

# Simple Linear Regression

Simple Linear Regression

$$y = b_0 + b_1 x_1$$

Dependent variable (DV)      Constant      Coefficient      Independent variable (IV)

In [ ]:

```
#import seaborn library เพื่อใช้ในการวาดภาพ
import seaborn as sns
iris = sns.load_dataset('iris') #อ่าน dataset ที่ชื่อ iris ซึ่งมีอยู่ใน seaborn library
iris.head() #แสดงข้อมูลส่วนหัวของข้อมูล iris
```

In [ ]:

```
%matplotlib inline
import seaborn as sns; sns.set()
sns.pairplot(iris, hue='species', height=1.5); #seaborn.pairplot => Plot pairwise relations
```

In [ ]:

```
X_iris = iris.drop('species', axis=1)
X_iris.shape
```

In [ ]:

```
y_iris = iris['species']
y_iris.shape
```

In [ ]:

```
import matplotlib.pyplot as plt
import numpy as np

rng = np.random.RandomState(42) #สร้างตัวเลขสุ่มโดยมีการกระจายค่าความน่าจะเป็น และกำหนด seed เป็น 42
x = 10 * rng.rand(50) #ทำการสุ่มค่า x จำนวน 50 ตัว
y = 2 * x - 1 + rng.randn(50) #ทำการสุ่มค่า y จำนวน 50 ตัว โดยที่ (2* x - 1) เป็นค่า constant ของสมการ
plt.scatter(x, y);
```

In [ ]:

```
from sklearn.linear_model import LinearRegression #เลือก model ชื่อ Linear Regression
model = LinearRegression(fit_intercept=True) #สร้าง model โดยมีการคำนวณจุดตัดแกน (intercept) ในสมการ
model #แสดงรายละเอียดของ model
```

In [ ]:

```
x #แสดงค่า x ที่ได้จากการสุ่ม
```

In [ ]:

```
x.shape
```

In [ ]:

```
y
```

In [ ]:

```
y.shape
```

In [ ]:

```
X = x[:, np.newaxis] #เปลี่ยนจาก row vector เป็น coloumn vector และเก็บไว้ที่ตัวแปร X  
X
```

In [ ]:

```
X.shape
```

In [ ]:

```
# features matrix อยู่ในรูปแบบของ [n_samples, n_features] x(50, 1)  
# target array อยู่ในรูปแบบของ [n_samples, n_targets] y(50,)  
model.fit(X, y)
```

In [ ]:

```
model.coef_ #แสดงค่าสัมประสิทธิ์ (coefficient) ของ model
```

In [ ]:

```
model.intercept_ #แสดงจุดตัดแกนของ model
```

In [ ]:

```
xfit = np.linspace(-1, 11) #สร้างอาร์เรย์ที่เป็นตัวเลขเรียงกัน โดยเริ่มต้นที่ -1 และสิ้นสุดที่ 11  
xfit
```

In [ ]:

```
#เปลี่ยนจาก row vector เป็น coloumn vector ให้ features matrix อยู่ในรูปแบบของ [n_samples, n_feat  
Xfit = xfit[:, np.newaxis]  
yfit = model.predict(Xfit) #ทำนายค่า y
```

In [ ]:

```
plt.scatter(x, y)
```

In [ ]:

```
plt.plot(xfit, yfit);
```

In [ ]:



```
plt.scatter(x, y)
plt.plot(xfit, yfit);
```