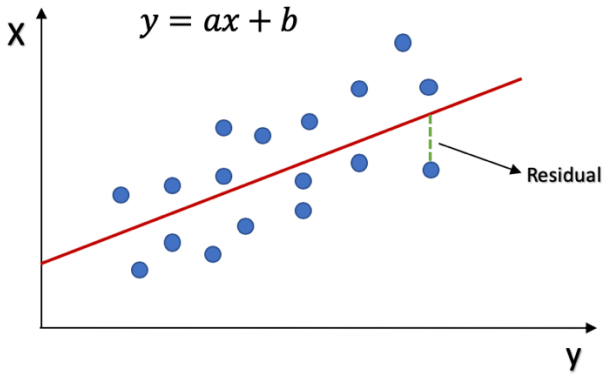
線性迴歸(Linear Regression)

線性回歸（Linear regression）是統計上在找多個自變數(independent variable)和依變數(dependent variable)之間的關係建出來的模型。只有一個自變數和一個依變數的情形稱為簡單線性回歸(Simple linear regression)，大於一個自變數的情形稱為多元回歸(multiple regression)。

簡單線性回歸: y=β0+β1x

β0：截距(Intercept)，β1：斜率(Slope)為 x 變動一個單位 y 變動的量，如下圖:



範例實作

線性回歸簡單來說，就是將複雜的資料數據，擬和至一條直線上，就能方便預測未來的資料。

先從簡單的線性回歸舉例，$y = ax + b$ ，a 稱為斜率，b 稱為截距。以下範例我們假設 `a=3 b=15`，
首先載入資料集，`numpy` 是 Python 套件中支援高階大量的維度陣列與矩陣運算的函式庫，此外也針對陣列運算提供大量的數學函式。另外 `matplotlib` 是專門做資料視覺化的函式庫，能夠有效的將各種資料使用不同方式呈現，例如直條圖、離散分佈圖...等

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

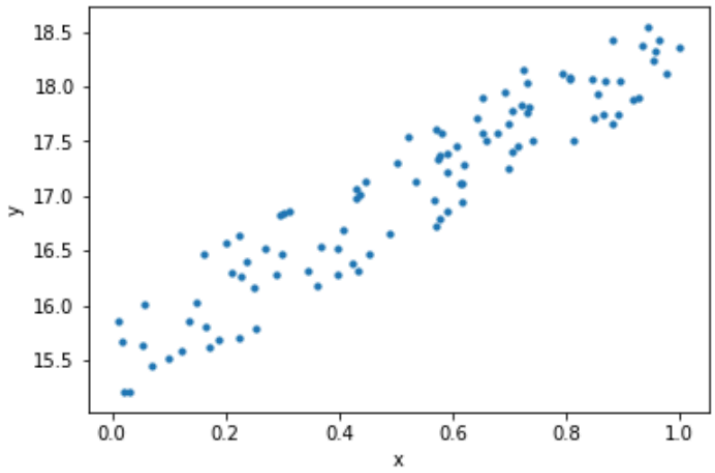
接著我們建立一個目標函式，這個函式是電腦要學習的目標基準，這邊範例就設定一個單線性的線條 y=ax+b ，其中 x 為 input; y 為 output; a 為 斜率，這邊設定 3(同學可以自由調整); b 為截距，為了要讓學習資料離散所以我們加上 noise 隨機數可以將數據分散。最後我們建立 100 個 0~1 之間的隨機數作為我們的輸入 x。

```
# generate random data-set
np.random.seed(0)
noise = np.random.rand(100, 1)
x = np.random.rand(100, 1)
y = 3 * x + 15 + noise
# y=ax+b Target function  a=3, b=15
```

數入和輸出都建立好後我們可以透過 matplotlib 將資料視覺化。

```
# plot
plt.scatter(x,y,s=10)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

透過視覺化可以觀察資料的分布狀態，我們可以看到數據點由左下至右上的規律。由於我們人眼一看就能看出來，但機器沒有眼睛他必須透過機器學習方式來找出資料間的規律性。



資料有了輸入和輸出(desire output)後我們就能機器學習了！這邊引入 sklearn 套件。sklearn 在 python 中提供大量常見的機器學習演算法和許多實用的資料集合，其中今天就是要使用到它的現性回歸的演算法(LinearRegression)。

.fit() 這個函式就是執行機器學習裡面分別放輸入和輸出，數據量過大的話通常這邊要等待一段時間。由於今天的範例比較簡單所以機器馬上就訓練好了。.predict() 函式就是透過訓練好的模型來預測輸出值，裡面的參數放輸入即可。

```
from sklearn.linear_model import LinearRegression
model = LinearRegression(fit_intercept=True)

model.fit(x, y)

pred = model.predict(x)

plt.scatter(x, y,s=10)
plt.plot(x, pred, color="r")
```