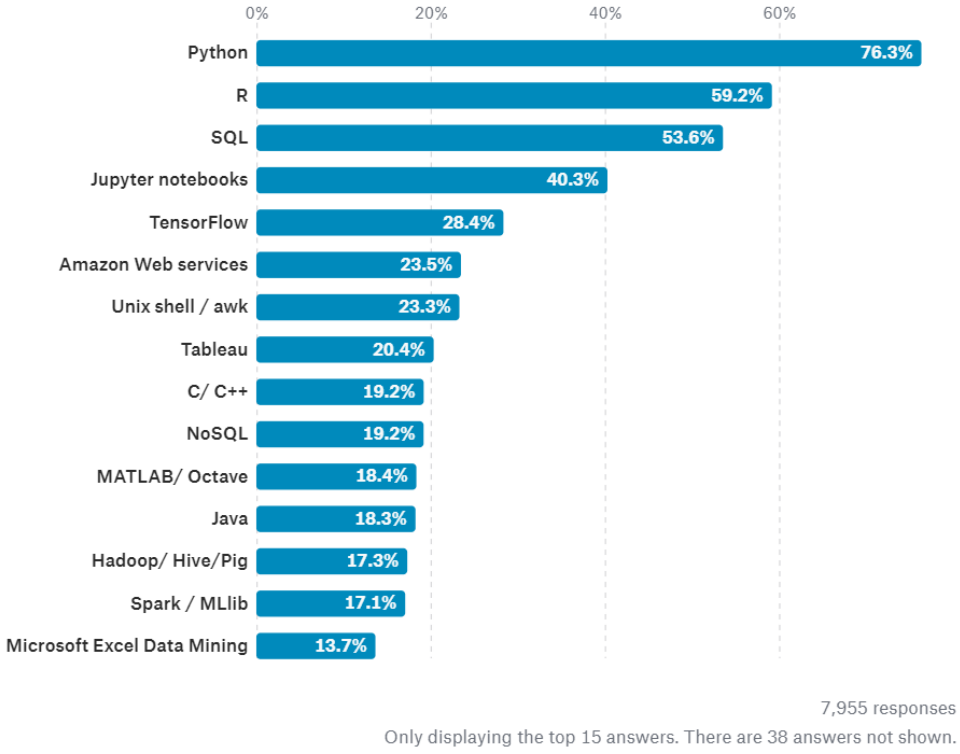
INTRO TO R PROGRAMMING

Motivations

https://www.kaggle.com/surveys/2017

What tools you use at work?



7,955 responses

Only displaying the top 15 answers. There are 38 answers not shown.

R Programmer Salaries in New York, NY

Salary estimated from 2,477 employees, users, and past and present job advertisements on Indeed in the past 36 months. Last updated: July 10, 2018

Location

New York

Salary not bad!

Popular Jobs	Average Salary	Salary Distribution
Programmer 119 salaries reported Programmer jobs in New York, NY	**$92,801** per year	
Programmer Analyst 253 salaries reported Programmer Analyst jobs in New York, NY	**$80,458** per year	
Data Scientist 1,133 salaries reported Data Scientist jobs in New York, NY	**$148,040** per year	
Senior Programmer 59 salaries reported Senior Programmer jobs in New York, NY	**$90,658** per year	
Data Analyst 493 salaries reported Data Analyst jobs in New York, NY	**$77,539** per year	

Data

Data Analyst

Data Analyst

Our Main Tasks

1. Clean
2. Transform
3. Summarise
4. Model
5. Visualize

School teaches us what to learn but ..

NEVER teaches HOW to learn.

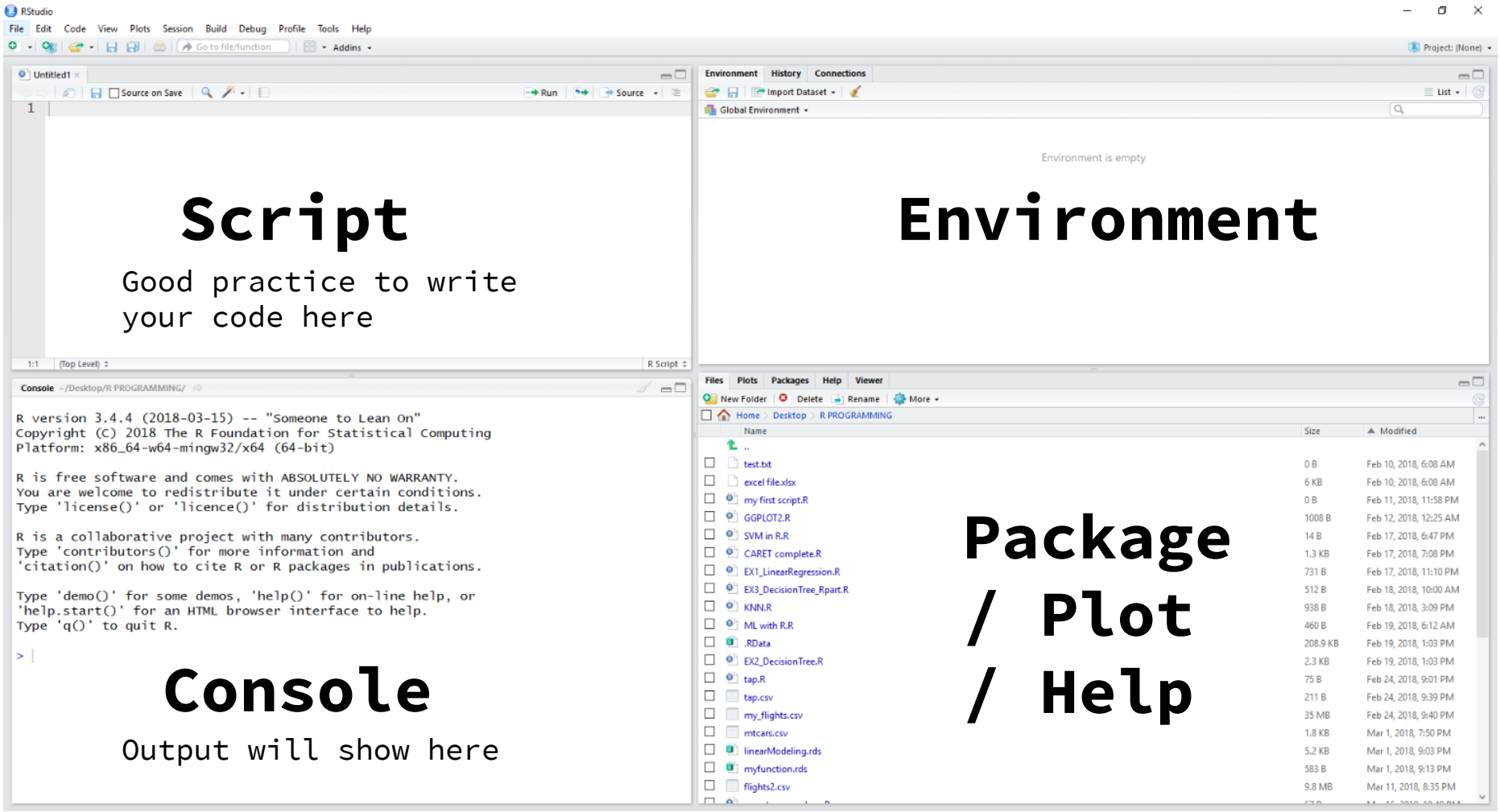
Attitudes

Learning should be HARD.

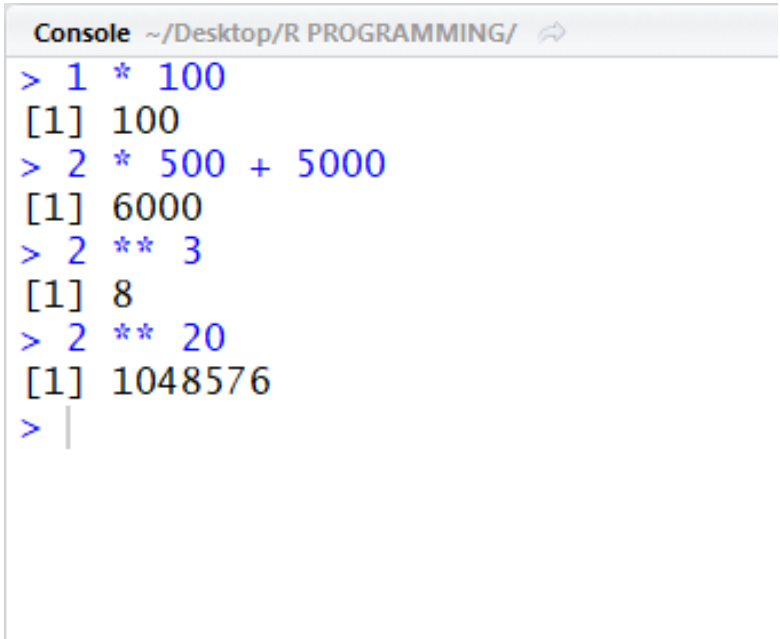
What we’re going to learn today?

- R Basics
- Data Types
- Data Frame
- Data Transformation
- Data Visualization
- Data Modeling


INTRO TO
R & RSTUDIO


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled1
Source on Save
Run
Source
1
Script
Good practice to write your code here
Environment History Connections
Import Dataset
Global Environment
Environment is empty
Environment
Console ~/Desktop/R PROGRAMMING/
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Console
Output will show here
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home Desktop R PROGRAMMING
Name Size Modified
test.txt 0 B Feb 10, 2018, 6:08 AM
excel file.xlsx 6 KB Feb 10, 2018, 6:08 AM
my first script.R 0 B Feb 11, 2018, 11:58 PM
GGPLOT2.R 1008 B Feb 12, 2018, 12:25 AM
SVM in R.R 14 B Feb 17, 2018, 6:47 PM
CARET complete.R 1.3 KB Feb 17, 2018, 7:08 PM
EX1_LinearRegression.R 731 B Feb 17, 2018, 11:10 PM
EX3_DecisionTree_Rpart.R 512 B Feb 18, 2018, 10:00 AM
KNN.R 938 B Feb 18, 2018, 3:09 PM
ML with R.R 460 B Feb 19, 2018, 6:12 AM
.RData 208.9 KB Feb 19, 2018, 1:03 PM
EX2_DecisionTree.R 2.3 KB Feb 19, 2018, 1:03 PM
tap.R 75 B Feb 24, 2018, 9:01 PM
tap.csv 211 B Feb 24, 2018, 9:39 PM
my_flights.csv 35 MB Feb 24, 2018, 9:40 PM
mtcars.csv 1.8 KB Mar 1, 2018, 7:50 PM
linearModeling.rds 5.2 KB Mar 1, 2018, 9:03 PM
myfunction.rds 583 B Mar 1, 2018, 9:13 PM
flights2.csv 9.8 MB Mar 11, 2018, 8:35 PM
Package / Plot / Help

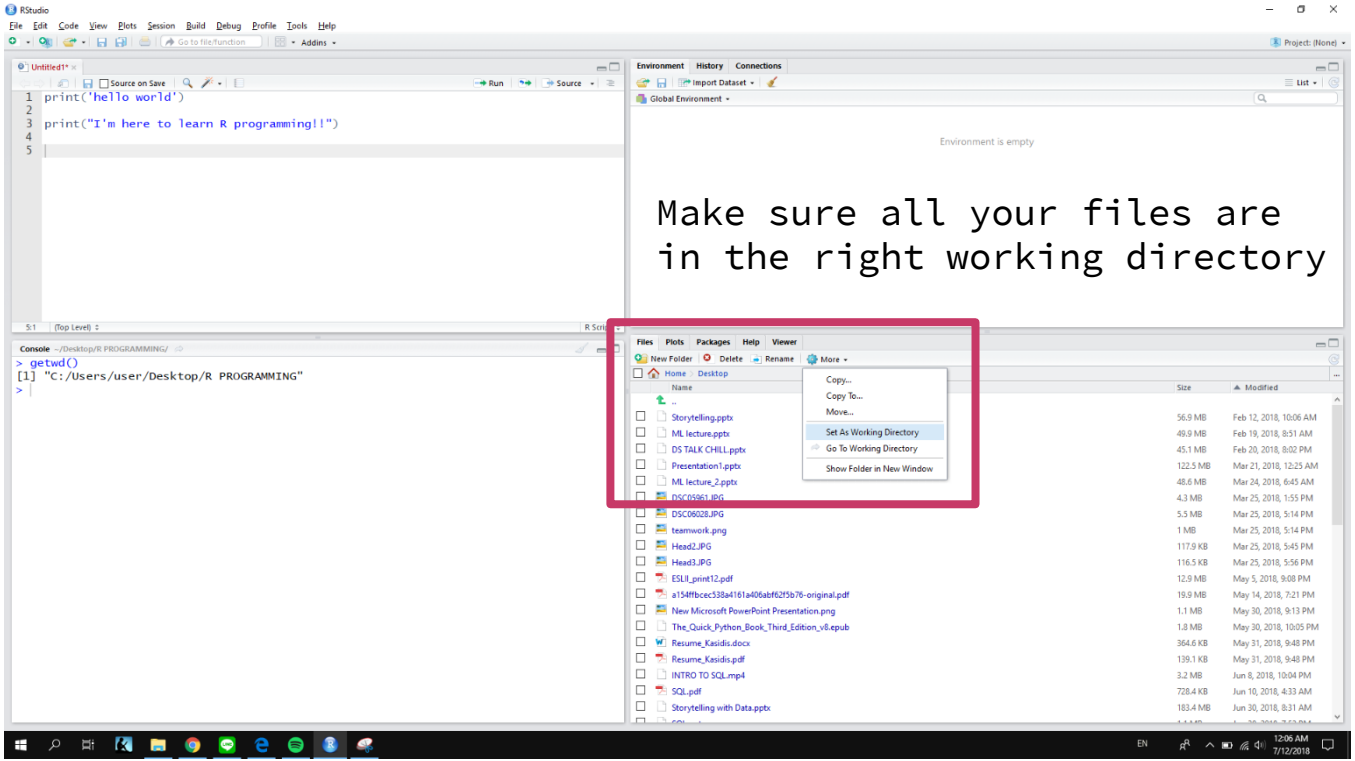
R is (advanced) calculator



Basic Operations in R

Addition (+)
Subtraction (-)
Multiply (*)
Division (/)
Power (**)
Modulo (%%)

Set Working Directory



Assign Variables

```
money <- 2000
food_expense <- 500

left_over <- money - food_expense
```

Good Variable Names

```
# use underscore to connect words
hello_world
my_expense
student_scores
total_sales
```

Keyboard Shortcuts

CTRL+ENTER to run code in that line

CTRL+SHIFT+ENTER to run script

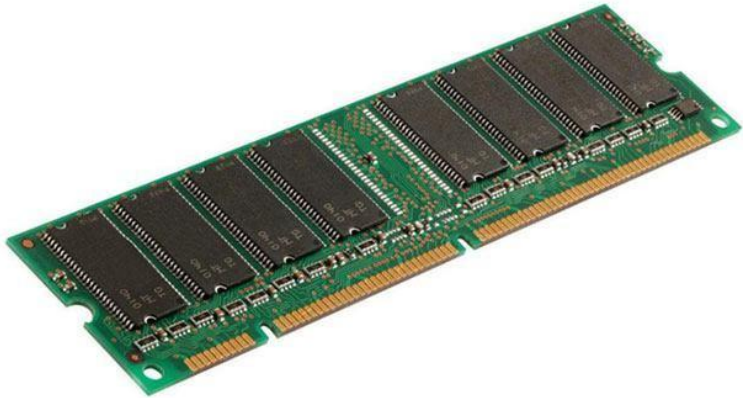
CTRL+L to clear script

CTRL+1 or **CTRL+2** to switch cursor

F1 when you need help about function


WHAT YOU SHOULD
KNOW ABOUT R

- R is **object-oriented programming** language
- R is case sensitive



- R keeps data in memory (RAM)
- R uses single core CPU (default)
- Everything in R is a function call

Basic Data Types

1. **Numeric** -- 1.23 2.57 300 200 120.2
2. **Character** -- “Hello” “Data” “Rockie”
3. **Logical** -- TRUE, FALSE
4. **Factor** -- male/female

Basic Data Structures

1. **Vector** -- `c(100,200,300)`
2. **List** -- `list(x, y)`
3. **Matrix** -- `matrix(1:9, ncol = 3, byrow = T)`
4. **DataFrame** -- What we use the most!!

Subsetting

There are 3 ways to subset data in R

1. By position
2. By name
3. By condition (logic)

If you see this
[ ] it’s subsetting.

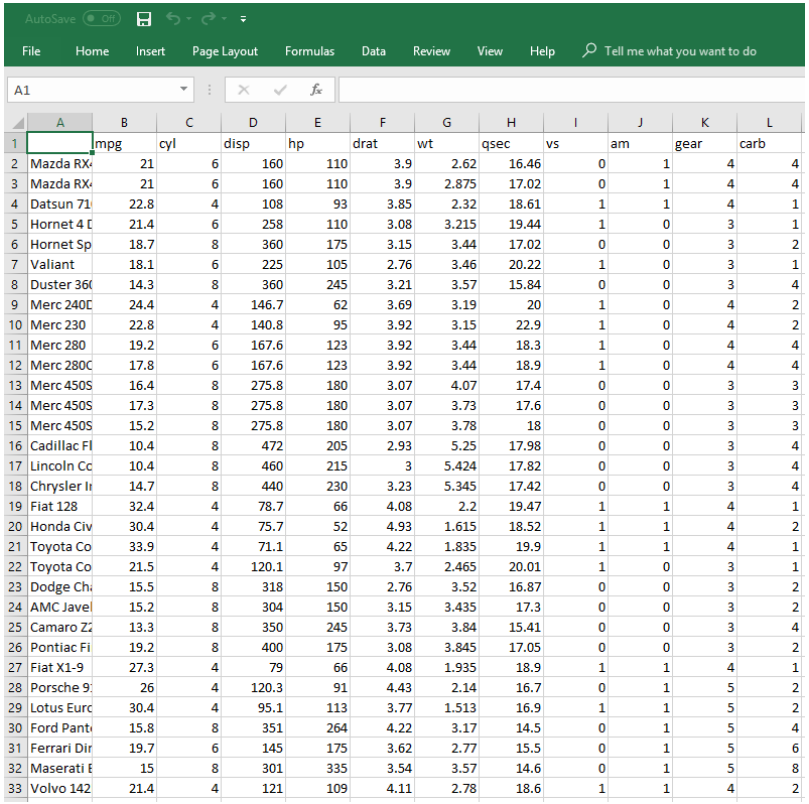
Examples

```
# create a vector
set.seed(123)
x <- rnorm(20)
print(x)
```

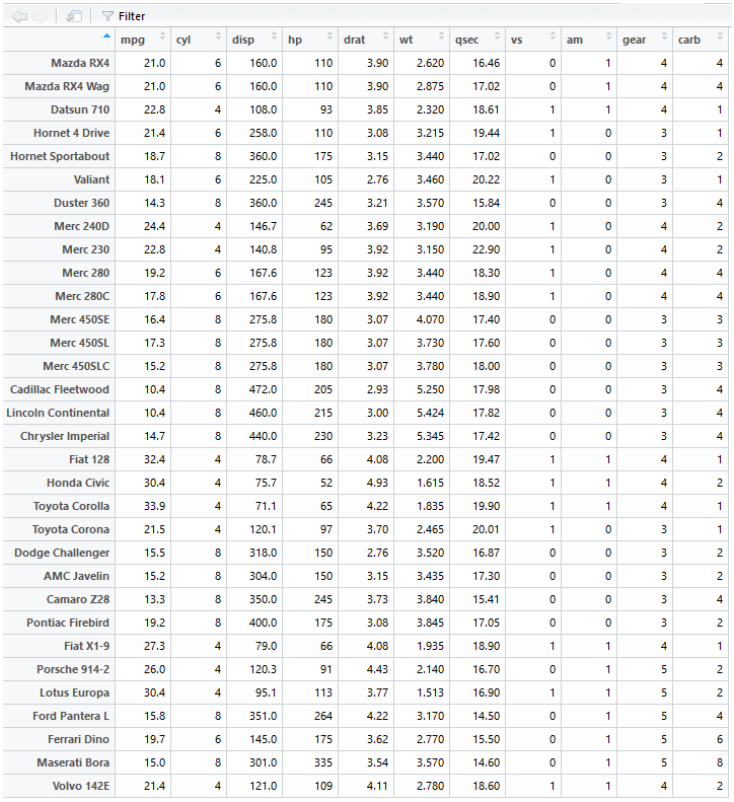
```
 [1] -0.56047565 -0.23017749  1.55870831  0.07050839  0.12928774
 [6]  1.71506499  0.46091621 -1.26506123 -0.68685285 -0.44566197
[11]  1.22408180  0.35981383  0.40077145  0.11068272 -0.55584113
[16]  1.78691314  0.49785048 -1.96661716  0.70135590 -0.47279141
```

```
# subset by position
x[1:5]
x[c(1:5, 10)]

# subset by condition/logic
x[x > 0]
X[x*2 >= 3]
```

What is DataFrame?



		mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
2	Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
3	Mazda RX4	21	6	160	110	3.9	2.875	17.02	0	1	4	4
4	Datsun 71	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
5	Hornet 4 D	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
6	Hornet Sp	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
7	Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1
8	Duster 36(	14.3	8	360	245	3.21	3.57	15.84	0	0	3	4
9	Merc 240D	24.4	4	146.7	62	3.69	3.19	20	1	0	4	2
10	Merc 230	22.8	4	140.8	95	3.92	3.15	22.9	1	0	4	2
11	Merc 280	19.2	6	167.6	123	3.92	3.44	18.3	1	0	4	4
12	Merc 280C	17.8	6	167.6	123	3.92	3.44	18.9	1	0	4	4
13	Merc 450S	16.4	8	275.8	180	3.07	4.07	17.4	0	0	3	3
14	Merc 450S	17.3	8	275.8	180	3.07	3.73	17.6	0	0	3	3
15	Merc 450S	15.2	8	275.8	180	3.07	3.78	18	0	0	3	3
16	Cadillac Fl	10.4	8	472	205	2.93	5.25	17.98	0	0	3	4
17	Lincoln Cc	10.4	8	460	215	3	5.424	17.82	0	0	3	4
18	Chrysler Ir	14.7	8	440	230	3.23	5.345	17.42	0	0	3	4
19	Fiat 128	32.4	4	78.7	66	4.08	2.2	19.47	1	1	4	1
20	Honda Civ	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
21	Toyota Co	33.9	4	71.1	65	4.22	1.835	19.9	1	1	4	1
22	Toyota Co	21.5	4	120.1	97	3.7	2.465	20.01	1	0	3	1
23	Dodge Cha	15.5	8	318	150	2.76	3.52	16.87	0	0	3	2
24	AMC Javel	15.2	8	304	150	3.15	3.435	17.3	0	0	3	2
25	Camaro Z2	13.3	8	350	245	3.73	3.84	15.41	0	0	3	4
26	Pontiac Fi	19.2	8	400	175	3.08	3.845	17.05	0	0	3	2
27	Fiat X1-9	27.3	4	79	66	4.08	1.935	18.9	1	1	4	1
28	Porsche 9	26	4	120.3	91	4.43	2.14	16.7	0	1	5	2
29	Lotus Euro	30.4	4	95.1	113	3.77	1.513	16.9	1	1	5	2
30	Ford Pant	15.8	8	351	264	4.22	3.17	14.5	0	1	5	4
31	Ferrari Din	19.7	6	145	175	3.62	2.77	15.5	0	1	5	6
32	Maserati E	15	8	301	335	3.54	3.57	14.6	0	1	5	8
33	Volvo 142	21.4	4	121	109	4.11	2.78	18.6	1	1	4	2



	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

DataFrame Basics

```
library(tidyverse)

glimpse(mtcars)
dim(mtcars)
head(mtcars)
tail(mtcars)
summary(mtcars)
nrow(mtcars)
ncol(mtcars)
names(mtcars)

mean(mtcars$mpg)
median(mtcars$mpg)
sd(mtcars$mpg)
sum(mtcars$mpg)
table(mtcars$am)

complete.cases(mtcars)
drop_na(mtcars)
```

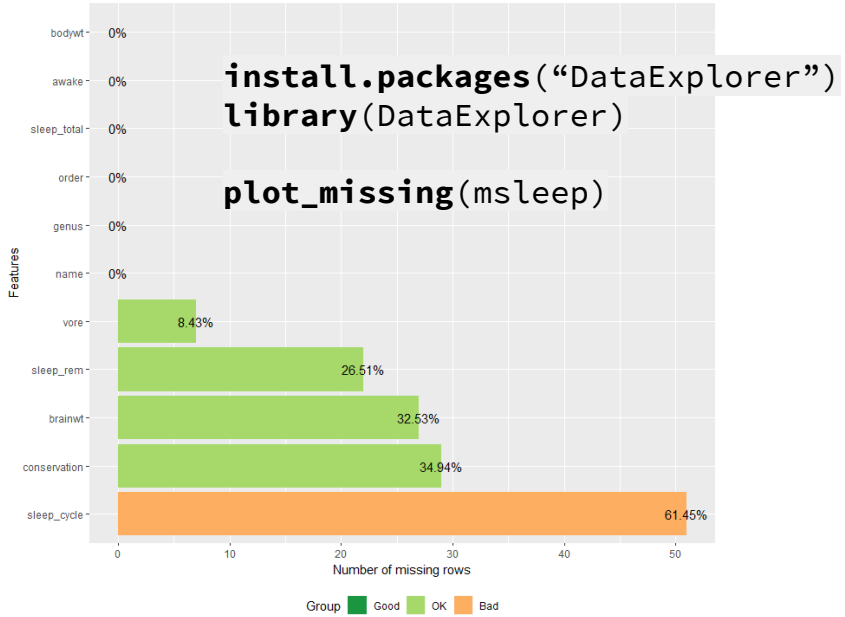
Let’s CLEAN some data

```
library(tidyverse)

glimpse(msleep)
```

Quiz

- Any NA values?
- How many complete cases in dataset?

```
install.packages(“DataExplorer”)
library(DataExplorer)

plot_missing(msleep)
```

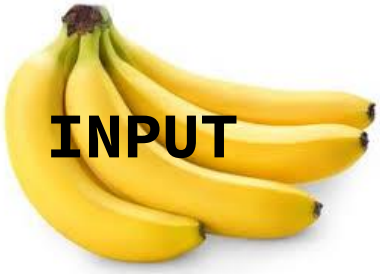


Imputation

- Mean
- Median

Replace missing value
with mean or median


FUNCTION



FUNCTION

```
output <- make_smoothie(🍌)

print(output)
```

Leaderboard

15,520
indexed packages

2,207,054
indexed functions

Most downloaded packages

Name	Direct downloads	Indirect downloads	Total

Next >

Most active maintainers

Name	Direct downloads	Indirect downloads	Total

Next >

Newest packages

washeR	Wed Jul 11 2018 11:00:03
rt.test	Tue Jul 10 2018 17:30:03
stablelearner	Tue Jul 10 2018 17:20:03
SubgrPlots	Tue Jul 10 2018 17:10:09
syllabifyr	Tue Jul 10 2018 17:10:02
peakPantheR	Tue Jul 10 2018 17:00:03
DiscreteFDR	Tue Jul 10 2018 16:50:07
countgmifs	Tue Jul 10 2018 16:40:28
memor	Tue Jul 10 2018 16:40:25

Newest updates

pavo	Wed Jul 11 2018 16:20:03
bmlm	Wed Jul 11 2018 16:10:03
nabor	Wed Jul 11 2018 16:00:02
did	Wed Jul 11 2018 15:40:03
xkcd	Wed Jul 11 2018 15:30:02
bigmatch	Wed Jul 11 2018 15:20:02
llama	Wed Jul 11 2018 14:30:03
desctable	Wed Jul 11 2018 14:20:02
metansue	Wed Jul 11 2018 13:40:03


Hadley Wickham
Chief Data Scientist @RStudio

Function Anatomy

```
function_name(arg1, arg2, …)
```

Write an R function is simple

```
sum_two_nums <- function(a,b) {
    return(a+b)
}
```

Your First R Function

```
roll_dices <- function(){
    ...
}
```


DATA
TRANSFORMATION

Five Verbs from dplyr

```
install.packages("dplyr")
library(dplyr)
```

- select() -- เลือก column
- filter() -- เลือก row
- arrange() -- เรียงข้อมูล
- mutate() -- สร้างตัวแปร (คอลั่มใหม่)
- summarise() -- สรุปผลสถิติ

select()

```
select(df, column1, column2, …)

select(mtcars, wt, hp, mpg)
select(mtcars, 1:3)
select(mtcars, starts_with(“a”))
select(mtcars, ends_with(“p”))
select(mtcars, contains(“a”))
```

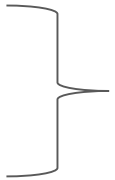
filter()

```
filter(df, conditions)
```

```
filter(mtcars, mpg > 20)
filter(mtcars, mpg > 20 & gear == 5)
filter(mtcars, mpg > 30 | mpg < 15)
filter(mtcars, hp %in% 100:200)
filter(mtcars, carb %in% c(1,8))
```

arrange()

```
arrange(df, column1, column2, …)

arrange(mtcars, mpg)

arrange(mtcars, desc(mpg))
arrange(mtcars, -mpg)
```



Descending order

pipe operator

```
mtcars %>%
    select(wt, hp, mpg) %>%
    filter(mpg > 20) %>%
    arrange(-mpg) %>%
    head(3)
```

mutate()

```
mutate(df, new_column = ...)
```

```
mtcars %>%
   mutate(hp_power = hp ** 2,
          hp_sqrt = sqrt(hp),
             hp_100 = hp + 100) %>%
   head(10)
```

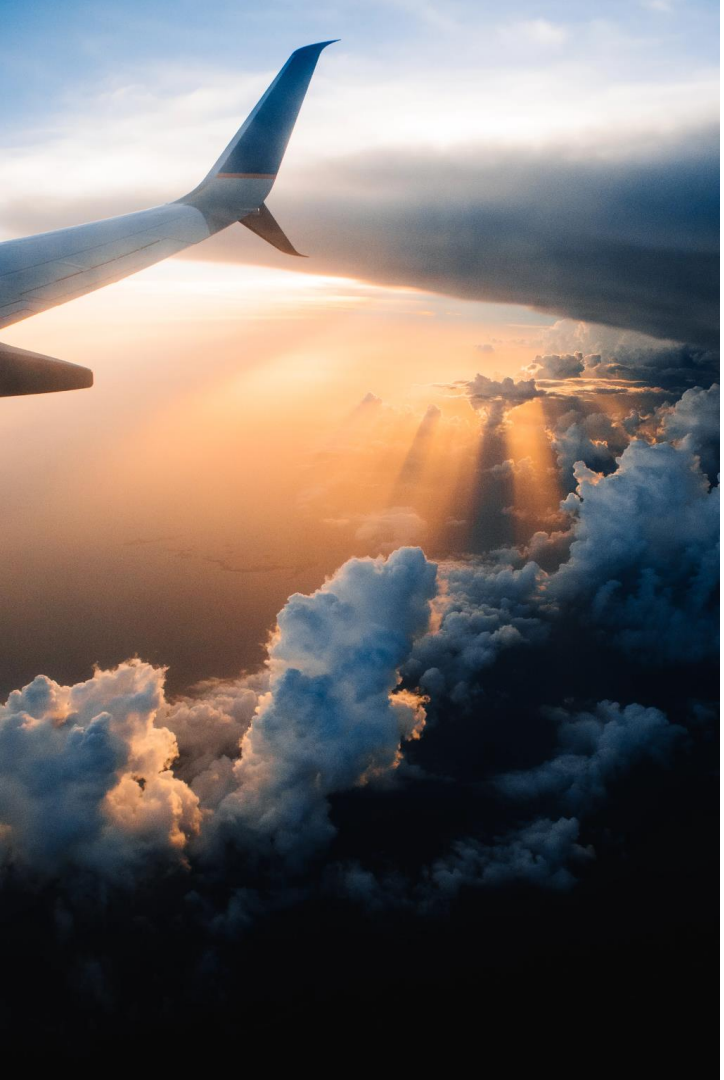
summarise()

```
summarise(df, statistics_function)
```

```
mtcars %>%
   summarise(avg_mpg = mean(mpg),
             sd_mpg = sd(mpg),
             max_weight = max(wt),
             min_weight = min(wt))
```

group_by()

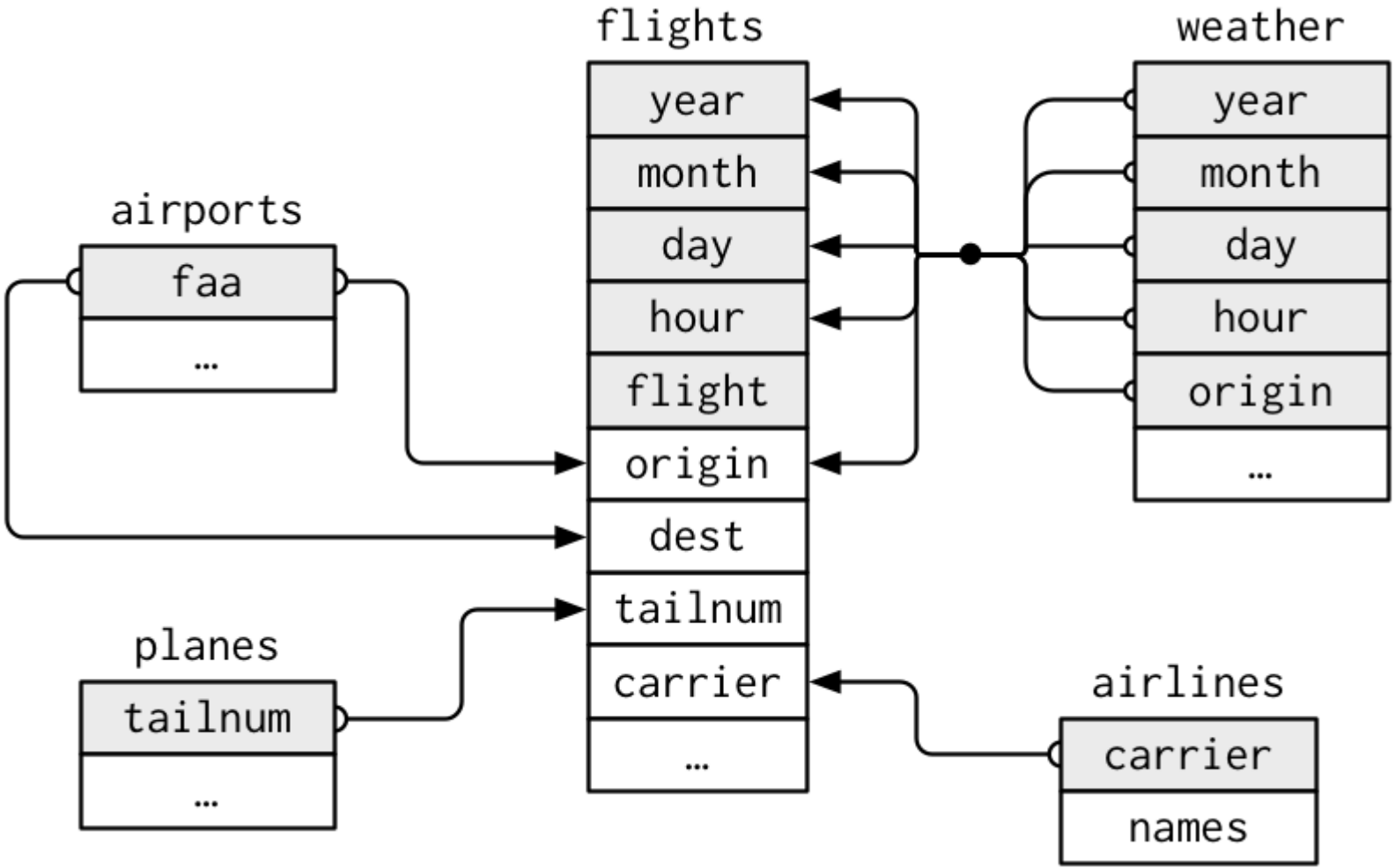
```
mtcars %>%
    group_by(factor(gear)) %>%
  summarise(avg_mpg = mean(mpg),
            sd_mpg = sd(mpg),
            max_weight = max(wt),
            min_weight = min(wt),
            n = n())
```



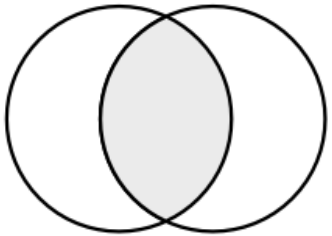
Analyzing NYC flights

```
install.packages("nycflights13")
library(nycflights13)
library(tidyverse)

glimpse(flights)
```

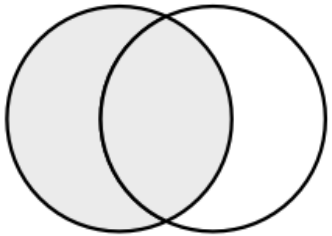
Challenge:
10 สายการบินที่บินเยอะที่สุดในเดือนกันยายนปี 2013 ชื่อว่าอะไรบ้าง?


flights
year
month
day
hour
flight
origin
dest
tailnum
carrier
…
weather
year
month
day
hour
origin
…
airports
faa
…
planes
tailnum
…
airlines
carrier
names

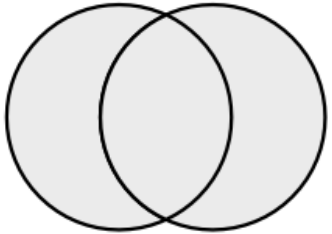
Four main types of JOIN



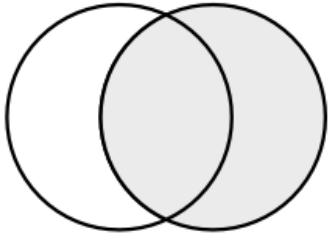
inner_join(x, y)



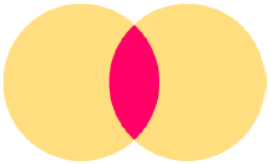
left_join(x, y)



full_join(x, y)



right_join(x, y)



INNER JOIN

Customer

ID	Name
1	Toy
2	Hello
3	World
4	SQL
5	Awesome

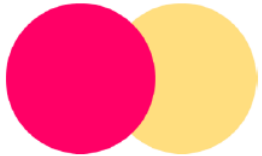
Age

ID	Age
1	29
2	30
4	18
6	25
7	26

Result

ID	Name	Age
1	Toy	29
2	Hello	30
4	SQL	18

ผลลัพธ์ออกมาเฉพาะ ROW ที่ matched กันได้ของสองตารางเท่านั้น



LEFT JOIN

ตารางซ้ายมือยังอยู่เหมือนเดิม แต่จะเชื่อมตารางขวาใน row ที่ matched

Customer

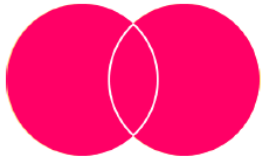
ID	Name
1	Toy
2	Hello
3	World
4	SQL
5	Awesome

Age

ID	Age
1	29
2	30
4	18
6	25
7	26

Result

ID	Name	Age
1	Toy	29
2	Hello	30
3	World	*NULL*
4	SQL	18
5	Awesome	*NULL*



FULL OUTER JOIN

Customer

ID	Name
1	Toy
2	Hello
3	World
4	SQL
5	Awesome

Age

ID	Age
1	29
2	30
4	18
6	25
7	26

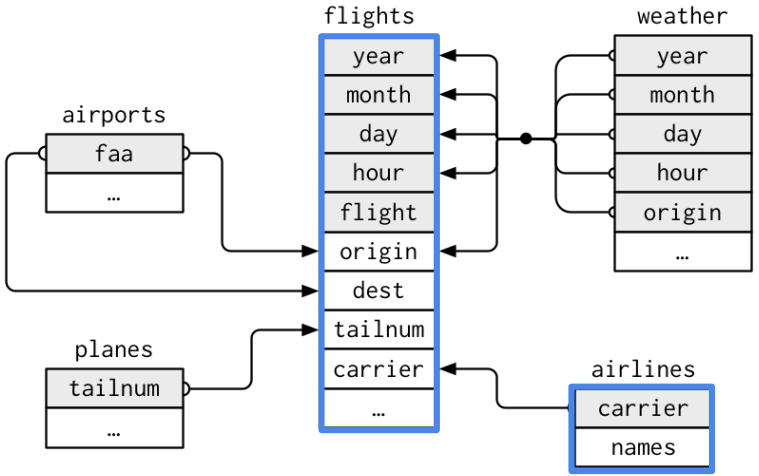
Result

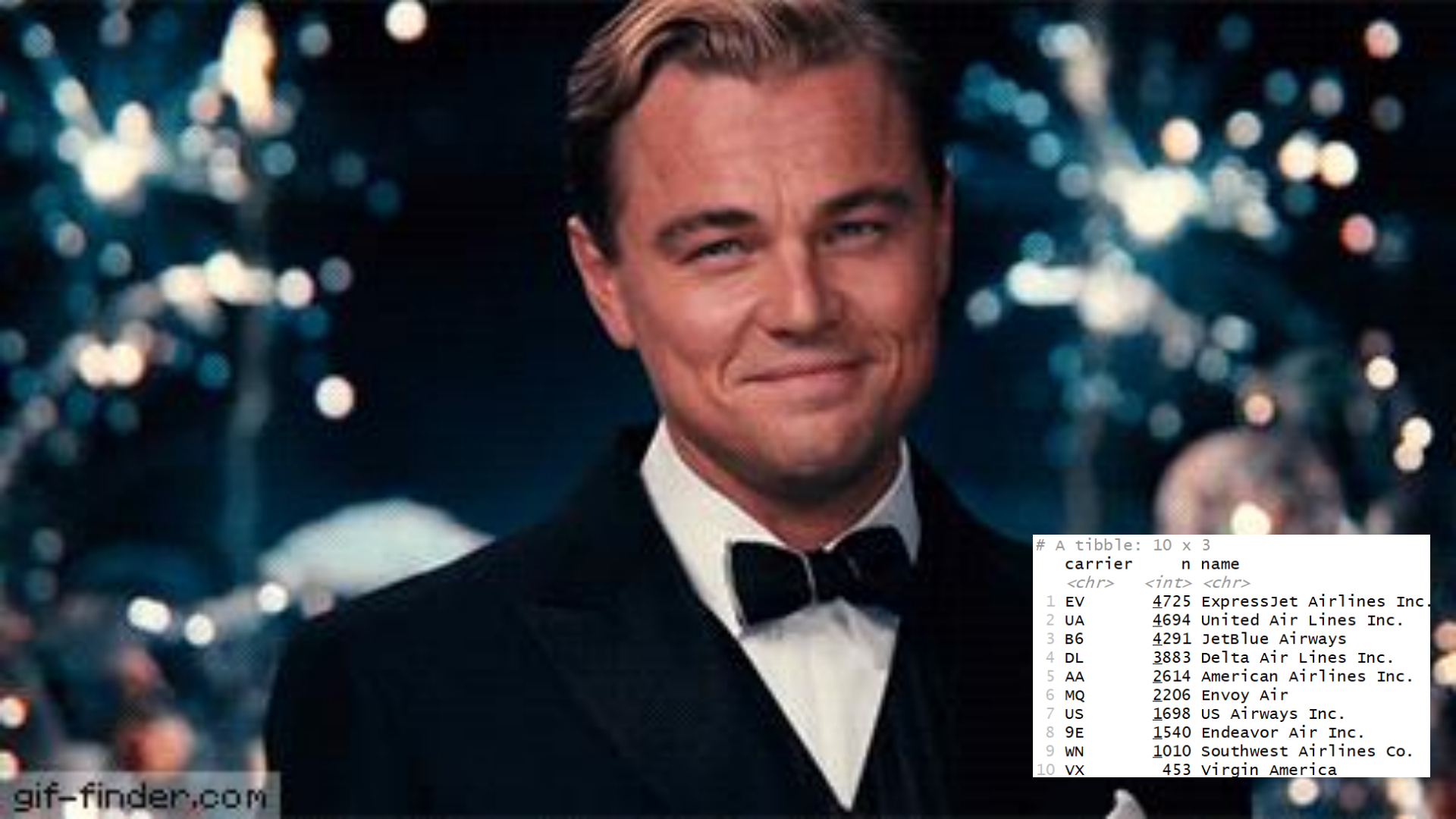
ID	Name	Age
1	Toy	29
2	Hello	30
3	World	*NULL*
4	SQL	18
5	Awesome	*NULL*
6	*NULL*	25
7	*NULL*	26

```
> flights %>%
+     filter(month == 9) %>%
+     group_by(carrier) %>%
+     summarise(n = n()) %>%
+     arrange(desc(n)) %>%
+     head(10)
# A tibble: 10 x 2
   carrier     n
   <chr>   <int>
 1 EV       4725
 2 UA       4694
 3 B6       4291
 4 DL       3883
 5 AA       2614
 6 MQ       2206
 7 US       1698
 8 9E       1540
 9 WN       1010
10 VX        453
```

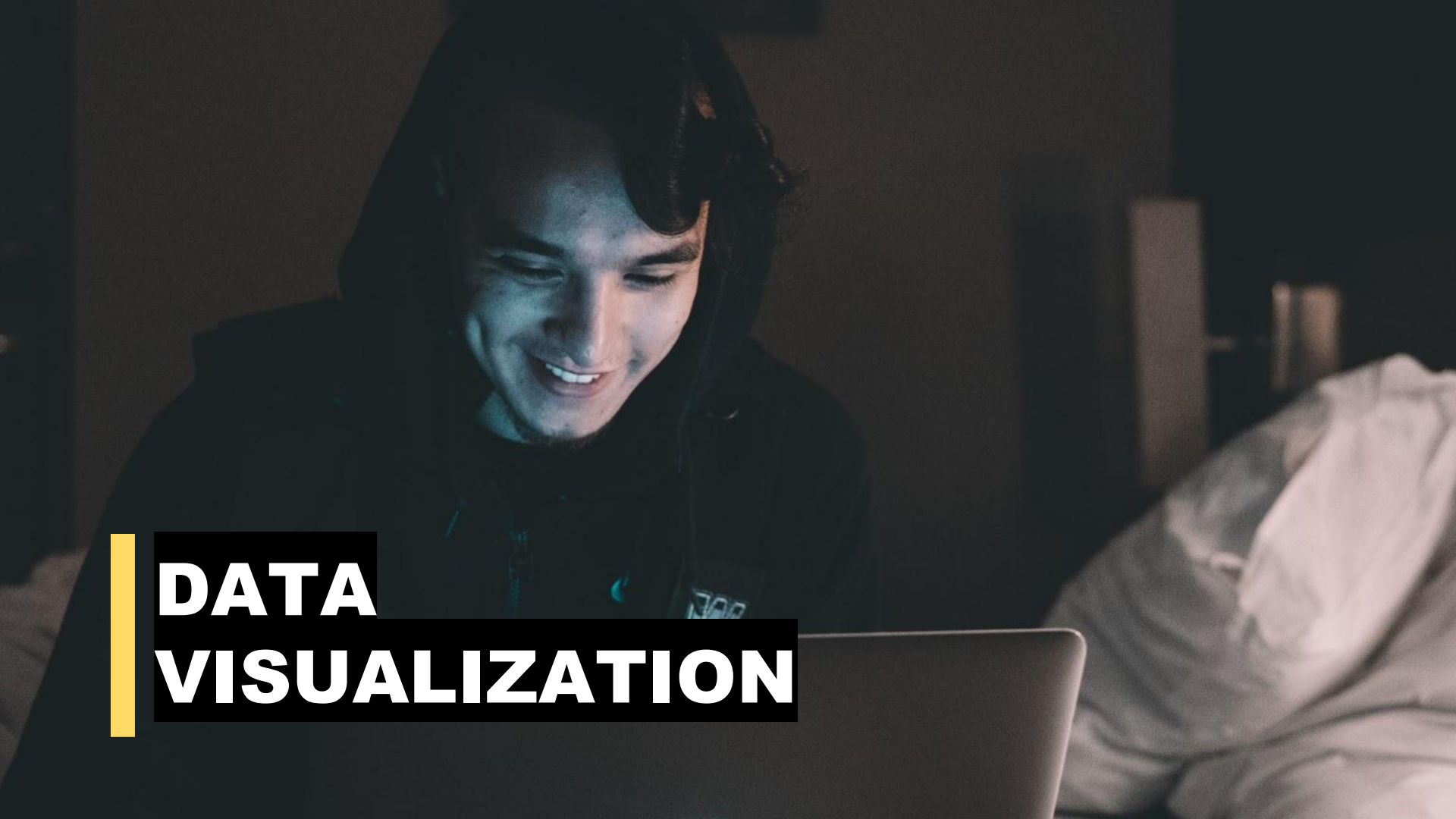
New Column

แสดงชื่อสายการบินเต็มๆ



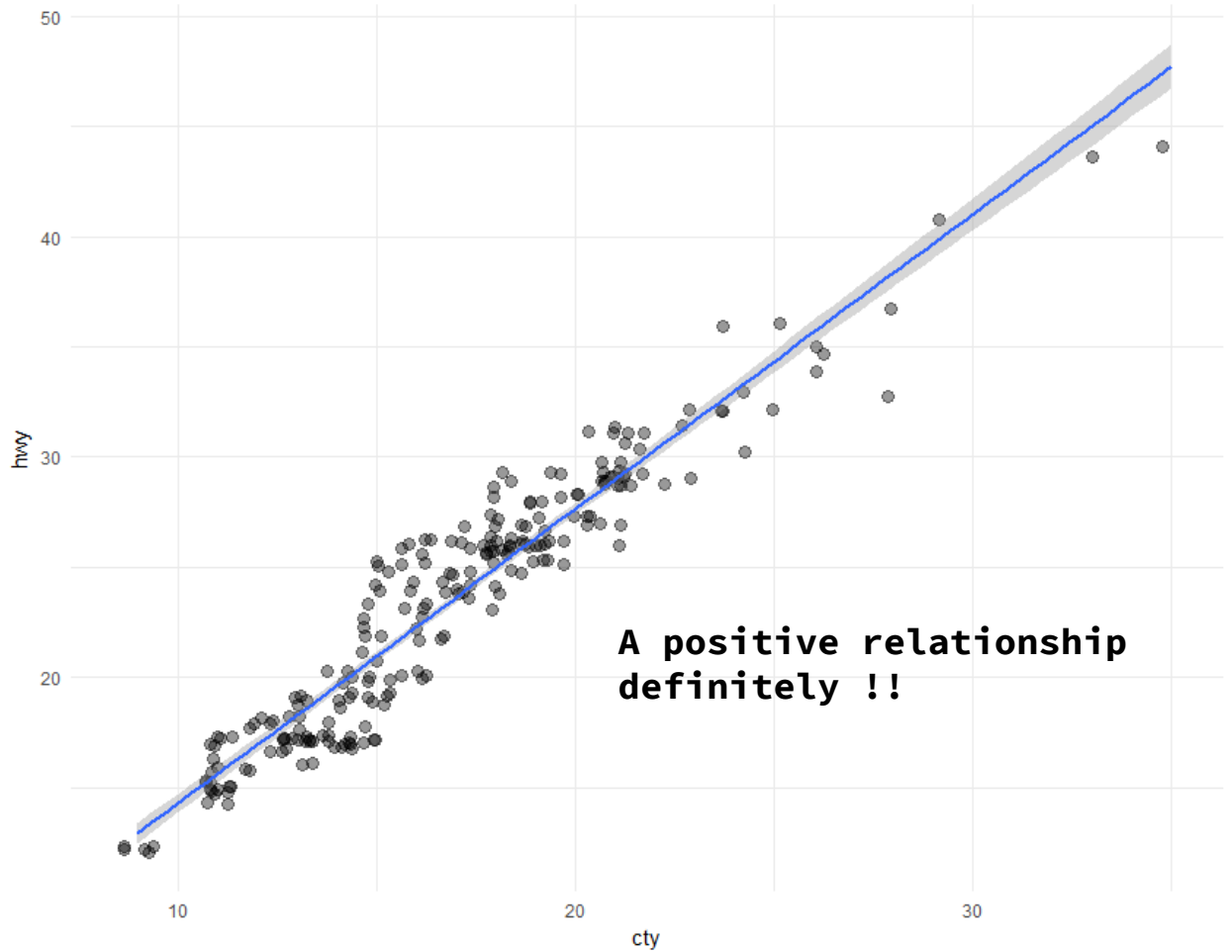


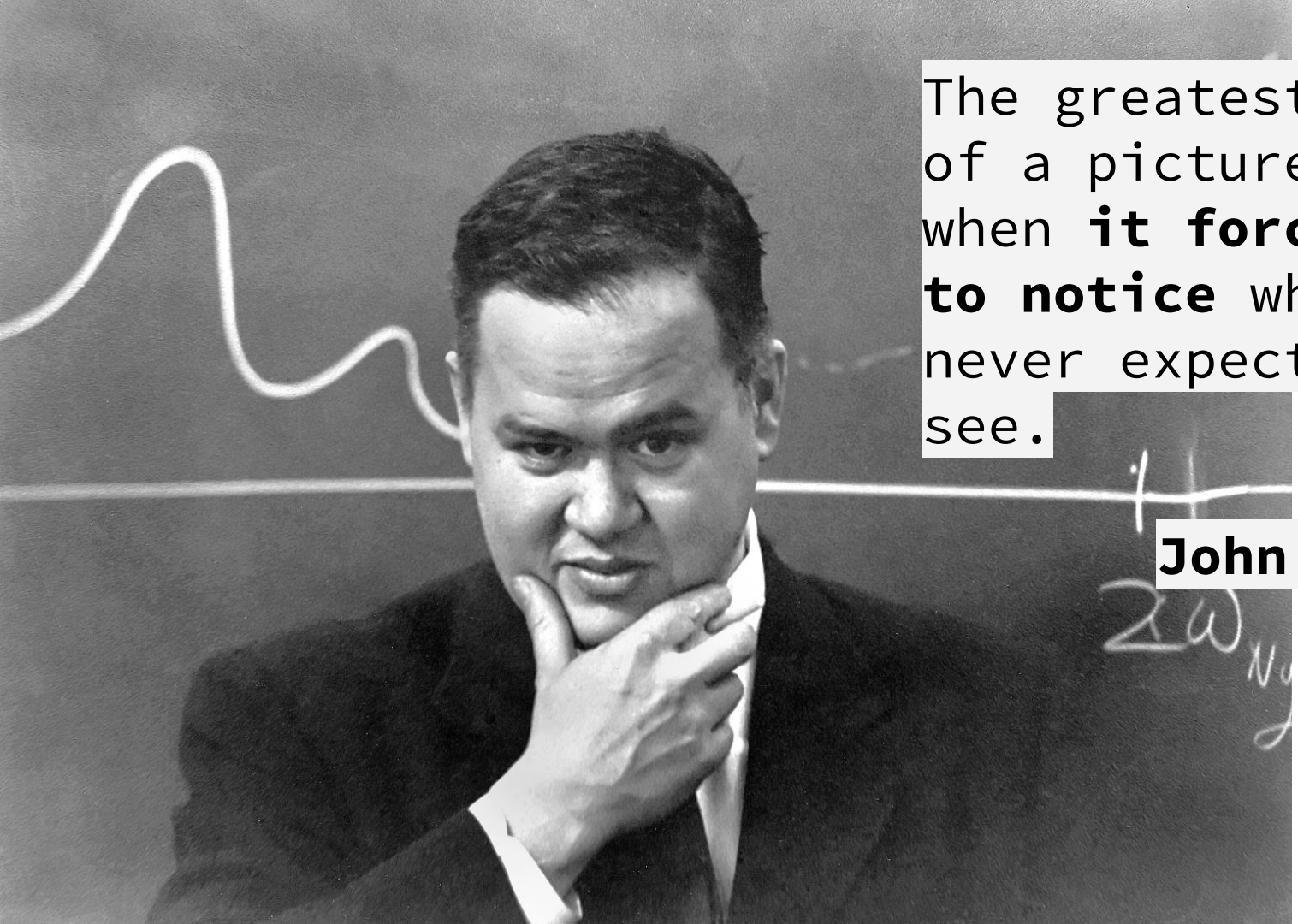

```
# A tibble: 10 x 3
   carrier     n name
   <chr>   <int> <chr>
 1 EV       4725 ExpressJet Airlines Inc.
 2 UA       4694 United Air Lines Inc.
 3 B6       4291 JetBlue Airways
 4 DL       3883 Delta Air Lines Inc.
 5 AA       2614 American Airlines Inc.
 6 MQ       2206 Envoy Air
 7 US       1698 US Airways Inc.
 8 9E       1540 Endeavor Air Inc.
 9 WN       1010 Southwest Airlines Co.
10 VX        453 Virgin America
```


DATA
VISUALIZATION

What's the relationship between cty and hwy?

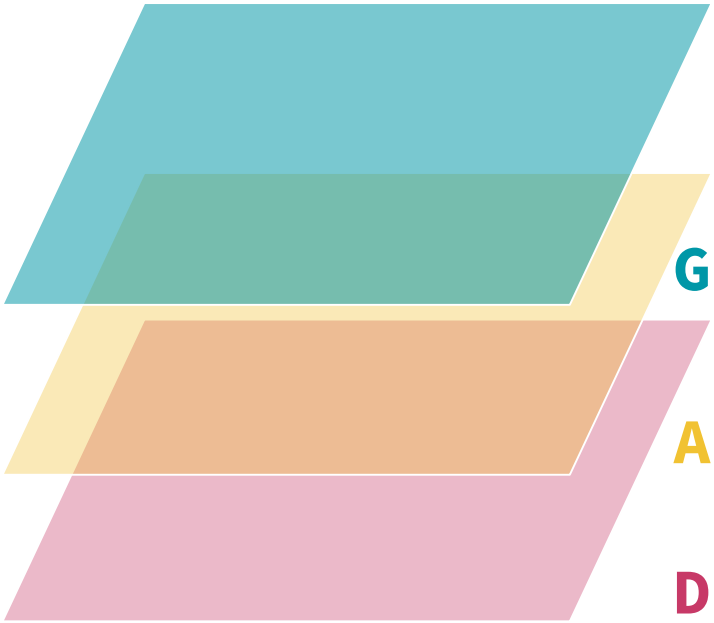
```
> head(mpg, 15)
# A tibble: 15 x 11
   manufacturer model    displ  year   cyl trans    drv     cty   hwy fl
   <chr>        <chr>    <dbl> <int> <int> <chr>    <chr> <int> <int> <cl
 1 audi         a4         1.8  1999     4 auto(l~ f        18    29 p
 2 audi         a4         1.8  1999     4 manual~ f        21    29 p
 3 audi         a4         2    2008     4 manual~ f        20    31 p
 4 audi         a4         2    2008     4 auto(a~ f        21    30 p
 5 audi         a4         2.8  1999     6 auto(l~ f        16    26 p
 6 audi         a4         2.8  1999     6 manual~ f        18    26 p
 7 audi         a4         3.1  2008     6 auto(a~ f        18    27 p
 8 audi         a4 quat~   1.8  1999     4 manual~ 4        18    26 p
 9 audi         a4 quat~   1.8  1999     4 auto(l~ 4        16    25 p
10 audi         a4 quat~   2    2008     4 manual~ 4        20    28 p
11 audi         a4 quat~   2    2008     4 auto(s~ 4        19    27 p
12 audi         a4 quat~   2.8  1999     6 auto(l~ 4        15    25 p
13 audi         a4 quat~   2.8  1999     6 manual~ 4        17    25 p
14 audi         a4 quat~   3.1  2008     6 auto(s~ 4        17    25 p
15 audi         a4 quat~   3.1  2008     6 manual~ 4        15    25 p
# ... with 1 more variable: class <chr>
```

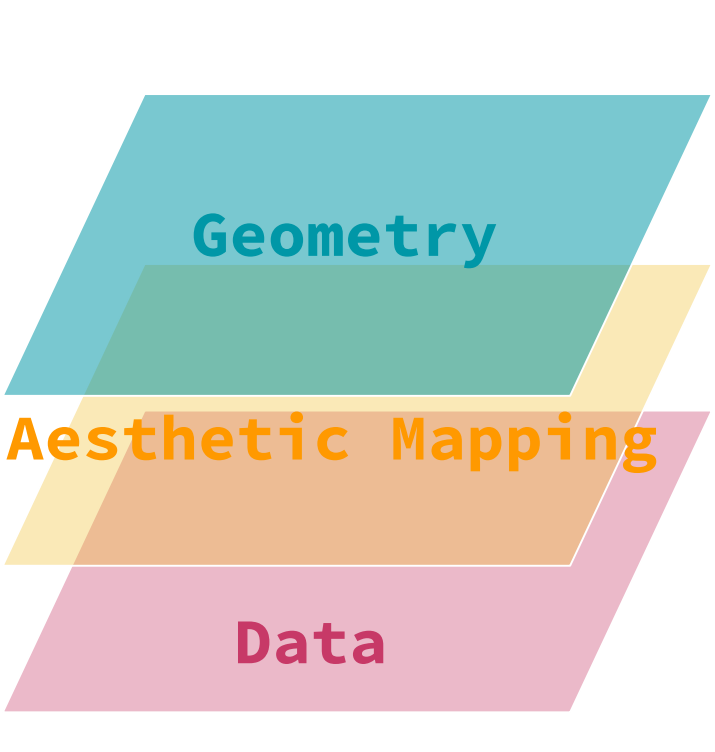

50
40
30
20
hwy
10
20
30
cty
A positive relationship
definitely !!



The greatest value of a picture is when **it forces us to notice** what we never expected to see.

John Tukey


Geometry
Aesthetic Mapping
Data




```
install.packages("ggplot2")
library(ggplot2)

ggplot(data = mpg,
       mapping = aes(cty, hwy)) +
  geom_point()
```

```
ggplot(data = mpg,
       mapping = aes(cty, hwy)) +
    geom_point()
```

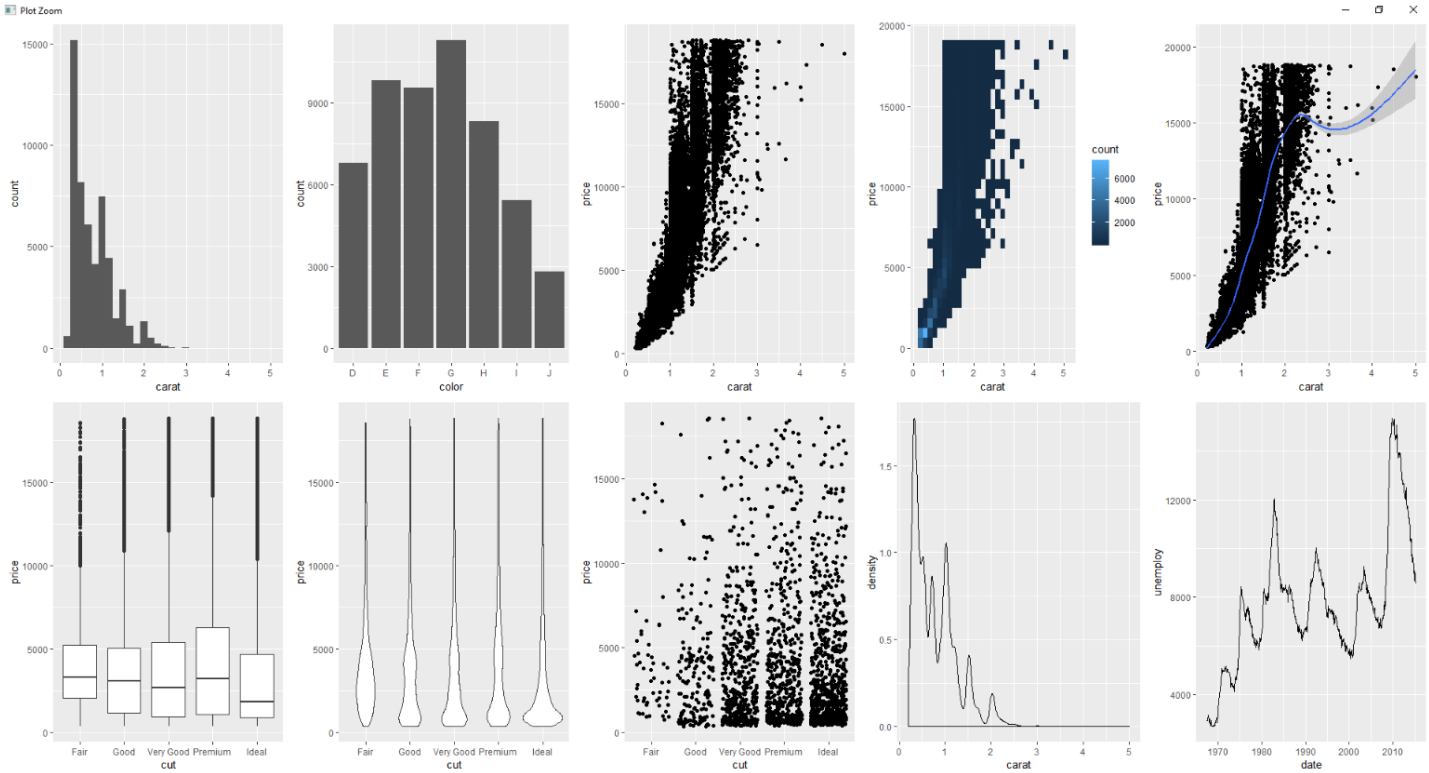
A Few Basic Plots

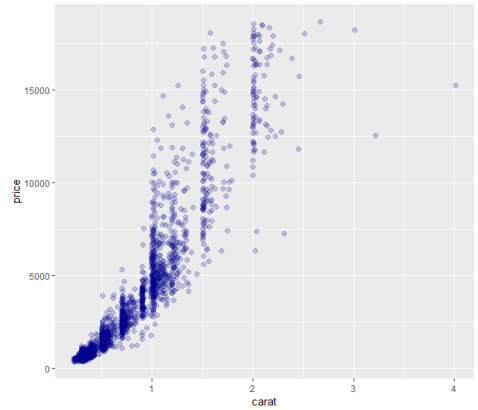
- Histogram
- Bar Plot
- Scatter Plot
- Box Plot
- Line Plot
- Jitter
- Violin
- Bin2d
- Density
- Smoother

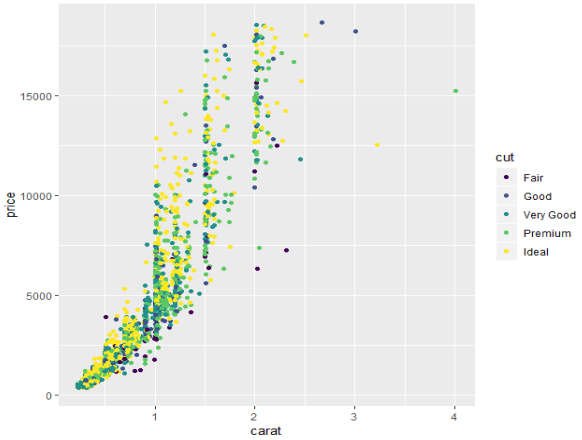
Common Graphs

```
install.packages("gridExtra")
library(gridExtra)
library(ggplot2)

glimpse(diamonds)

# histogram
p1 <- ggplot(diamonds, aes(carat)) +
                    geom_histogram()

# bar plot
p2 <- ggplot(diamonds, aes(color)) +
    geom_bar()

# point
p3 <- ggplot(diamonds, aes(carat, price)) +
    geom_point()

# bin2d
p4 <- ggplot(diamonds, aes(carat, price)) +
    geom_bin2d()

# point + smooth
p5 <- ggplot(diamonds, aes(carat, price)) +
    geom_point() +
    geom_smooth()

# boxplot
p6 <- ggplot(diamonds, aes(cut, price)) +
    geom_boxplot()

# violin
p7 <- ggplot(diamonds, aes(cut, price)) +
    geom_violin()

# jitter
p8 <- ggplot(sample_n(diamonds,2000),
aes(cut, price)) +
    geom_jitter()

# density
p9 <- ggplot(diamonds, aes(carat)) +
    geom_density()

# line
p10 <- ggplot(economics, aes(date,
unemploy)) +
    geom_line()
```

```
# arrange grid
grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, ncol=5)
```







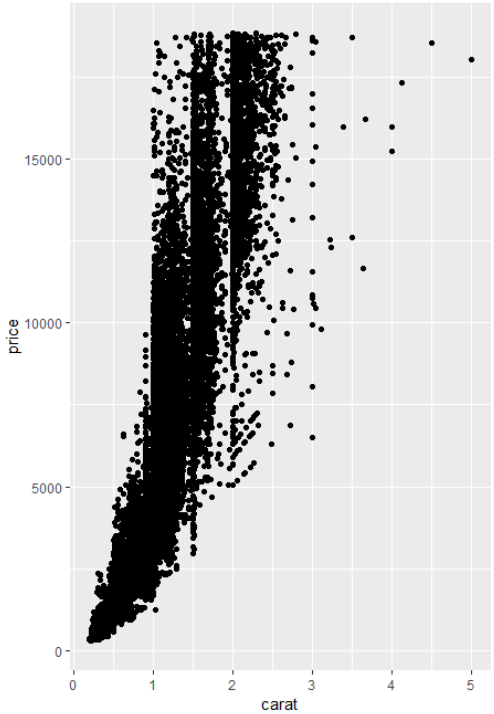
Setting vs. Mapping

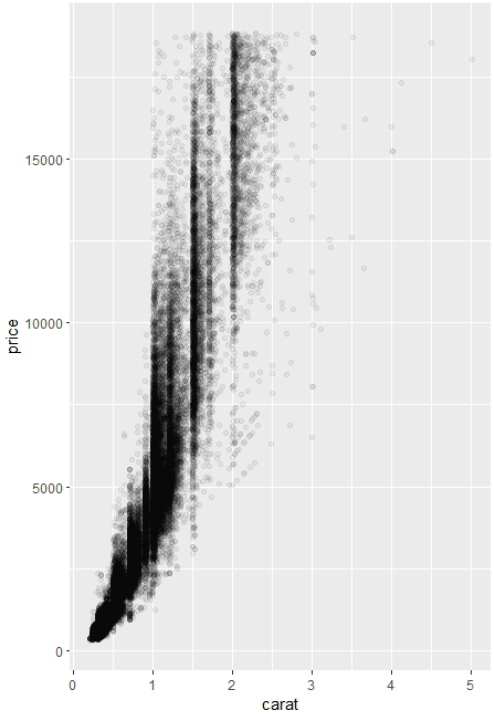
```
ggplot(data = diamonds,
       mapping = aes(carat, price)) +
         geom_point(alpha=1/5, size=2, col="darkblue")
```

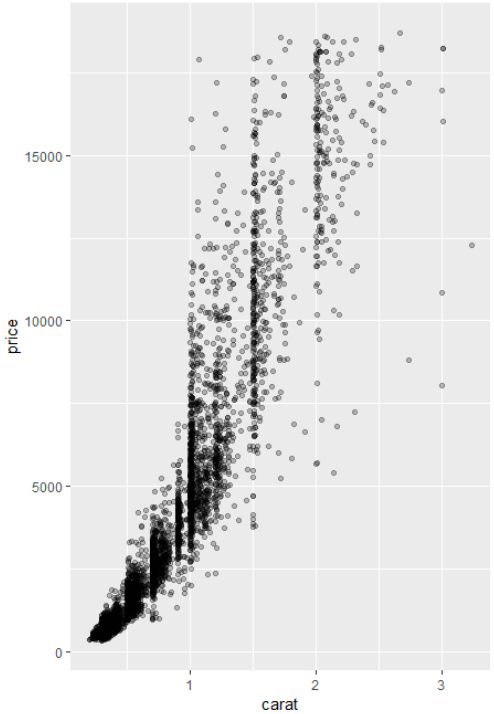
Mapping จะเกิดขึ้นใน aes() เท่านั้นนะคร้าบ

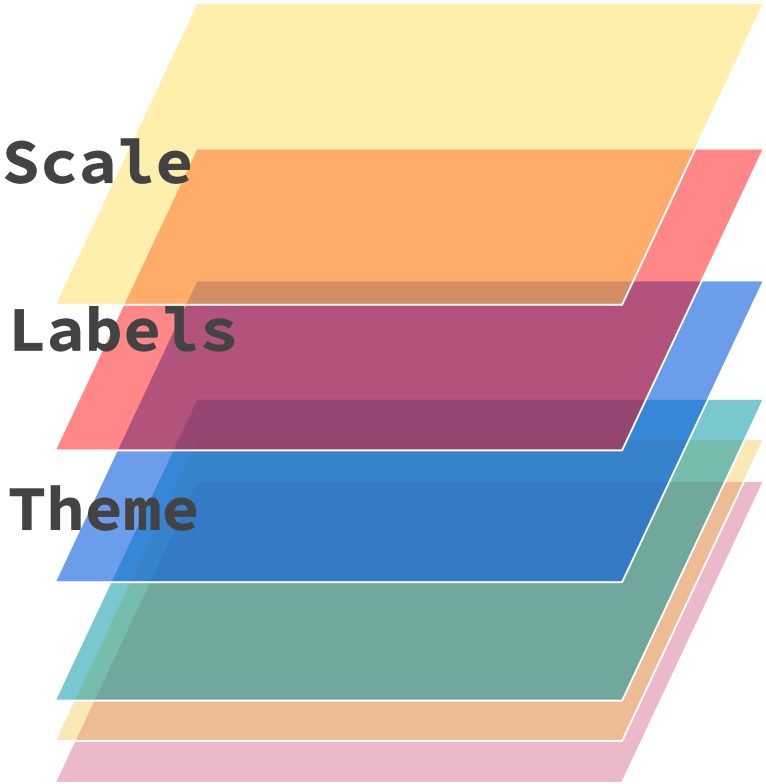
```
ggplot(data = diamonds,
       mapping = aes(carat, price, col=cut)) +
         geom_point()
```

Dealing with **OVERPLOTTING**



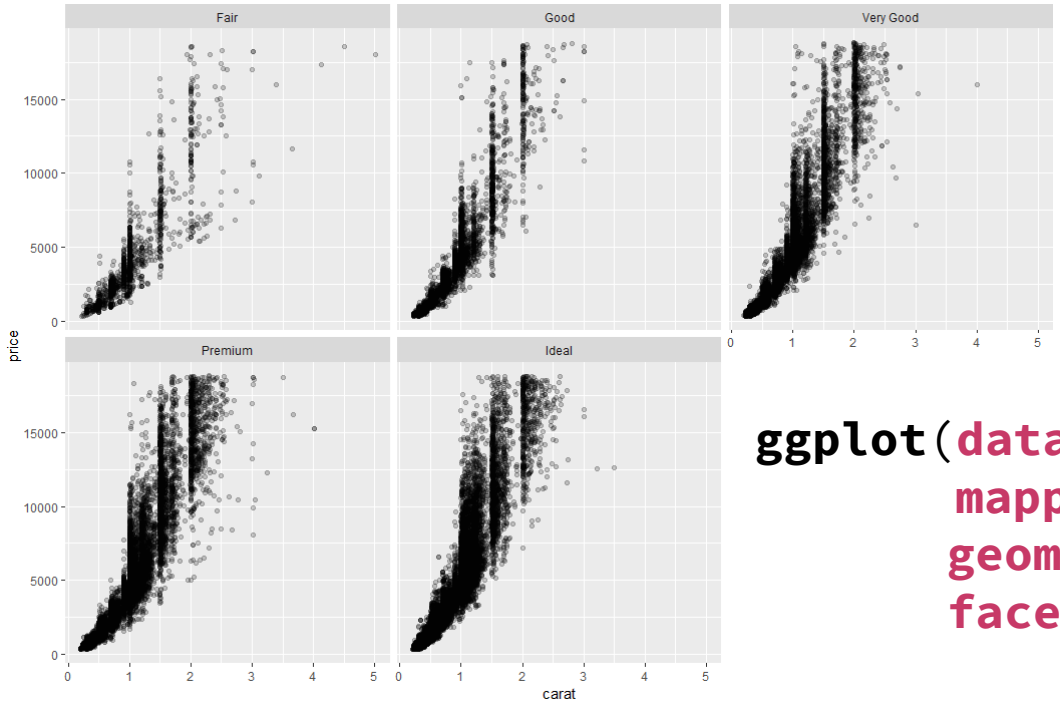




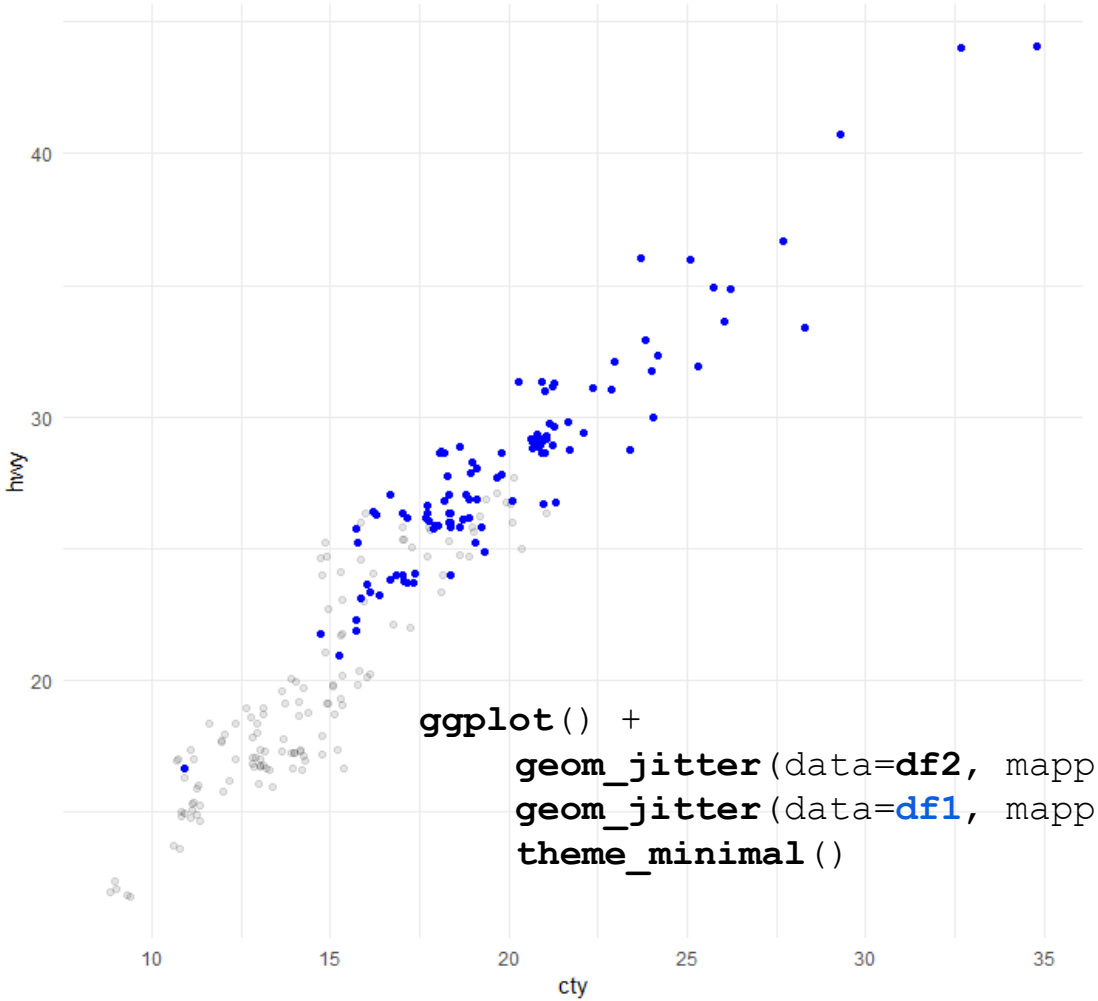



```
ggplot(data = mpg,
       aes(cty, hwy, col=drv)) +
       geom_point() +

       theme_minimal() +

       labs(title = 'Scatter
Plot',
                    x = 'city mpg',
                    y = 'highway mpg') +

    scale_color_manual(values =
c('red', 'gold', 'blue'))
```

facet

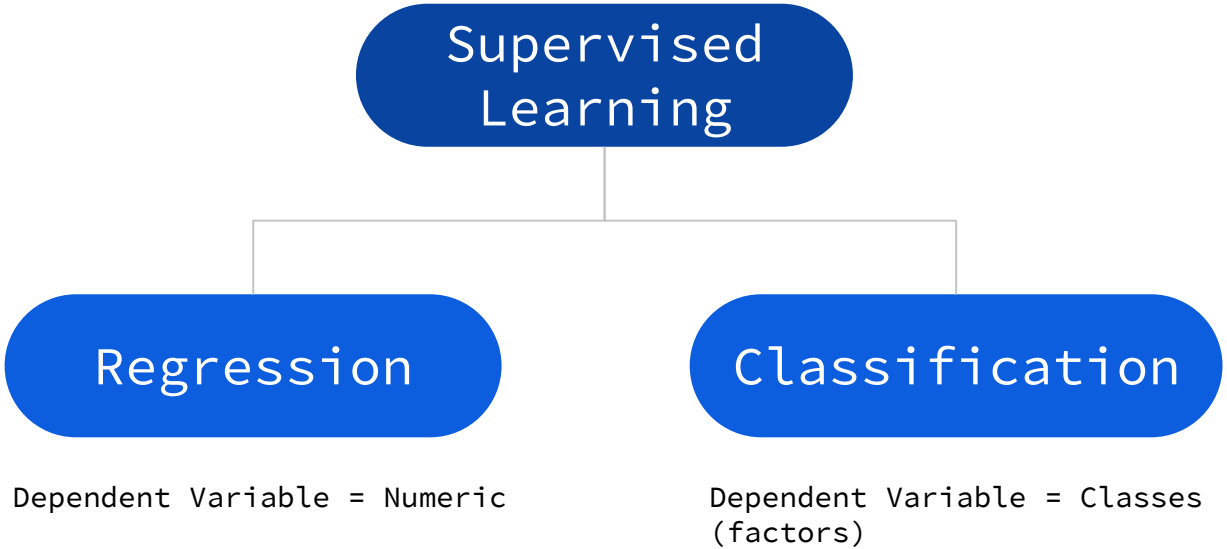


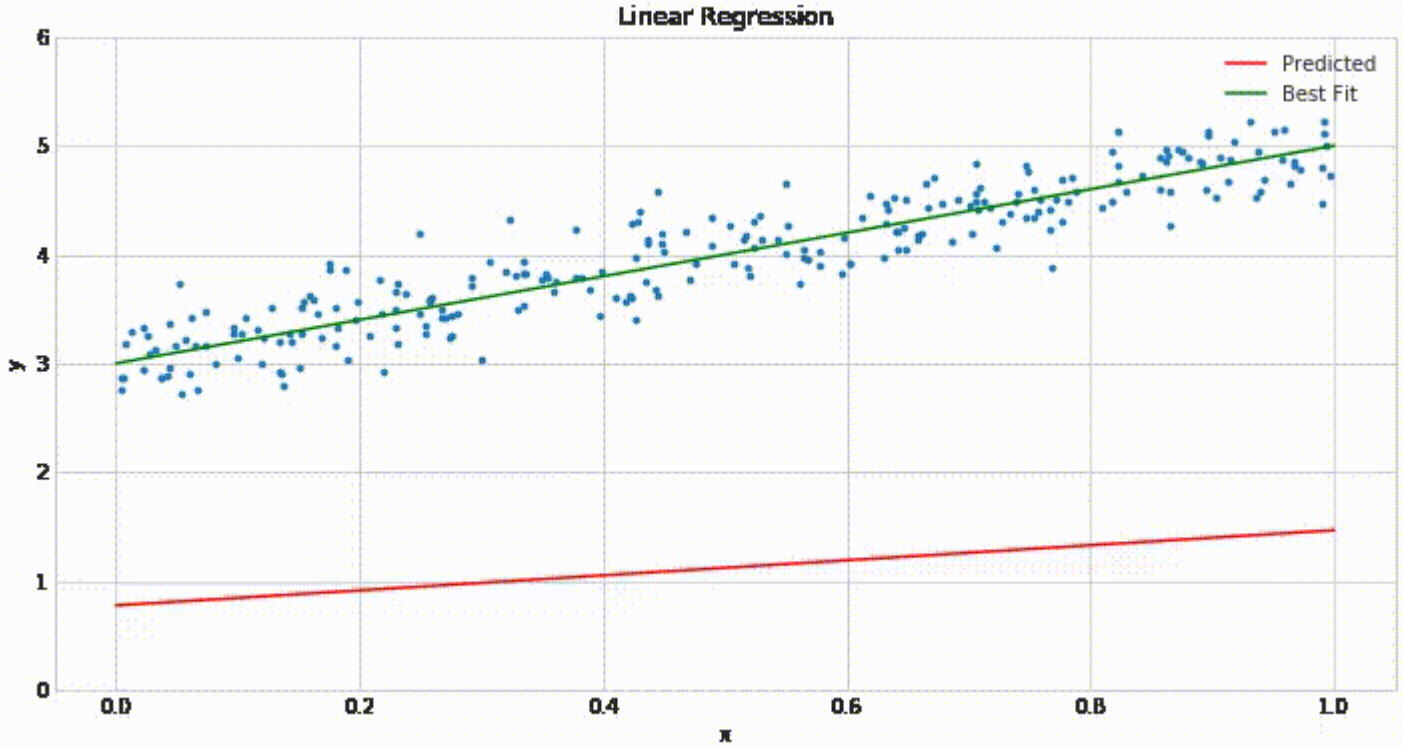

```
ggplot(data = diamonds,
       mapping = aes(carat, price)) +
       geom_point() +
       facet_wrap(~cut)
```



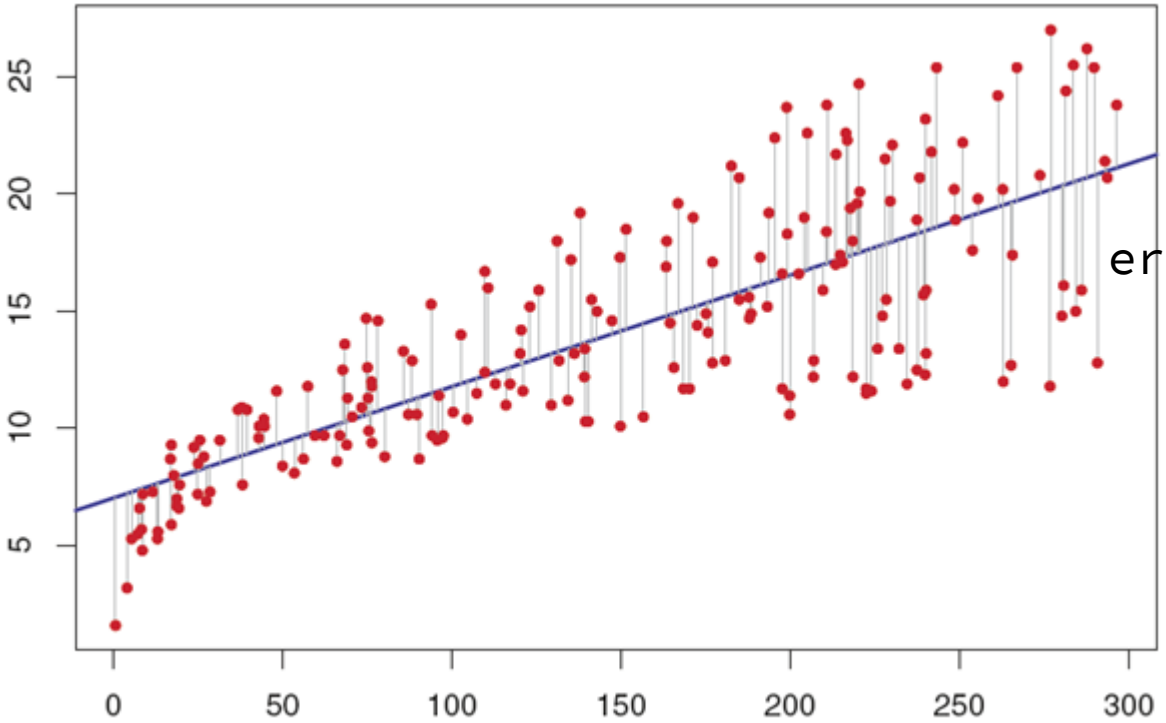

```
ggplot() +
     geom_jitter(data=df2, mapping=aes(cty, hwy), alpha=1/10) +
     geom_jitter(data=df1, mapping=aes(cty,hwy), col='blue') +
     theme_minimal()
```

LINEAR REGRESSION MODELING

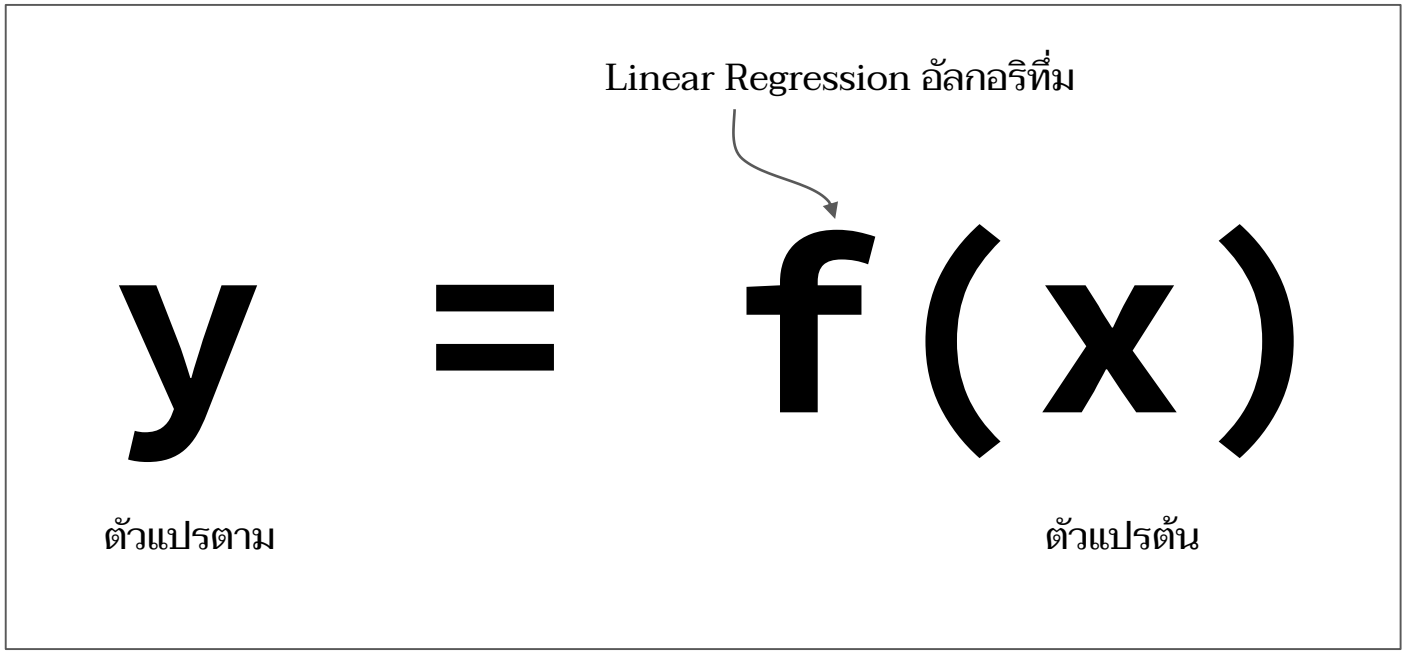

Supervised Learning
Regression
Classification
Dependent Variable = Numeric
Dependent Variable = Classes (factors)


Linear Regression
Predicted
Best Fit
y
x
0
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
1.0

Linear Regression aims to minimize sum(all_errors)



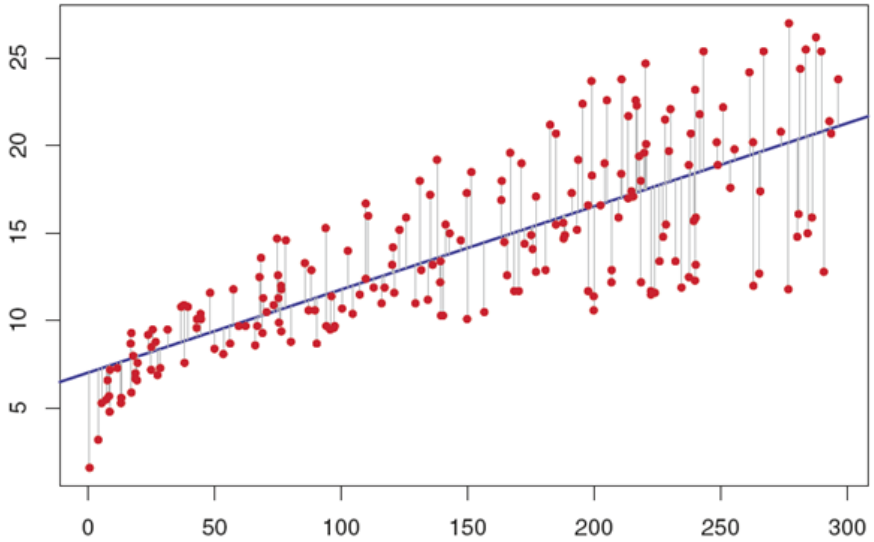
Model 101



One independent variable

$$y = b0 + b1.x1$$

จุดตัดแกนตั้ง ความชัน



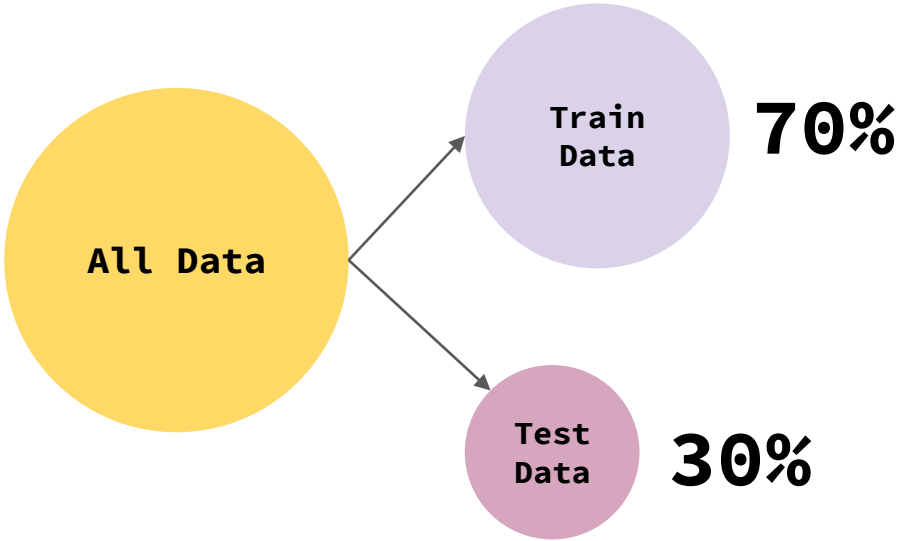
More than one independent variables

$$y = b0 + b1.x1 + b2.x2 + b3.x3 + ...$$

Let’s build your first model in R

Steps to Build (any) a Model

1. Split Data
2. Train Model
3. Test Model



We build model that can be used in **the future**.

GENERALIZATION

Practice
predicting house price in Boston

Price = f(x …)

Build Linear Regression in R

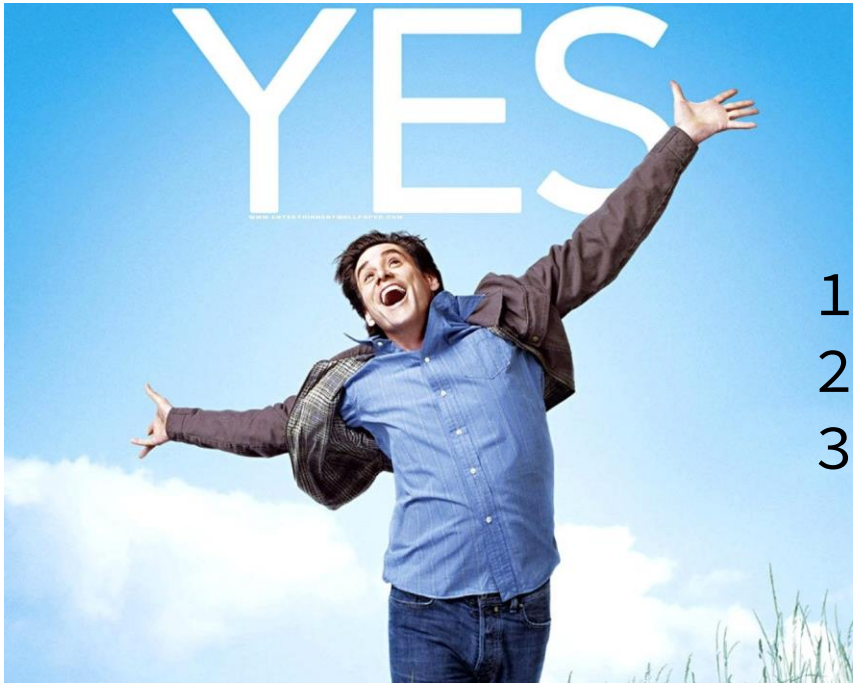
```
# install MASS
install.packages("MASS")
library(MASS)

# dataset
glimpse(MASS::Boston)

# [1] split data
set.seed(123)
index <- sample(1:nrow(Boston), 0.7*nrow(Boston),
                replace = FALSE)
train_data <- Boston[index, ]
test_data <- Boston[-index, ]
```

Build Linear Regression in R

```
# [2] train model
reg_model <- lm(medv ~ crim + rad + black + tax,
                data = train_data)

# [3] test model
predictions <- predict(reg_model, test_data)


# compute RMSE (Root Mean Square Error)
sqrt(mean((test_data$medv - predictions)**2))
```

Can we improve the model?

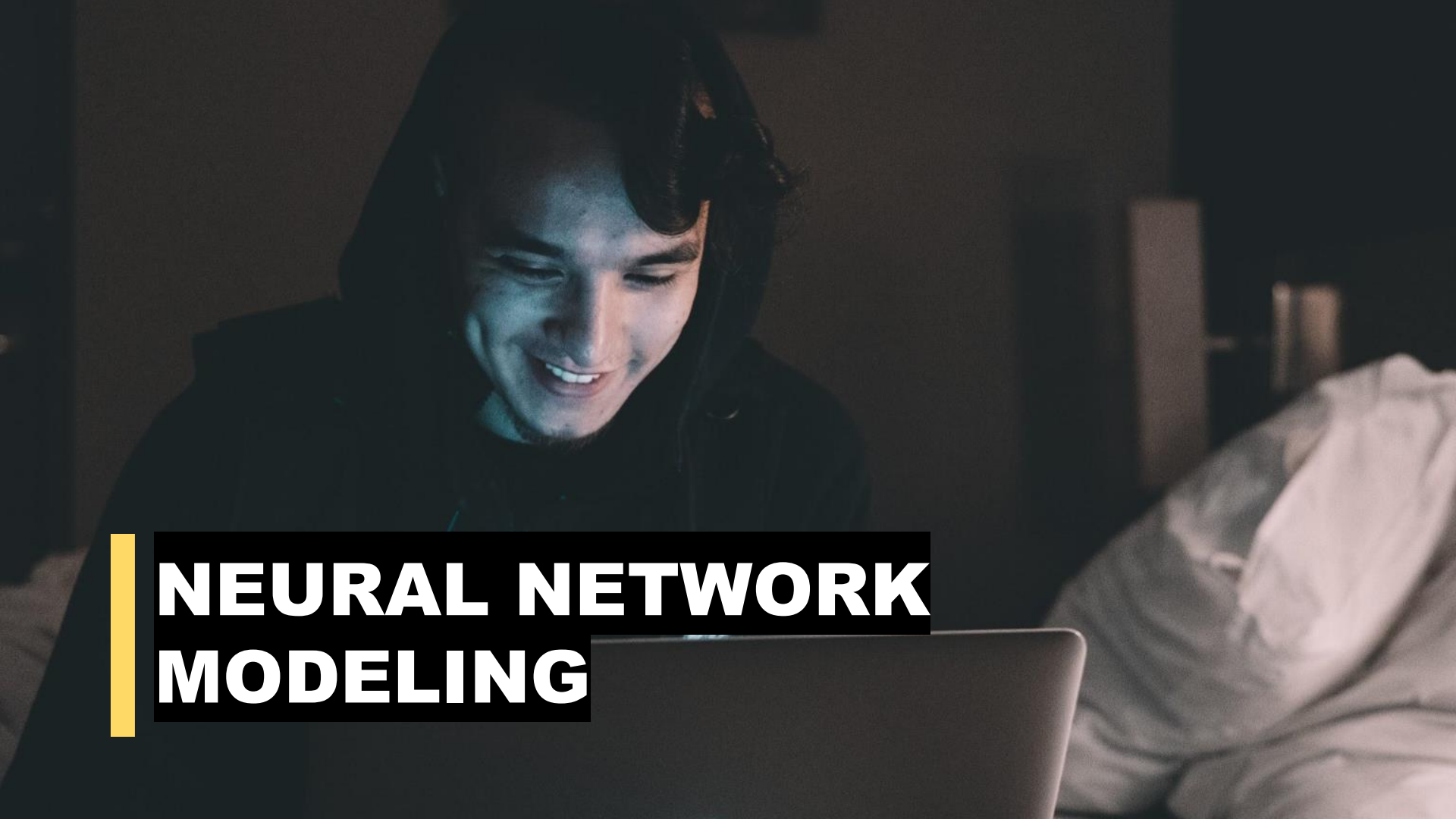
i.e. lower the RMSE value



1. Collect More Data
2. Add more X’s variables
3. Try other algorithms

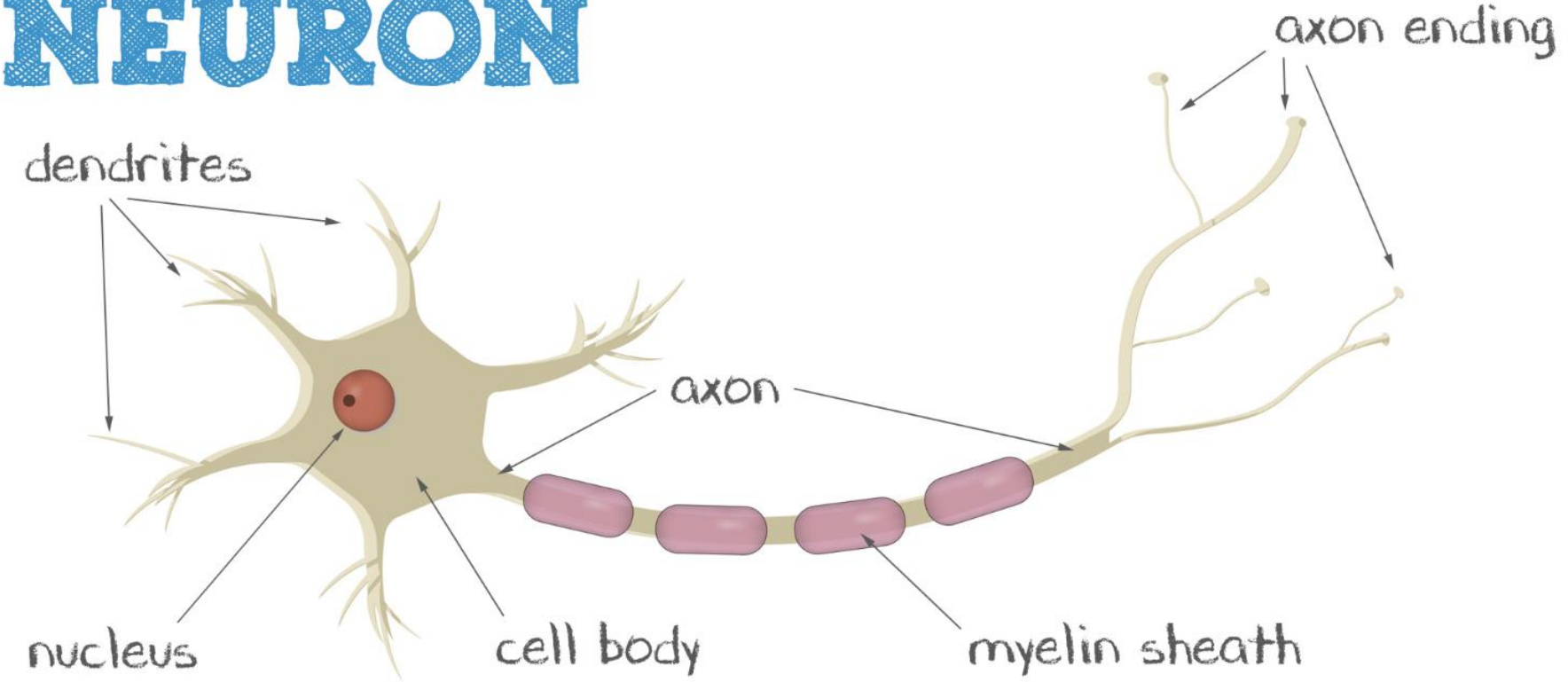
Try This !!


```
reg_model <- lm(medv ~ .,
                data = train_data)
```

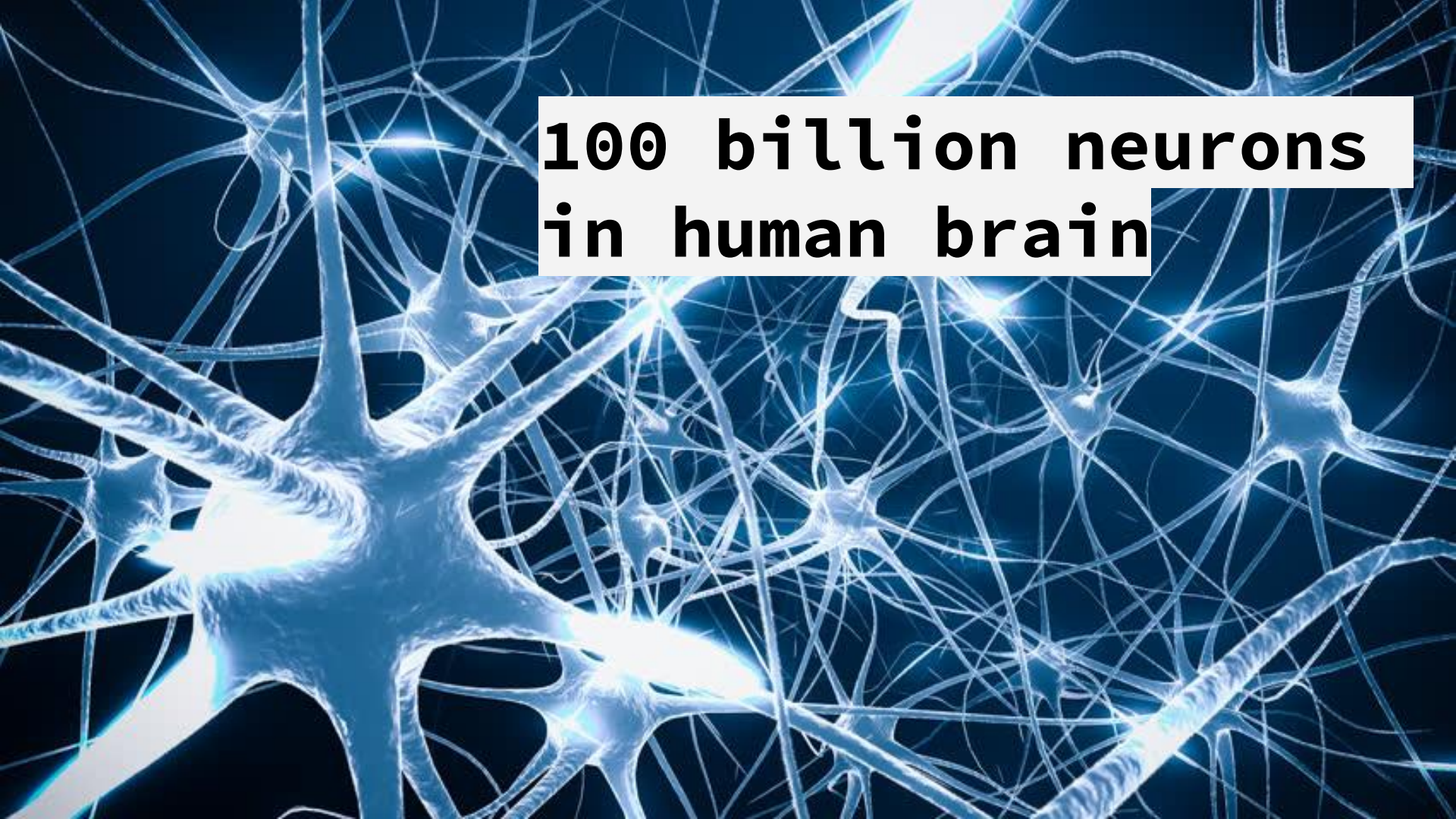

NEURAL NETWORK
MODELING

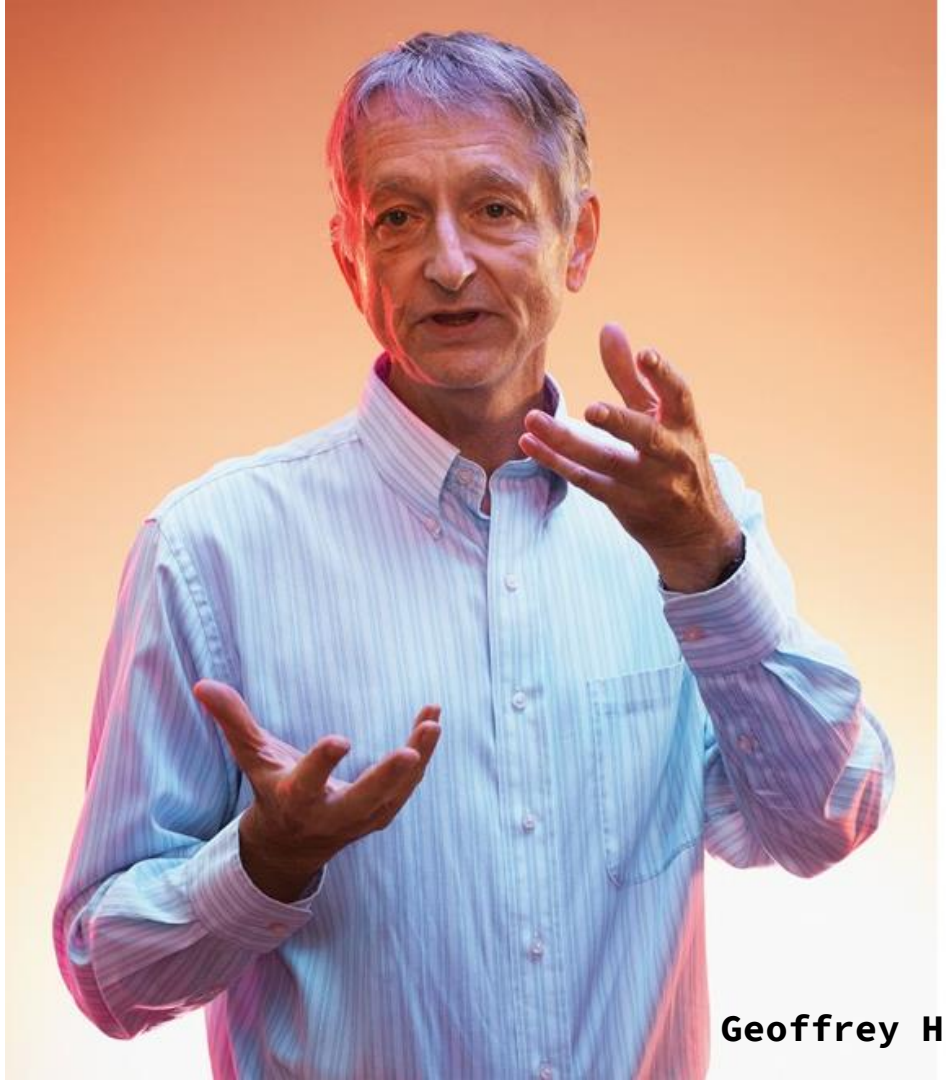

How our brain learn?

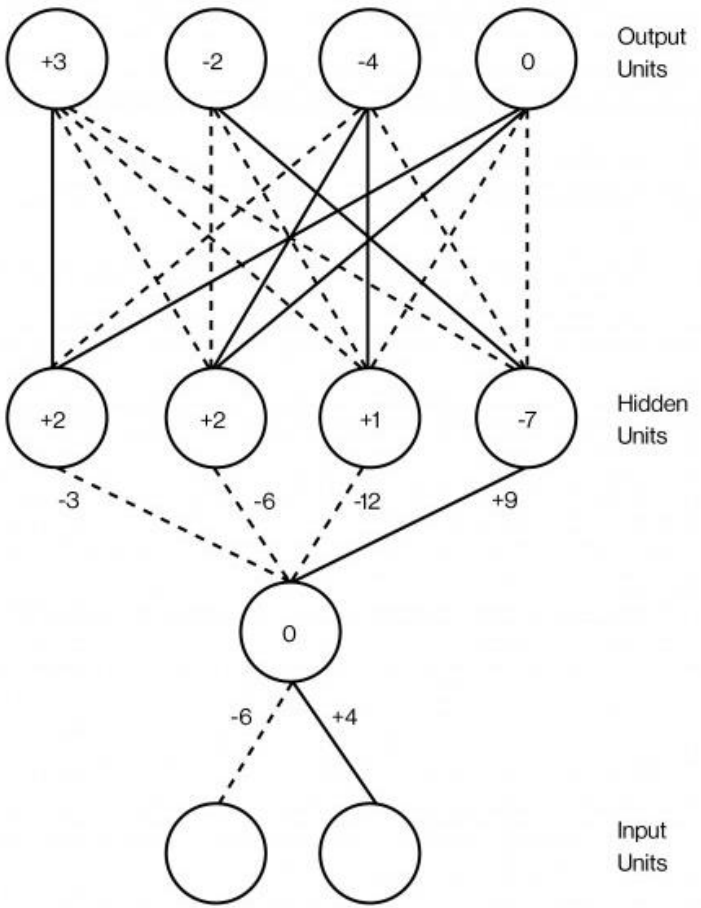
NEURON



http://gooeybrains.com/wp-content/uploads/2016/08/Neuron-image.jpg


100 billion neurons
in human brain

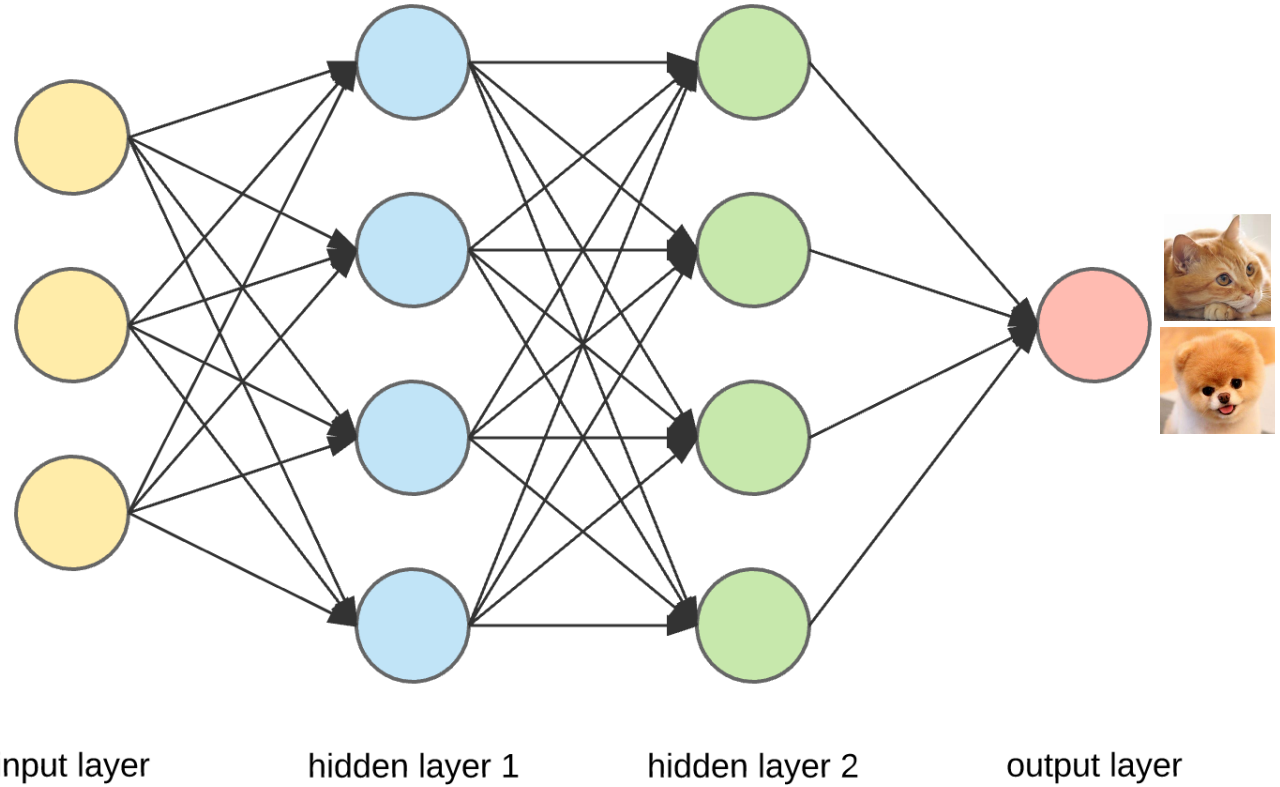




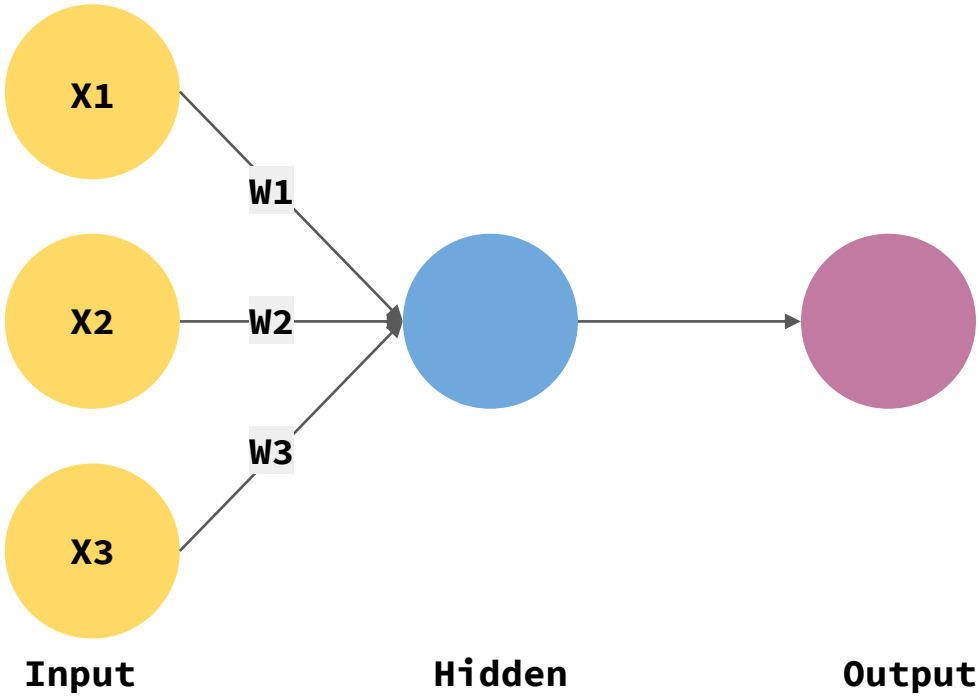
Geoffrey Hinton

CAT OR DOG?

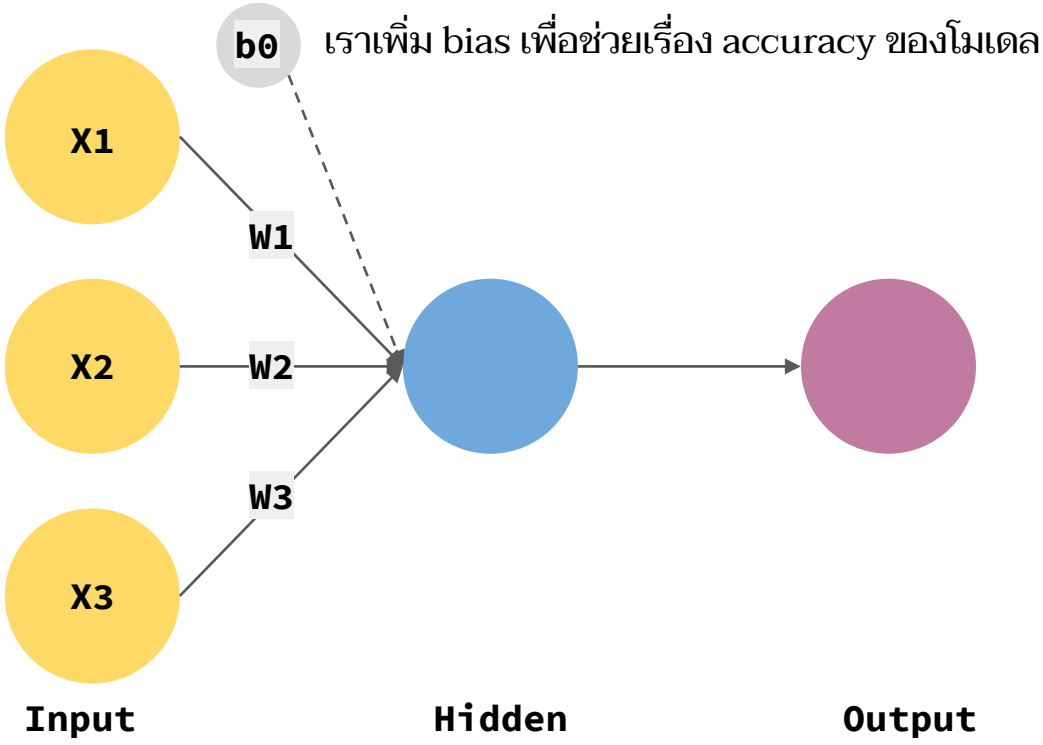
Neural Network Architecture



How Neuron Works (Perceptron)



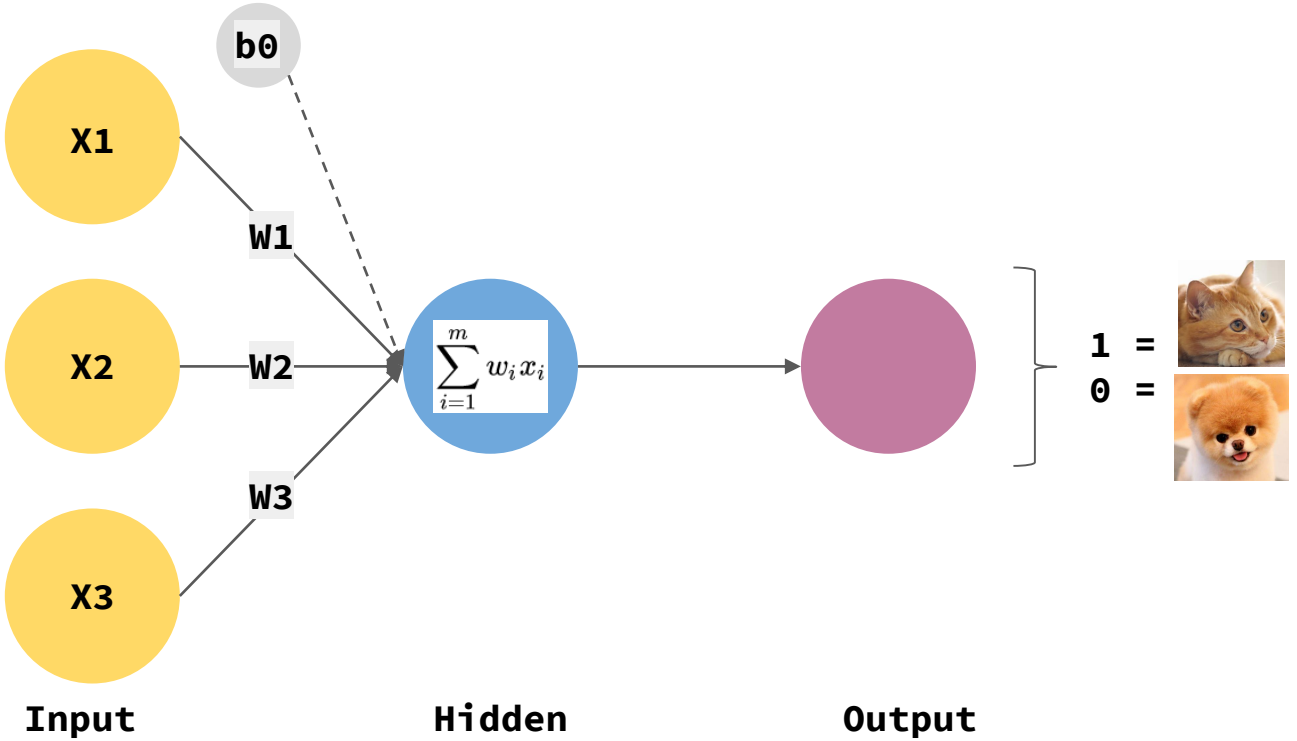
How Neuron Works (Perceptron)



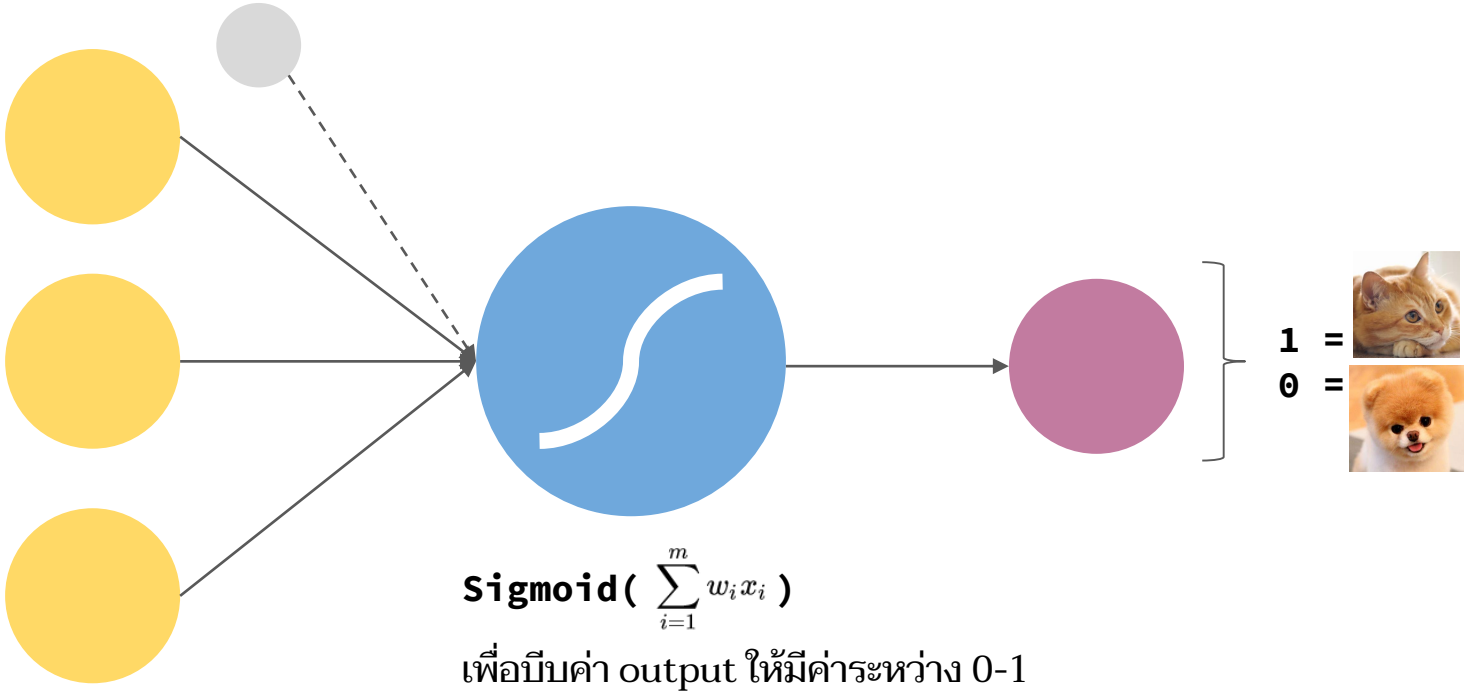
dot_product = b0 + w1.x1 + w2.x2 + w3.x3

dot_product = $\sum_{i=1}^{m} w_i x_i$ m คือจำนวน training sample ทั้งหมดของเรา

How Neuron Works (Perceptron)



Activation Function

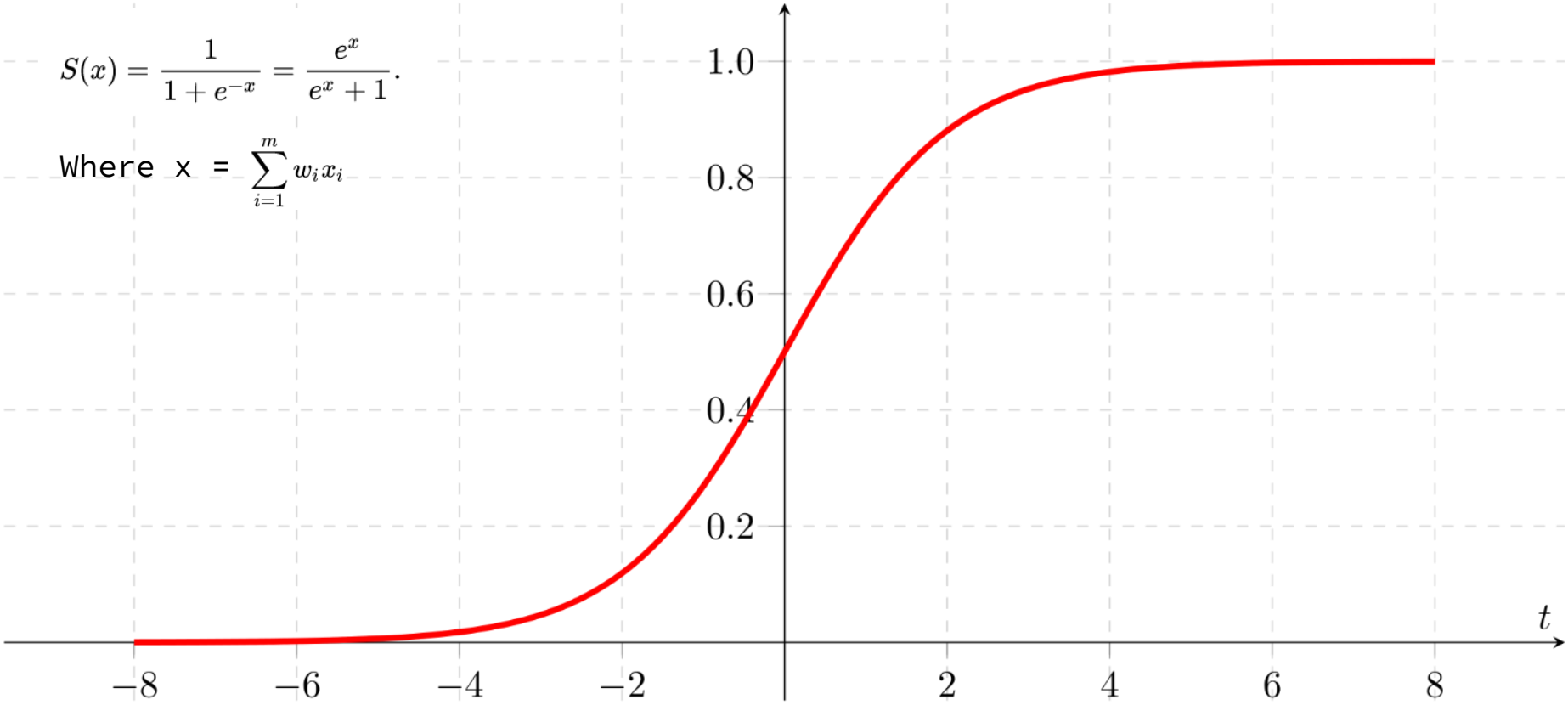


Sigmoid($\sum_{i=1}^{m} w_i x_i$ **)**

เพื่อบีบค่า output ให้มีค่าระหว่าง 0-1

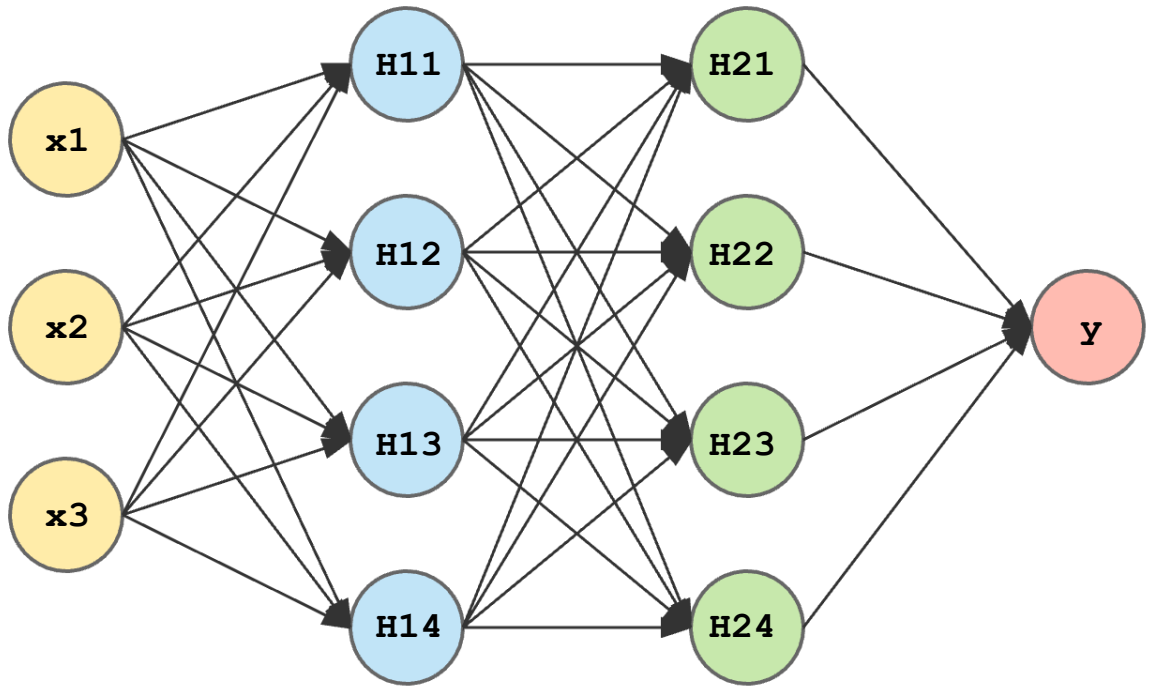
$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$

Where x = $\sum_{i=1}^{m} w_i x_i$

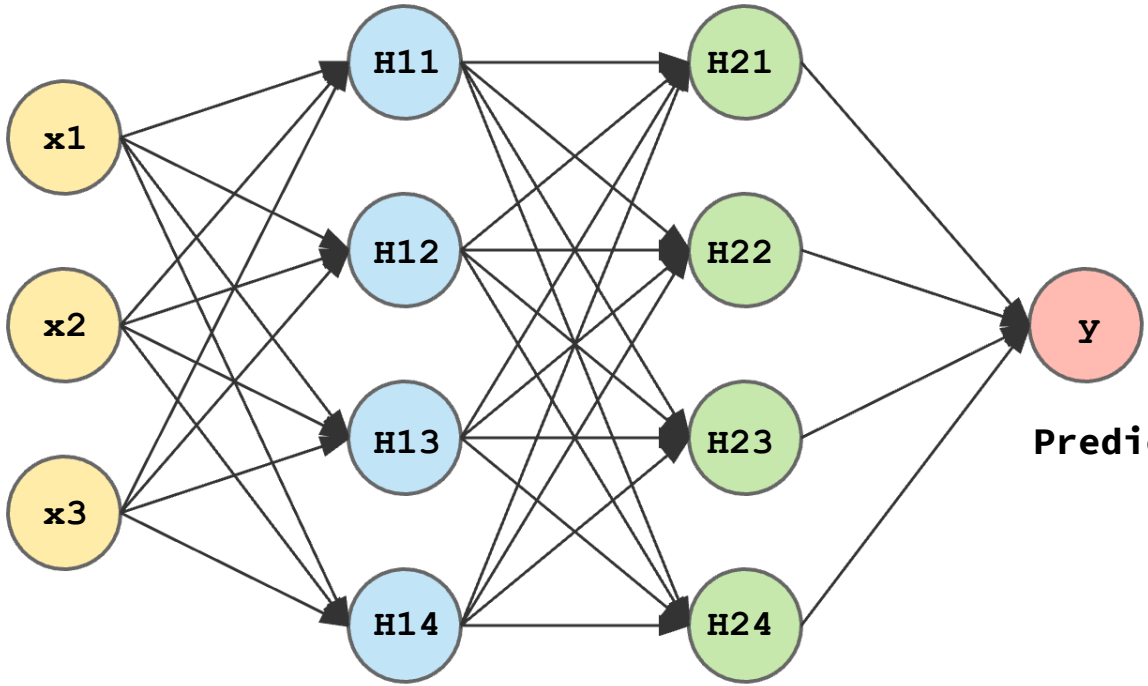


https://en.wikipedia.org/wiki/Sigmoid_function

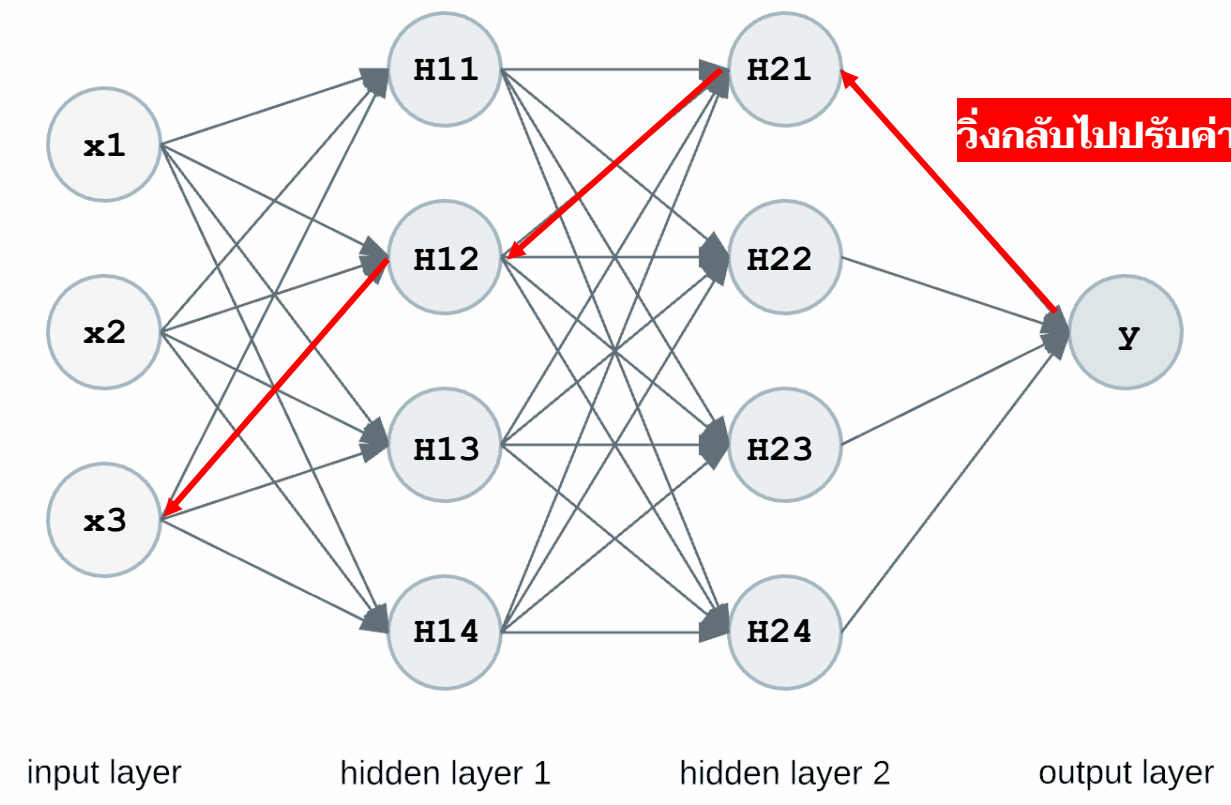
Forward Pass



Forward Pass

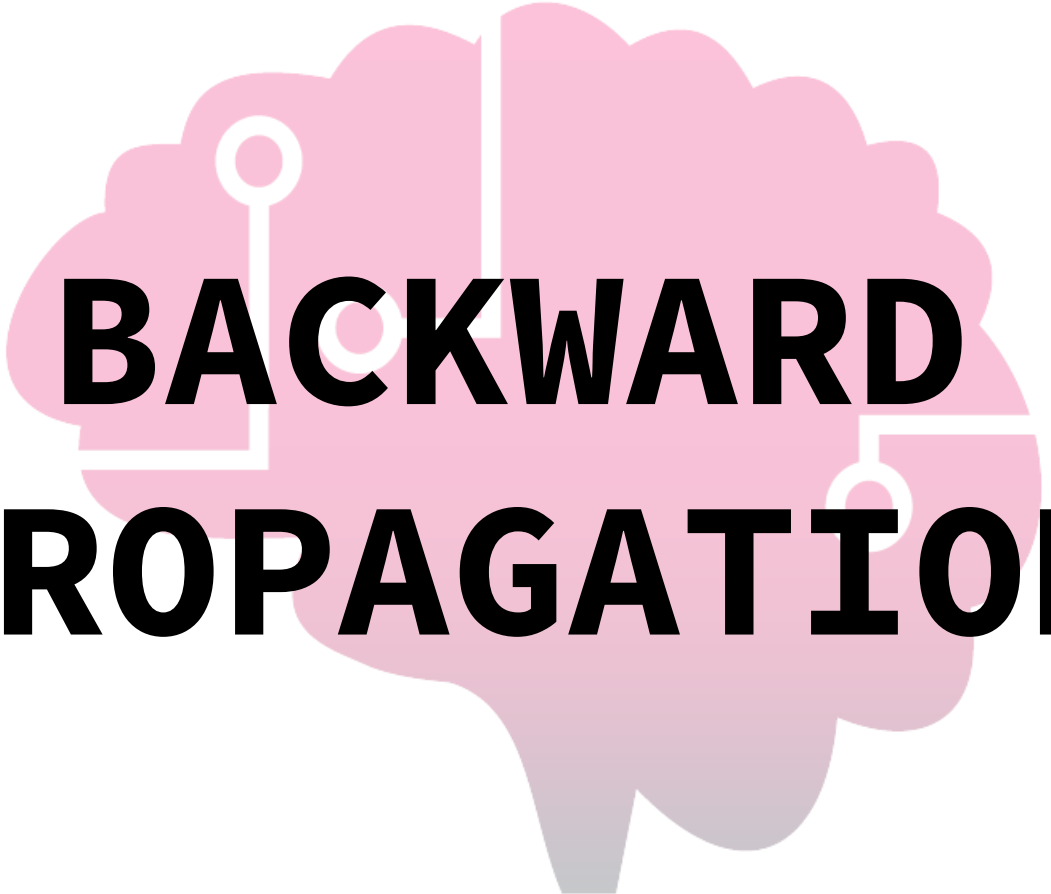


Backward Pass



Steps to build Neural Network

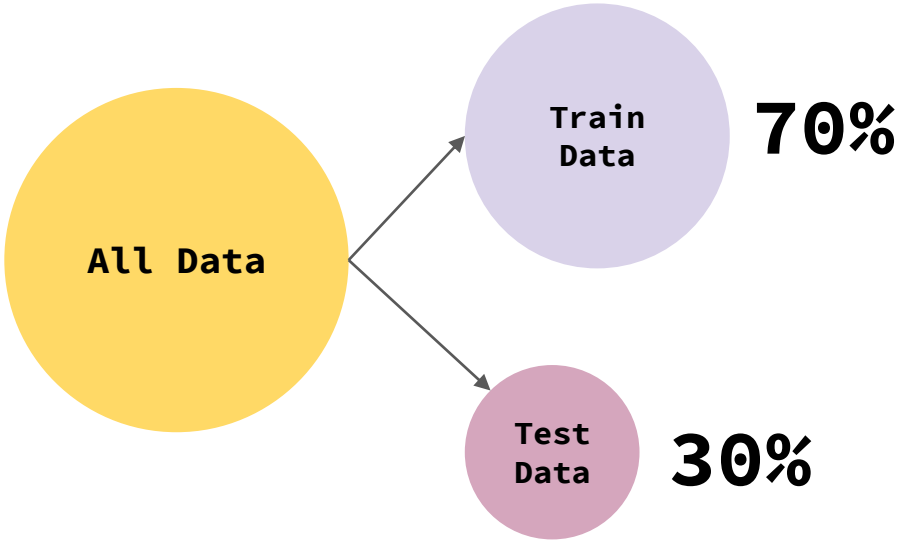
1. Define architecture
2. Forward pass
3. Compare prediction vs. actual y
4. **Backward pass (optimize weights)**
5. Wait and be amazed with results!!



BACKWARD PROPAGATION

Steps to Build (any) a Model

1. Split Data
2. Train Model
3. Test Model



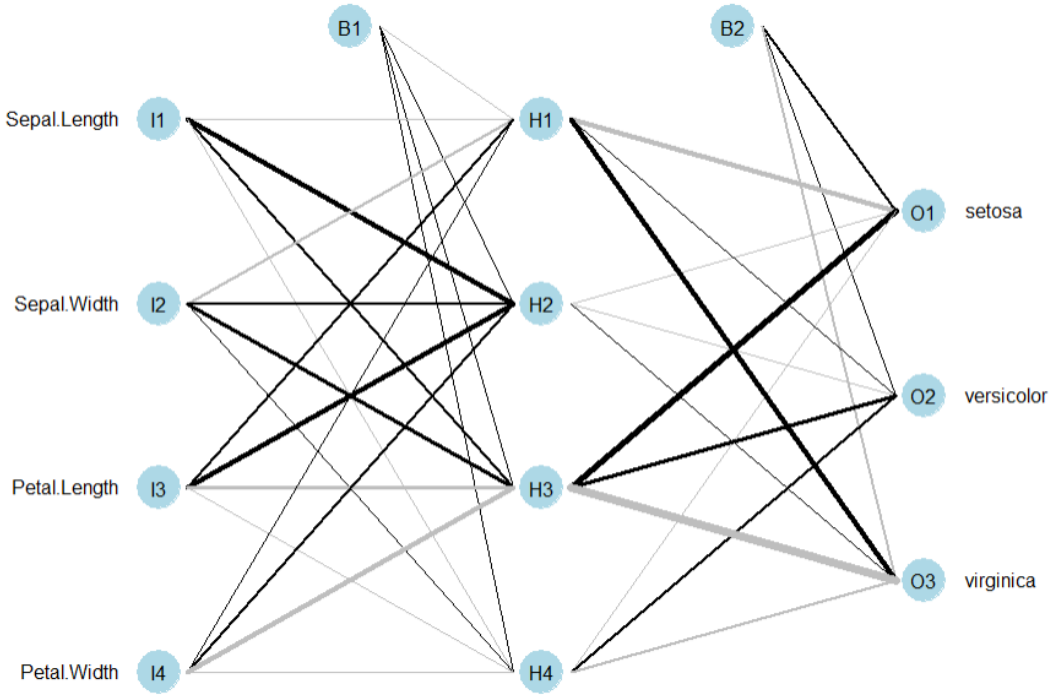
Build NN in R

```
# install nnet
install.packages("nnet")
library(nnet)
library(tidyverse)

# dataset
glimpse(iris)

# [1] split data
set.seed(123)
index <- sample(1:nrow(iris), 0.7*nrow(iris), replace=FALSE)
train_data <- iris[index, ]
test_data <- iris[-index, ]
```

Build NN in R

```
# [2] train model
set.seed(123)
nn_model <- nnet(Species ~ Petal.Length + Petal.Width,
                 data = train_data, size = 4)

summary(nn_model)

# We can plot our nn_model using package NeuralNetTools

install.packages("NeuralNetTools")
library(NeuralNetTools)
plotnet(nn_model)
```

Build NN in R




```
plotnet(nn_model)
```

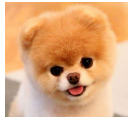
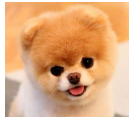
Build NN in R

```
# [3] test model
predictions <- predict(nn_model, test_data, type = “class”)
print(predictions)

confusion_mat <- table(test_data$Species, predictions)
print(confusion_mat)
```

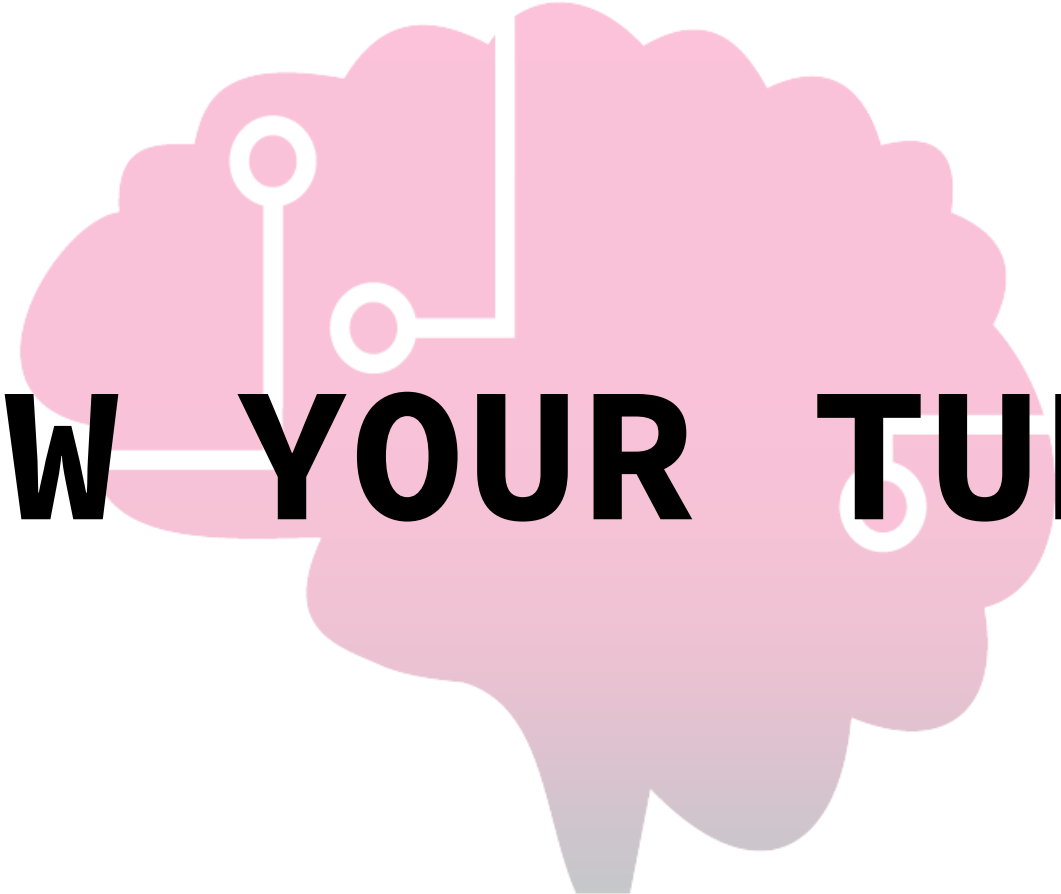
```
           predicted_y
            setosa versicolor virginica
setosa          15          0         0
versicolor       0         11         4
virginica        0          1        14
```

%Accuracy

Confusion Matrix Explained

	Predicted y=0 	Predicted y=1 
Actual y=0 	**CORRECT**	FALSE
Actual y=1 	FALSE	**CORRECT**

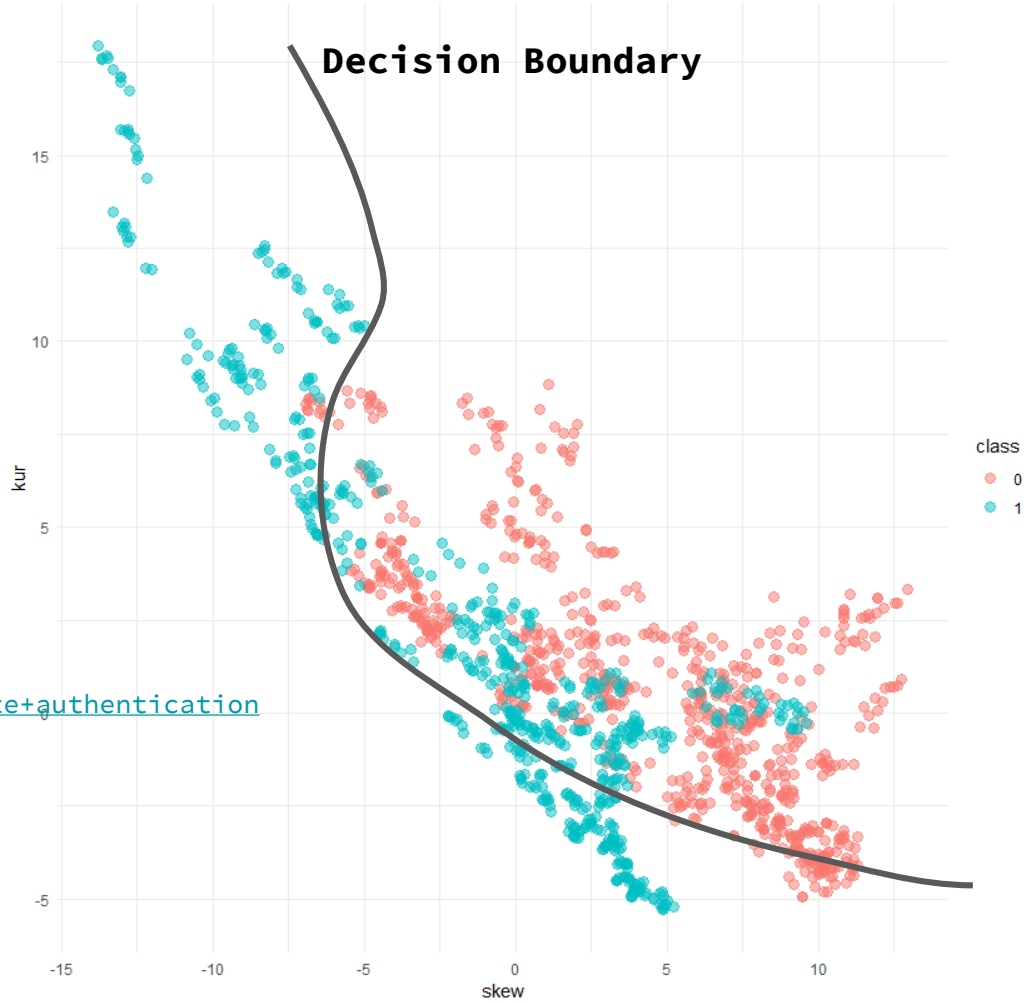
```
sum(diag(confusion_mat)) / sum(confusion_mat)
```



NOW YOUR TURN

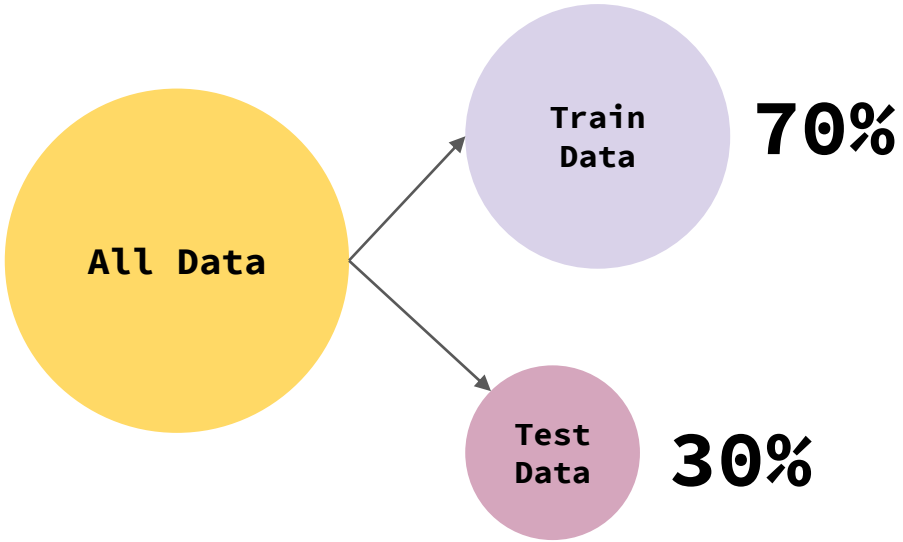
PROJECT: BANK NOTE

https://archive.ics.uci.edu/ml/datasets/banknote+authentication



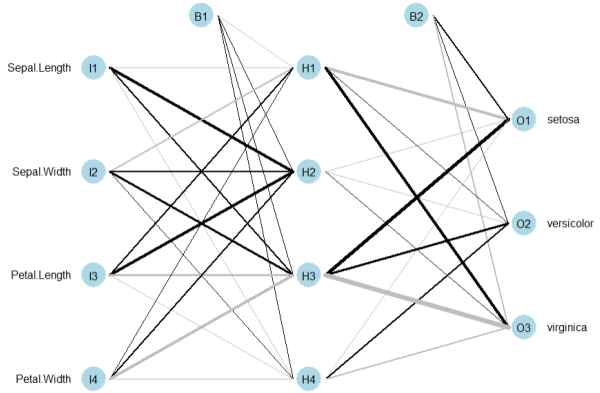
```
## IMPORT DATA FROM THE UCI PORTAL

my_df <- read.table("https://archive.ics.uci.edu/ml/machine-learning-
databases/00267/data_banknote_authentication.txt", sep = ",", header = FALSE,
col.names = c("var","skew","kur","entropy","class"))

head(my_df)
glimpse(my_df)
my_df$class <- as.factor(my_df$class)

ggplot(my_df, aes(skew,kur, col=class)) +
    geom_point(size = 3, alpha = 1/2) +
    theme_minimal()
```

STEP 1



ปล. อย่าลืมตั้งค่า set.seed() ด้วยนะคร้าบ

STEP 2-3

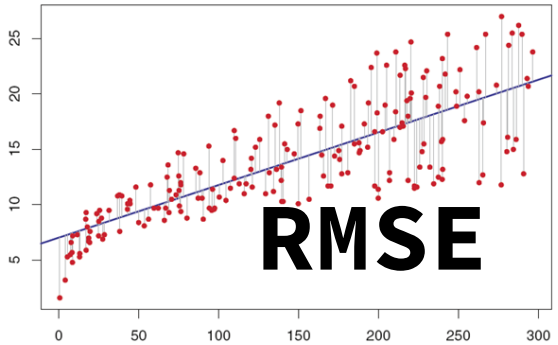


- สร้างโมเดลด้วย train_data
- ทดสอบโมเดลด้วย test_data
- ประเมินผลการทำนายด้วย confusion matrix

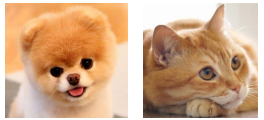
HOW DOES YOUR MODEL PERFORM?

How we evaluate models depends of the type of problem.

Regression



Classification

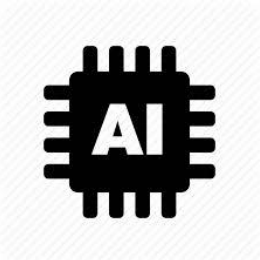


Accuracy

From confusion matrix

What we’ve learned is called:

ARTIFICIAL NEURAL NETWORK (ANN)

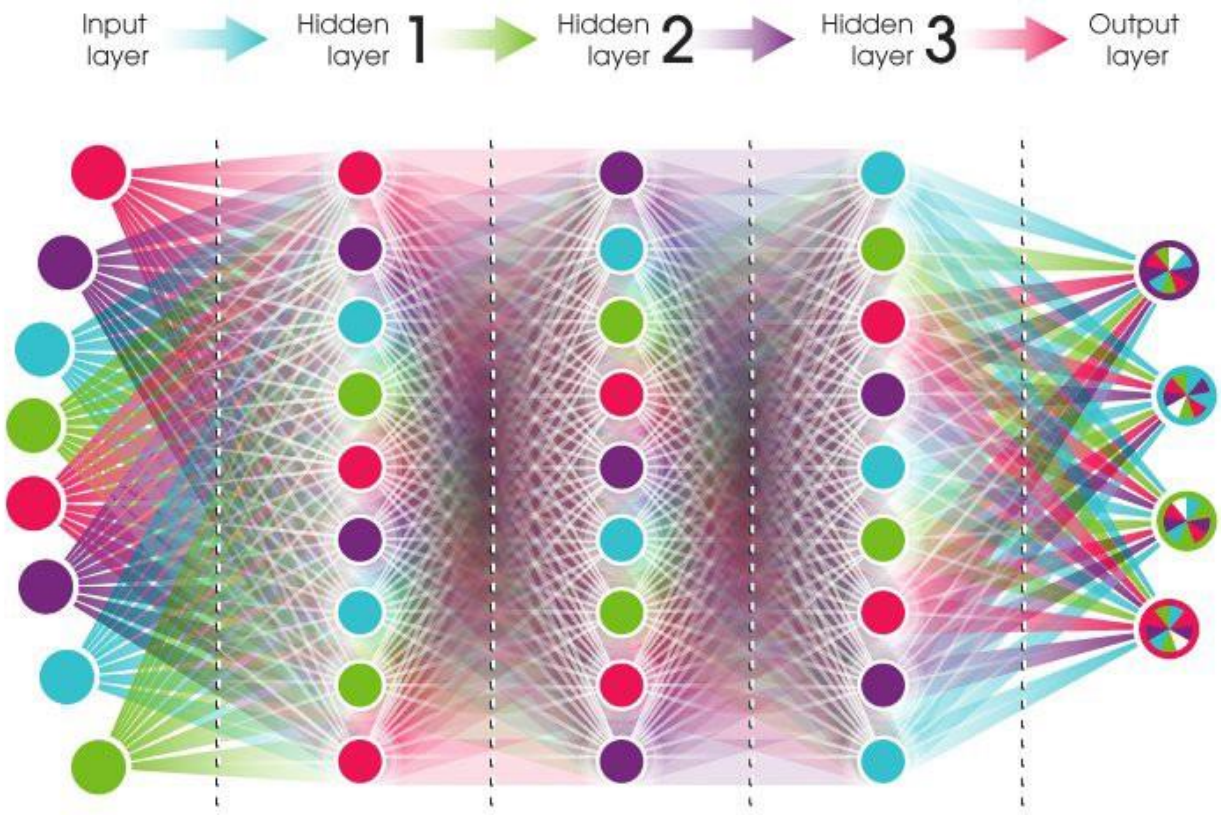


Modern architecture is so deep

Deep neural network

Deep Learning

DEEP NEURAL NETWORK



neuralnetworksanddeeplearning.com - Michael Nielsen, Yoshua Bengio, Ian Goodfellow, and Aaron Courville, 2016.

Graphics Processing Unit (GPU)

Data Analyst

What we learn today

1. Clean
2. Transform
3. Summarise

dplyr

4. Model
5. Visualize

ggplot2
Neural network


TV
14

DATAROCKIE SCHOOL

Blogs Login Sign Up

Never Stop Learning

Data Science | Statistics | Programming

สมัครเรียนฟรี

Featured Courses

Thank you very much kub :)

DataRockie