EDA: Outlier 及處理

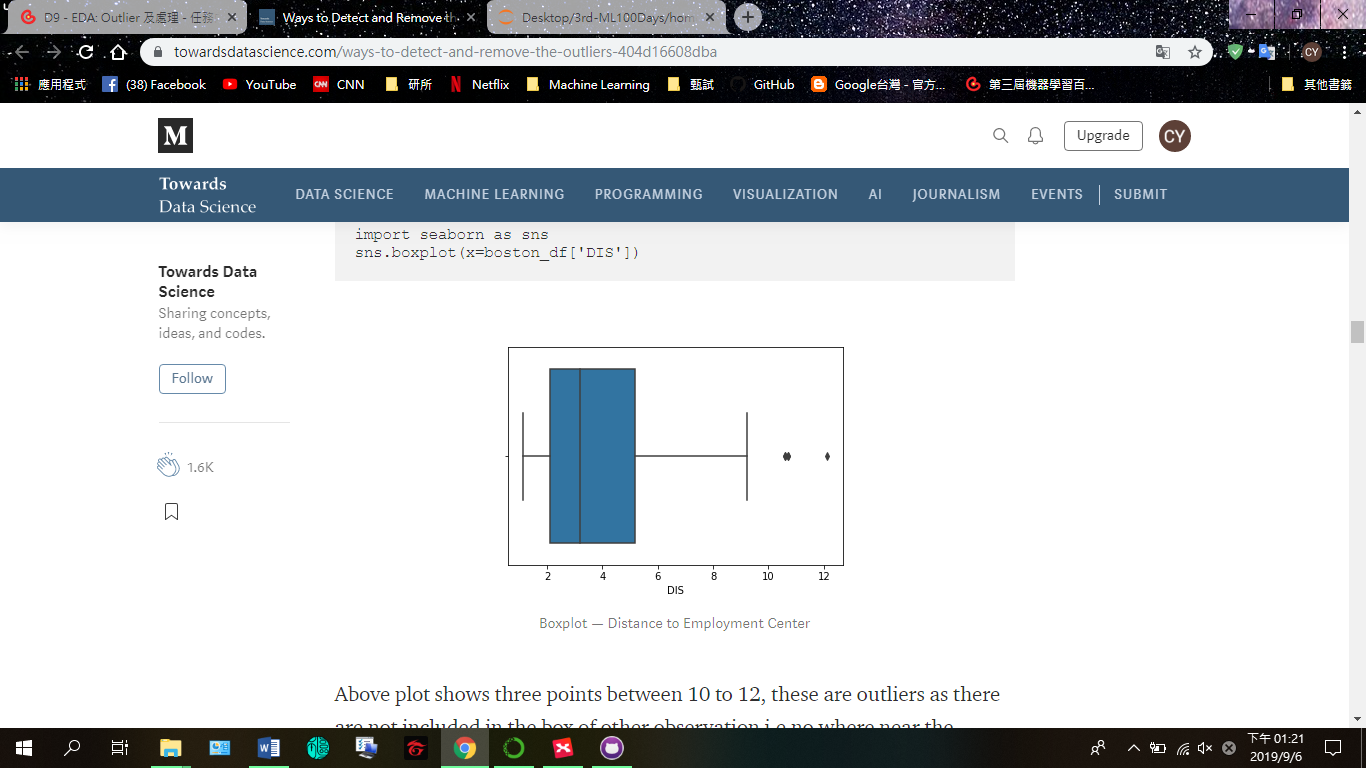
在開始用資料來作圖之前可以先處理資料，因為我們所得到的資料不一定完全正確，所以這是個必須的步驟，而這個步驟分成兩大部分 : 處理NA的格子、處理outlier。

* Outlier

Outlier有個兄弟我必需去分辨他們兩個，**outlier**產生的原因是因為”資料在紀錄上真的有錯誤”，但是有的時候是他的兄弟**varience**他的出現是因為在現實中就真的有那麼一筆特別跟其他數據不一樣的數據，而他們分辨的方法有很多以下就介紹兩種。

1. visualization methods
   1. **Box plot**

是最簡單的方法，就是利用**Box plot，然後觀察該圖，只要是在主體以外的部分就把該筆資料判別為outlier，也就是下圖中被紅框圍起來的部分。**



* **Scatter plot**

這個方法非常的直覺，就是把資料挑出兩個feature來作圖，觀察所有資料分布的情況，離群體最遠的部分就把他們當作outlier。

* **Z-Score**

這個方法的原理就是計算每個feature的統計數值，利用其平均數跟標準差來做一個評分的算式，而我們利用此方法得到的分數來作為判斷標準，每個點得到的分數只要是超出**-3~+3**的範圍我們就判定該筆資料為outlier，作法如下:

#得到各資料點的分數

from scipy import stats  
import numpy as npz = np.abs(stats.zscore(boston\_df))  
print(z)

#找出不再正常範圍內的資料點們

print(np.where(z > 3))

#去除不符合條件的資料點，同時建立一個array，來裝整理完的資料

#all的功能是對指定的條件做T/F判斷，在此指定條件就是(z < 3)

boston\_df\_o = boston\_df\_o[(z < 3).all(axis=1)]

* **IQR score**

這個方法是利用"四分位數"作為架構，以Q3-Q1的值作為可忍受範圍的大小，做法如下:

#得到所有feature的IQR

Q1 = boston\_df\_o1.quantile(0.25)  
Q3 = boston\_df\_o1.quantile(0.75)  
IQR = Q3 - Q1  
print(IQR)

#再以Q3和Q1作為基準點向外擴1.5\*IQR，如果不再該範圍內即認定為outlier

print(boston\_df\_o1 < (Q1 - 1.5 \* IQR)) |(boston\_df\_o1 > (Q3 + 1.5 \* IQR))

#移除不符合條件的資料點

boston\_df\_out = boston\_df\_o1[~((boston\_df\_o1 < (Q1 - 1.5 \* IQR)) |(boston\_df\_o1 > (Q3 + 1.5 \* IQR))).any(axis=1)]

boston\_df\_out.shape