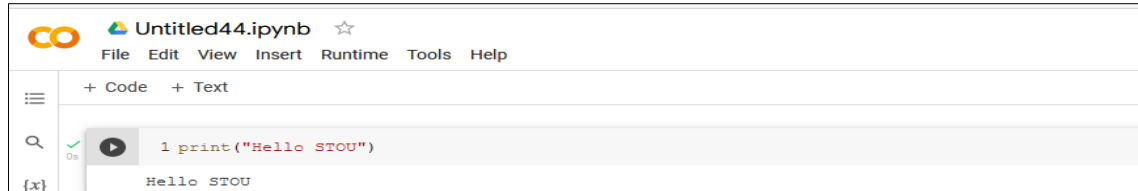


## 1 LAB1-Introduction to Python on Google Colab

### 1.1. `print("Hello STOU")`



### 1.2. #Single line comment `print("Hello STOU") #first command`

### 1.3. #Multiline comments `x = 5` `y = "Google"` `z = "Colab"` `print(x)` `print(y)` `print(z)`

### 1.4. #Python basic operators `num1 = 10 # int` `num2 = 5.2 # float` `num3 = num1+num2` `print(num3)`

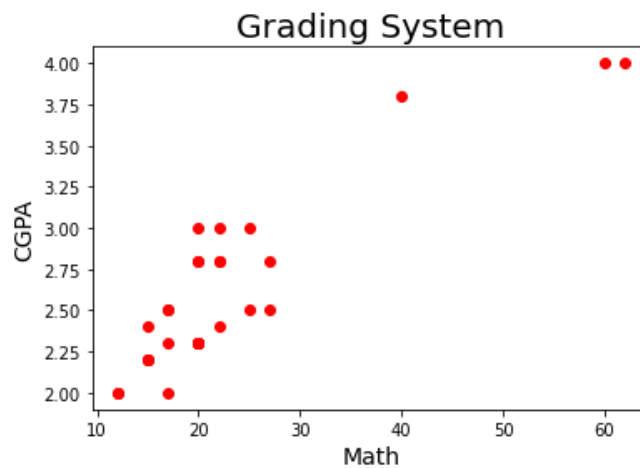
### 1.5. #Python concatenate strings `firstname="URAI"` `lastname="PAITON"` `fullname = firstname+lastname` `print(fullname)`

### 1.6. #Modules `import pandas as pd #all files` `import matplotlib.pyplot as plt` `import numpy as np`

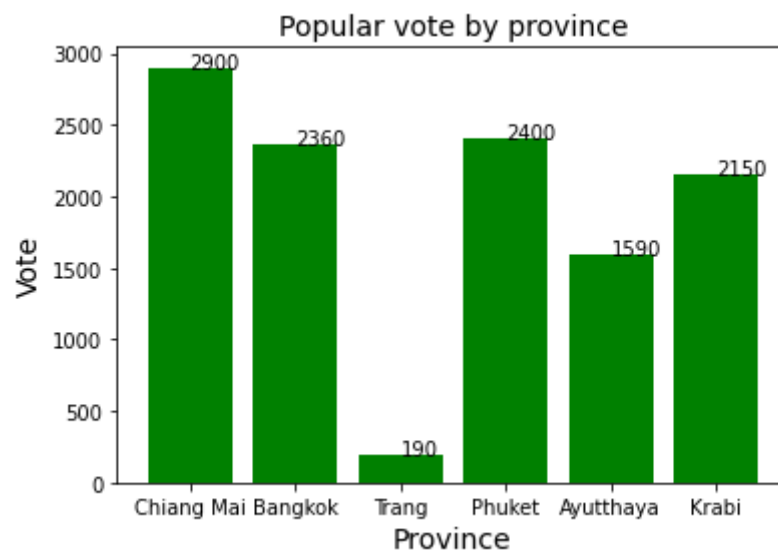
### 1.7. #Read File `data = pd.read_csv('/content/sample_data/GradingSystem.csv')` `print(data)`

### 1.8. #count all rows `count_data = len(data)` `print("Count all rows:" + str(count_data))`

```
1.9. #Plot Graph
x = data['Math']
y = data['CGPA']
plt.scatter(x,y,color='red')
plt.xlabel('Math', fontsize=14)
plt.ylabel('CGPA', fontsize=14)
plt.show()
```



## 2 LAB2- Python Libraries for Data Science (votebyprovince.csv)



```
2.1 import..... # Matplotlib
```

```
2.2 import..... # Pandas
```

```
2.3 data = pd.read_csv('อ่านไฟล์ votebyprovince.csv')
```

```
2.4 print(data)
```

	province	vote
0	Chiang Mai	2900
1	Bangkok	2360
2	Trang	190
3	Phuket	2400
4	Ayutthaya	1590
5	Krabi	2150

## 2.5 data.sort\_values(by=['vote'],ascending=True)

```
1 data.sort_values(by=['vote'],ascending=True)
```

	province	vote
2	Trang	190
4	Ayutthaya	1590
5	Krabi	2150
1	Bangkok	2360
3	Phuket	2400
0	Chiang Mai	2900

## 2.6 #Plot Graph

2.7 **x = data[ ?? ]** #ข้อมูลจังหวัด

2.8 **y = data[ ?? ]** #ข้อมูลคะแนนโหวต

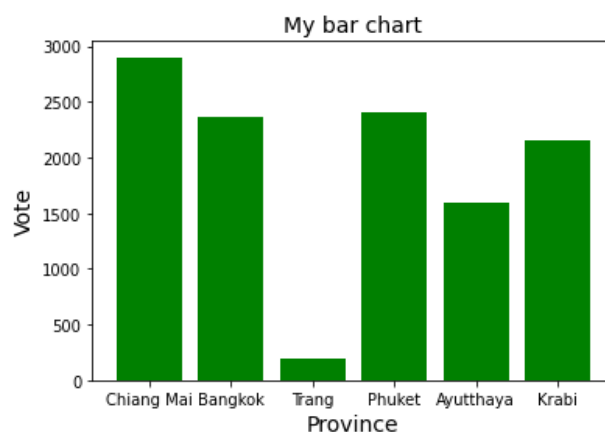
2.9 **plt.bar(x,y,color='green')**

2.10 **plt.xlabel('Province', fontsize=14)**

2.11 **plt.ylabel('Vote', fontsize=14)**

2.12 **plt.title('Popular vote by province', fontsize=14)**

2.13 **plt.??)** #แสดงกราฟ



## 2.14 #Plot Graph with label

2.15 **def addlabels(x,y):**

2.16 **for i in range(len(x)):**

2.17 **plt.text(i,y[i],y[i])**

```

2.18 x = data['province']
2.19 y = data['vote']
2.20 plt.bar(x,y,color='green')
2.21 addlabels(x, y)
2.22 plt.xlabel('Province', fontsize=14)
2.23 plt.ylabel('Vote', fontsize=14)
2.24 plt.title('Popular vote by province', fontsize=14)
2.25 plt.show()

```

### 3 LAB3- grading system (GradingSystem.csv)

```

✓ [33] 1 #ทดสอบคะแนนคณิตศาสตร์ = 30
0s    2 x = 30    #คะแนนคณิตศาสตร์
      3 def predictCGPAscore():
      4     a = 1.64536225 #จุดตัด
      5     b = 0.04122299 #สัมประสิทธิ์ถดถอย
      6     error = 0
      7     y = a + np.sum(b*X) + 0
      8     print(y)
      9 predictCGPAscore()

2.88205195

```

3.1 pip install scikit-learn

3.2 #การนำเข้าโมดูลต่างๆ

3.3 import numpy as np

3.4 import pandas as pd

3.5 import matplotlib.pyplot as plt

3.6 import sklearn #ติดตั้ง sklearn ด้วยคำสั่ง pip install scikit-learn

3.7 from sklearn.linear\_model import LinearRegression

3.8 from sklearn.metrics import mean\_squared\_error, r2\_score

3.9 #read data

3.10 pd.set\_option('display.max\_rows', None)

3.11 dataset = pd.read\_csv('content/sample\_data/GradingSystem.csv')

3.12 print(dataset)

	StudentNo	Science	Math	CGPA
0	1	22	27	2.5
1	2	23	17	2.3
2	3	31	25	2.5
3	4	27	22	2.4
4	5	32	20	2.3
5	6	32	20	2.3
6	7	23	15	2.2
7	8	18	20	2.8
8	9	26	22	2.8
9	10	30	17	2.5
10	11	30	20	2.3
11	12	30	15	2.4
12	13	30	27	2.8
13	14	30	22	3.0
14	15	30	20	3.0
15	16	62	25	3.0
16	17	50	17	2.0
17	18	50	20	2.3
18	19	50	12	2.0
19	20	50	20	2.3
20	21	50	15	2.2

3.13 **#set data x,y**

3.14 `x = dataset['Math']`

3.15 `y = dataset['CGPA']`

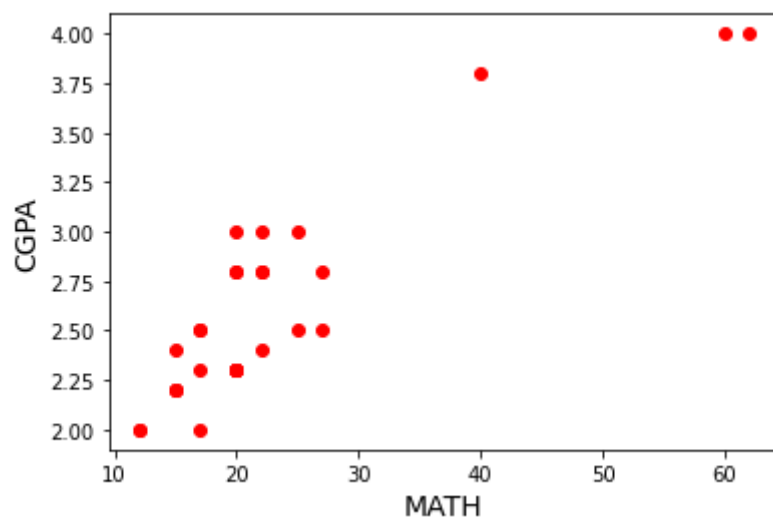
3.16 **#plot graph**

3.17 `plt.scatter(x,y,color='red')`

3.18 `plt.xlabel('MATH', fontsize=14)`

3.19 `plt.ylabel('CGPA', fontsize=14)`

3.20 `plt.show()`



3.21 **#กำหนดค่า x,y**

3.22 `x = dataset.iloc[:, 2].values.reshape(-1, 1)` #อาร์เรย์ตัวแปรอิสระ

3.23 `y = dataset.iloc[:, 3].values.reshape(-1, 1)` #อาร์เรย์ตัวแปรตาม

### 3.24 #การแบ่งข้อมูลออกเป็น 70:30

3.25 `from sklearn.model_selection import train_test_split`

3.26 `x_train, x_test, y_train, y_test = train_test_split(x,y,train_size=0.7, test_size=0.3,random_state=0)`

### 3.27 #การสร้างแบบจำลองการถดถอยเชิงเส้นอย่างง่าย

3.28 `regression_model = LinearRegression()`

3.29 `regression_model.fit(x_train,y_train)`

### 3.30 #การทำนายข้อมูล

3.31 `y_predicted = regression_model.predict(x_test)`

3.32 `y_predicted`

### 3.33 #การแสดงผลค่าจริง และค่าทำนาย

3.34 `df = pd.DataFrame({'Actual': [y_test], 'Predicted': [y_predicted]})`

3.35 `print(df)`

### 3.36 #การวัดประสิทธิภาพของแบบจำลองการถดถอยเชิงเส้นอย่างง่าย

3.37 `rmse = mean_squared_error(y_test, y_predicted)`

3.38 `r2_score = r2_score(y_test,y_predicted)`

### 3.39 #การแสดงผลค่าผลการวัดประสิทธิภาพของแบบจำลอง

3.40 `print('The intercept is:', regression_model.intercept_)`

3.41 `print('The coefficient is:',regression_model.coef_)`

3.42 `print('The rmse is:',rmse)`

3.43 `print('The r2_score is:',r2_score)`

```

➤ The intercept is: [1.64536225]
  The coefficient is: [[0.04122299]]
  The rmse is: 0.05792675297327944
  The r2_score is: 0.32195563716248055

```

### 3.44 #ทดสอบทำนายคะแนนคณิตศาสตร์ = 30

3.45 `X = 30` #คะแนนคณิตศาสตร์

3.46 `def predictCGPAScore():`

3.47 `a = 1.64536225` #จุดตัด

3.48 `b = 0.04122299` #สัมประสิทธิ์ถดถอย

3.49 `error = 0`

3.50 `y = a + np.sum(b*X) + 0`

3.51 `print (y)`

3.52 `predictCGPAScore()`

ตอบ.....

**3.53 #ทดสอบทำนายคะแนนคณิตศาสตร์ = 20**

```

3.54 X = 30 #คะแนนคณิตศาสตร์
3.55 def predictCGPAScore():
3.56     a = 1.64536225 #จุดตัด
3.57     b = 0.04122299 #สัมประสิทธิ์ถดถอย
3.58     error = 0
3.59     y = a + np.sum(b*X) + 0
3.60     print (y)
3.61 predictCGPAScore()

```

ตอบ.....

**4 LAB4- student dropout (Student.csv)**

```

[[41  0]
 [ 2  7]]

```

	precision	recall	f1-score	support
0	0.95	1.00	0.98	41
1	1.00	0.78	0.88	9
accuracy			0.96	50
macro avg	0.98	0.89	0.93	50
weighted avg	0.96	0.96	0.96	50

```

0.96

```

**4.1 #การนำเข้าโมดูลต่างๆ**

```

import pandas as pd #data processing
import numpy as np #linear algebra
import matplotlib.pyplot as plt #data visualization
import seaborn as sns #statistical data visualization
import sklearn as sk #machine learning model
import sklearn.metrics as metrics

from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

```

**4.2 #อ่านไฟล์ Student.csv**

```
4.3 studentdata = pd.read_csv('/content/sample_data/Student.csv')
```

```
4.4 studentdata.head(5)
```

	StudentYear	HighSchoolGrade	Age	Familymember	Gender	CGPA	Address	CoreCourseScore	ElectiveCourseScore	Result
0	1	3.0	18	3.0	M	2.5	Central	1	1	0
1	1	3.0	18	5.0	F	2.3	Central	1	1	0
2	3	3.0	20	6.0	F	2.5	Central	1	1	0
3	1	3.0	18	2.0	F	2.4	Central	1	1	0
4	1	3.0	18	3.0	M	2.3	Southern	1	1	0

```
4.5 #การหาคุณลักษณะสำคัญ (feature selection)
```

```
4.6 all_features = [name for name in studentdata.columns if studentdata[name].dtype == 'object']
```

```
4.7 all_features
```

```
4.8 #แปลง gender และ address ให้เป็นตัวเลข
```

```
4.9 all_features=[name for name in studentdata.columns if studentdata[name].dtype == 'object']
```

```
4.10 from sklearn.preprocessing import LabelEncoder
```

```
4.11 le=LabelEncoder()
```

```
4.12 for i in list(all_features):
```

```
4.13     studentdata[i]=le.fit_transform(studentdata[i])
```

```
4.14 for x in all_features:
```

```
4.15     print(x,"=",studentdata[x].unique())
```

```
Gender = [1 0]
Address = [0 3 2 4 1]
```

```
4.16 #แสดงข้อมูลหลังเปลี่ยนเป็นตัวเลข
```

```
4.17 studentdata.head(5)
```

	StudentYear	HighSchoolGrade	Age	Familymember	Gender	CGPA	Address	CoreCourseScore	ElectiveCourseScore	Result
0	1	3.0	18	3.0	1	2.5	0	1	1	0
1	1	3.0	18	5.0	0	2.3	0	1	1	0
2	3	3.0	20	6.0	0	2.5	0	1	1	0
3	1	3.0	18	2.0	0	2.4	0	1	1	0
4	1	3.0	18	3.0	1	2.3	3	1	1	0

```
4.18 #หาค่า chi2
```

```
4.19 from sklearn.feature_selection import chi2
```

```
4.20 studentdata.fillna(0, inplace=True)
```

```
4.21 X = studentdata.drop('Result',axis=1)
```

```
4.22 y = studentdata['Result']
```

```
4.23 chi_scores = chi2(X,y)
```

```
4.24 chi_scores
```

```
4.25 #แสดงคุณลักษณะข้อมูลที่สัมพันธ์กับผลลัพธ์
```

```
4.26 p_values = pd.Series(chi_scores[1],index = X.columns)
```



4.27 `p_values.sort_values(ascending = True , inplace = True)`

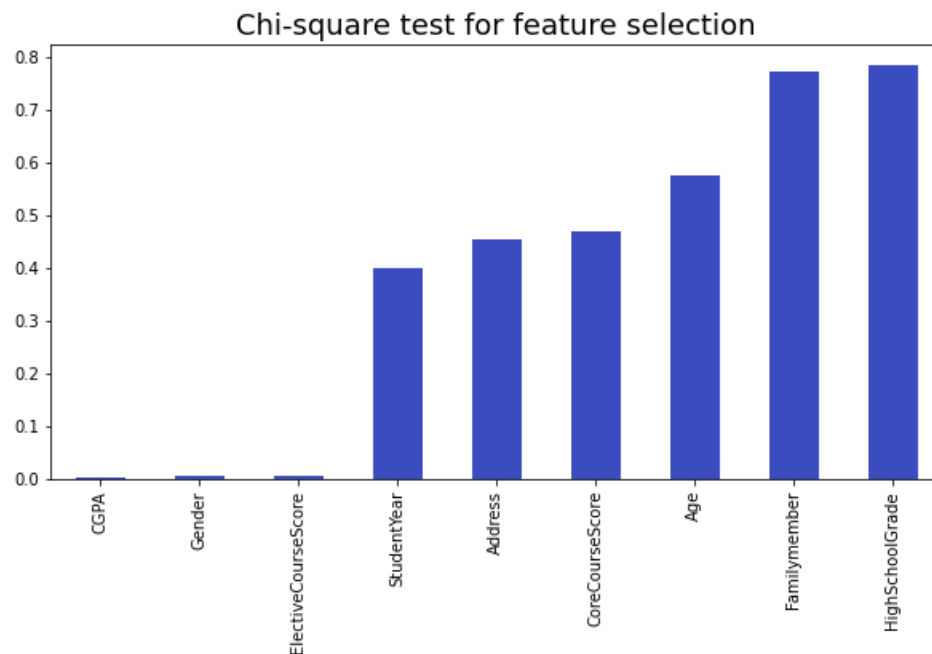
4.28 `p_values`

```
CGPA          0.001333
Gender        0.004478
ElectiveCourseScore  0.005175
StudentYear   0.398843
Address       0.453260
CoreCourseScore  0.469059
Age           0.574372
Familymember  0.773231
HighSchoolGrade 0.785089
dtype: float64
```

4.29 `#สร้างกราฟ#แสดงคุณลักษณะข้อมูลที่สัมพันธ์กับผลลัพธ์`

`p_values.plot.bar(figsize = (10,5), cmap="coolwarm")`

`plt.title('Chi-square test for feature selection', size=18)`



4.30 `#นำข้อมูลที่ไม่เกี่ยวข้องกับการสร้างแบบจำลองออก`

4.31 `newfeature = studentdata.columns.tolist()`

4.32 `newfeature.remove('CGPA')`

4.33 `newfeature.remove('Gender')`

4.34 `newfeature.remove('ElectiveCourseScore')`

4.35 `newfeature`

```
['StudentYear',
 'HighSchoolGrade',
 'Age',
 'Familymember',
 'Address',
 'CoreCourseScore',
 'Result']
```

#### 4.36 #สร้าง Algorithm ชื่อว่า Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

4.37 #การแบ่งชุดข้อมูลนักเรียนออกเป็น 80:20

4.38 X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1)

#### 4.39 #การสร้างแบบจำลองต้นไม้ตัดสินใจ

```
modelDT = DecisionTreeClassifier(random_state=43)
```

```
#การฝึกสอนข้อมูล
```

```
modelDT.fit(X_train, y_train)
```

```
#การทำนายข้อมูล
```

```
predictions = modelDT.predict(X_test)
```

#### 4.40 #การสร้างคอนฟิวชันเมทริกซ์สำหรับวัดประสิทธิภาพแบบจำลอง

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

```
print(confusion_matrix(y_test, predictions))
```

```
print(classification_report(y_test, predictions))
```

```
print(accuracy_score(y_test, predictions))
```

```
[[41  0]
 [ 2  7]]
```

	precision	recall	f1-score	support
0	0.95	1.00	0.98	41
1	1.00	0.78	0.88	9
accuracy			0.96	50
macro avg	0.98	0.89	0.93	50
weighted avg	0.96	0.96	0.96	50

```
0.96
```