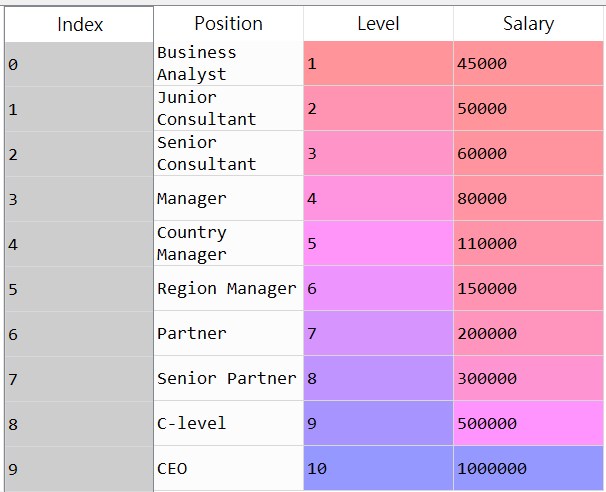
資料集：大公司的人資團隊，要雇用潛在新員工，在面試的過程中要給薪水，潛在員工說他有20年以上的經驗過去年薪160K，所以希望不低於原薪水，為了驗證真假，根據前公司提供他們職等的薪水，用他的年資，去判斷他說的年薪真假與否，並且知道他做過區域經理兩年並且知道平均四年可以往上升一階，所以大約在6.5的等級上



所以資料集有三個欄位，但我們只需要level和salary兩欄，

X =dataset.iloc[:,1].values 不行，如同前兩周提到的，在建立機器學習時X必須要視特徵矩陣，不能是vector，而Y則是要是vector

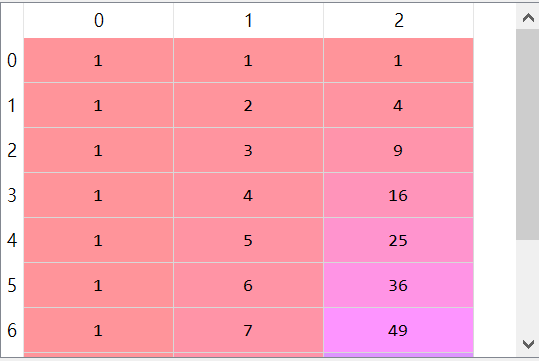
X =dataset.iloc[:,1:2].values 所以要改成，一樣是一欄，因為python是0開始，並且含下不含上

在這裡不做train set和test set 兩個原因，一個是資料筆數太少，二是我們需要很精確的預測結果，因為要用來談薪水，所以需要盡可能多的資訊。

為了比較一般線性和多項式回歸的差異，作者在這邊建立了兩個方程式，一個簡單線性回歸取名為lin\_reg，一個多項式回歸取名為poly\_reg，而多項式回歸在執行前要利用預處理的library將多項式的class PolynomialFeatures讀入，這個物件是用來轉換特徵矩陣去增加平方項、三次方等任何你想增加的，作者將原本的X矩陣轉換(一個型態的變數)為X。

PolynormialFeatures() 作者在這邊只設定一個主要的參數degree，用來設定你要幾次方的特徵矩陣，預設是2也就是平方。

然後再利用fit\_transform去產生轉換後的新矩陣

比較前後的差異

前面的是原先的員工等級，後面是利用多項式去產出的三個欄位，這裡可以看到原本的X會在中間代表X1，第三行是X1的平方項，第一行則是常數(截距項)

然後再利用先產生的X\_poly去執行複回歸

再來分別對兩個模型做視覺化呈現，對實際資料做散佈圖，然後兩個模型作配適線

簡單線性的就如同前面的一樣，看的出來配適的不是很好

而多項式回歸看起來表現比較好

分別那CEO 和等級6.5來比較，

CEO預測年薪90萬實際100萬

6.5，線性回歸預測為年薪30萬，多項式回歸預測為年薪20萬，而求職者說他在前公司有16萬，看起來他是很誠實的

如果我們希望可以再多項式回歸上表現更號，我們可以用dergee =3下去做。

在前段的地方不再凸起

將區間切為0.1一格，但看不出差異。