




Statistics 101 (Intro to Statistics)

Tags	Statistics
Class	
Finished Yet?	<input checked="" type="checkbox"/>
Knowledge	 The Sixth Sprint: Statistics

Lesson 1: Intro to Statistics

-หนังสือที่แนะนำให้อ่าน: Naked Statistics เขียนโดย Charles Wheelan

-pdf:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d3139a06-1f0e-4484-b398-8c2d27494d37/Live_-_Basic_Statistics.pdf

-Google Sheets: https://docs.google.com/spreadsheets/d/1ERs9eFcHPqWtyL-mmwClwXpinqrBRyvtm_Md_UQcpYA/edit#gid=0

-การเริ่มเรียนสถิติ ควรเข้าใจใน Research Design, Random, และ Representative

-การสุ่มที่ดีที่สุด คือการสุ่มแบบ Random Sampling (ประชากรที่เราสนใจ มีโอกาสโดนสุ่มเท่า ๆ กัน)

-Representative: สัดส่วนข้อมูลของกลุ่มตัวอย่างจะหน้าตาเหมือนกับประชากร (Sample Represents Population)

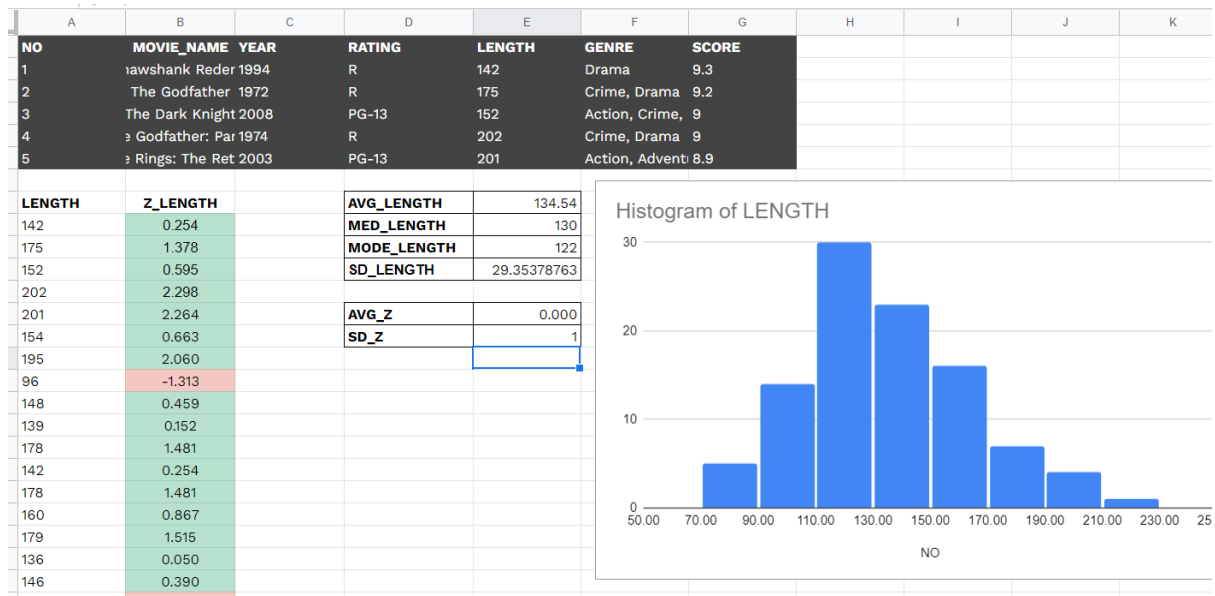
-สำมะโนประชากร (Census) ของไทยเก็บทุก ๆ 10 ปี

-ก่อนที่จะนำมาคำนวณทางสถิติ ต้องมั่นใจก่อนว่าข้อมูลของเรามีคุณภาพจริง ๆ (ขนาดไม่สำคัญเท่ากับคุณภาพในการสุ่มตัวอย่าง)

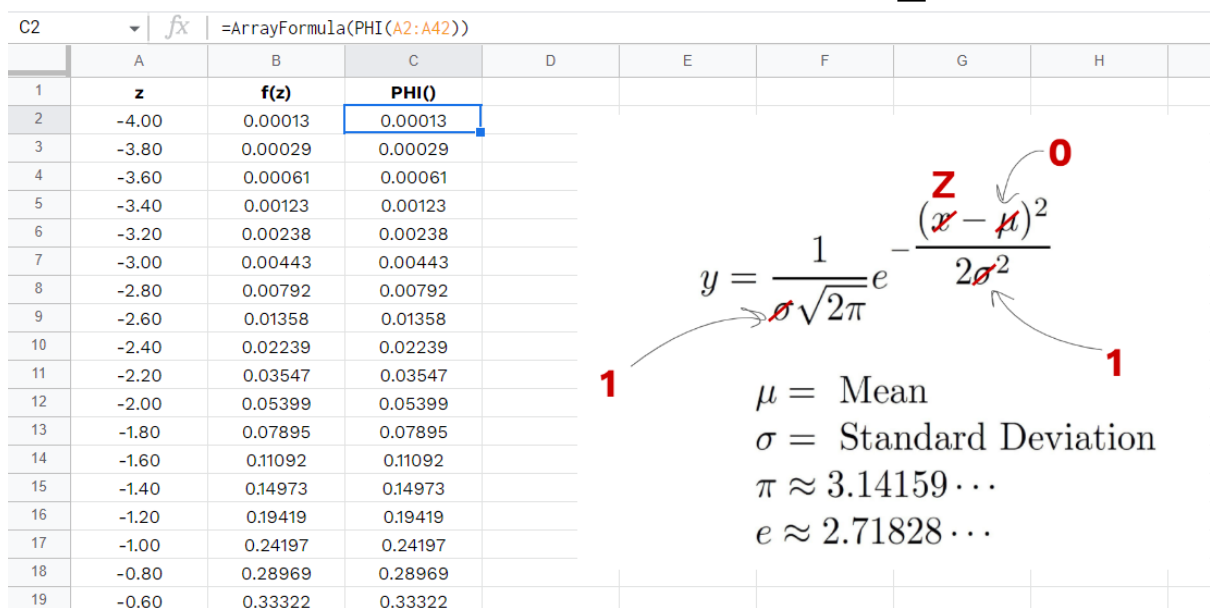
*Quality of our sample is of paramount importance

-ยกตัวอย่างง่าย ๆ เราทำซูป แล้วอยากรู้ว่าซูปอร่อยไหม เราคงไม่กินซูปทั้งหม้อ แต่เราจะหยิบขึ้นมา تذลิชิมแทน (Sampling) ถ้าซูปในช้อนที่เรา تذลิชิมอร่อยแล้ว ซูปทั้งหม้อก็น่าจะอร่อยเหมือนกัน (Inference)

- เราทำ Inference (การตัดสินใจ การสรุปผล) อิงจากข้อมูลที่เรามี ซึ่งข้อมูลที่เรามีนั้น อาจไม่ถูกต้องหรือครบถ้วน 100%
- ในยุคนี้ องค์กรใหญ่ไม่จำเป็นต้องสุ่มตัวอย่างเพื่อคำนวณทางสถิติ สมมติ DTAC มีผู้ใช้ 20 ล้านคน ทาง DTAC ไม่จำเป็นต้องสุ่มตัวอย่าง เพราะมีข้อมูลของลูกค้าทุกคนใน Database อยู่แล้ว จึงสามารถนำมาคำนวณได้ทันที
- Big Data คือการใช้ข้อมูลทั้งหมดมาวิเคราะห์แบบไม่สุ่มตัวอย่าง มีข้อมูลเท่าไรเอามาใช้ตัดสินใจหมดเลย
- Central Tendency คือการวัดค่ากลางข้อมูล (Mean, Median, Mode)
 1. Mean: ค่าเฉลี่ย (นำข้อมูลทั้งหมดมาบวกกัน แล้วหารด้วยจำนวนข้อมูล)
 2. Median: ค่าที่อยู่ตรงกลาง
 3. Mode: ค่าที่มีมากที่สุดในช่วงข้อมูล (เช่น 1 2 2 3 Mode = 2)
- ถ้า Graph เป็น Normal Distribution สามารถใช้ค่ากลางได้ทั้ง 3 ค่า
- Skewed Distribution มี 2 ประเภท
 1. Negative Skew (ระฆังเบ้ซ้าย หางไปทางซ้าย) $Mean < Median < Mode$
 2. Positive Skew (ระฆังเบ้ขวา หางไปทางขวา) $Mode < Median < Mean$
- Median จะทนต่อ Outlier
- ใช้ Mean ถ้าเป็น Normal Distribution
- ใช้ Median ถ้าเป็น Skewed Distribution
- ในแทบทุกประเทศ การกระจายรายได้ไม่สมมาตร (รวยกระจุก จนกระจาย) ซึ่ง GDP Per Capita วัดด้วยค่าเฉลี่ย และคนที่มียาได้เยอะมาก ๆ ไม่กี่คนสามารถ Skew GDP Per Capita ให้เพี้ยนไปได้
- GDP โตขึ้น แต่ก็มีเรื่องเงินเฟ้อ และต้นทุนการใช้ชีวิตสูงขึ้น
- วิกฤตเศรษฐกิจจะกลับมาทุก ๆ 12 ปี
- Variability (การวัดการกระจายตัวข้อมูล) ใช้ S.D. (Standard Deviation) และ Variance (Square Root ของ S.D.) ยิ่ง S.D. มีค่ามากเท่าไร ข้อมูลก็ยิ่งกระจายตัวมากเท่านั้น
- Position คือการวัดตำแหน่งข้อมูล ใช้ Box Plot เราจะแบ่งข้อมูลเป็น Quartile ที่ 2 (Median)
- Outlier: Extreme Value (ค่าสูง/ต่ำเกินไปในช่วงข้อมูล)
- $IQR = Q3 - Q1$ (เป็น Robust Statistics เหมือน Median ที่ทนต่อ Outlier)
- ค่าไหนที่มีค่ามากกว่า $Q3 + 1.5 * IQR$ อาจกำหนดให้เป็น Outlier ได้
- Z-Score $[(X - Mean) / SD]$ เปลี่ยนข้อมูลดิบเป็นค่ามาตรฐาน
- ค่า Z เป็นได้ทั้งลบและค่าบวก ถ้าค่าเป็น 0 เท่ากับว่าค่าของข้อมูลดิบเท่ากับค่าเฉลี่ย
- สมมติว่า เพื่อนในห้องสูง 169.2 ซม. แต่เราสูง 180 ซม. แล้วเราได้ค่า Z เท่ากับ +1.63 มีความหมายว่า เราสูงกว่าคนอื่นในห้องอยู่ 1.63 เท่าของ S.D.
- ถ้าข้อมูลของเรากระจายตัวปกติ เราสามารถหาพื้นที่ใต้กราฟได้ ดังนั้น เราสามารถ Normalize พื้นที่ใต้กราฟให้เป็น 1 ได้
- ถ้าข้อมูลของเรามีการกระจายตัวที่ดี 68.2% จะอยู่ในช่วง ± 1 ของ S.D., 95% จะอยู่ในช่วง ± 2 , และ 99.7% จะอยู่ในช่วง ± 3 ตามลำดับ
- กราฟที่ถูก Normalized ด้วย Z-Score จะมีค่าเฉลี่ยเป็น 0 และ S.D. เป็น 1 แต่ไม่ได้เปลี่ยน Distribution ของข้อมูล

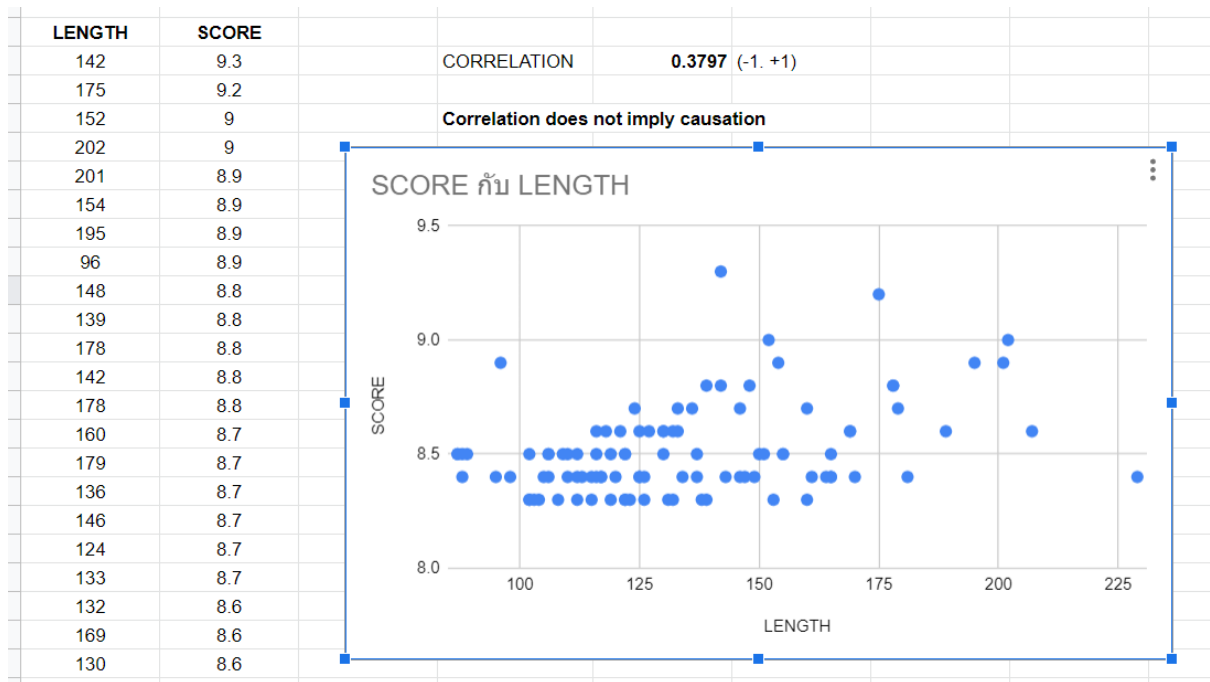


-เราสามารถใช =PHI() ในการคำนวณ Normal Distribution ได้



-Pearson Correlation Coefficient ใช้หาความสัมพันธ์ของข้อมูล 2 ชุดที่เป็นตัวเลขเหมือนกัน

-=CORREL() สามารถใช้หา Correlation Coefficient ได้ ซึ่งเราจะกำหนดข้อมูลที่เป็นตัวแปรตามก่อน จากนั้นค้นด้วยจุลภาค แล้วตามด้วยตัวแปรต้น



*Correlation does not imply causation

-Correlation Coefficient ใช้วัด Linear Association ดังนั้น เราจะใช้ Correlation Coefficient วัดข้อมูลที่มีกราฟเป็นเส้นตรง แต่ถ้า Plot Graph แล้วได้เป็นเส้นโค้ง ก็จะไม่เหมาะกับการใช้ Pearson Correlation Coefficient

-การเลือกใช้สถิติแบบต่าง ๆ จะขึ้นกับชุดข้อมูลเป็นหลัก

-My Google Sheets

:https://docs.google.com/spreadsheets/d/1l3oEBD_fTXj7Ru1HYt98DzMPCmBh0SW2_dEFWu9zclQ/edit?usp=sharing

Lesson 2: Intro to AB Testing

pdf:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8504addc-613e-4b7a-bdad-d68f1dae0c33/RCT_AB_Testing.pdf

Google Sheets: <https://docs.google.com/spreadsheets/d/1zHH10L6z2-4UeSAsfrl00eIMvFoPkpWT-KGK14ZjN5M/edit#gid=0>

-การทำ AB Testing เป็นพื้นฐานที่สำคัญมาก

-Type of Study:

1. Observational (Correlation) [เกิดจากการ Observe เช่น แบบสอบถาม หรือการทำ Survey เป็นต้น]
2. Experimental (Causation) [x ทำให้เกิด y ถ้า x เปลี่ยน y จะเปลี่ยนแปลง]

*Correlation does not imply causation

- Spurious Correlation: ความสัมพันธ์แบบปลอม ๆ ไม่ได้เกี่ยวข้องกันจริง ๆ
 - Lurking Variable: ตัวแปรที่เราไม่เห็น แต่ส่งผลต่อความสัมพันธ์ที่เราเห็น
 - Randomized Control Trials (RCT): Golden Standard to test causation
 - เปรียบเทียบ 2 กลุ่ม กลุ่มหนึ่งเป็นกลุ่มปกติ อีกกลุ่มทำ Treatment ออก Action ให้มีปัจจัยต่างจากกลุ่มแรก แล้วนำมาวัดผล
 - เช่น แบ่งกลุ่มเป้าหมาย 2 กลุ่ม คนกลุ่มหนึ่งเห็นโฆษณา แต่อีกกลุ่มไม่เห็น แล้วนำมาเทียบกันตอนจบ
- [RCT]

Keyword	Meaning
Test Group	The group that could potentially be exposed to the treatment
Control Group	The group that won't be exposed to the treatment
Treatment	The variable that the test group is exposed to
Result	The true impact of an ad. Also known as "Lift"

[AB Testing]

Keyword	Meaning
AB Testing	This type of experiment lets you test different versions of your ads so you can see what works best and improve future campaigns
Test Groups	The random, non-overlapping audiences in an experiment
Ad versions	This is what happens when Facebook duplicates your ads and changes the variable you choose
Results	Show which ads produced the lowest cost per result

*อธิบายง่าย ๆ คือ RCT กลุ่มหนึ่งเห็นโฆษณา แต่อีกกลุ่มไม่เห็น ส่วน AB Testing กลุ่มทั้งสองกลุ่มเห็นโฆษณาทั้งคู่ แต่เห็นคนละ Version (A และ B)

-Ideal คือทำทั้ง 2 แบบ เช่นเรื่องโฆษณา RCT ทำเพื่อให้เห็นว่าโฆษณาของเรามี Impact หรือเปล่า ส่วน AB Testing ทำเพื่อดูว่าโฆษณาแบบไหนมี Impact มากกว่ากัน

-Descriptive Statistics

-Uniform Distribution: ทุกค่าในช่วงมีโอกาสถูกสุ่มได้เท่ากัน

-NORMINV(RAND(), mean, SD) = สุ่มค่าที่อยู่บน Normal Distribution ที่มีค่า mean และ SD เท่ากับค่าที่เราตั้งไว้ ใช้สร้างข้อมูลเบื้องต้นได้

-Confidence Interval [ได้ใช้เยอะมาก]

-=CONFIDENCE.NORM(alpha, SD, Sample Size) ใช้หา Margin of Error (เช่น ความมั่นใจ 95% ใช้ Alpha = 0.05 ถ้ามั่นใจ 99% ใช้ Alpha = 0.01 เป็นต้น)

-สร้าง Upper Bound และ Lower Bound ด้วย Mean + Margin of Error และ Mean - Margin of Error ตามลำดับ

-T-test ใช้ในการเปรียบเทียบค่าเฉลี่ย แต่ต้องวัดการกระจายตัวของข้อมูลก่อนโดยใช้ F-test เปรียบเทียบ Variance

*ต้องทำ F-test ก่อน T-test เสมอ

[F-test]

-H0 = Equal variance assumed

-H1 = Equal variance not assumed

-If p-value ≤ 0.05 , reject H0

[T-test]

-H0: mean store1 = store2

-H1: mean store1 \neq store2

-if p-value \leq 0.05, reject H0

-ทุกวันนี้เราสามารถใช้ Simulation ช่วยได้ ไม่ต้องเขียนสูตรเหมือนสมัยก่อน

-Paired T-test: เทียบค่าเฉลี่ยของคน 2 กลุ่ม ซึ่งคน 2 กลุ่มนี้เป็นคนกลุ่มเดียวกัน แต่เป็นคนละช่วงเวลา (เช่น ก่อน เทรน-หลัง เทรน)

-Paired T-test ไม่ต้องทำ F-test หา Variance ก่อน

-H0: mean before = mean after

-H1: mean before \neq mean after

-if p-value \leq 0.05, Reject H0

-My Google Sheets: <https://docs.google.com/spreadsheets/d/1UnlsjLYGtoyuhWS8Pe9Lzsonyy2TMY-stTDXJFgyJPg/edit#gid=0>

Lesson 3: Correlation and Linear Regression

-mtcars dataset:

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ba04513a-5d24-4000-9b81-8cfc6778e39a/Untitled.xlsx>

-pdf:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/00e181a4-3306-4f4e-ab45-7283adccc6d6/Live_-_Correlation_and_Linear_Regression.pdf

-Correlation (Pearson Correlation) คือสถิติสำหรับหาความสัมพันธ์ของตัวแปรที่เป็นตัวเลข 2 ตัว

-ค่าวิ่งอยู่ระหว่าง -1 ถึง +1 และเครื่องหมายบวกลบจะบอกทิศทางความสัมพันธ์ของตัวแปรสองตัว

-ยิ่งค่าเข้าใกล้ |1| ความสัมพันธ์ก็จะยิ่งสูง

-Chart ที่เราใช้ในการหา Correlation คือ Scatter Plot

-ใช้วัดความสัมพันธ์แบบเส้นตรง (Linear Correlation)

-ใน Excel จะมี Function ชื่อ =CORREL() ใช้หาค่า Correlation Coefficient จากข้อมูล 2 column ได้

-Analysis Toolpak ใน Excel ช่วยหาค่า Correlation Coefficient ในหลาย column พร้อม ๆ กันได้ จากนั้น เราสามารถใช้ Conditional Formatting ช่วยให้เห็นภาพชัดเจนขึ้น เช่น สมมติค่าที่เป็นบวก เป็นต้น

-0.77617											
	hp	mpg									
hp	1										
mpg	-0.77617	1									
	hp	mpg	cyl	disp	drat	wt	qsec	vs	am	gear	carb
hp	1.00										
mpg	-0.78	1.00									
cyl	0.83	-0.85	1.00								
disp	0.79	-0.85	0.90	1.00							
drat	-0.45	0.68	-0.70	-0.71	1.00						
wt	0.66	-0.87	0.78	0.89	-0.71	1.00					
qsec	-0.71	0.42	-0.59	-0.43	0.09	-0.17	1.00				
vs	-0.72	0.66	-0.81	-0.71	0.44	-0.55	0.74	1.00			
am	-0.24	0.60	-0.52	-0.59	0.71	-0.69	-0.23	0.17	1.00		
gear	-0.13	0.48	-0.49	-0.56	0.70	-0.58	-0.21	0.21	0.79	1.00	
carb	0.75	-0.55	0.53	0.39	-0.09	0.43	-0.66	-0.57	0.06	0.27	1.00

-Linear Regression: $y = b_0 + b_1x$ [b_0 = intercept, b_1 = slope (delta y / delta x)] ใช้กับข้อมูลที่เป็นเส้นตรง

*อย่า Extrapolate (สรุปผลเกินจากข้อมูลที่เรามี)

-The best fitted line = The lowest error

-Positive Coorelation = Slope เป็นบวก | Negative Coorelation = Slope เป็นลบ

-No Coorelation = Slope เป็น 0 (ข้อมูลทั้งสองไม่มีความเกี่ยวข้องกัน)

-ค่า R Square ใช้ในการวัดผลโมเดล มีค่าระหว่าง 0 ถึง 1 ยิ่งค่าเข้าใกล้ 1 ยิ่งดี เกิดจากค่า Correlation^2

SUMMARY OUTPUT						
Regression Statistics						
Multiple R	0.777488522					
R Square	0.604488402					
Adjusted R Square	0.590850071					
Standard Error	3.917366721					
Observations	31					
ANOVA	Overall Significance of the model					
	df	SS	MS	F	Significance F	
Regression	1	680.1664	680.1664	44.32275	2.66906E-07	<0.05 Significance
Residual	29	445.0271	15.34576			
Total	30	1125.194				
	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	30.21210626	1.679217	17.99179	2.83E-17	26.7777225	33.64649
hp	-0.068646443	0.010311	-6.65753	2.67E-07	-0.089734994	-0.04756

-P-value ถ้าน้อยกว่า 0.05 เท่ากับว่า โอกาสที่จะเกิด Extreme value มีน้อยกว่า 5% (ยิ่งน้อยยิ่งดี)

=LINEST() ใช้ในการหาค่า Coorelation Coefficient ของหลาย Column (แต่ใช้ Analysis Toolpak จะสะดวกกว่าและไม่ต้องเขียนสูตร)

-Objectives that we run model

1. Prediction

2. Inference (Why)

-โมเดลที่อธิบายได้ ส่วนใหญ่จะมีประสิทธิภาพไม่ค่อยดี แต่โมเดลที่อธิบายไม่ได้ ส่วนใหญ่จะมีประสิทธิภาพดี

-ในชีวิตจริง Regression อาจไม่ได้ใช้เยอะ เพราะมีโมเดลอื่น ๆ ที่มีประสิทธิภาพมากกว่า แต่ข้อดีของ Regression Model คือ สามารถอธิบายได้

-Residual = Error (ค่า Actual - ค่า Predicted)

-Objective ของ Linear Regression คือการ Minimize ค่า Sum Squared Error (ผลรวม Error ยกกำลังสอง)

-RMSE (Root Mean Squared Error) ใช้ในการวัดผล Linear Regression Model ยิ่งน้อยยิ่งดี

[Model Training Golden Rule]

-Full Data ต้องแบ่งเป็น Train Data กับ Test Data เพื่อวัดผลข้อมูล (Test Data คือข้อมูล Unseen Data ที่ตัวโมเดลไม่เคยเห็นในช่วงการเทรนโมเดล)

-Overfitting: โมเดลเข้ากับข้อมูล Train ได้ดีเกินไป ถ้ามีข้อมูลใหม่เข้ามา จะทำนายได้ไม่ดี (ค่า Error จะพุ่งสูงขึ้นมาก)

*อาจแก้ได้ด้วยการเพิ่ม Data หรือเปลี่ยนสัดส่วนการ Split Data และการทำ Cross Validation จะลดปัญหา Overfitting ไปได้เยอะมาก

```
##Correlation
library(ggplot2)

cor(mtcars$mpg, mtcars$hp)
cor(mtcars$mpg, mtcars$wt)

ggplot(mtcars, aes(hp, mpg)) + geom_point()
ggplot(mtcars, aes(wt, mpg)) + geom_point()
ggplot(mtcars, aes(wt, hp)) + geom_point()

cor(mtcars[, c("mpg", "wt", "hp")])

#Dplyr (Tidyverse)
library(dplyr)
cormat <- mtcars %>%
  select(mpg, wt, hp, am) %>%
  cor()

#Compute correlation (r) and significance test
cor(mtcars$mpg, mtcars$hp)
cor.test(mtcars$mpg, mtcars$hp)

##Linear Regression
#Simple linear regression
lmFit <- lm(mpg ~ hp, data = mtcars)

summary(lmFit)

#Prediction
lmFit$coefficients[[1]] + lmFit$coefficients[[2]] * 200

new_cars <- tibble(
  hp = c(250, 320, 400, 410, 450)
)

#Predict
new_cars$mpg_pred <- predict(lmFit, newdata = new_cars)
new_cars$hp_pred <- null
new_cars

summary(mtcars$hp)

#RMSE (Root Mean Squared Error)
#Multiple linear regression
```



```

#mpg = f(hp, wt, am)
#mpg = intercept + b0*hp + b1*wt + b2*am
lmFit_v2 <- lm(mpg ~ hp + wt + am, data = mtcars)
coefs <- coef(lmFit_v2)
coefs[[1]] + coefs[[2]]*200 + coefs[[3]]*3.5 + coefs[[4]] * 1

##Build full model
lmFit_full <- lm(mpg ~ ., data = mtcars)
mtcars$predicted <- predict(lmFit_full)
head(mtcars)

##Train RMSE
squared_error <- (mtcars$mpg - mtcars$predicted) ** 2
(rmse <- sqrt(mean(squared_error)))

##Randomly split Data
set.seed(42)
n <- nrow(mtcars)
id <- sample(1:n, size = n*0.7)
train_data <- mtcars[id, ]
test_data <- mtcars[-id, ]

#Train model
model1 <- lm(mpg ~ hp + wt + am + disp, data = train_data)
p_train <- predict(model1)
(rmse_train <- sqrt(mean((train_data$mpg - p_train) ** 2)))

#Test model
p_test <- predict(model1, newdata = test_data)
(rmse_test <- sqrt(mean((test_data$mpg - p_test) ** 2)))

#Print result
cat("RMSE of train data: ", rmse_train,
    "\nRMSE of test data: ", rmse_test)

##Logistic regression
mtcars %>% head()

str(mtcars)

#Convert am to factor
mtcars$am <- factor(mtcars$am,
                    levels = c(0,1),
                    labels = c("Auto", "Manual"))
class(mtcars$am)
table(mtcars$am)

#Randomly split Data
set.seed(42)
n <- nrow(mtcars)
id <- sample(1:n, size = n*0.7)
train_data <- mtcars[id, ]
test_data <- mtcars[-id, ]

#Train model
logit_model <- glm(am ~ mpg,
                  data = train_data,
                  family = "binomial")
p_train <- predict(logit_model, type = "response") #probability
train_data$pred <- if_else(p_train >= 0.5, "Manual", "Auto")
mean(train_data$am == train_data$pred)

#Test model
p_test <- predict(logit_model,
                  newdata = test_data,
                  type = "response")
test_data$pred <- if_else(p_test >= 0.5, "Manual", "Auto")
mean(test_data$am == test_data$pred)

```

-my excel:

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/16a2b5b7-5ed4-4180-9f09-4e00e6abcf95/Untitled.xlsx>

Lesson 4: Logistic Regression

Google Sheets: <https://docs.google.com/spreadsheets/d/1-FEDMGWvWJFJ2-caBgaMLV2TMN4L7SA3jdIN27yFRWE/edit#gid=0>

pdf:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/79280ff6-5b74-4c89-b33e-b15a64a5cb14/Logistic_Regression.pdf

-เราจะใช้ Logistic Regression กับงานประเภท Classification (Predict binary outcome)

- $y = f(x) \rightarrow \text{divorced} = f(\text{happiness level})$

-Sigmoid Function: $S(z) = 1 / (1 + e^{-z})$

-Alternate: $S(z) = e^z / (1 + e^z)$ *ได้ผลลัพธ์เหมือนกัน

-ในการสร้าง ML Model ตัว Performance ของ Train กับ Test ควรจะมีค่าใกล้เคียงกัน (Comparable)

-โปรแกรมทางสถิติทุกโปรแกรมจะเช็คค่า Confidence ไว้ที่ 5% (0.05) ปกติที่ใช้กันจะมี 1%, 5% และ 10%

-การปรับ Threshold จะมีผลต่อความแม่นยำของ Model

-Model Evaluation: Model ของเรามีประสิทธิภาพแค่ไหน ? (ใช้ Test Data ในการทดสอบ Model)

-Confusion Matrix ใช้แสดงผลค่า 4 อย่าง

1. ทายถูก ถูกจริง (True Positive) [TP]
2. ทายถูก ดันผิด (False Positive) [FP]
3. ทายผิด ผิดจริง (True Negative) [TN]
4. ทายผิด ดันถูก (False Positive) [FN]

-ตารางการใช้งานค่า 4 อย่างใน Confusion Matrix:

Keywords	How to find?
Accuracy	$TP + TN / N$ [N = total number of dataset]
Precision	$TP / (TP + FP)$
Recall	$TP / (TP + FN)$
Harmonic Mean (F1)	$2 * (Precision * Recall) / (Precision + Recall)$

-เราสามารถใส่ rep(ค่า, จำนวนครั้งที่อยากทำซ้ำ) ใน R เพื่อแสดงค่าซ้ำ ๆ ได้

```
##Logistic regression example (Binary Classification)
library(tidyverse)
```

```

happiness <- c(10, 8, 9, 7, 8,
              5, 9, 6, 8, 7,
              1, 1, 3, 1, 4,
              5, 6, 3, 2, 0)
divorce <- c(0, 0, 0, 0, 0,
            0, 0, 0, 0, 0,
            1, 1, 1, 1, 1,
            1, 1, 1, 1, 1)

df <- tibble(happiness, divorce)

#Fit logistic regression to full data set
model <- glm(divorce ~ happiness, data = df, family = "binomial")

summary(model)

#Predict & evaluate model
df$prob_divorce <- predict(model, type = "response")
df$pred_divorce <- ifelse(df$prob_divorce >= 0.5, 1, 0)

#Confusion Matrix
conM <- table(df$pred_divorce, df$divorce,
             dnn = c("Predicted", "Actual"))

#Model Evaluation
cat("Accuracy =", conM[1, 1] + conM[2, 2] / sum(conM) )
cat("Precision =", conM[2, 2] / ( conM[2, 1] + conM[2, 2]) )
cat("Recall =", conM[2, 2] / ( conM[1, 2] + conM[2, 2]) )
cat("F1 =", 2 * ( (0.9 * 0.9) / (0.9 + 0.9) ) )

```

Homework

-Using logistic regression to predict the survival rate of passengers in Titanic, and compare the results between train model and test model

```

##Logistic Regression Homework

#Load tidyverse library
library(tidyverse)

#Load titanic train data set
titanic_train <- read_csv("train (1).csv")

#Check column name
head(titanic_train)

#Drop NA
titanic_train <- na.omit(titanic_train)

#Check column count
nrow(titanic_train)

#Check if value in 'Survived' column is factor or not
str(titanic_train$Survived)

#Change the value in 'Survived' column from int to factor
titanic_train$Survived <- factor(titanic_train$Survived,
                                levels = c(0,1),
                                labels = c("Not Survived",
                                             "Survived"))

#Split data
set.seed(42)

```

```

n <- nrow(titanic_train)
id <- sample(1:n, size = n * 0.7) #Train 70% Test 30%
train_data <- titanic_train[id, ]
test_data <- titanic_train[-id, ]

#Train model
model <- glm(Survived ~ Pclass,
             data = train_data,
             family = "binomial")

#Predict & evaluate train model
train_data$prob_survive <- predict(model, type = "response")
train_data$pred_survive <- if_else(train_data$prob_survive >= 0.7,
                                  1, 0)

summary(train_data)
conM <- table(train_data$pred_survive, train_data$Survived,
             dnn = c("Predicted", "Actual"))

#create Variables to store accuracy, precision, recall, and F1 for conM
acc_train <- conM[1, 1] + conM[2, 2] / sum(conM)
pre_train <- conM[2, 2] / ( conM[2, 1] + conM[2, 2] )
rec_train <- conM[2, 2] / ( conM[1, 2] + conM[2, 2] )
f1_train <- 2 * ( (pre_train * rec_train) / (pre_train + rec_train) )

#Show accuracy, precision, recall, and F1 for conM
cat("accuracy_train = ", acc_train,
    "\nprecision_train = ", pre_train,
    "\nrecall_train = ", rec_train,
    "\nf1_train (Harmonic Mean) = ", f1_train)

##Predict & evaluate test model
test_data$prob_survive <- predict(model,
                                 newdata = test_data,
                                 type = "response")
test_data$pred_survive <- if_else(test_data$prob_survive >= 0.7,
                                  1, 0)

summary(test_data)
conMtest <- table(test_data$pred_survive, test_data$Survived,
                 dnn = c("Predicted", "Actual"))

#create Variables to store accuracy, precision, recall, and F1 for conMtest
acc_test <- conMtest[1, 1] + conMtest[2, 2] / sum(conMtest)
pre_test <- conMtest[2, 2] / ( conMtest[2, 1] + conMtest[2, 2] )
rec_test <- conMtest[2, 2] / ( conMtest[1, 2] + conMtest[2, 2] )
f1_test <- 2 * ( (pre_test * rec_test) / (pre_test + rec_test) )

#Show accuracy, precision, recall, and F1 for conMtest
cat("accuracy_test = ", acc_test,
    "\nprecision_test = ", pre_test,
    "\nrecall_test = ", rec_test,
    "\nf1_test (Harmonic Mean) = ", f1_test)

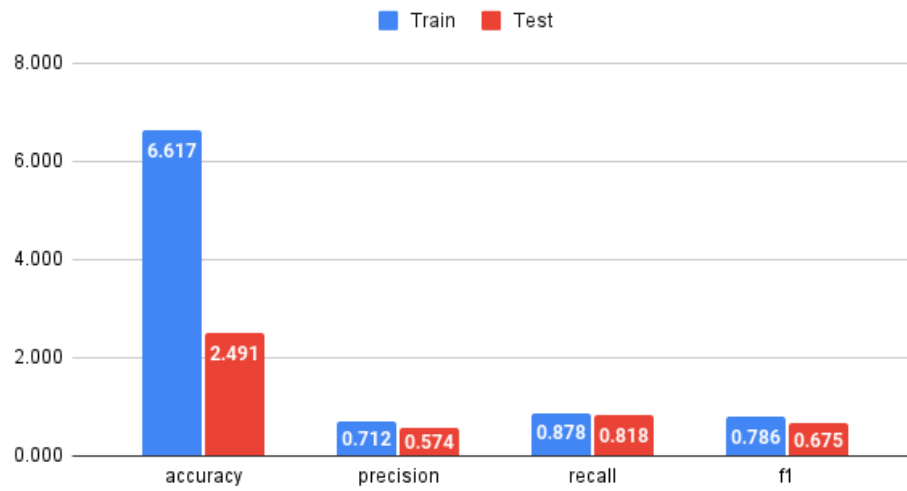
#Create data frame to store train and test parameters
titanic_logit_df <- data.frame(accuracy = c(acc_train, acc_test),
                              precision = c(pre_train, pre_test),
                              recall = c(rec_train, rec_test),
                              f1 = c(f1_train, f1_test),
                              row.names = c("train", "test"))

#export data frame as csv
write_csv(titanic_logit_df, "titanic_logit_test.csv")

```

-Visualize data using Google Sheets chart

Titanic Logistic Regression Train&Test Model Comparison



-In conclusion, The accuracy of train model is much higher than the accuracy of test model, so it is likely that the model is overfitting.