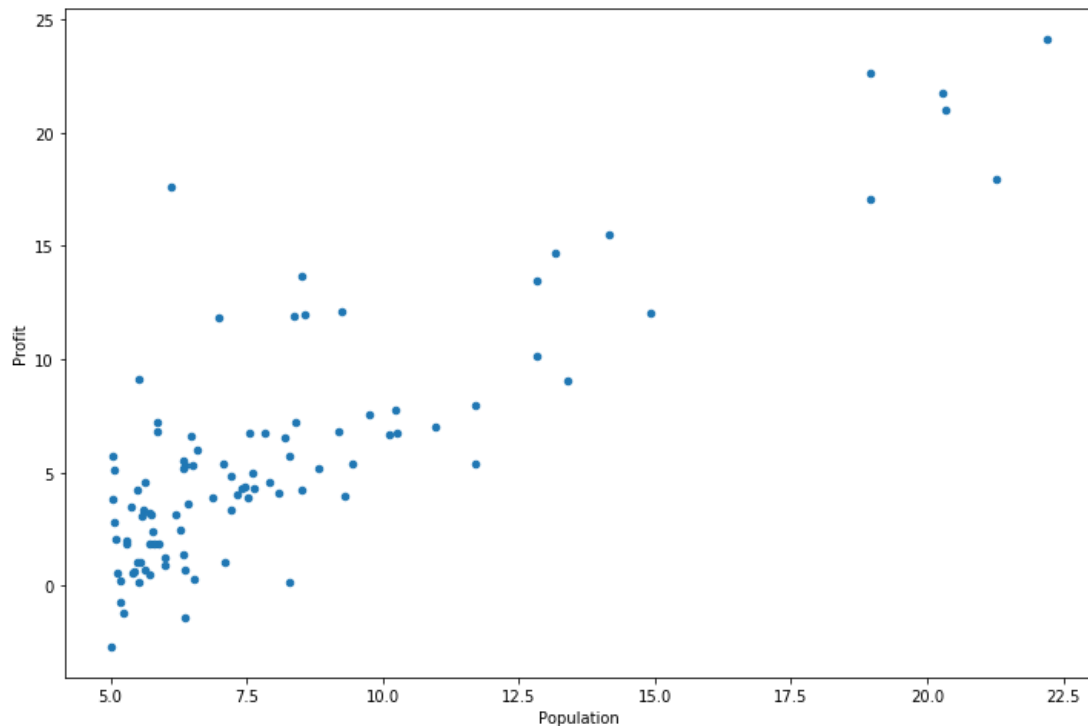


1.Linear regression with one variable

Exercise: Plotting the data



補充:

1.從 <https://pandas.pydata.org/> 舉例在 Pandas 套件下 Data Frame 本篇常見的命令: (pandas.DataFrame.xxx)

.describe()	描述統計,ex 平均值 中位數等等
.head()	顯示起始資料(預設 5 筆)
.tail()	顯示最後資料
.insert()	增加列或行標籤
.drop()	刪除列或行標籤

2. 用 read_csv 來讀取 ex1data1 的檔案

Exercise Gradient Descent

Cost function: $J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$

假設的 function: $h_{\theta}(x) = \theta^T x = \theta_0 + \theta_1 x_1$

Our model: $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$

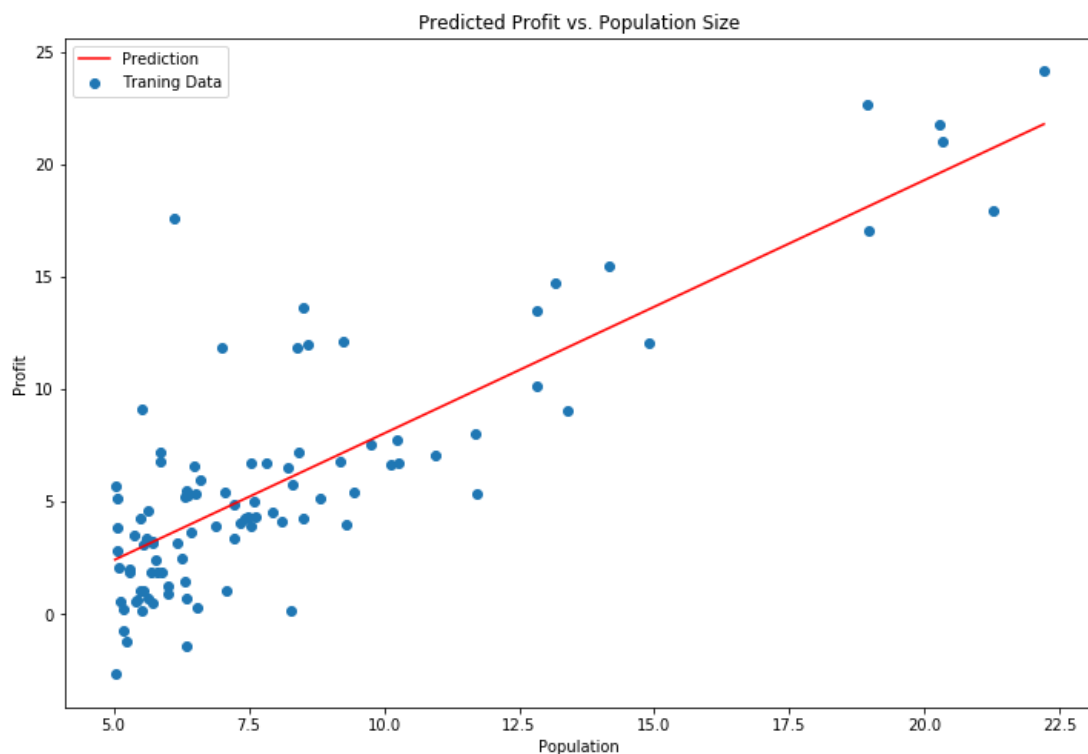
補充:

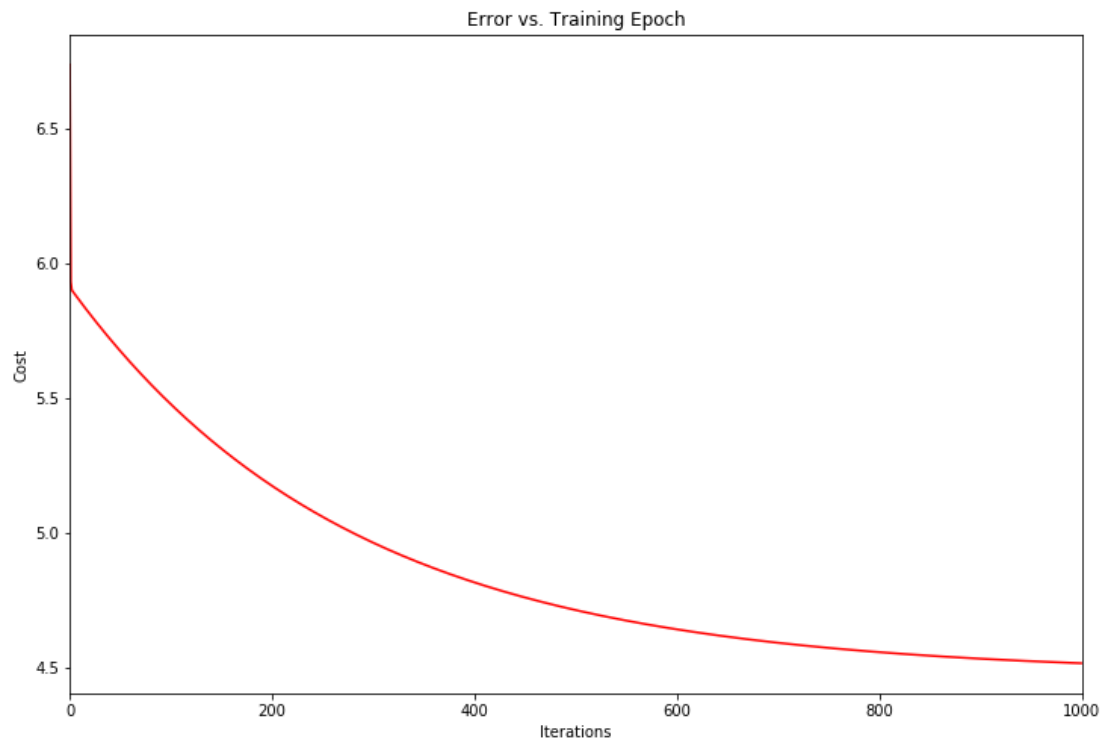
目標是使 CostFunction 最小,相當於人口乘上估計 Function 減去利潤的平方
在 `def computeCost` 下, 把 `costfunction` 表示出來.

在 `X` (training data) 下第一列增加一筆為 1 的向量,是之後的常數項

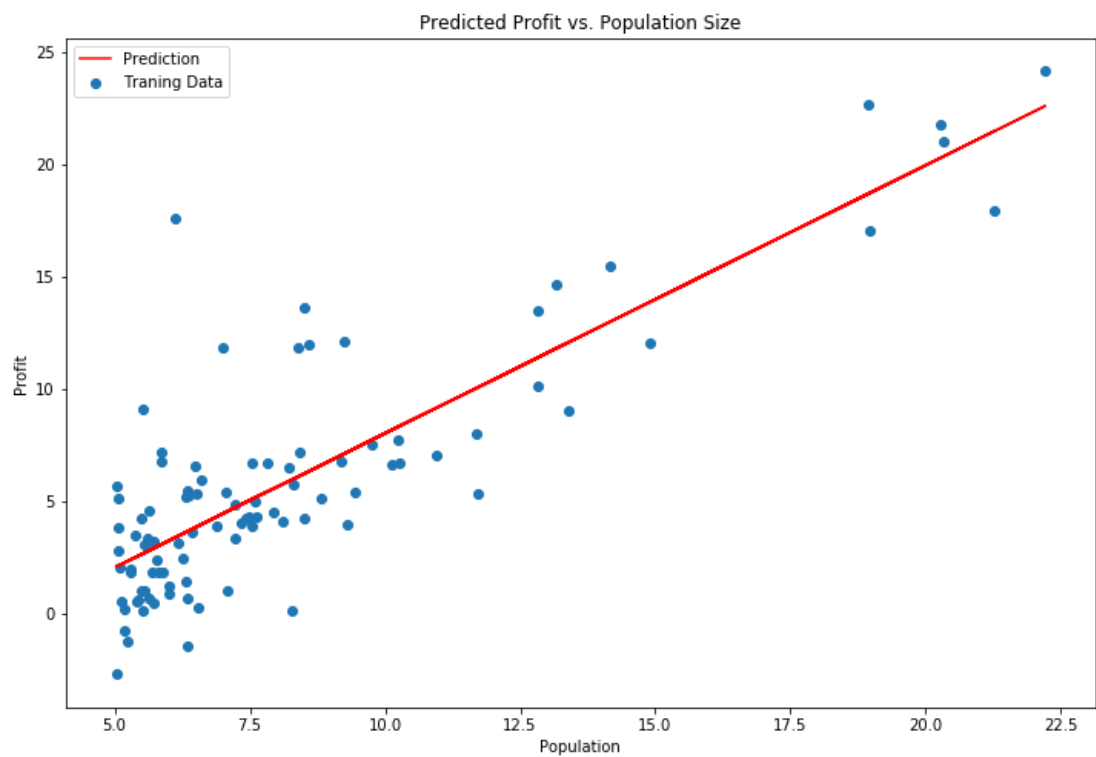
把 `X`, `y`, `theta` 轉為 `matrix` 來做計算,算出 `theta=0` 的初始值

在 `def gradientDescent` 下,回傳每次迭代的 `theta` 和 `cost`





Exercise: Implement scikit-learn's linear regression function



補充:

numpy.matrix.A1: 變為 flatten 的向量

sklearn.linear_model.LinearRegression:

LinearRegression 使用係數 $w = (w_1, \dots, w_p)$ 結合線性模型, 用最小二乘線性回歸方法觀察目標與通過線性近似預測的目標之間的平方餘數.

在方程式中找到 X, y 的最小平方和

做了 sklearn.linear_model.Lasso:

Lasso 也是 Linear regression family 的議員

不同點在於模型上沒有進行正則化 (權重懲罰)



Reference:

[1] <https://pandas.pydata.org/>

[2] <https://scikit-learn.org/>

[3] 常用屬性或方法 DataFrame:

<https://ithelp.ithome.com.tw/articles/10185922>

[4] What's the difference between Linear Regression, Lasso, Ridge, and ElasticNet in sklearn?

<https://towardsdatascience.com/whats-the-difference-between-linear-regression-lasso-ridge-and-elasticnet-8f997c60cf29>