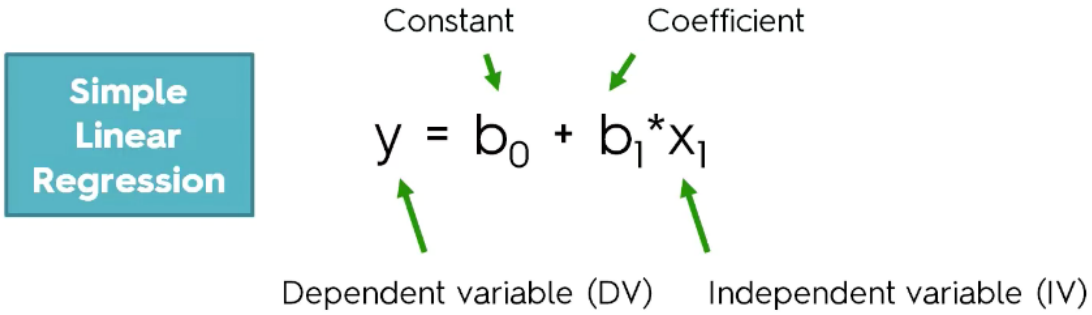
節 4 Simple Linear Regression

講座 19、20

	YearsExperience	Salary
2	1.1	39343
3	1.3	46205
4	1.5	37731
5	2	43525
6	2.2	39891
7	2.9	56642
8	3	60150
9	3.2	54445
10	3.2	64445
11	3.7	57189

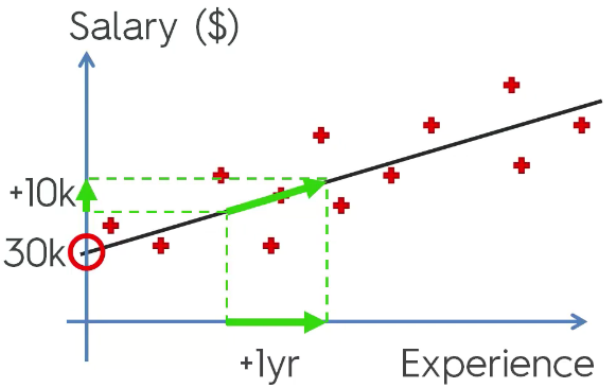
內含工作年資和薪資的資料，筆數 30 筆

講座 21、22



- ✧ 迴歸分析可做為評斷分類(變數間)的關聯程度
  eg. 薪資和年資的相關性
- ✧ 訪查、調研我們會知道年資越高薪資越高，但卻無法知道準確的量化程度
- ✧ 利用簡單線性迴歸模型找到那條具最佳解釋力的線性組合
  eg. 年資的增減如何影響薪資的結果
- ✧ 簡單線性迴歸模型
  y：相關變數、被解釋變數
  eg. 薪資 vs. 工作年資、考試分數 vs. 用功程度
  x1：獨立變數、解釋變數，內含假設 x 改變會影響 y 的變化
  順帶一提，當然 x 變數有可能根本不是影響 y 的直接因子
  (後面的課程會提到這個議題，變數的隱含關聯性)

Simple Linear Regression:



$y = b_0 + b_1 * x$

Salary = b_0 + b_1 *Experience

b1：x1 的係數、斜率項，指 x1 變動一單位影響 y 的變化(正負、大小)

b0：截距項，反應當 x1=0 時 y 的值

eg. 新鮮人剛入社會的薪資行情

節 23

```
# Importing the dataset
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
```

✧ 解釋變數 X：除了資料最後一欄，因為 X 只有一個，在此等同於[:, 0]

註：經討論，LinearRegression 要用 2D array，若使用 X = dataset.iloc[:, 0].values 產生的 1D array 會有 error，所以在這還是要用[:, :-1] or [:, 0:1]

另可用 shape 來看是幾維的 array，numpy.ndarray.shape(點它看說明)

✧ 被解釋變數 y：資料的第二欄，index 是 1

✧ .loc 用欄位來讀資料；.iloc 用位置來讀資料

eg. X = dataset.iloc[:, :-1].values vs. X = dataset.loc[:,'YearsExperience']

```
# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)
```

✧ 抽樣三分之一的資料作為驗證資料集，驗證資料集一般約占 20-30%

✧ random_state 則是設定固定的 seed，若想要跑出來的結果一樣可以設定相同 seed

節 24

```
# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

✧ 用普通最小平方法找到這條線，因為影響薪資的因素不僅有年資，可能也會因為學歷等影響，所以實際值和預測值(由這條線性組合預測得之)之間存在誤差，藉由最小化誤差平方的方式找

到最適的簡單迴歸模型

✧ LinearRegression()：class，其中的參數設定皆為選擇性，固可都用預設值執行
參數設定說明：
LinearRegression(fit_intercept=True, normalize=False, copy_X=True, n_jobs=1)
fit_intercept：計算截距；normalize：正規化；
copy_X：True，X_train 值不會被覆蓋； n_jobs：要開多少 CPU 來運算

✧ fit()：fit(X, y, sample_weight=None)
其中 y 為 Target values，就是被解釋變數 y 的意思

節 25

```
# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

✧ 用剛建好的模型對驗證資料集作預測，所以 X_test 要放入 regressor.predict()參數中

- predict()：用 predict 的 method 作成驗證資料集的預測 y

節 26

```
# Visualising the Training set results
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()

# Visualising the Test set results
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

✧ Training set
- 對於實際值和簡單線性迴歸模型所計算出來的線性組合(亦可稱為最佳配適線 best fitting line)可利用作圖了解模型的好壞
- plt.scatter()：作散布圖，實際年資和薪資的散布情形
- plt.plot()：畫最佳配適線，用 X_train 資料集和 X_train 帶入模型所預測的結果得此線
- plt.title()
設定圖名稱

- plt.xlabel
  設定 X 軸名稱
- plt.ylabel
  設定 y 軸名稱
- plt.show()
  印出結果
  註：可將其視為圖層堆疊的結束點，另外雖然在 spyder 不論有沒有下這項指令都會秀出圖，但其他編譯器則必須有 show()才會將圖 print
- 補充
  1. S=20：點的大小
  2. C='r'：顏色(b->blue、c->cyan、g->green.....)
  3. Marker='x'：註記型態
  4. linewidth='20'：線寬
  5. edgecolors='b'：點的外框顏色

✧ Test set

- plt.plot()：驗證資料同樣要和模型運算出的最佳配適線比較，所以其中參數和 Training set 相同