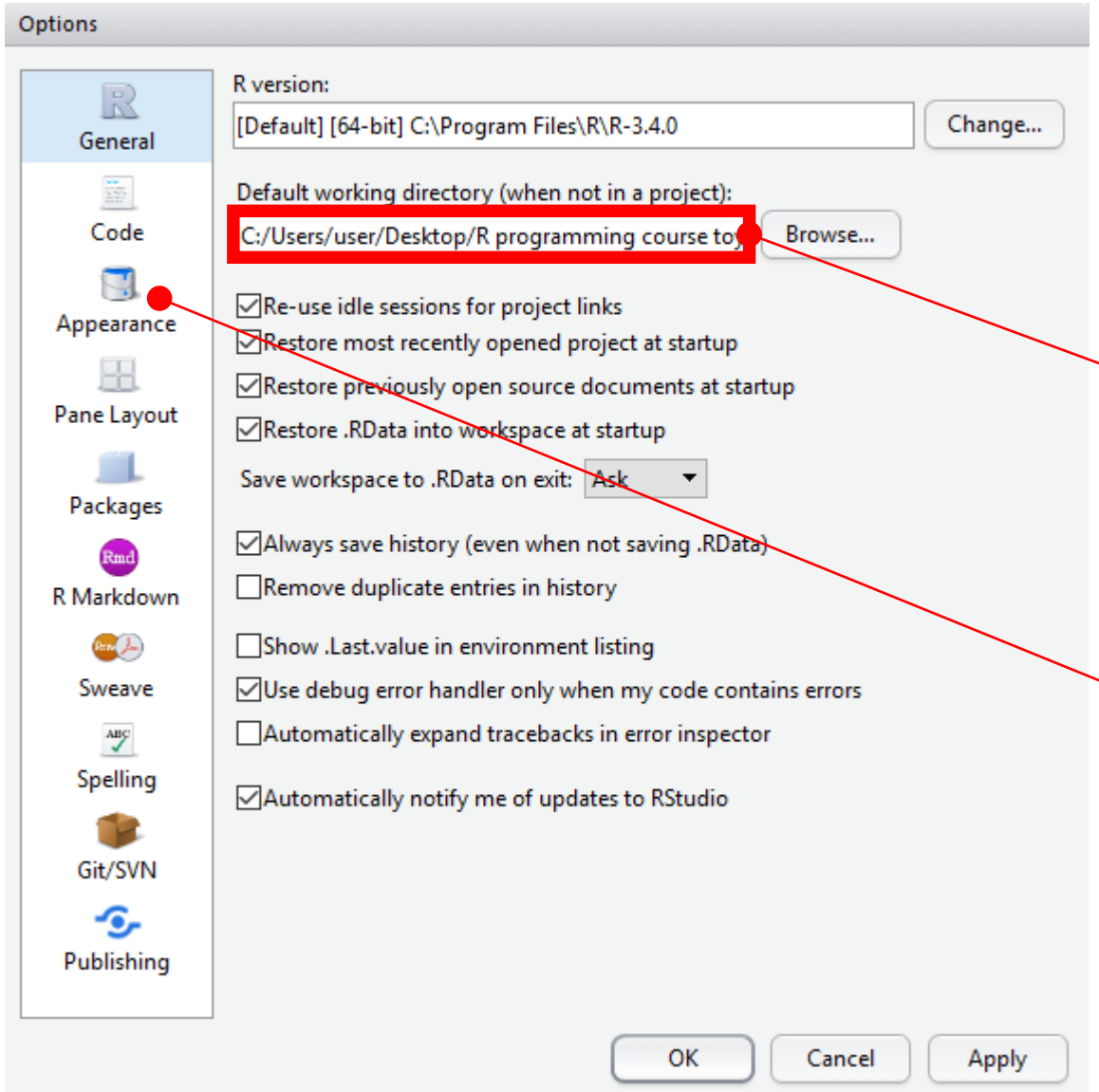

“
I’m not a teacher
but an awakener.
”
Robert Frost


Before we start, 5 things you should know about R
1
R เป็น object oriented programming
2
Code R สามารถเขียนเก็บ เซฟไว้ใช้ได้เรื่อยๆ
3
R ใช้การเรียกฟังชั่นต่างๆเพื่อประมวลผล
4
สามารถดาวโหลด package เพื่อเพิ่มประสิทธิภาพ
5
R เป็น case sensitive ตัวพิมพ์ใหญ่เล็กมีผล


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1
Source on Save
Run
Source
1
พิมพ์โค้ดตรงนี้
Script
Ctrl + Enter
Console
ผลจะปริ้นตรงนี้
1:1
(Top Level)
R Script
Console C:/Users/user/Desktop/R programming course toy/
Environment
History
Import Dataset
List
Global Environment
Environment is empty
Environment
ตัวแปรต่างๆอยู่ตรงนี้
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
C: Users user Desktop R programming course toy
Name
Size
Modified
.RData 41 KB Jan 18, 2017, 10:29 PM
.Rhistory 10.5 KB Feb 17, 2017, 10:11 PM
basic statistics.R 4.2 KB Jan 15, 2017, 1:14 PM
ccc.html 1.5 MB Jan 15, 2017, 5:14 PM
ccc.Rmd 977 B Jan 15, 2017, 5:13 PM
example.csv 96 B Jan 14, 2017, 12:30 PM
example2.csv 147 B Jan 14, 2017, 12:33 PM
kkk.Rmd 467 B Jan 15, 2017, 5:15 PM
quant mod 2.jpg 262.8 KB Jan 18, 2017, 9:37 PM
quantmod.JPG 55.7 KB Jan 18, 2017, 9:37 PM
R Basics 15 01 2017.pptx 14 MB Jan 14, 2017, 11:31 AM
R Basics Final 4 Feb 2017.pdf 5.1 MB Feb 4, 2017, 9:15 AM
R Basics Final.pptx 26.2 MB Feb 4, 2017, 9:14 AM
R course Jan 2017.R 6.2 KB Jan 15, 2017, 8:04 AM
Rplot.png 12.7 KB Jan 18, 2017, 9:37 PM
Plot, Help
ดูกราฟ อ่านคู่มือ
EN
10:14 PM
2/17/2017

Setup your R studio :D

Basic



ก่อนเริ่มใช้ RStudio ครั้งแรก ให้ทุกคนเข้าไปที่ **Tool > Options** แล้วตั้งค่าต่างๆก่อนครับ

- ตั้งค่า working directory ตรงนี้
- ปรับหน้าตาของ RStudio ได้ตรงนี้เลย

พิมพ์ `getwd( )` เพื่อเรียกดู directory

R เป็นเหมือนเครื่องคิดเลขเบื้องต้น

```
Console C:/Users/user/Desktop/R programming course toy/
> 1+1
[1] 2
> 2+2
[1] 4
> 2*10
[1] 20
> 10+10*100
[1] 1010
```

เราสร้าง object ใน R ด้วยเครื่องหมาย assign operator <-

```
Console C:/Users/user/Desktop/R programming course toy/
> a <- 10
> b <- 20
> a + b
[1] 30
> a * b
[1] 200
> (a * b) + 1000
[1] 1200
```

`<-` is assign operator, we can use `=` as well

R can handle a number of data types

- Numeric
- Integer
- Character
- Factor
- Logical
- Matrix
- Data Frame
- Date
- Time

ใช้ฟังชั่น class( ) ในการตรวจสอบ object ที่เราสร้างขึ้นมา

```
Console C:/Users/user/Desktop/R programming course toy/
> class(1)
[1] "numeric"
> class(1L)
[1] "integer"
> class("Hello World!")
[1] "character"
> class(TRUE)
[1] "logical"
> class(mat)
[1] "matrix"
> class(mtcars)
[1] "data.frame"
```

Useful Functions

is.numeric( )	as.numeric( )
is.logical( )	as.logical( )
is.character( )	as.character( )
is.matrix( )	as.matrix( )
is.data.frame( )	as.data.frame( )

Create vectors and do some calculations

Basic

ใช้ฟังชั่น `c ( )` ในการสร้าง vector

```
Console C:/Users/user/Desktop/R programming course toy/
> a = c(1:5)
> b = c(6:10)
> a
[1] 1 2 3 4 5
> b
[1]  6  7  8  9 10
> a+b
[1]  7  9 11 13 15
> a-b
[1] -5 -5 -5 -5 -5
> a*b
[1]  6 14 24 36 50
> c = c(a,b)
> c
 [1]  1  2  3  4  5  6  7  8  9 10
```

Element-wise computation

Useful Functions

c( ) stands for concatenate

seq(from = , to = , by =)

rep(x, times =)

Logical operators in R

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Exactly equal to
!=	Not equal to
!x	Not x
x | y	x or y
x & y	x and y
isTRUE(x)	Tests whether x is TRUE

Source: R in action (2nd Edition)

R ใช้เครื่องหมาย == เพื่อเช็คว่าทั้งสองฝั่งมีค่าเท่ากันไหม

```
Console C:/Users/user/Desktop/R programming course toy/
> pi == 3
[1] FALSE
> pi == 3.14
[1] FALSE
> pi == pi
[1] TRUE
> 5/2 == 2.5
[1] TRUE
> "Hello" == "hello"
[1] FALSE
> NA == 100
[1] NA
> NA == NA
[1] NA
```

R = case sensitive

NA = missing value

Three ways to subset data in R

Basic

```
Console C:/Users/user/Desktop/R programming course toy/
> my.vector = c(1,13,20,50,100,120)
> my.vector
[1]   1  13  20  50 100 120
> my.vector[2]
[1] 13
> my.vector[c(2,6)]
[1]  13 120
```

1. By Position

เรา subset ข้อมูลใน R ได้สามแบบ เพื่อดึงค่าที่เราต้องการ

```
Console C:/Users/user/Desktop/R programming course toy/
> new = rep(my.vector, times = 2)
> new
 [1]   1  13  20  50 100 120   1  13  20  50 100 120
> new[new > 20]
[1]  50 100 120  50 100 120
```

2. By Logic

```
Console C:/Users/user/Desktop/R programming course toy/
> my.vector
[1]   1  13  20  50 100 120
> names(my.vector) = c("thailand", "japan", "korea", "laos", "UK", "australia")
> my.vector
 thailand     japan     korea      laos        UK australia
        1        13        20        50       100       120
> my.vector["korea"]
korea
   20
> my.vector[c("UK", "korea")]
   UK korea
  100    20
```

3. By Name

OK, I think you now know how a function works in R

Basic

อันนี้คือไฮไลท์ของโปรแกรม R เลย คือการเรียกใช้งาน function เพื่อสร้าง output ที่เราต้องการ

```
Console C:/Users/user/Desktop/R programming course toy/
> seq(from = 0, to = 20, by = 4)
[1]  0  4  8 12 16 20
> rep(1:5, times = 2)
 [1] 1 2 3 4 5 1 2 3 4 5
> c(1:5, 6:10)
 [1]  1  2  3  4  5  6  7  8  9 10
> class("TRUE")
[1] "character"
> is.numeric("DataRockie")
[1] FALSE
> is.character("DataRockie")
[1] TRUE
```

function (.... input ....) → output

Useful Functions

help(....)

?function_name

You can always get some helps in R

Now let's do some basic statistics by using functions

Basic

Create a numeric vector using `c(1:50)` and then compute some statistics.

```
Console C:/Users/user/Desktop/R programming course toy/
> x = c(1:50)
> mean(x)
[1] 25.5
```

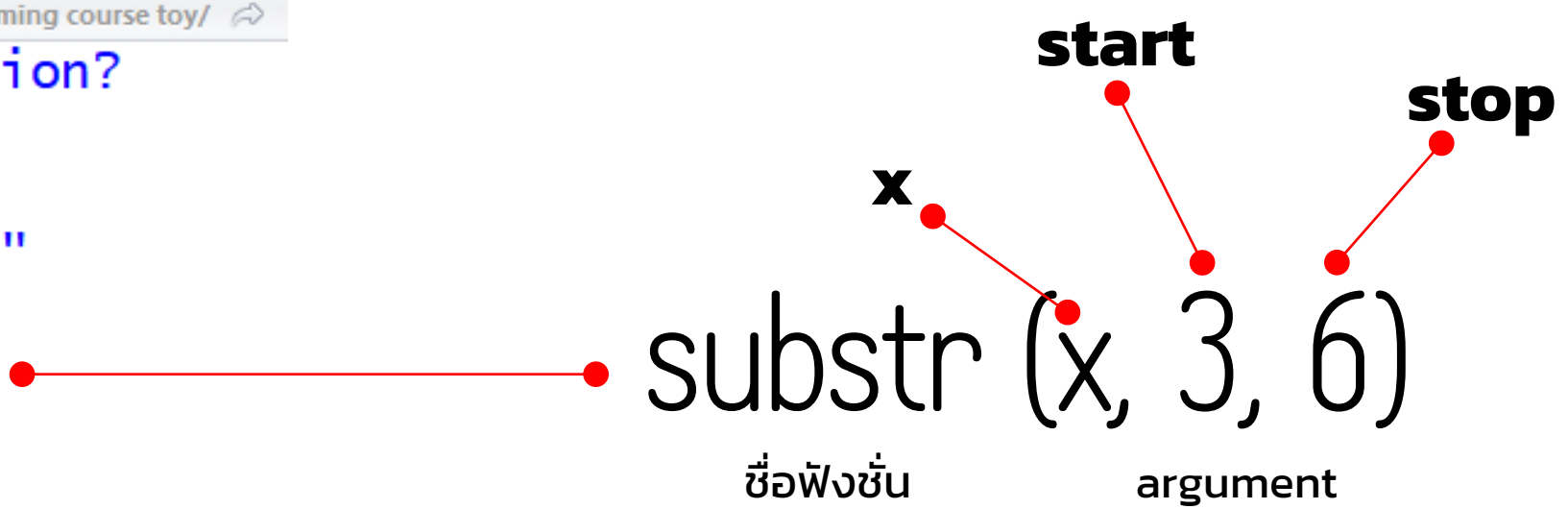
Let's try `rnorm( )` in the console. What does it do?

Useful Functions	
mean(x)	fivenum(x)
sd(x)	summary(x)
median(x)	cor(x, y)
max(x)	cov(x, y)
min(x)	rnorm(...)

Explain: What is FUNCTION?

Basic

```
Console C:/Users/user/Desktop/R programming course toy/
> ## What is function?
> help("substr")
>
> x <- "I love you"
>
> substr(x, 3, 6)
[1] "love"
> substr(x, 1, 1)
[1] "I"
> substr(x, 8, 10)
[1] "you"
```



ถ้าอยากรู้ว่าฟังชั่นไหนใช้ยังไงให้เราเรียก help(xxx) ขึ้นมาดู
หรือเสิช google ดูตัวอย่างวิธีการใช้ได้เลย

Time to write your first function

Basic

มาลองฝึกเขียนฟังชั่น

addtwo(a, b)

มี 2 arguments (a,b) เพื่อหาผลรวมของ a,b ออกมาครับ

Template

```
# template to write a function
# use "function" followed by arguments in ( )
# and use { } ... inside the { } is what you want R to do

function.name <- function(a,b) {
    #write what you want R to do here
}
```

```
Console C:/Users/user/Desktop/R programming course toy/
> addtwo(2,3)
[1] 5
> addtwo(5,5)
[1] 10
> addtwo(5,1000)
[1] 1005
> addtwo(-5, -3)
[1] -8
```

First challenge today!

Please write function "howlong50" takes one argument (which is your birth year) and return how long (years) before you get 50 years old

```
Console C:/Users/user/Desktop/R programming course toy/
> howlong50(1988)
[1] 21
> howlong50(1990)
[1] 23
> howlong50(2000)
[1] 33
```

If this happens ... then what ? Or else?

Basic

Example 1

```
hundred <- function(a,b) {
    if(a+b > 100) {
        print("yes")
    } else {
        print("no")
    }
}
```

```
Console C:/Users/user/Desktop/R programming course toy/
> hundred(50,20)
[1] "no"
> hundred(50,50)
[1] "no"
> hundred(50,51)
[1] "yes"
> hundred(50,560)
[1] "yes"
```

Example 2

```
what.is.it <- function(a) {
    if(a > 0) {
        print("positive")
    } else if(a < 0) {
        print("negative")
    } else {
        print("zero")
    }
}
```

```
Console C:/Users/user/Desktop/R programming course toy/
> what.is.it(100)
[1] "positive"
> what.is.it(-100)
[1] "negative"
> what.is.it(0)
[1] "zero"
```

Use this template to write if ... else statement

Template

```
if ( this is true ) {
        do this
} else {
        do this
}
```

Control Structure in R just like other programming lang.

Basic

FOR LOOP

```
> for (i in 1:10) {
+     print("Hello World!")
+ }
[1] "Hello World!"
[1] "Hello World!"
[1] "Hello World!"
[1] "Hello World!"
[1] "Hello World!"
[1] "Hello World!"
[1] "Hello World!"
[1] "Hello World!"
[1] "Hello World!"
[1] "Hello World!"
```

for (i in 1:10) อันนี้เราเรียกว่า sequence
สั่งปริ้น 10 รอบ

WHILE LOOP

```
> x <- 0
> while (x < 5) {
+     print(paste("x value is", x, "still less than 5 so I print"))
+     x <- x + 1
+ }
[1] "x value is 0 still less than 5 so I print"
[1] "x value is 1 still less than 5 so I print"
[1] "x value is 2 still less than 5 so I print"
[1] "x value is 3 still less than 5 so I print"
[1] "x value is 4 still less than 5 so I print"
```

while (x < 5)
ถ้า x ยังน้อยกว่า 5 ให้เราปริ้นประโยคนี้

Now let's create a matrix and a data frame from scratch

Basic

```
Console C:/Users/user/Desktop/R programming course toy/
> m = matrix(1:9, byrow = TRUE, nrow = 3)
> m
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
> class(m)
[1] "matrix"
> m2 = as.data.frame(m)
> m2
  V1 V2 V3
1  1  2  3
2  4  5  6
3  7  8  9
> class(m2)
[1] "data.frame"
> rownames(m2) = c("hello", "hi", "ni hao")
> colnames(m2) = c("American", "British", "Chinese")
> m2
       American British Chinese
hello         1       2       3
hi            4       5       6
ni hao        7       8       9
```

Matrix

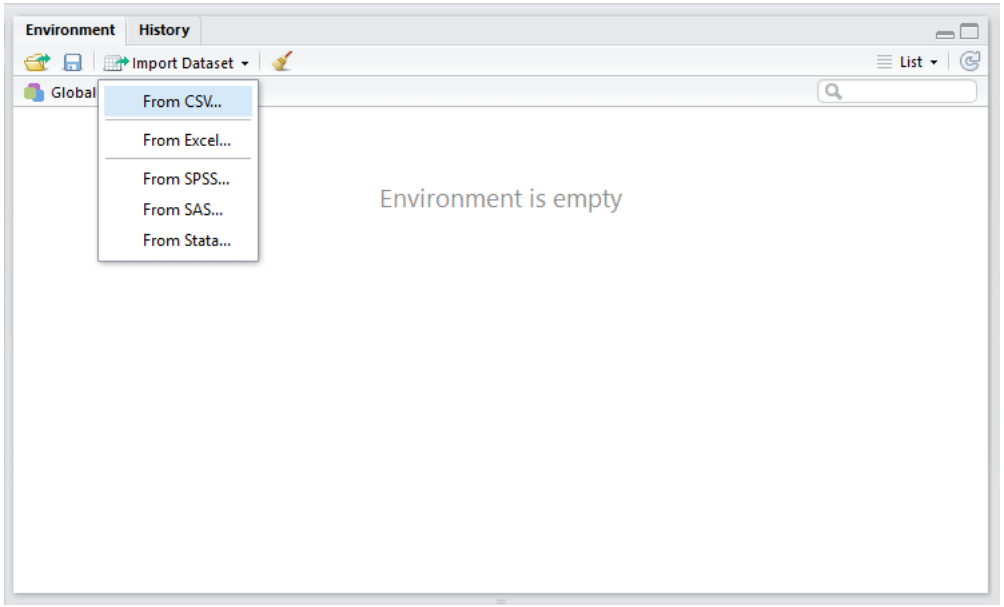
Data Frame

Let's try to **subset** a matrix / data frame by using [,]

Import and Export data in R

Basic

R สามารถ import ข้อมูลได้หลายแบบ



อ่านข้อมูล

read.csv("file name.csv")

write.csv(df, "file name.csv")

เขียนข้อมูล

ก่อนที่จะใช้ read.csv ต้อง make sure ก่อนว่า file csv นั้นอยู่ใน working directory เราแล้ว

Let's work with data frame a little bit

Basic

Untitled1* × mtcars ×

Filter

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Type `View(mtcars)` in console

Type `help(mtcars)` to read the description of this data set

Useful Functions

str(x)	names(x)
dim(x)	ncol(x)
summary(x)	nrow(x)

If you want to calculate mean of hp column of this data frame, you can write `mean(mtcars$hp)`

FACTOR

```
Console C:/Users/user/Desktop/R programming course toy/
> my.vec <- c(1,1,1,3,2,3,3,3,2,1,1)
> animals <- factor(my.vec,
+                   levels = c(1,2,3),
+                   labels = c("cat", "dog", "fish"))
> animals
 [1] cat  cat  cat  fish dog  fish fish fish dog  cat  cat
Levels: cat dog fish
```

ก่อนเราจะรันโมเดล หรือรันผลสถิติอะไรก็ตาม อย่างแรกต้องเปลี่ยน type ของตัวแปรเราให้ถูกต้องก่อน โดยเฉพาะตัวแปรที่เป็น categorical ควรเปลี่ยนเป็น factor ก่อนนะครัช

What is am in mtcars dataframe?

OKAY !! Let's do some analyses with data frame

1. ค่าเฉลี่ย HP ของ data set เราเป็นเท่าไร?

2. รถยนต์คันไหน มีค่า mpg สูงสุด? (เปลืองน้ำมัน)
   Hint: ใช้ which.max(...) แล้วค่อย subset ข้อมูล row นั้น

3. ความสัมพันธ์ระหว่าง mpg vs. wt เป็นอย่างไร?
   Hint: ใช้ plot(x, y)

Unleash your R capabilities

Install new packages to perform new tasks

Basic

```
# install new package
install.packages("dplyr")

# use library() to call package
every time you restart R studio
library(dplyr)
```

Install.packages("...")

Data Transformation

Basic

Written by Hadley Wickham, Data Scientist

dplyr

Main

- select( )
- filter( )
- arrange( )
- mutate( )
- summarise( )

- group_by( )
- rename( )
- sample_n( )
- count( )
- summarise_all( )

Main

- pipe operator (%>%)

select( ) ใช้ในการเลือก column

filter( ) ใช้ในการเลือก row

arrange( ) ใช้เรียงข้อมูล

mutate( ) ใช้สร้างตัวแปรใหม่

summarise( ) ใช้สรุปผลทางสถิติ

group_by( ) จับกลุ่มตัวแปร

rename( ) เปลี่ยนชื่อ column

sample_n( ) สุ่มตัวอย่าง

count( ) นับความถี่

summarise_all( ) สรุปผลทุกตัวแปร

These two functions make data transformation easy

dplyr

Select column

```
# select
select(m, mpg, cyl)
select(m, 1:5)
select(m, -1)
select(m, contains("a"))
select(m, starts_with("a"))
select(m, ends_with("p"))
```

Filter row

```
# filter
filter(m, mpg >= 30)
filter(m, hp >= 200, am == 1)
filter(m, hp > 150 & am == 1)
filter(m, hp > 150 | am == 1)
filter(m, cyl %in% 1:4)
filter(m, cyl %in% c(4,8))
filter(m, hp < 100 | hp > 200)
```

Another three main verbs in dplyr

Arrange values

```
# arrange
arrange(m, mpg)
arrange(m, desc(mpg))
arrange(m, cyl, hp)
```

Create new variable

```
# mutate
mutate(m, GC = gear * carb)
```

Compute statistics

```
# summarise
summarise(m, mean_wt = mean(wt, na.rm = TRUE))

# summarise_all
summarise_all(m, mean)
m %>% summarise_all(n_distinct)
```

Time to combine your functions

dplyr

Chain your code

```
# piping
# we use %>% to chain code

m %>%
  1 select(wt, mpg) %>%
  2 filter(wt >= 3.5) %>%
  3 mutate(new.variable = wt/mpg) %>%
  4 arrange(desc(new.variable))
```

Output

```
> m %>%
+     select(wt, mpg) %>%
+     filter(wt >= 3.5) %>%
+     mutate(new.variable = wt/mpg) %>%
+     arrange(desc(new.variable))
      wt  mpg new.variable
1  5.424 10.4    0.5215385
2  5.250 10.4    0.5048077
3  5.345 14.7    0.3636054
4  3.840 13.3    0.2887218
5  3.570 14.3    0.2496503
6  3.780 15.2    0.2486842
7  4.070 16.4    0.2481707
8  3.570 15.0    0.2380000
9  3.520 15.5    0.2270968
10 3.730 17.3    0.2156069
11 3.845 19.2    0.2002604
```

Useful trick: after %>% hit "enter" to start a new line. This would make your code more readable.

Supplementary functions

Grouped Statistics

```
Console C:/Users/user/Desktop/R programming course toy/
> m %>%
+     group_by(am) %>%
+     summarise(mean_wt = mean(wt))
# A tibble: 2 × 2
     am  mean_wt
  <dbl>    <dbl>
1     0 3.768895
2     1 2.411000
```

Use `group_by()` and now you can tell automatic (0) or manual (1) has higher average weight

Rename column

```
Console C:/Users/user/Desktop/R programming course toy/
> m %>%
+     select(wt) %>%
+     rename(weight = wt) %>%
+     head(5)
                  weight
Mazda RX4          2.620
Mazda RX4 Wag      2.875
Datsun 710         2.320
Hornet 4 Drive     3.215
Hornet Sportabout  3.440
```

Second challenge today!

Install this package to get flights data frame

```
# download this package
install.packages("nycflights13")
require(nycflights13)

df <- flights
str(df)
```

Your boss wants to see the average distance grouped by unique carriers in February

Can you get this table for me?

```
# A tibble: 15 × 2
   carrier avg_distance
     <chr>        <dbl>
1       HA     4983.0000
2       VX     2492.7122
3       AS     2402.0000
4       F9     1620.0000
5       UA     1435.7301
6       AA     1350.2714
7       DL     1226.9959
8       B6     1056.8906
9       WN      949.7278
10      FL      691.0000
11      MQ      565.0470
12      US      527.2474
13      EV      525.0656
14      9E      467.8931
15      YV      229.0000
```


TV
14

Bonus Tracks!! What if you want to join two tables?

dplyr

สมมติเราอยากจะดึงชื่อสายการบิน มาแปะใส่ flights dataframe ของเรา แต่ชื่อมันอยู่อีก table นึง?

```
Console C:/Users/user/Desktop/R programming course toy/
> flights2 <- select(flights, tailnum, flight, carrier)
> left_join(flights2, airlines, by = "carrier")
# A tibble: 336,776 x 4
   tailnum flight carrier                    name
     <chr>  <int>   <chr>                   <chr>
 1  N14228   1545      UA    United Air Lines Inc.
 2  N24211   1714      UA    United Air Lines Inc.
 3  N619AA   1141      AA   American Airlines Inc.
 4  N804JB    725      B6          JetBlue Airways
 5  N668DN    461      DL     Delta Air Lines Inc.
 6  N39463   1696      UA    United Air Lines Inc.
 7  N516JB    507      B6          JetBlue Airways
 8  N829AS   5708      EV ExpressJet Airlines Inc.
 9  N593JB     79      B6          JetBlue Airways
10  N3ALAA    301      AA   American Airlines Inc.
# ... with 336,766 more rows
```

เทียบเท่ากับการทำ **vlookup( )** ใน Excel เลย

ตารางหลัก

อยากดึงค่าจาก ตาราง y ไปแปะใส่ x

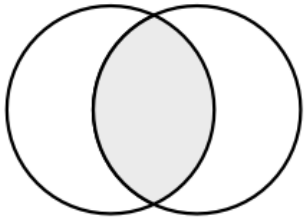
left_join(x, y, by = “xxx”)

=vlookup(x, table y, column)

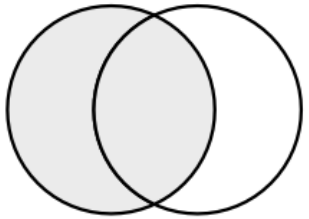
Different Types of Joins

dplyr

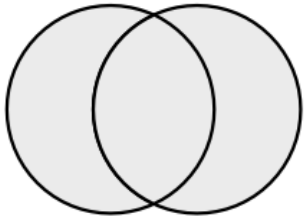
อ่านเพิ่มเติมเรื่อง relational data ได้ที่
http://r4ds.had.co.nz/relational-data.html



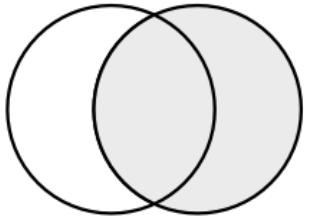
inner_join(x, y)



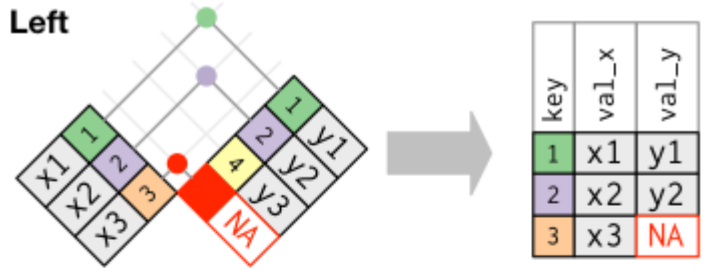
left_join(x, y)

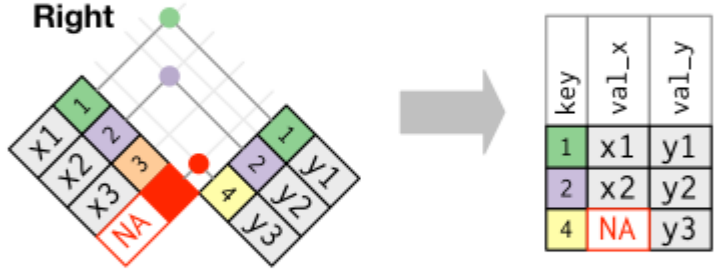


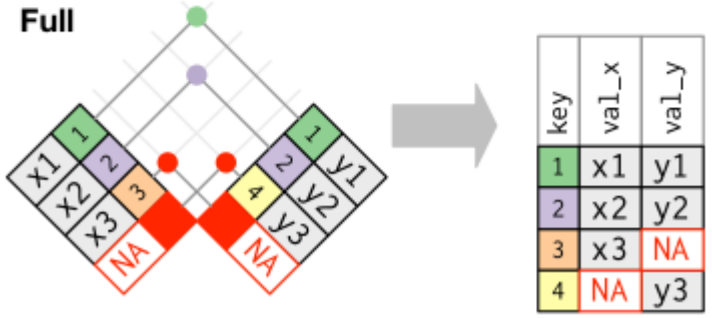
full_join(x, y)



right_join(x, y)



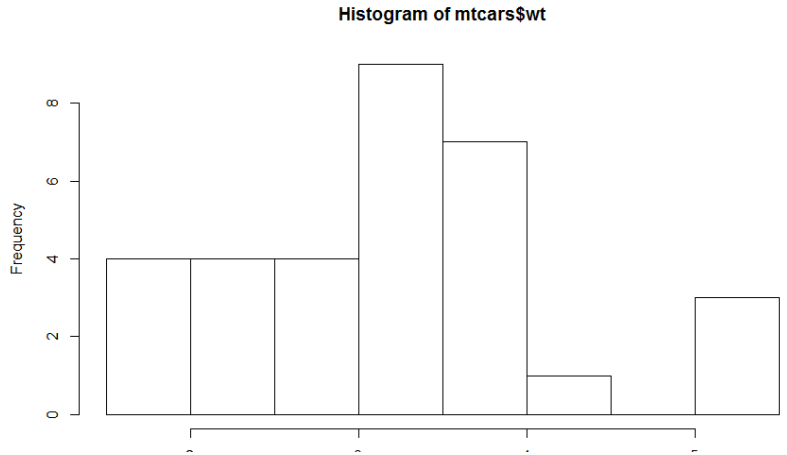





Basic
Data Visualization
A simple graph has brought more information to the data analyst's mind than any other device
John Tukey

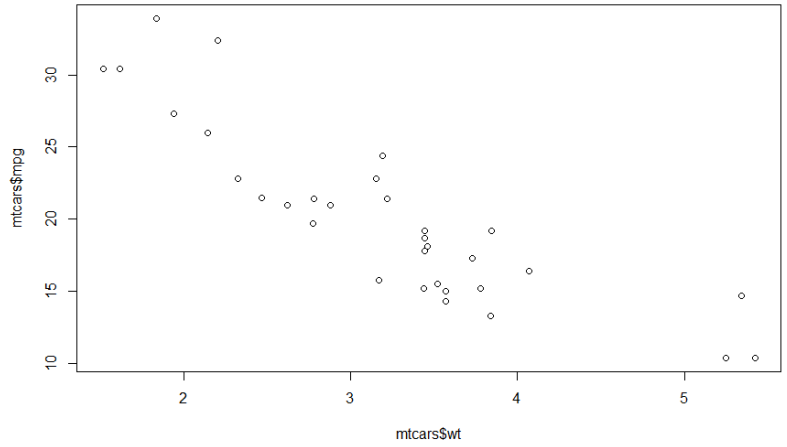
R comes with base graphic, simple to use, quick plot

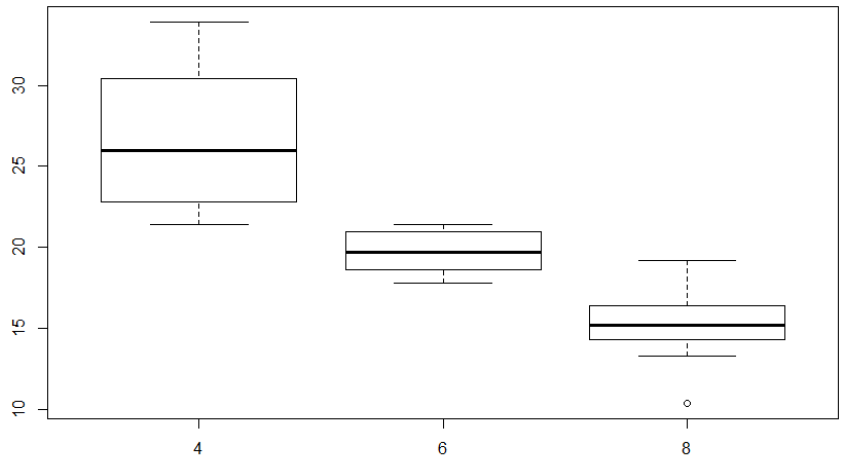
Base R



Remember: You can see arguments in each function by typing ?function.name

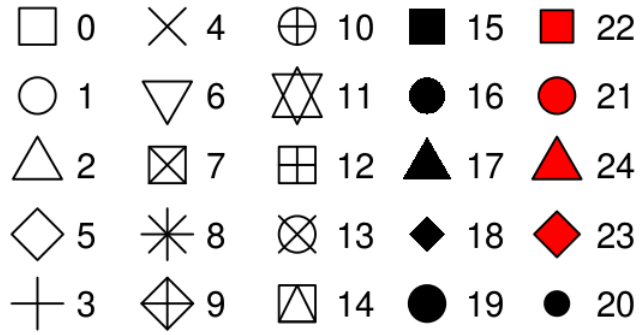
```
hist(mtcars$wt)
plot(mtcars$wt, mtcars$mpg)
boxplot(mtcars$mpg ~ mtcars$cyl)
barplot(table(mtcars$cyl))
```





Base R: plot() function

pch ใช้เปลี่ยน shape ต่างๆของ scatter plot ของเรา




```r
par(mfrow = c(1,1))
plot(mtcars$wt, mtcars$mpg, pch = 16)

plot(mtcars$wt, mtcars$mpg, pch = 17, col = "red",
     xlab = "weight",
     ylab = "mile per gallon",
     main = "Scatter plot wt x mpg",
     xlim = c(1,5))

index <- which(mtcars$wt > 5)
points(mtcars$wt[index], mtcars$mpg[index], pch = 16, col = "red")
```


Basic
Data Visualization
The most powerful graphical package ever created on Earth.

Written by Hadley Wickham again :P, Data Scientist

ggplot2

Template

Graphics in R built on this principle, 4 elements

Plot = **df** + **variables** + **geometry** + **scale**

Template

```
# install package
install.packages("ggplot2")
require(ggplot2)

# template for basic plot
# ggplot uses + to add layer while ggvis use %>%
ggplot(data = mpg) +
    geom_point(mapping = aes(cty, hwy))
```

```
last_plot()
ggsave("plot.png", width = 10, height = 10)
```

Plot Template: we use it all the time!!

Data Frame

Variables

```
ggplot(data, aes(x, y) +
  geom_point()
```

Geometry

อยากจะ plot กราฟสวยๆ นึกถึง ggplot เลยครัช

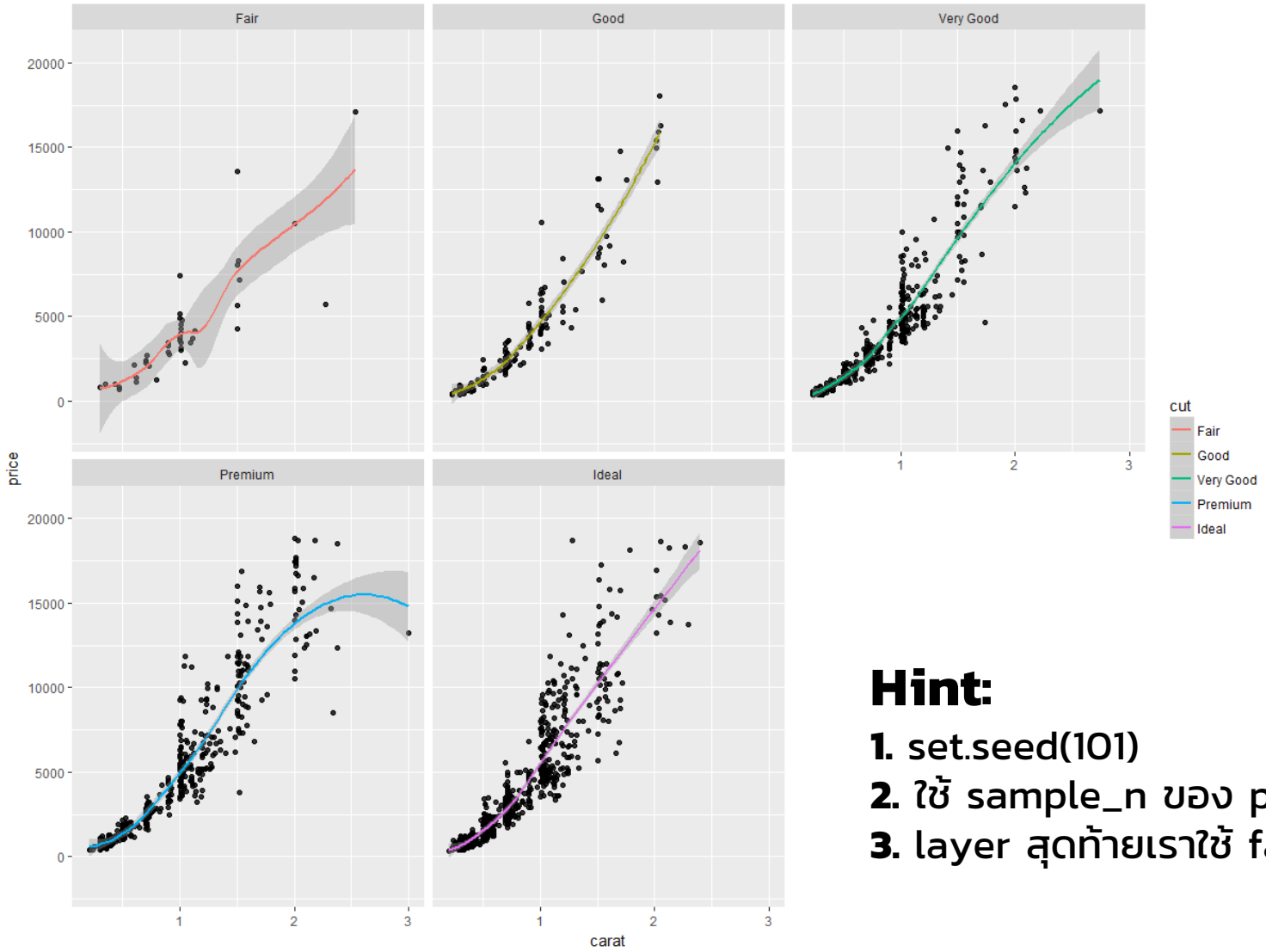
Plot Template: additional layers

```
ggplot(data, aes(x, y) +
    geom_point() +
    scale_color_xxx() +
    facet_grid() +
    coord_flip() +
    theme_bw() +
    labs()
```

Main

Additional

Plot Template: additional layers



QUIZ TIME !!

ลองสร้าง plot ด้านซ้าย หน่อยฮะ ใช้ data frame = diamonds ของแพ็คเกจ ggplot2

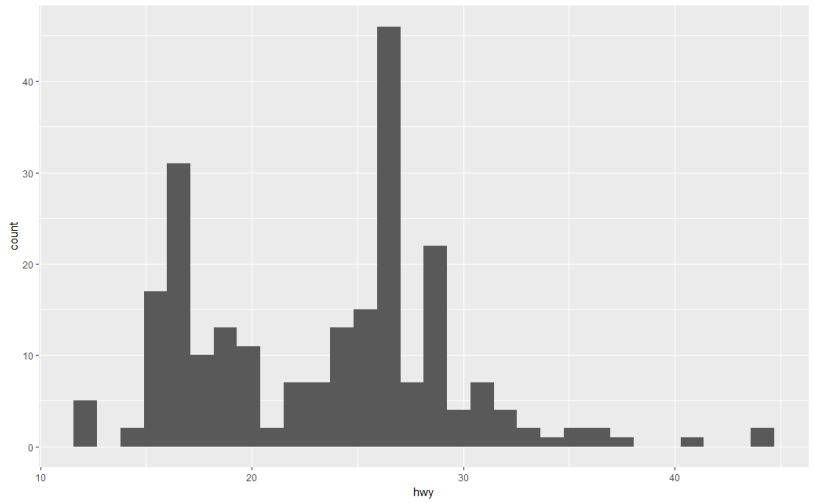
Hint:

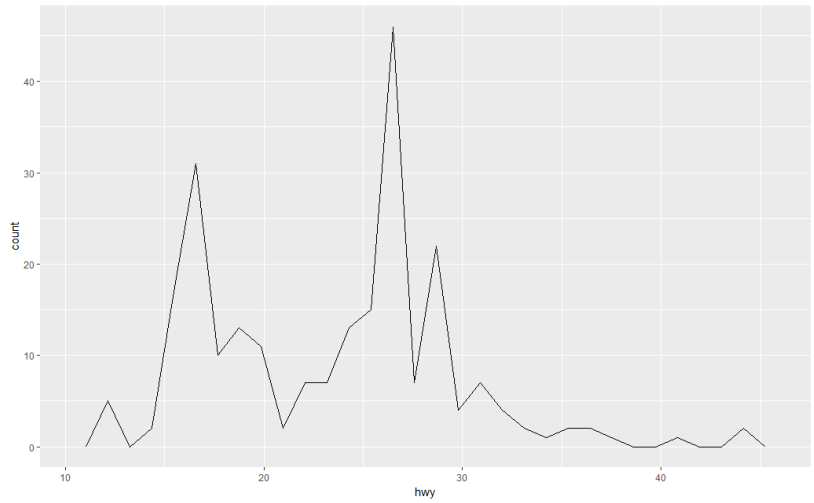
1. set.seed(101)
2. ใช้ sample_n ของ package dplyr เพื่อสุ่มช้อมูล n=2000
3. layer สุดท้ายเราใช้ facet_wrap(~cut)

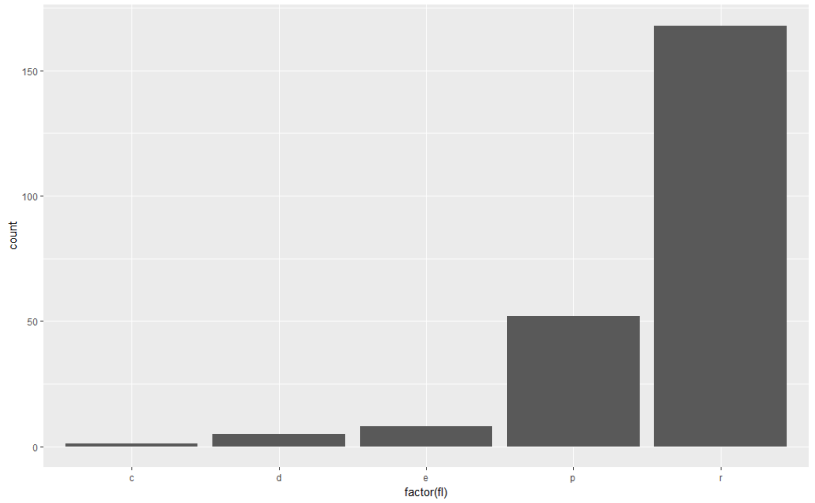
One, Two, Three or more variables. Discrete or Continuous

ggplot2

```
# one variable

c = ggplot(data = mpg, mapping = aes(hwy))

c + geom_area(stat = "bin")
c + geom_histogram()
c + geom_histogram(binwidth = 10, alpha = 0.5, fill = "blue")
c + geom_freqpoly(size = 0.5)

ggplot(mpg, aes(factor(fl))) + geom_bar()
```





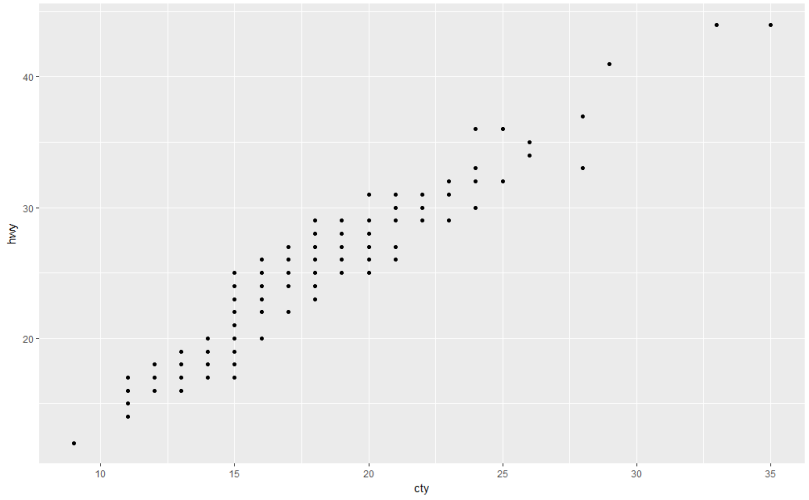


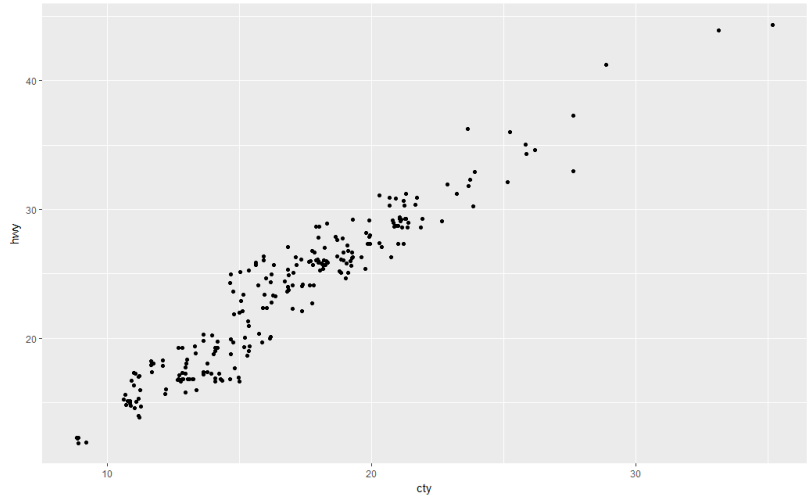
Two variables

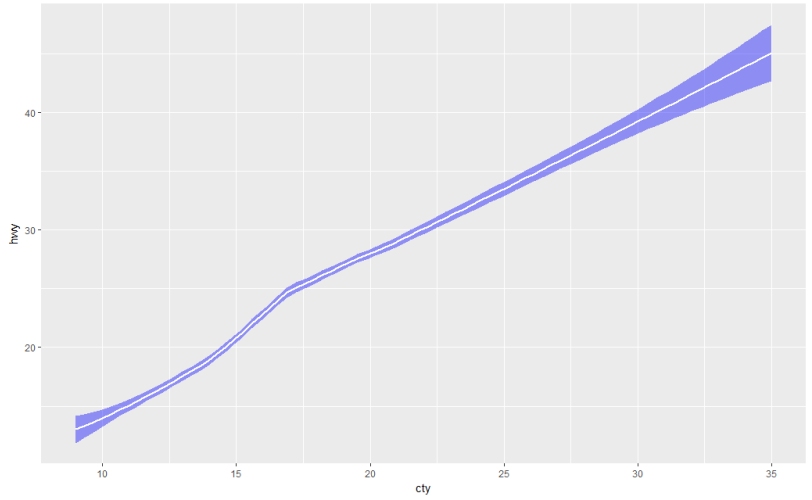
ggplot works just like ggvis, multiple layers

```
# continuous x continuous
e = ggplot(mpg, aes(cty, hwy))

e + geom_point()
e + geom_jitter() # add noise to data
e + geom_smooth(method = loess, color = "white", fill = "blue") # method =
lm, glm
e + geom_smooth(method = lm)
```

```
# discrete (X) x continuous (Y)
f = ggplot(mpg, aes(class, hwy))
f + geom_col()
f + geom_boxplot()
f + geom_violin()

# continuous function

i = ggplot(economics, aes(date, unemploy))
i + geom_area()
i + geom_line()
```



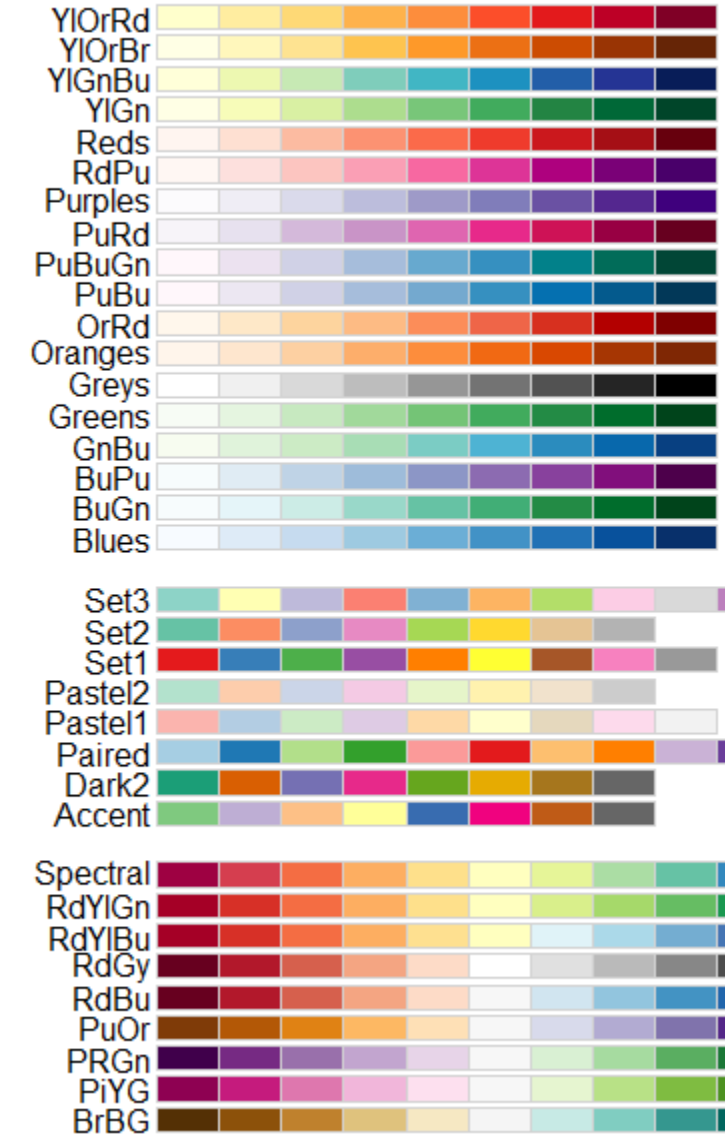




We can adjust the scales to change color

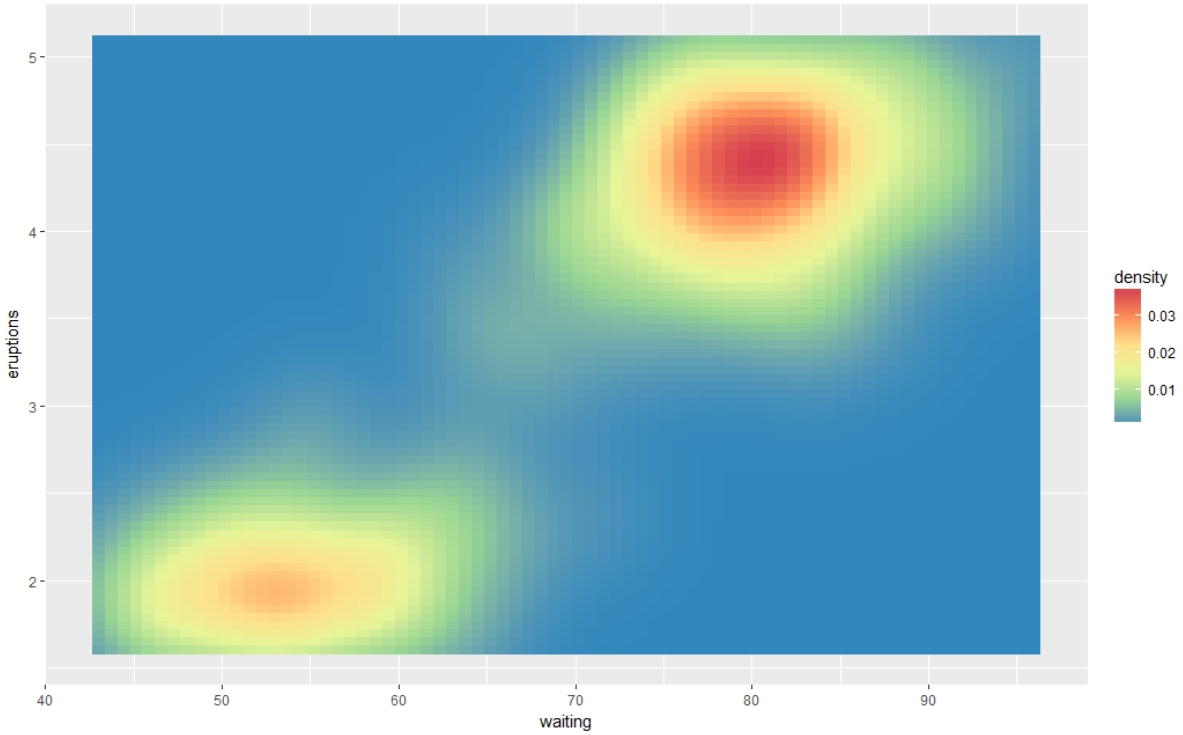

```
# scales adjustment
# scale_fill_brewer()

(n = d + geom_bar(aes(fill = factor(fl))))

install.packages("RColorBrewer")
require(RColorBrewer)
display.brewer.all()

n + scale_fill_manual(
    values = c("darkblue", "skyblue", "royalblue", "blue", "navy"),
    limits = c("c", "d", "e", "p", "r"),
    name = "fuel",
    labels = c("C", "D", "E", "P", "R"))

n + scale_fill_grey()
n + scale_fill_brewer(palette = "Blues")
n + scale_fill_brewer(palette = "Set3")

# scale_color_brewer()

require(dplyr)

data = sample_n(diamonds, size = 1000)
(x <- ggplot(data = data,
             mapping = aes(carat, price)) + geom_point(aes(color=clarity)))

x + scale_color_brewer()
x + scale_color_brewer("diamonds\nclarity")
x + scale_color_brewer(palette = "Spectral")
```

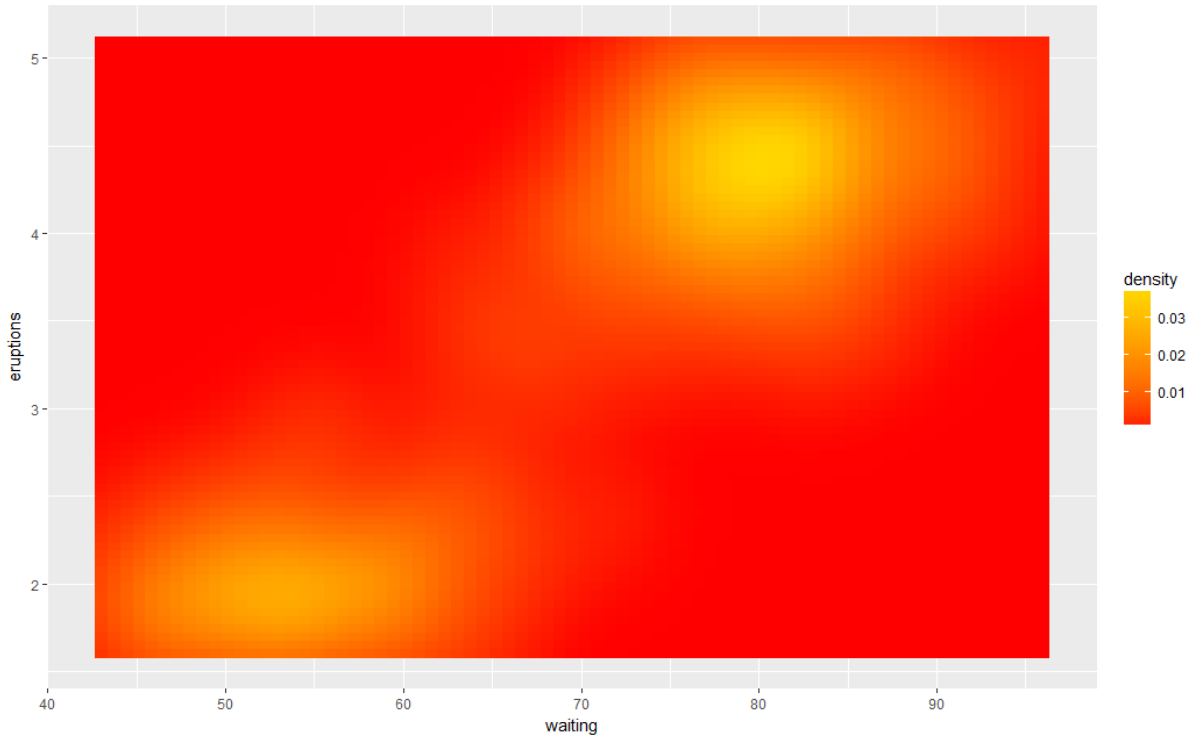
Package RColorBrewer



Three variables, ggplot can create beautiful graphic


```
# use distriller variant with continuous data

(v = ggplot(faithfuld) +
   geom_tile(aes(waiting, eruptions, fill = density)))

v + scale_fill_distiller()
?scale_fill_distiller
v + scale_fill_distiller(palette = "Spectral")

v + scale_fill_gradient(low = "red", high = "gold")
```

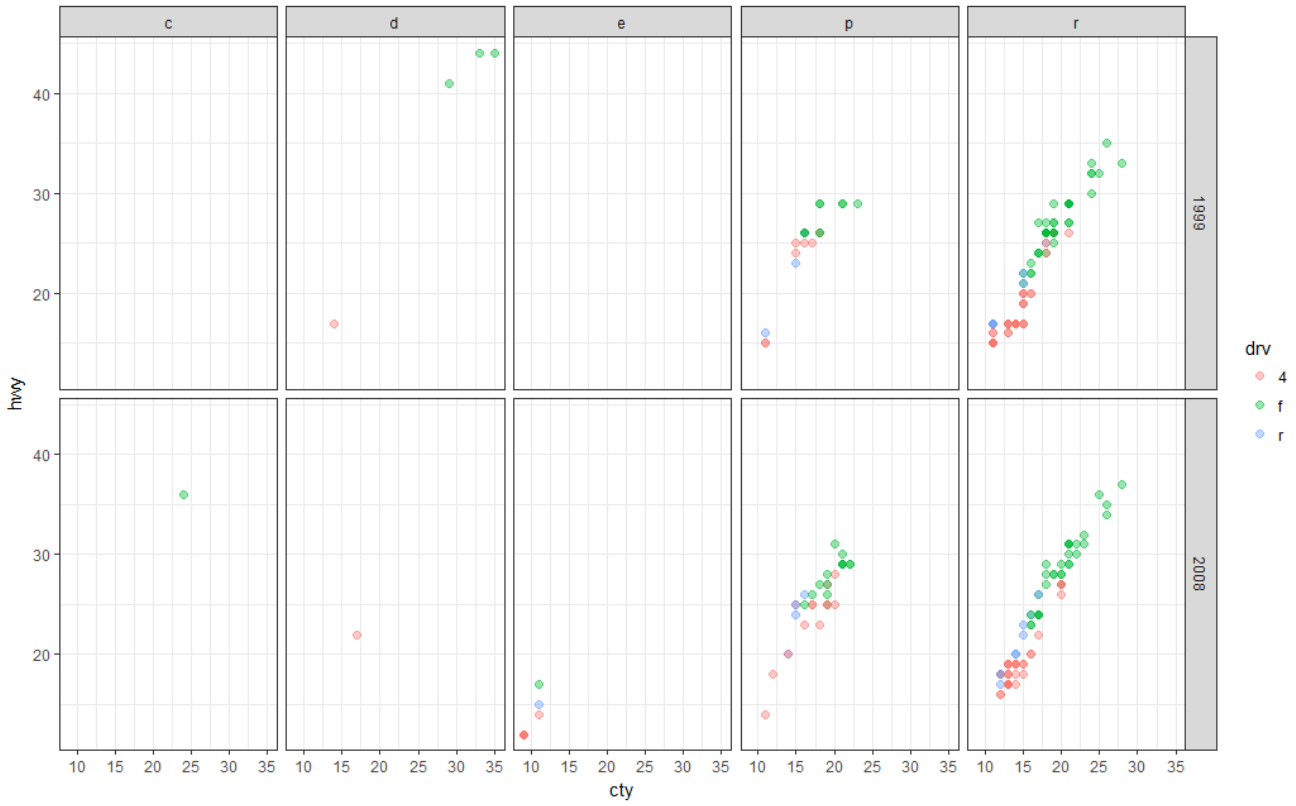




Perhaps, the most useful ggplot function is FACET

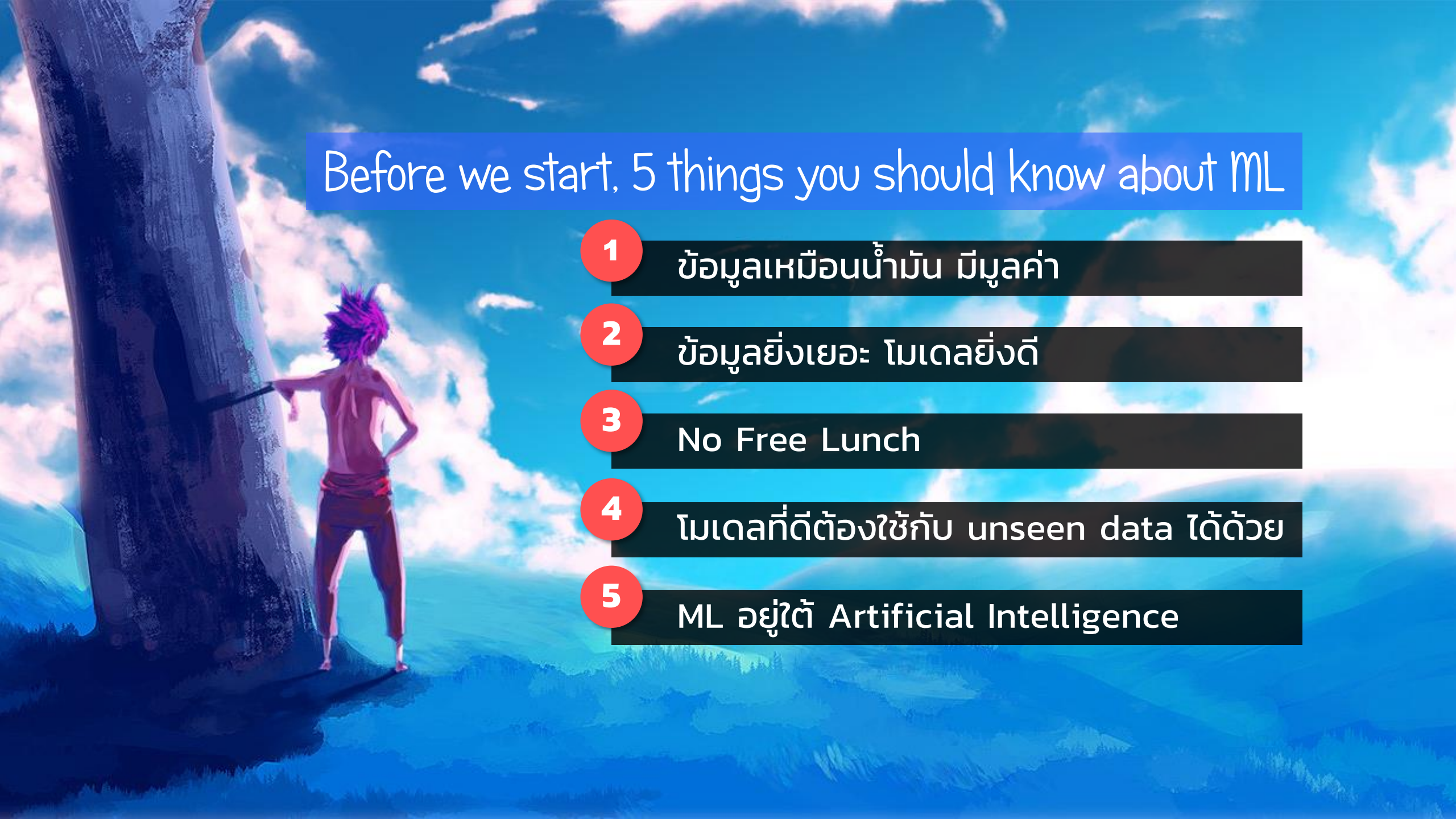

```
# add theme

(k = ggplot(data, aes(carat, price, color = clarity)) + geom_point())

(k = k + scale_color_brewer())

k + theme_classic()
k + theme_bw()
k + theme_dark()
k + theme_void()
k + theme_minimal()
k + theme_grey()

# coordinate systems

d + coord_cartesian(xlim = c(0,5), ylim = c(0,200))
d + coord_flip() # to vertical

# add labels

k + labs(x = "carat degree",
         y = "price of diamonds",
         title = "scatter plot of caret x price",
         subtitle = "more explanation puts here",
         caption = "and also caption yeah")

# faceting
(t = ggplot(data = mpg,
            mapping = aes(cty, hwy)) +
      geom_point(aes(color = drv), size = 2, alpha = .40) +
      theme_bw())

t + facet_grid(. ~ fl) # facet into column base on fl
t + facet_grid(year ~ .) # facet into rows base on year
t + facet_grid(year ~ fl) # matrix plots
t + facet_wrap(~ fl)
```

Facet breaks down the plots into grid by the variables you select



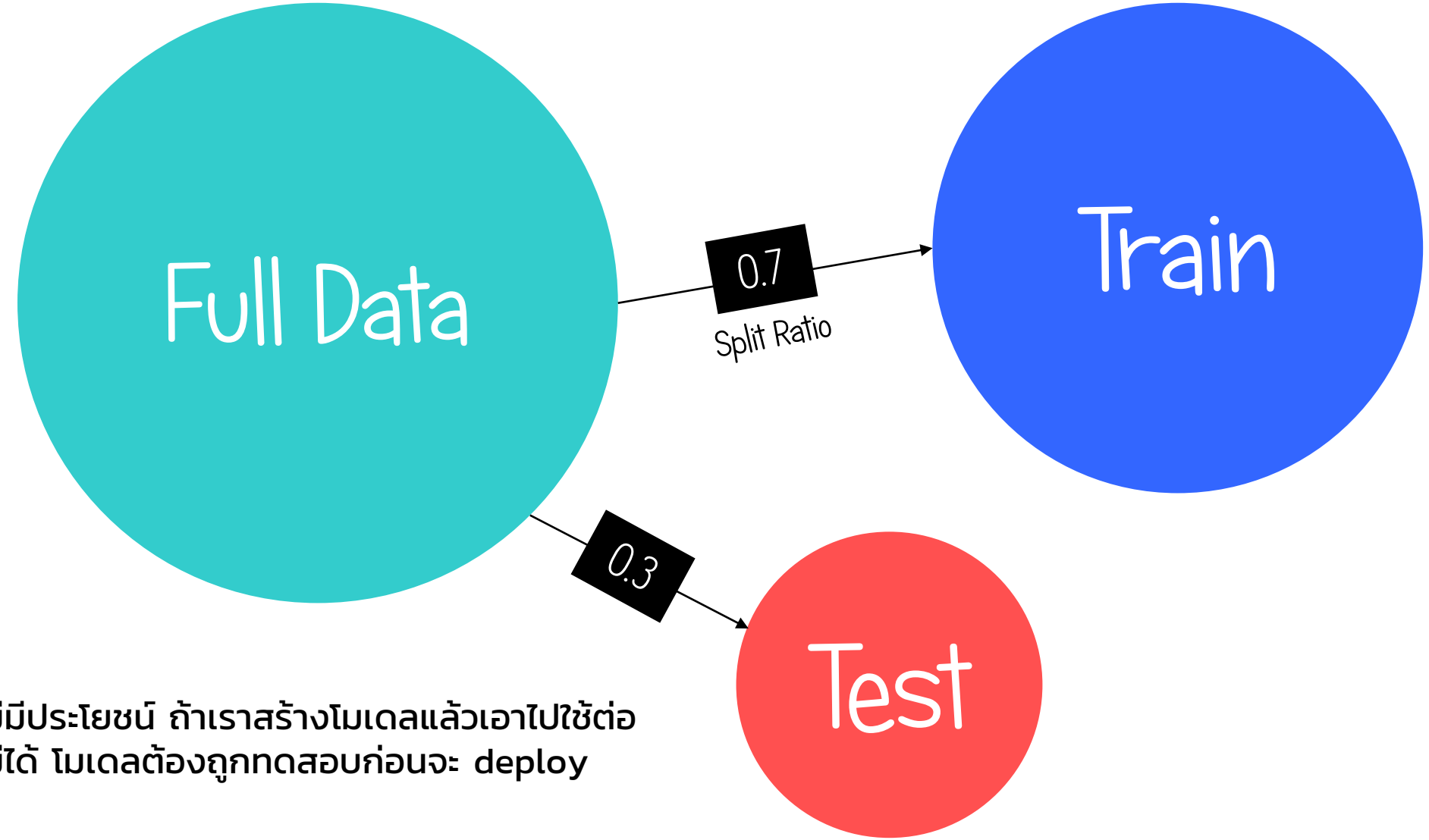
“You have already learned
two of the most useful skills in R.
Well done guys !!”


Machine
Learning


Before we start, 5 things you should know about ML
1
ข้อมูลเหมือนน้ำมัน มีมูลค่า
2
ข้อมูลยิ่งเยอะ โมเดลยิ่งดี
3
No Free Lunch
4
โมเดลที่ดีต้องใช้กับ unseen data ได้ด้วย
5
ML อยู่ใต้ Artificial Intelligence

After you've cleaned your data, SPLIT it !!!

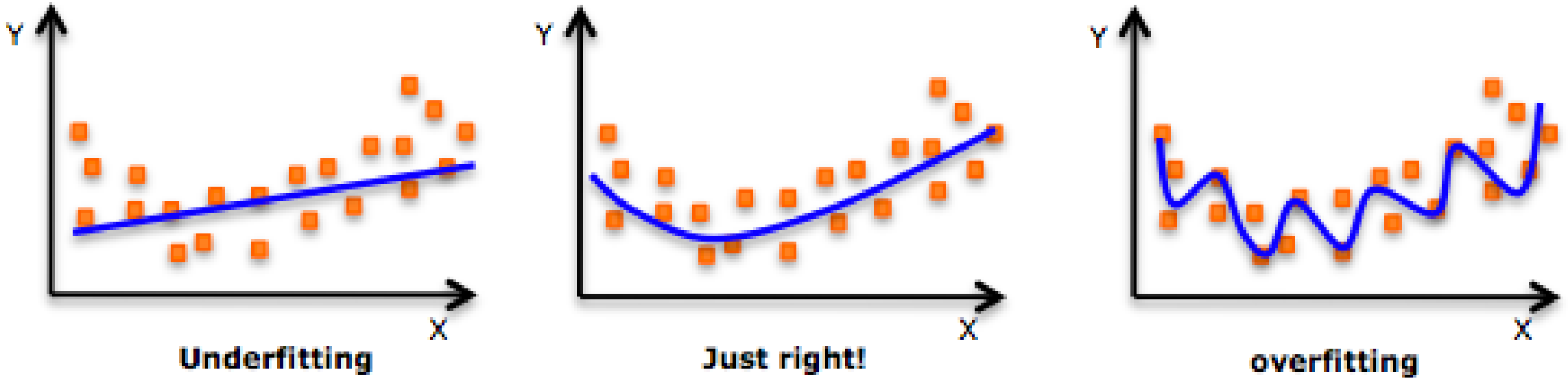
ML



ไม่มีประโยชน์ ถ้าเราสร้างโมเดลแล้วเอาไปใช้ต่อ
ไม่ได้ โมเดลต้องถูกทดสอบก่อนจะ deploy

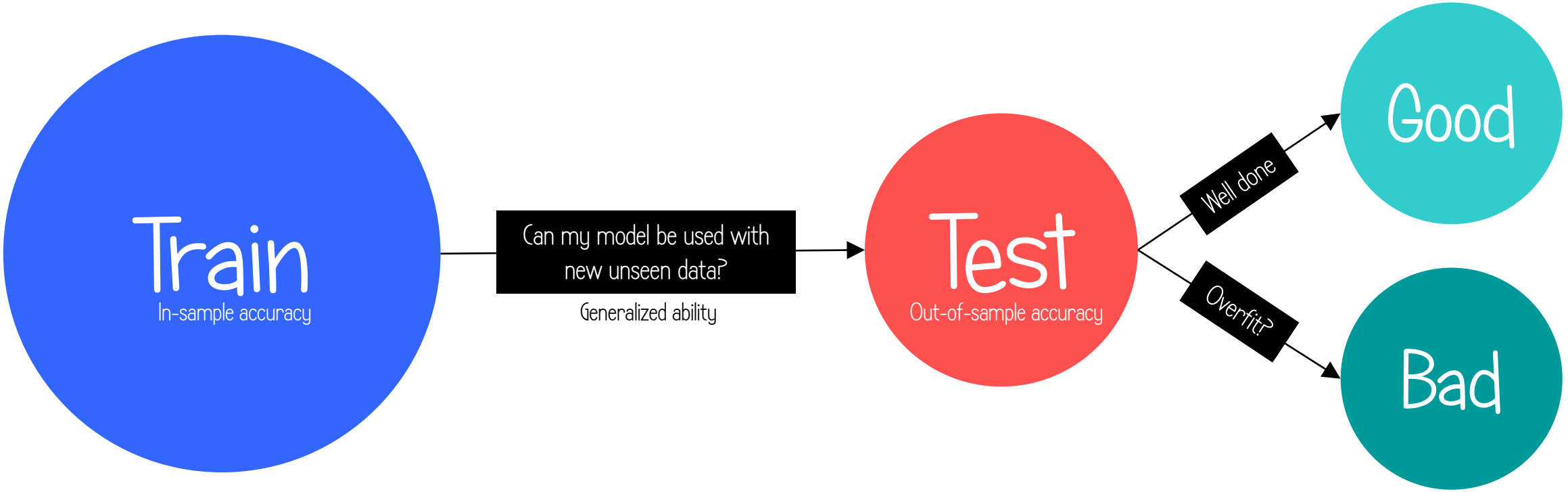
OVERFITTING the most concerned problem in ML

Overfitting คือการที่เราสร้างโมเดล
ที่ฟิตกับ training data เกินไป



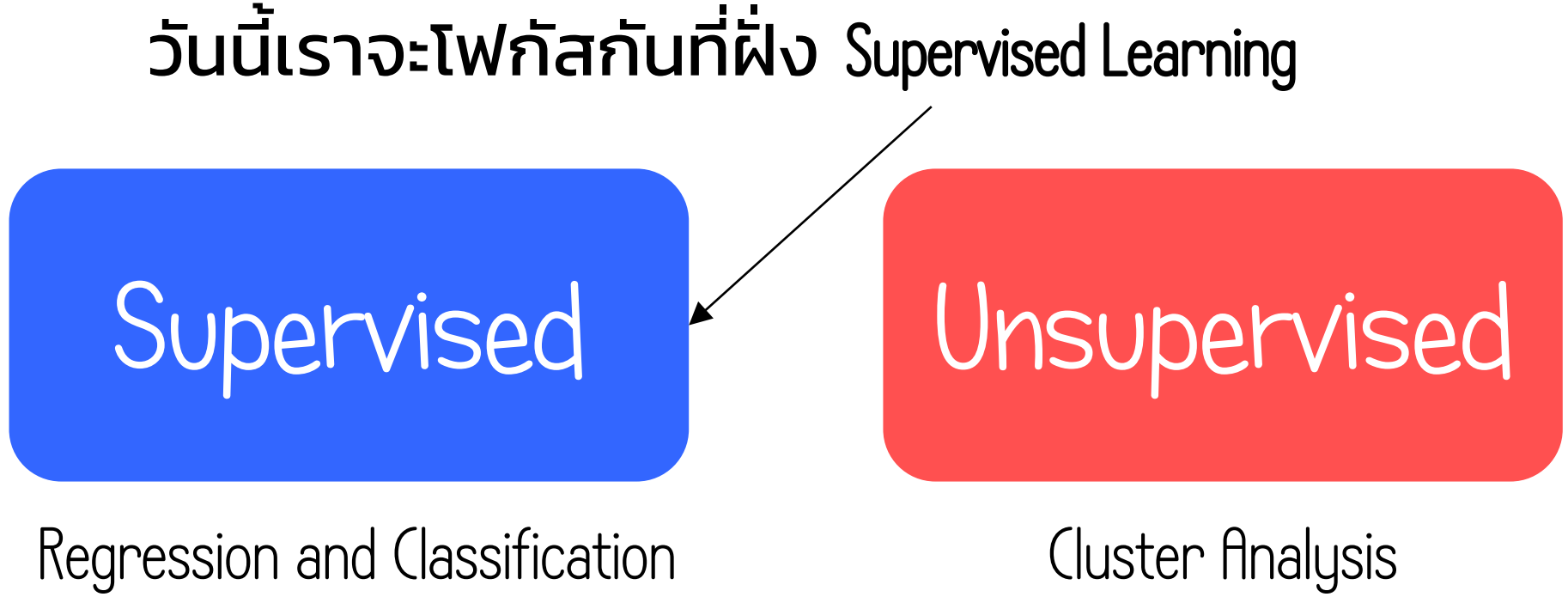
ปล. **Underfitting** คือ โมเดลเราทำงานไม่ดีทั้ง train และ test เลย

Then you build model on TRAIN, validate it on TEST



Remark: There is a method called "cross validation" like k-fold to validate your model, more robust, generally produce better results

Broad domains of learning algorithms



machine learning มีหลายชื่อมากเลย อย่าง **predictive analytics** หรือ **data mining** ก็ถือว่าเป็น ML เหมือนกัน (overlap)

Template we use to build model in R

ML

เราใช้ตัวหนอนในการสร้างโมเดล

function (**y** ~ . , **data** = …)

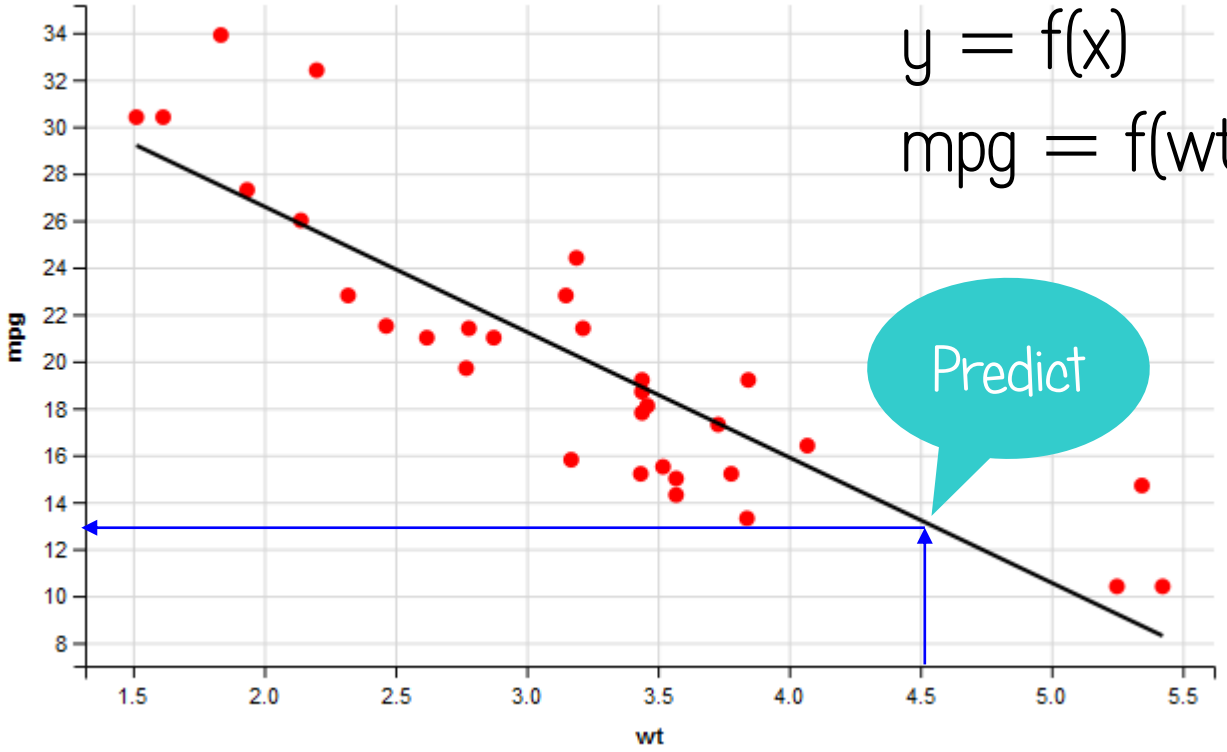

Model 1
Linear Regression

Linear regression predicts quantitative outcome

ML

Linear Regression Theory

y



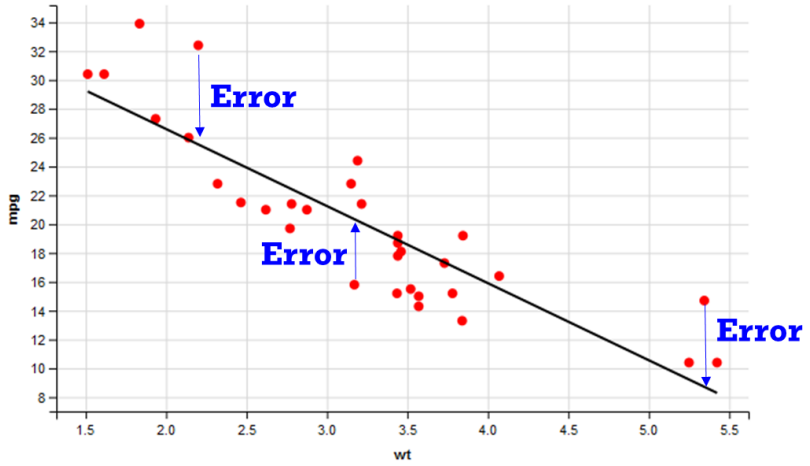
x

$y = f(x)$

$mpg = f(wt)$

Still, there will be errors.

No learner is perfect.



Let’s build our first linear regression algorithm

ML

```r
#### LINEAR REGRESSION ####

diamonds

# split data
set.seed(101)
sample <- sample.split(diamonds$price, SplitRatio = 0.70)
train <- subset(diamonds, sample == TRUE)
test <- subset(diamonds, sample == FALSE)

# training
lm.model <- lm(price ~ . , data = train)
summary(lm.model)

# in-sample-error (RMSE)
rmse1 <- sqrt(mean(lm.model$residuals^2))
rmse1

# out-of-sample error
p <- predict(lm.model, test)
rmse2 <- sqrt(mean((p-test$price)^2))
```

ขั้นตอนแรกคือการ split data

Train Model

Test Model

Interpret the coefficients. Correlation != Causation

ML

```
> summary(lm.model)

Call:
lm(formula = price ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-21668.7   -616.6   -191.8    393.0  10557.4

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5523.6683   532.2663  10.378  < 2e-16 ***
carat       11413.9982    59.3087 192.451  < 2e-16 ***
cut.L         626.2919    27.5613  22.724  < 2e-16 ***
cut.Q        -317.6103    22.0695 -14.391  < 2e-16 ***
cut.C         159.7539    18.9278   8.440  < 2e-16 ***
cut^4         -13.4005    15.1423  -0.885  0.37618
color.L     -2030.2114    20.9864 -96.740  < 2e-16 ***
color.Q      -685.4824    19.1461 -35.803  < 2e-16 ***
color.C      -172.0185    17.8763  -9.623  < 2e-16 ***
color^4        51.5210    16.4418   3.134  0.00173 **
color^5       -98.2225    15.5664  -6.310 2.82e-10 ***
color^6       -39.8854    14.2021  -2.808  0.00498 **
clarity.L    4239.4197    37.1791 114.027  < 2e-16 ***
clarity.Q   -1977.6819    34.7133 -56.972  < 2e-16 ***
clarity.C    1033.8836    29.6971  34.814  < 2e-16 ***
clarity^4    -369.4348    23.7396 -15.562  < 2e-16 ***
clarity^5     248.1224    19.3621  12.815  < 2e-16 ***
clarity^6      -4.9447    16.7758  -0.295  0.76819
clarity^7      86.1837    14.7845   5.829 5.61e-09 ***
depth         -62.4064     6.6256  -9.419  < 2e-16 ***
table         -23.5194     3.5316  -6.660 2.78e-11 ***
x            -986.9980    50.5608 -19.521  < 2e-16 ***
y              -0.9457    20.6168  -0.046  0.96341
z            -111.2507    73.6852  -1.510  0.13110
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1165 on 38453 degrees of freedom
Multiple R-squared:  0.9204,    Adjusted R-squared:  0.9204
F-statistic: 1.934e+04 on 23 and 38453 DF,  p-value: < 2.2e-16
```

Positive — X positively correlate with Y

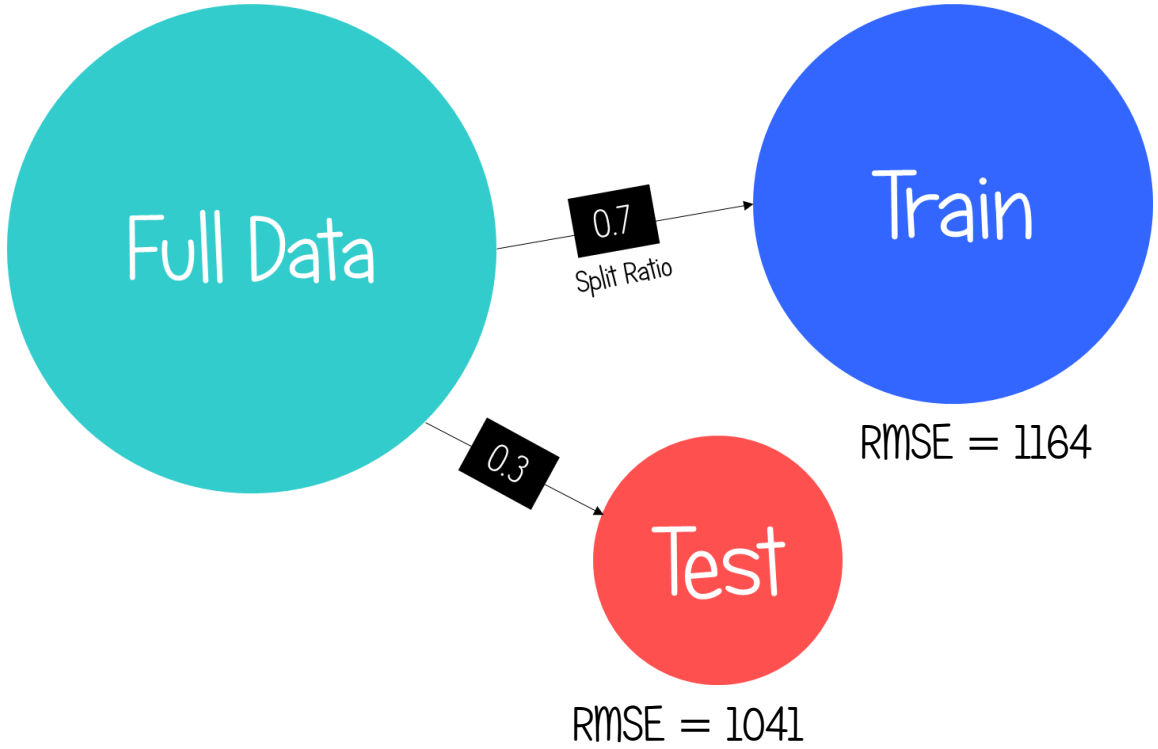
Negative — X negatively correlate with Y

นักสถิติ

- ดูค่า **coefficient** ว่าเป็น +/-
- ดูค่า **std. error** ว่าค่ามันเหวี่ยงแค่ไหน
- และสุดท้ายดู **p-value** ว่าตัวแปรไหนมีนัยสำคัญบ้าง

Predict and Evaluate our trained model

ถ้า in-sample error < out-of-sample error โมเดลเรามีแนวโน้มที่จะ Overfitting



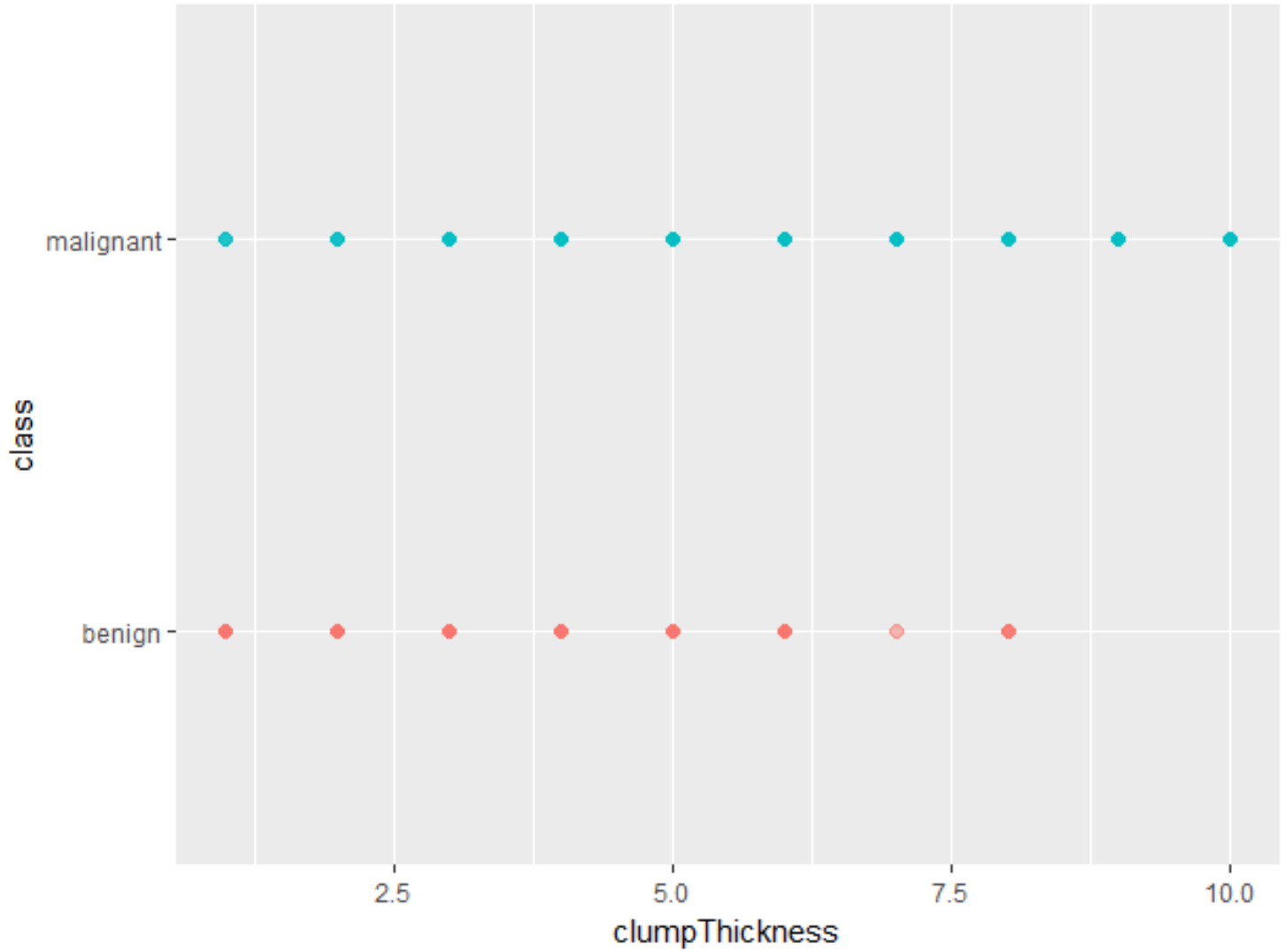
Train RMSE = 1164
Test RMSE = 1041

Linear Regression สร้างง่าย แต่เป็นโมเดลที่มี **high bias** (assumption ในการสร้างค่อนข้างเยอะ overfit ง่าย)


Model 2
Logistic Regression

Predicting 1 / 0 outcome

ML



Logistic แค่ปรับโมเดล Linear regression นิดหน่อยด้วย sigmoid function ทำให้โมเดลสามารถ predict ความน่าจะเป็นของ class 1/0 ได้

เด๋วเราจะสร้างโมเดลที่ใช้ predict มะเร็งเต้านมในผู้หญิง (Malignant)

How do we evaluate the model?

สิ่งที่เราทำนาย

ความจริง

	Benign	Malignant
Benign	True negative	False positive
Malignant	False negative	True positive

Overall Accuracy = True positive + True negative / Total n

And that’s we call Type I and Type II error

ML

Type I Error

อันนี้คือ False Positive

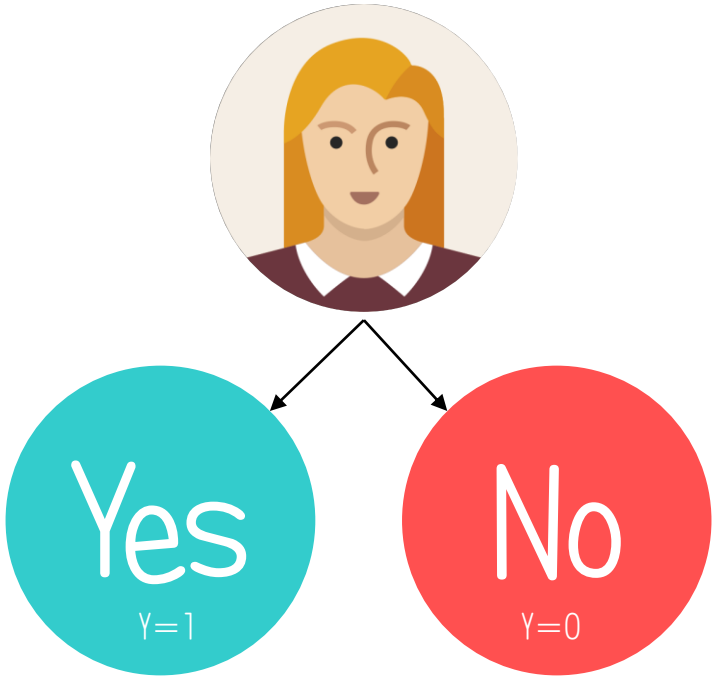
Type II Error

อันนี้คือ False Negative

Logistic is popular algorithm for classification problem

ML

```
#### LOGISTIC REGRESSION ####
# load data
df <- read.csv("breast cancer.csv")

# check missing value
mean(is.na(df))
clean.df <- na.omit(df)

# split data
set.seed(101)
sample <- sample.split(clean.df$class, SplitRatio = 0.70)
train <- subset(clean.df, sample == TRUE)
test <- subset(clean.df, sample == FALSE)

# train model
logit.model <- glm(class ~ ., data =  train, family = "binomial")
summary(logit.model)
```



Interpret the coefficients.

ML

```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)              -12.3828     2.1834  -5.671 1.42e-08 ***
clumpThickness             0.7297     0.2344   3.113  0.00185 **
sizeUniformity             0.1204     0.3556   0.338  0.73501
shapeUniformity            0.2936     0.3803   0.772  0.44014
maginalAdhesion            0.3729     0.1661   2.245  0.02478 *
singleEpithelialCellSize   0.2000     0.2723   0.734  0.46275
bareNuclei                 0.5139     0.1514   3.394  0.00069 ***
blandChromatin             0.4395     0.2602   1.689  0.09122 .
normalNucleoli             0.2347     0.1606   1.461  0.14392
mitosis                    0.4122     0.4270   0.965  0.33436
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 618.587  on 477  degrees of freedom
Residual deviance:  47.964  on 468  degrees of freedom
AIC: 67.964

Number of Fisher Scoring iterations: 9
```

Positive — X positively correlate with Y=1

Negative — X negatively correlate with Y=1

Evaluate Trained Model using “Confusion Matrix”


```
> table(train$class, p1 > 0.5)

           FALSE TRUE
benign       305    6
malignant      5  162
> (162 + 305) / nrow(train)
[1] 0.9769874
```

เรา evaluate model classification ด้วยตาราง confusion matrix

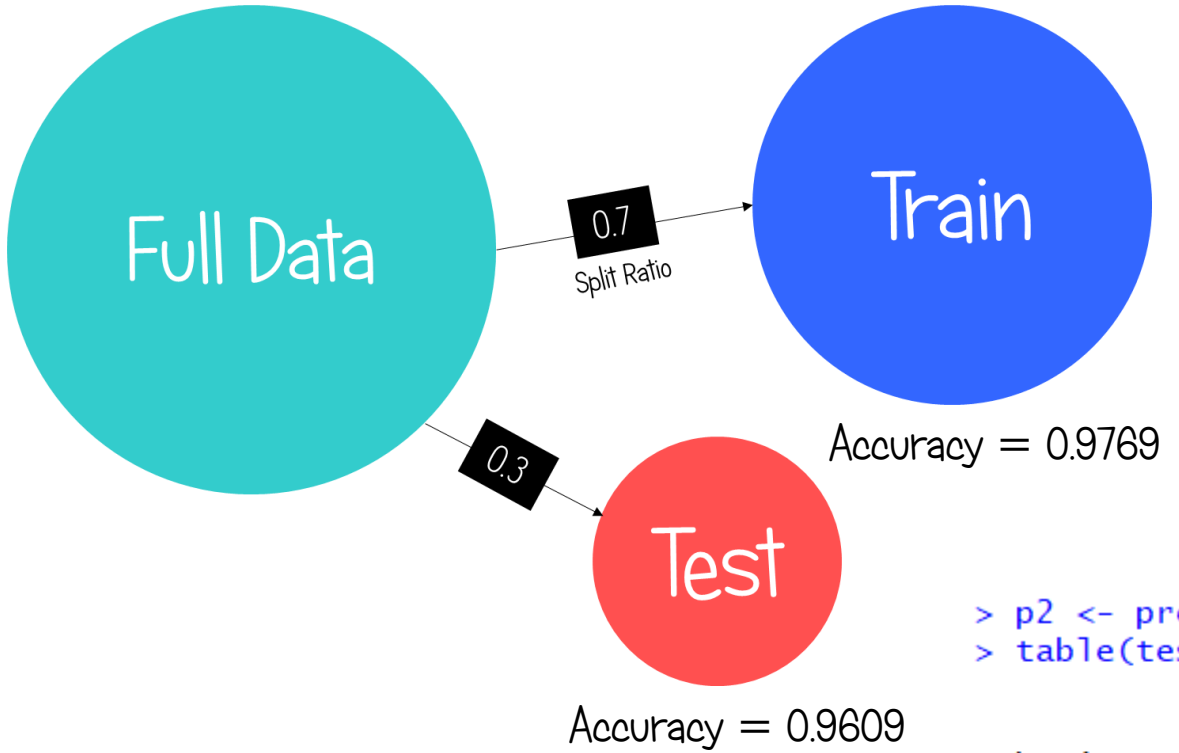
Confusion Matrix

โมเดลทายถูก accuracy ประมาณ 97.7%

Actual \ Predicted	B	M
B	305	6
M	5	162

Predict the unseen cases, and compare Accuracy

ML




```
> p1 <- predict(logit.model, type = "response")
> table(train$class, p1 > 0.5)

            FALSE TRUE
  benign      305    6
  malignant     5  162
> (162 + 305) / nrow(train)
[1] 0.9769874
```

```
> p2 <- predict(logit.model, test, type = "response")
> table(test$class, p2 > 0.5)

            FALSE TRUE
  benign      129    4
  malignant     4   68
> (68 + 129) / nrow(test)
```


Model 3
Decision Tree

Build model to predict who will survive during titanic inc.

ML

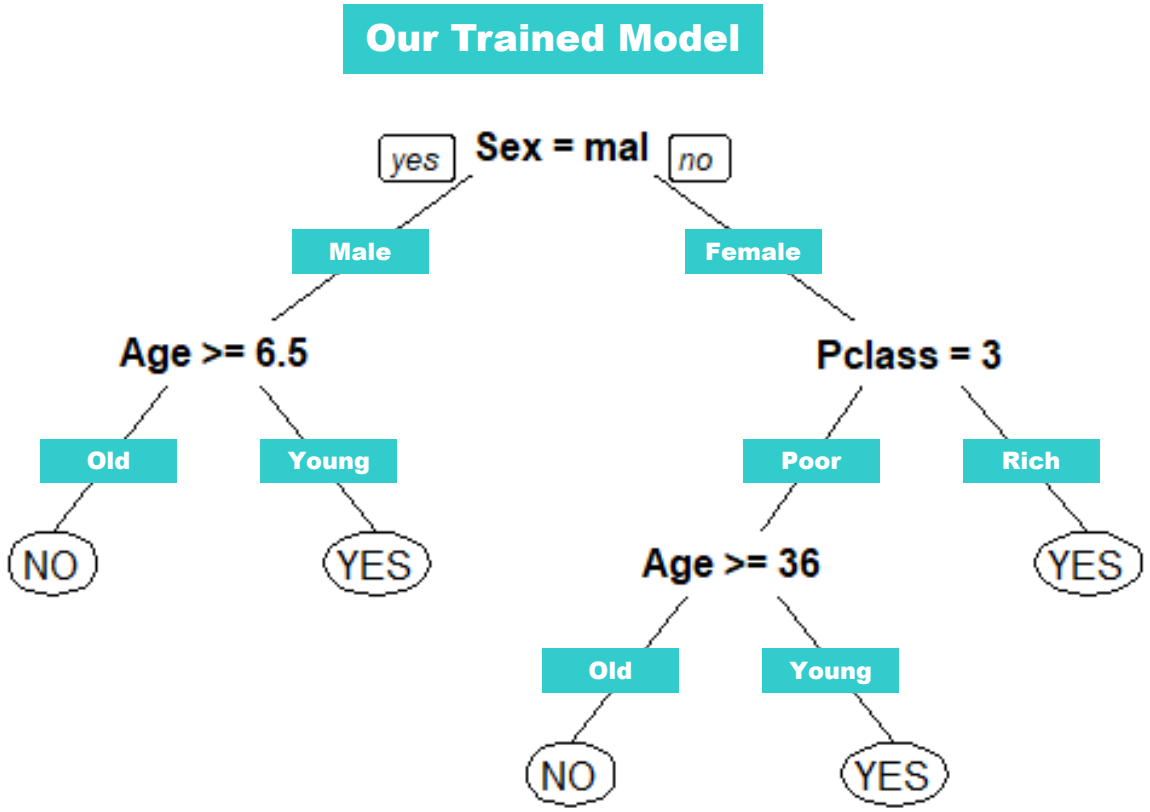
Decision Tree เป็นโมเดลที่ใช้ได้หลายงานมากๆ ปรับตัวได้ดี (ใช้ได้ทั้งการแก้ปัญหา regression / classification) สร้างง่าย แต่ข้อเสียคือเรื่อง accuracy ไม่สูงนัก

```r
# read data
df <- read.csv("titanic full data.csv")
str(df)

# review class of all variables
df$Pclass <- factor(df$Pclass)

# split data
require(caTools)
set.seed(101)
sample <- sample.split(df$Survived, SplitRatio = 0.70)

train <- subset(df, sample == TRUE)
test <- subset(df, sample == FALSE)

# training
require(rpart)
require(rpart.plot)

set.seed(101)
tree.model <- rpart(Survived ~ Pclass + Age + Sex,
                    data = train,
                    method = "class",
                    minbucket = 10)

prp(tree.model)
```

สร้างโมเดลด้วย rpart(y ~ .)

Plot Tree, Predict Test, & Evaluate Model

ML



Confusion Matrix

Train Model

```
> table(train$Survived, p.train)
     p.train
       NO YES
  NO  334  50
  YES  66 173
> (334 + 173)/ nrow(train)
[1] 0.8138042
```

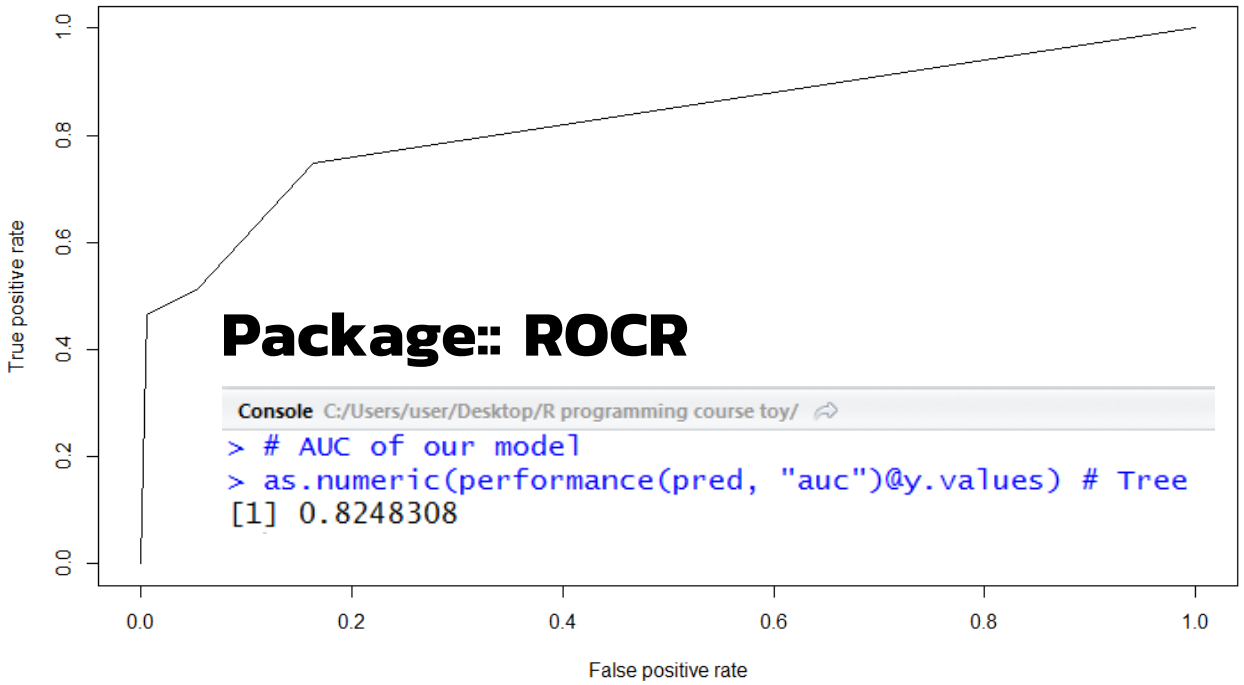
Test Model

```
> table(test$Survived, p)
     p
       NO YES
  NO  139  26
  YES  31  72
> (139 + 72) / nrow(test)
[1] 0.7873134
```

In-sample vs. Out-of-sample errors are quite comparable

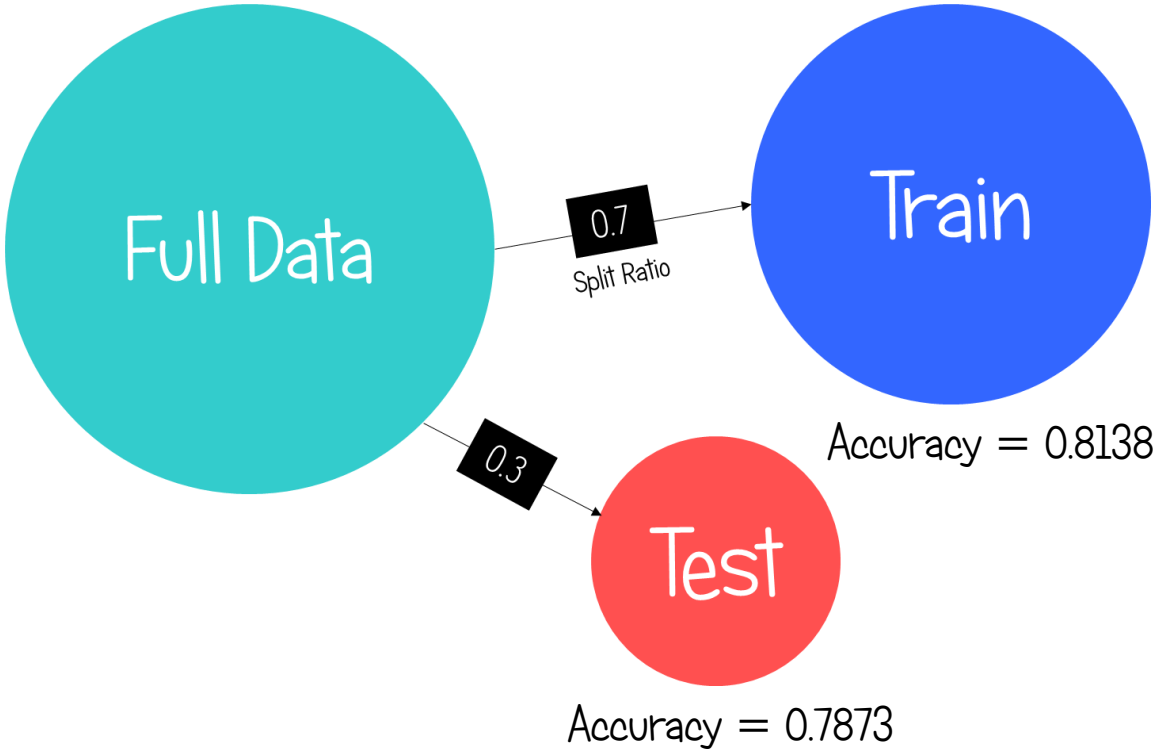
นอกจาก **confusion matrix** เรายังใช้ **ROC** ในการดูว่าโมเดล classification ของเราทำงานดีไหม

```
# ROC
pred = prediction(PredictROC[,2], test$Survived)
perf = performance(pred, "tpr", "fpr")
plot(perf)
```



Package:: ROCR

```
Console C:/Users/user/Desktop/R programming course toy/
> # AUC of our model
> as.numeric(performance(pred, "auc")@y.values) # Tree
[1] 0.8248308
```



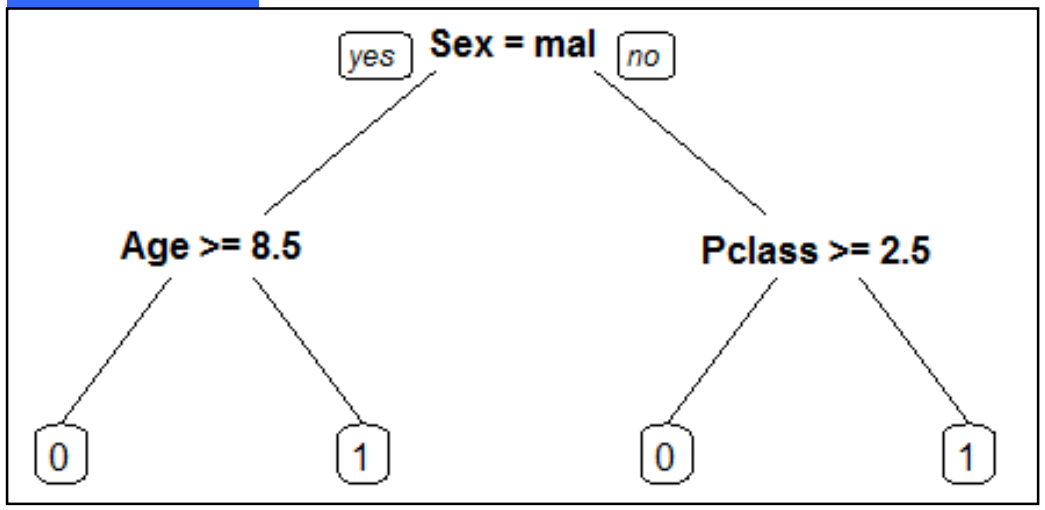
You need to try different models, see what best for you.

ML

Trade off between model accuracy, speed, simplicity

```
Console C:/Users/user/Desktop/R programming course toy/
> as.numeric(performance(pred, "auc")@y.values) # Tree
[1] 0.8162989
> as.numeric(performance(LogitROC, "auc")@y.values) # Logistic
[1] 0.8506043
```

Tree



Logistic

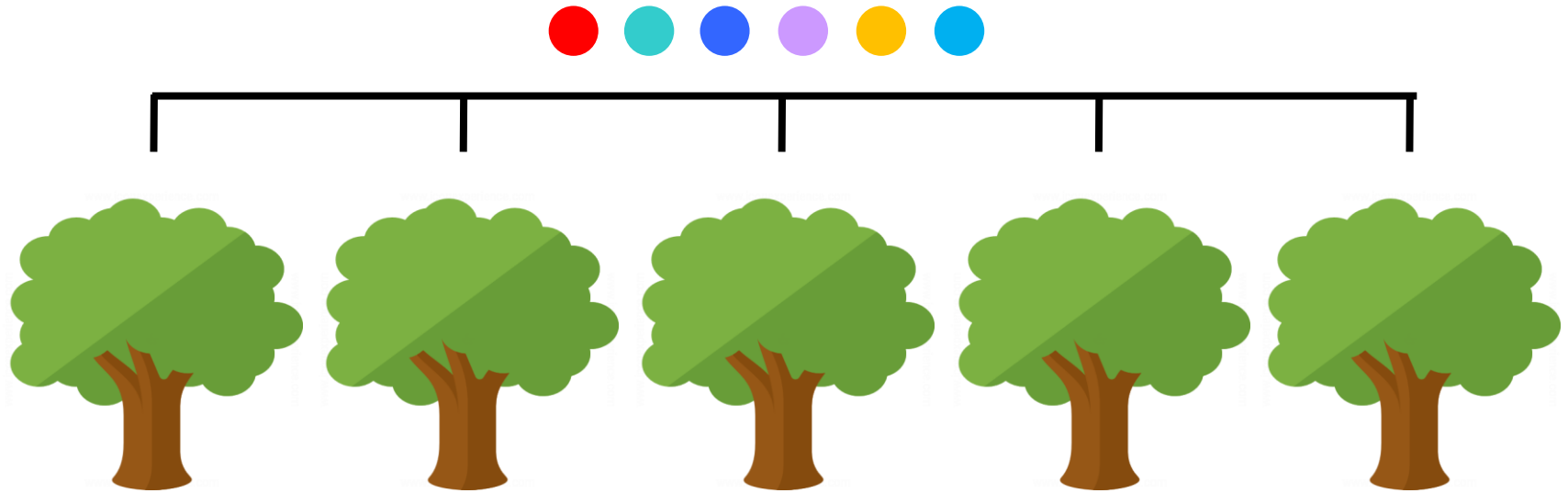
```
> myLogit

Call:  glm(formula = Survived ~ Sex + Age + Pclass, family = "binomial",
    data = train)

Coefficients:
(Intercept)      Sexmale          Age       Pclass
    5.13640     -2.51424     -0.03852     -1.31633

Degrees of Freedom: 499 Total (i.e. Null);  496 Residual
  (123 observations deleted due to missingness)
Null Deviance:      674.6
Residual Deviance: 449.9        AIC: 457.9
```


Model 4
Random Forest

Ideas behind Random Forest learners

ML

ข้อจำกัดของ Decision Tree คือโมเดลมัน **overfit** ง่ายมากๆ วิธีแก้คือเรา grow many trees แล้วค่อยมาหา mean accuracy ของโมเดลเราตอนจบ = **ensemble model**



First split

Second split

เราสุ่ม set of predictor เพื่อสร้าง tree แต่ละชั้น ต้องสุ่มใหม่ตลอดเลย ถึงเรียกมันว่า random

Script to perform Random Forest

ML

Random Forest เป็นเหมือนกล่องดำ อธิบายยากแต่ accuracy ดี

```r
#### RANDOM FOREST ####

# load data
df <- read.csv("breast cancer.csv")

# check missing value
mean(is.na(df))
clean.df <- na.omit(df)

# split data
require(caTools)
set.seed(101)
sample <- sample.split(clean.df$class, SplitRatio = 0.70)
train <- subset(clean.df, sample == TRUE)
test <- subset(clean.df, sample == FALSE)

# train
require(randomForest)
set.seed(101)
forestModel <- randomForest(class ~., data = train, importance = TRUE)

forestModel

(301 + 162) / nrow(train)

# test
forestPredict <- predict(forestModel, test)
1 - mean(forestPredict != test$class)
table(test$class, forestPredict)
```

Package:: randomForest

```
> importance(forestModel, type = 2)
                        MeanDecreaseGini
clumpThickness                  8.061072
sizeUniformity                 54.887089
shapeUniformity                49.758278
maginalAdhesion                 7.799246
singleEpithelialCellSize       19.829797
bareNuclei                     31.249610
blandChromatin                 23.948837
normalNucleoli                 19.497496
mitosis                         1.732731
```

เราสามารถเรียกดูค่า importance ของตัวแปรได้

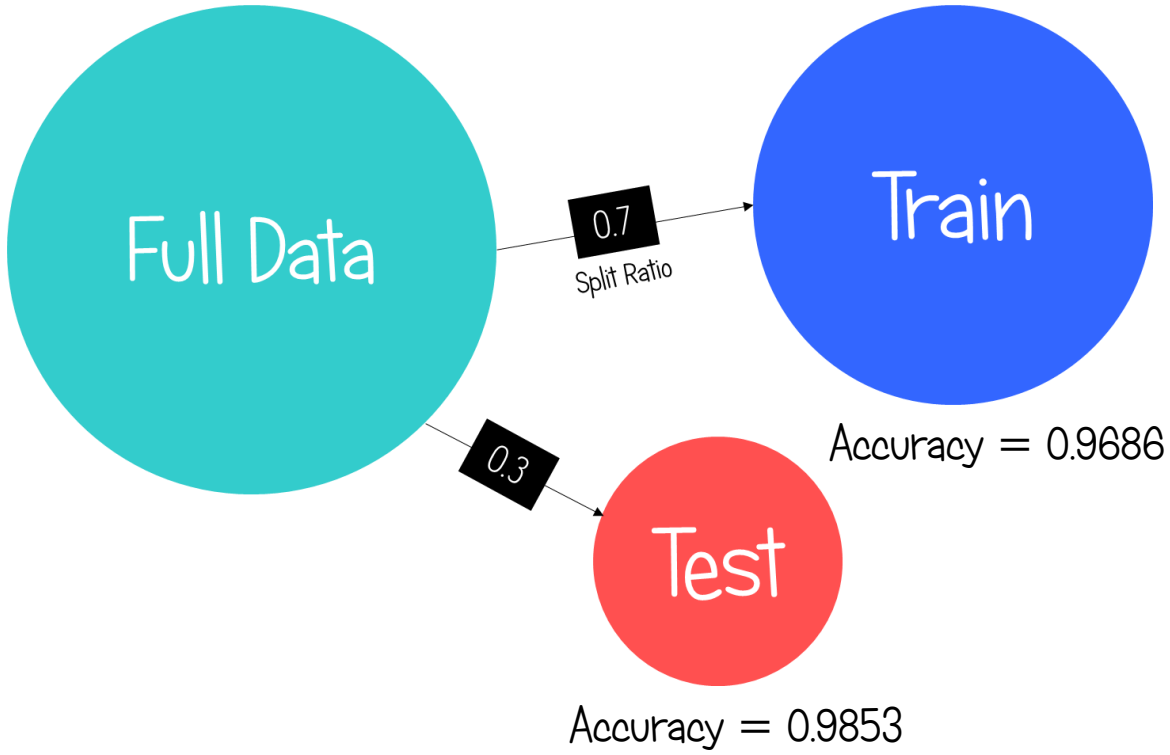
Advanced: ค่า gini ใช้วัดเรื่อง node impurities

Accuracy is higher than Decision Tree significantly

ML

```
Call:
 randomForest(formula = class ~ ., data = train, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 3.14%
Confusion matrix:
          benign malignant class.error
benign       301        10  0.03215434
malignant      5       162  0.02994012
> (301 + 162) / nrow(train)
[1] 0.9686192
```

```
> table(test$class, forestPredict)
           forestPredict
            benign malignant
  benign       132         1
  malignant      2        70
> (132 + 70) / nrow(test)
[1] 0.9853659
```



No Free Lunch

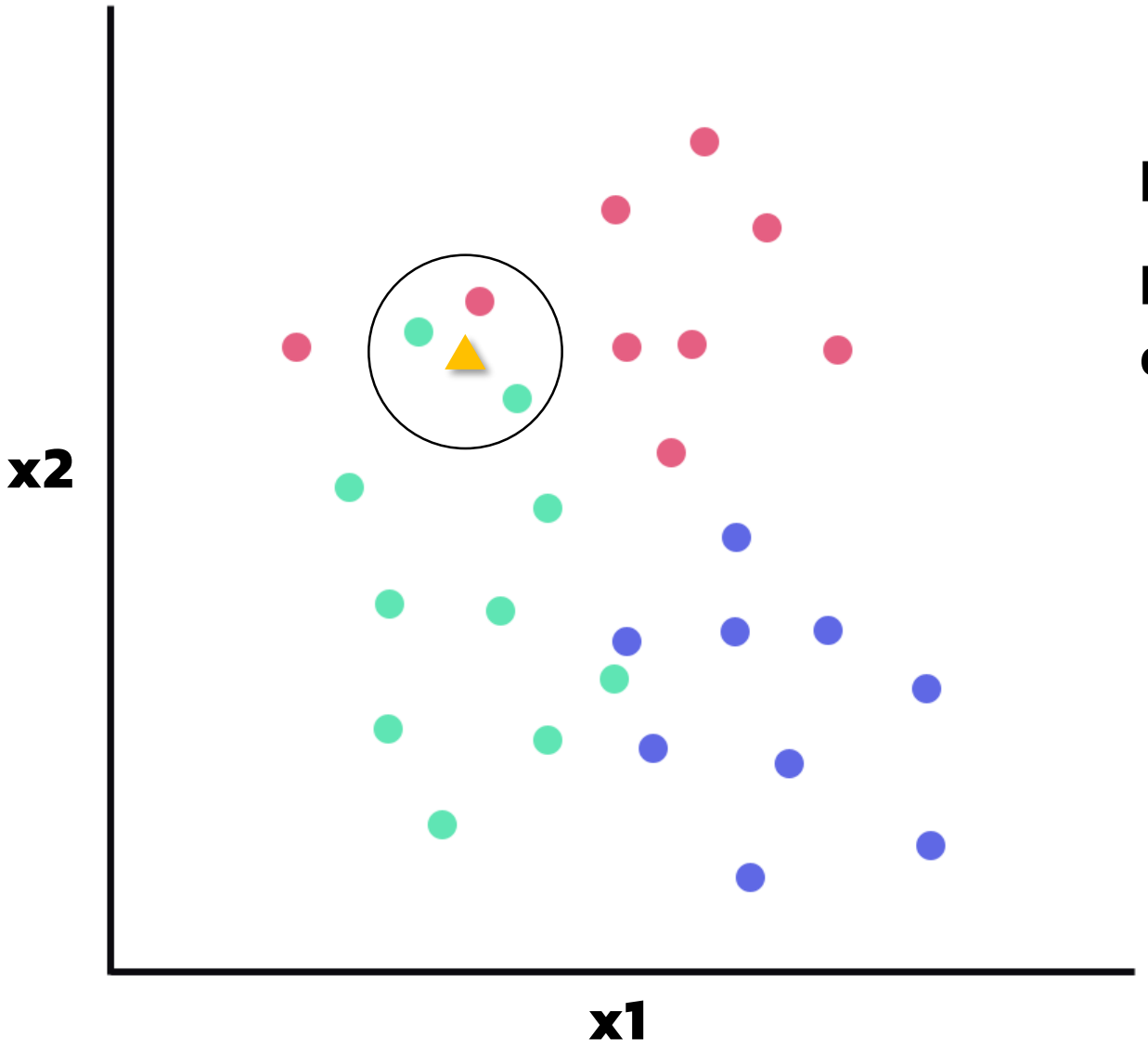
แต่ละโมเดลมีข้อดีข้อเสียไม่เหมือนกัน หน้าที่เราคือฟิตมันหลายๆโมเดล ถถถ+

Decision Tree อธิบายง่าย acc ต่ำ
Random Forest อธิบายไม่ได้ acc สูง

Model 5

KNN

One of the simplest ML model anyone can do



kNN ชื่อเต็มว่า **K-Nearest Neighbor**

K คือจำนวน data points ที่เราใช้ในการ classify unseen data ▲

▲ **เป็นสีอะไรดี?**

Script to perform KNN


```r
install.packages('caTools')
install.packages('class')
require(caTools)
require(class)

## normalize data
scale.data <- scale(iris[1:4])
final.data <- cbind(scale.data, iris[5])

## split data
sample <- sample.split(final.data$Species, SplitRatio = 0.70)
train <- subset(final.data, sample == TRUE)
test <- subset(final.data, sample == FALSE)

## build model
set.seed(101)
fit.model <- knn(train[1:4], test[1:4], train$Species, k = 3)
mean(test$Species != fit.model)
```

สำหรับ knn ให้ดาวโหลด **package:: class**

Computers love normalized data

Split data

Train model

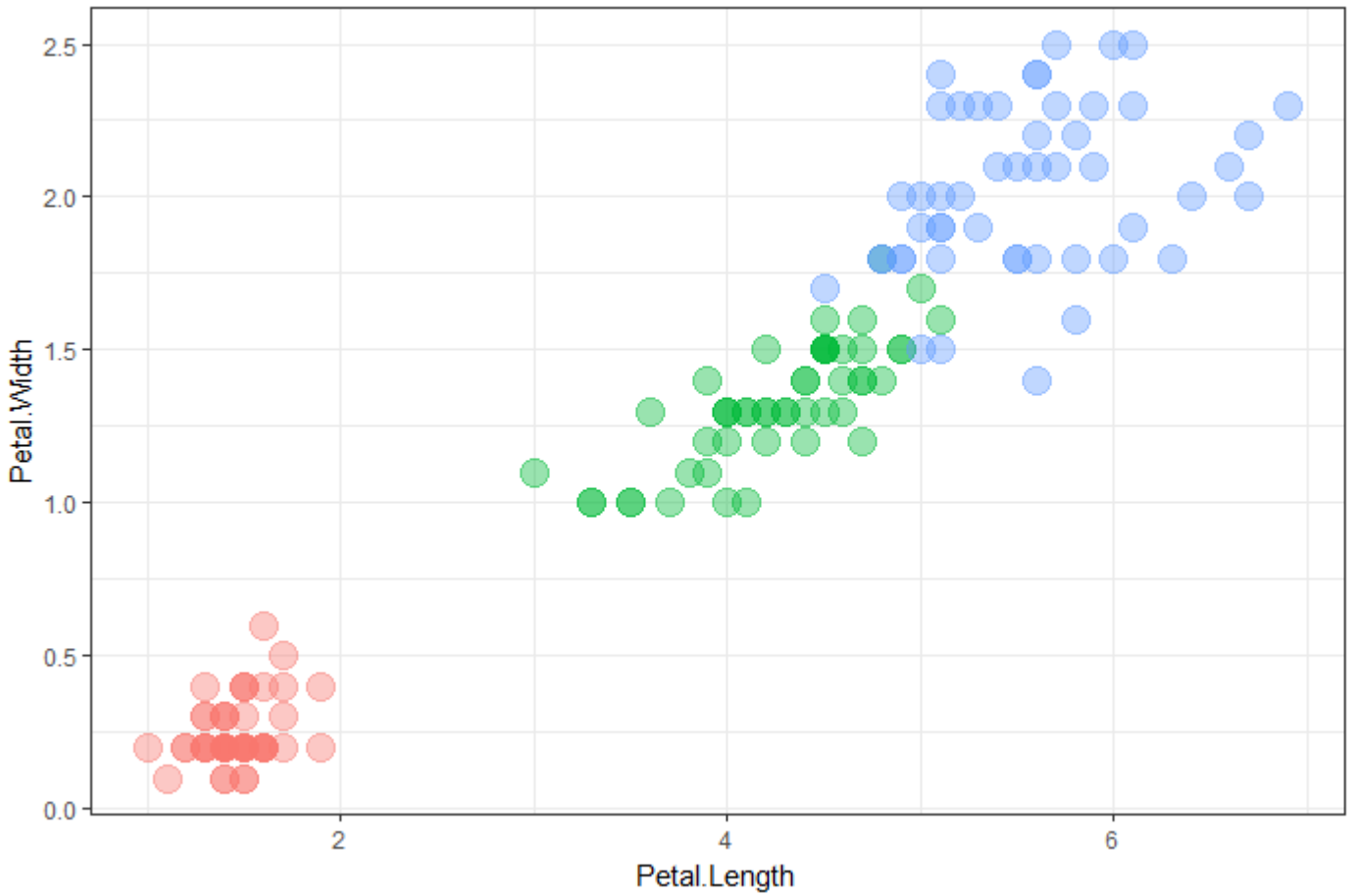
Model 6

K-MEANS

What if you don't know the RIGHT answer?

ML

ถ้าเกิด data ของเราไม่มี label (y) เราต้องเปลี่ยนไปใช้ unsupervised learning แทน




```
ggplot(iris,
       aes(Petal.Length, Petal.Width)) +
    geom_point(aes(color = Species),
               size = 5,
               alpha = 0.4) +
    theme_bw()
```

สมมติถ้าเราลบ Species ออกจาก dataframe iris แล้วให้คอมพิวเตอร์ช่วยจับกลุ่มข้อมูลเราใหม่ มันจะทำได้ดีแค่ไหน?

Script to perform K-MEANS

ML

เราใช้ kmeans(data, center, nstart) ในการสร้างโมเดลง่ายๆ

```
Console C:/Users/user/Desktop/R programming course toy/
> require(class)
> # set seed
> set.seed(101)
> iris.cluster <- kmeans(iris[, 1:4], centers = 3, nstart = 25)
> print(iris.cluster)
K-means clustering with 3 clusters of sizes 62, 50, 38

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.901613    2.748387     4.393548    1.433871
2     5.006000    3.428000     1.462000    0.246000
3     6.850000    3.073684     5.742105    2.071053

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [75] 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 3 3 3 1 3 3 3 3
[112] 3 3 1 1 3 3 3 3 1 3 1 3 1 3 3 1 1 3 3 3 3 3 1 3 3 3 1 3 3 3 1 3 3 3 1 3
[149] 3 1

Within cluster sum of squares by cluster:
[1] 39.82097 15.15100 23.87947
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

อันนี้คือ prediction ของเรา

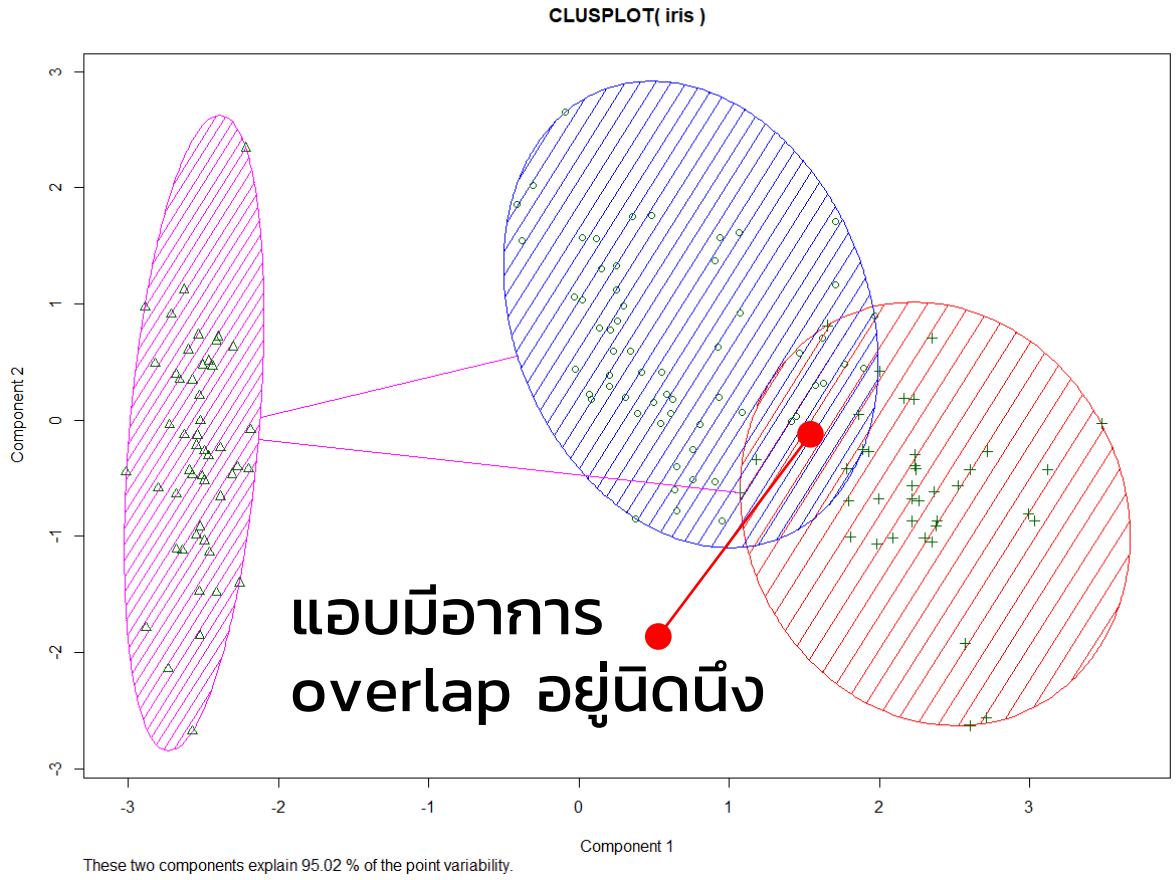
Script to perform K-MEANS

ML

```
Console C:/Users/user/Desktop/R programming course toy/
> library(cluster)
> clusplot(iris, iris.cluster$cluster, color = TRUE, shade = TRUE)
> table(iris$Species, iris.cluster$cluster)

             1  2  3
  setosa     0 50  0
  versicolor 48  0  2
  virginica  14  0 36
> (50 + 48 + 36) / nrow(iris) # accuracy
[1] 0.8933333
```

นี้ขนาดไม่รู้ว่า class จริงๆเป็นยังไง ยัง predict ได้แม่นยำตั้ง 89.3% !!

Virginica ดูจะ classify ยากสุดเลย เพราะโมเดลเราไปสับสนกับ Setosa ด้วย




CARET
Train model like PRO

One of the most powerful package for supervised learning

ML

Max Kuhn

Classification and
REgression Training

You are ready to dive in :D

```
install.packages("caret")
require(caret)
```

http://topepo.github.io/caret/index.html

CROSS VALIDATION - Concept

ML

10-Fold Cross Validation = 9 Train + 1 Test

CROSS VALIDATION – How it works?

ML

เราจะรันโมเดลใช้ 9 Train + 1 Test ไปเรื่อยๆจนกว่าทุก fold จะถูกใช้ครบ → สุดท้ายเราจะใช้ data ทั้งหมดในการสร้าง final model

Just a glimpse on CARET package

ML

```
5  data(diamonds)
6  glimpse(diamonds)
7
8  #### build model using caret ####
9
10 model1 <- train(price ~., diamonds, method = "lm",
11                 trControl = trainControl(method = "cv",
12                                          number = 10,
13                                          verboseIter = TRUE))
```

Linear Regression

Cross Validation

CARET เหมือนกับ ggplot2 ที่มี template ในการสร้าง model ให้กับเรา

สิ่งเดียวที่ต้องเปลี่ยนคือ **method = “lm”** ที่ใช้ในการ define model ของเรา

and the Final model

เรารัน linear regression โดยใช้ 10-fold cross validation ในการทดสอบโมเดลของเรา

```
> print(mod1)
Linear Regression

53940 samples
    9 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 48547, 48546, 48546, 48547, 48546, 48545, ...
Resampling results:

  RMSE      Rsquared
  1130.955  0.9197637

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Performance

Remember

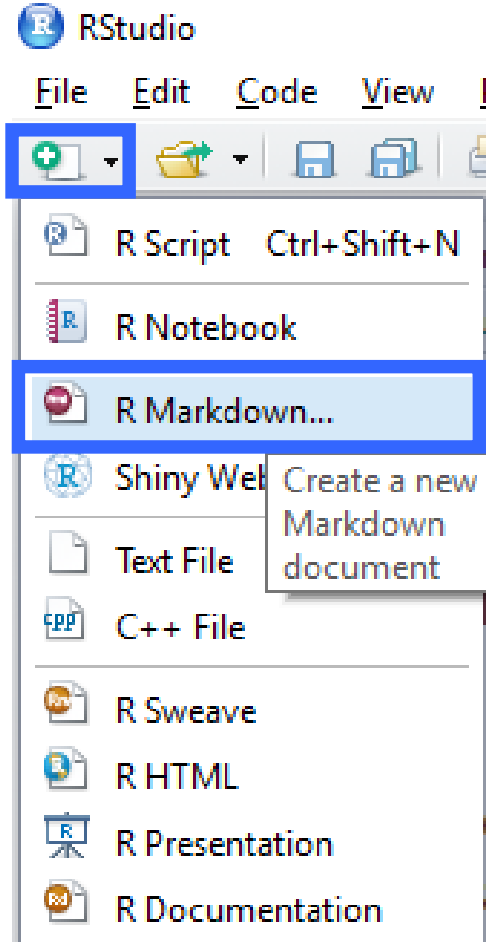
Insights and persistence are what counts

R markdown Report

Basic

We use markdown to produce report in R

RMD



Pandoc's Markdown

Write with syntax on the left to create effect on right (after render)

```
Plain text
End a line with two spaces
to start a new paragraph.
*italics* and **bold**
`verbatim code`
sub/superscript^2^~2~
~~strikethrough~~
escaped: \* \_ \\
endash: --, emdash: ---
equation: $A = \pi*r^{2}$
equation block:
$$E = mc^{2}$$
> block quote
# Header1  {#anchor}
## Header 2  {#css_id}
### Header 3  {.css_class}
#### Header 4
##### Header 5
###### Header 6
<!--Text comment-->
\textbf{Tex ignored in HTML}
<em>HTML ignored in pdfs</em>
<http://www.rstudio.com>
[link](www.rstudio.com)
Jump to [Header 1](#anchor)
image:
```

Plain text
End a line with two spaces
to start a new paragraph.
italics and **bold**
`verbatim code`
sub/superscript$^2_2$
~~strikethrough~~
escaped: * _ \
endash: –, emdash: —
equation: $A = \pi * r^2$
equation block:

$$E = mc^2$$

> block quote

Header1

Header 2

Header 3

Header 4

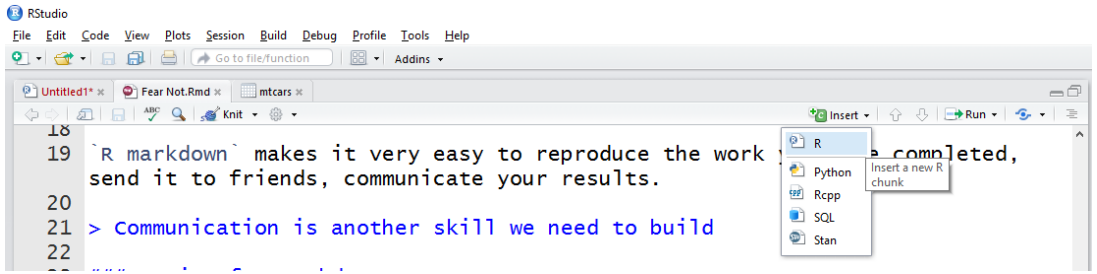
Header 5

Header 6

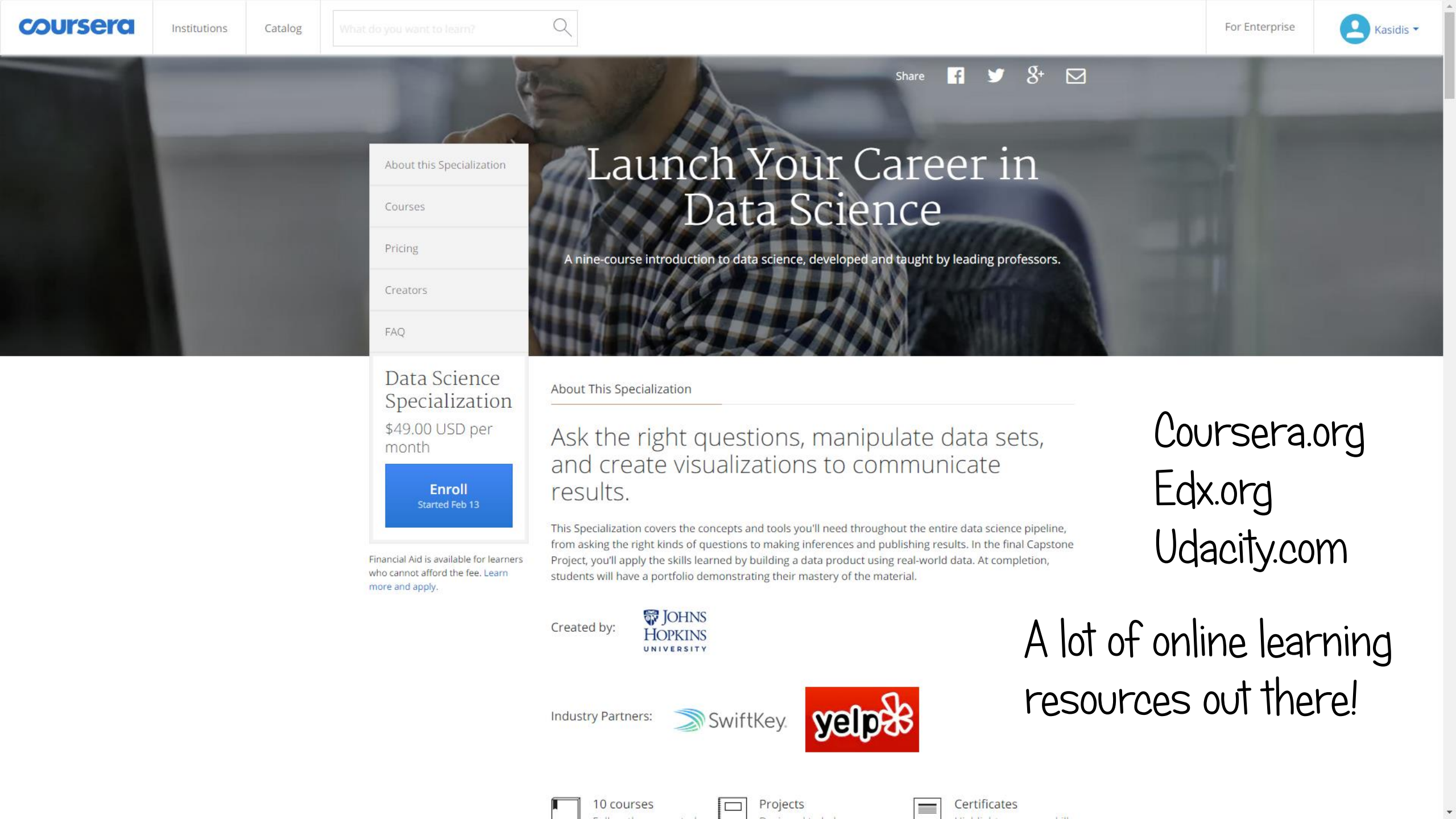
HTML ignored in pdfs

http://www.rstudio.com
link
Jump to Header 1
image:

You can insert R code chunk by using "insert" icon on the top right



Remark: R markdown cheat sheet https://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf



Coursera.org
Edx.org
Udacity.com

A lot of online learning resources out there!

Congratulations on completing the course :D

Best of luck. Glad to meet you all today.


Thank You
R Basics for Starters
www.bitesize.studio
www.FB.com/datarockie