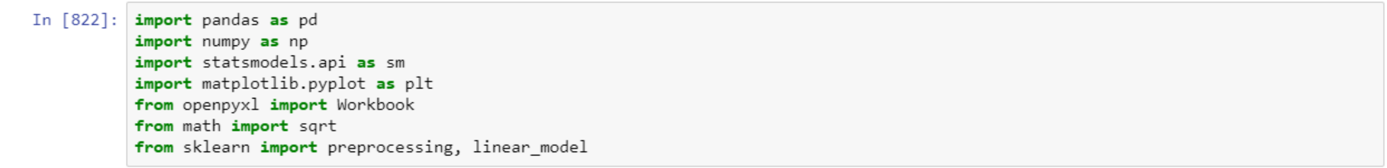
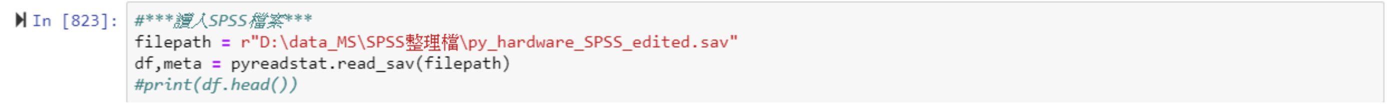
# python 與 SPSS 的確認流程圖

by 葉之晴

## 1. SPSS 整理變數類別後，利用 python 將全部的值讀入

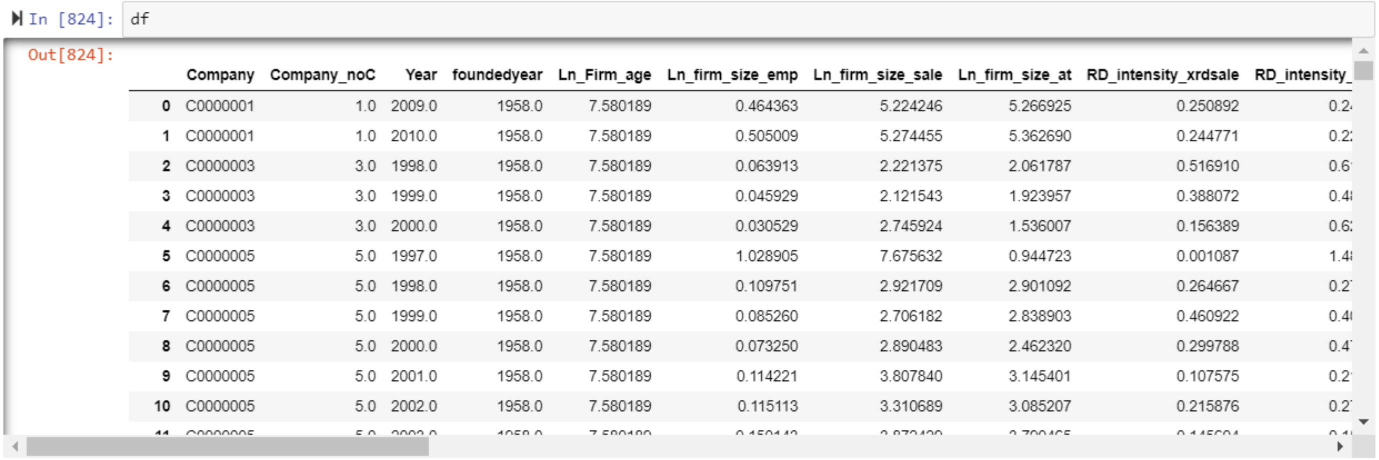


```
In [822]: import pandas as pd
          import numpy as np
          import statsmodels.api as sm
          import matplotlib.pyplot as plt
          from openpyxl import Workbook
          from math import sqrt
          from sklearn import preprocessing, linear_model
```



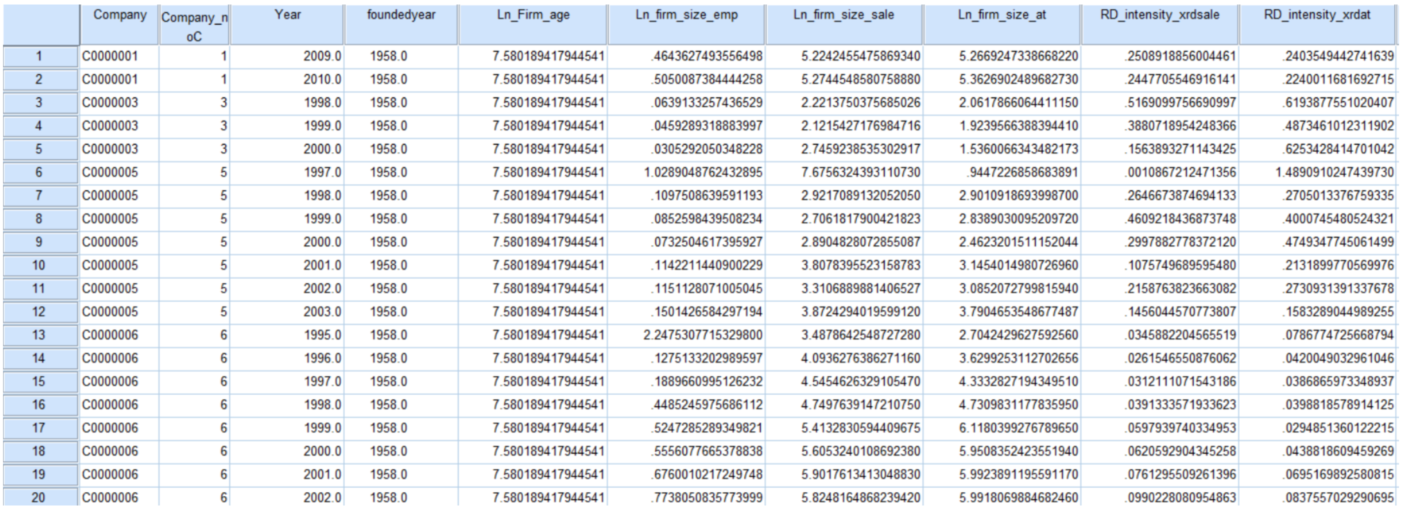
```
In [823]: #***讀入SPSS檔案***
          filepath = r"D:\data_MS\SPSS整理檔\py_hardware_SPSS_edited.sav"
          df,meta = pyreadstat.read_sav(filepath)
          #print(df.head())
```

# python output 總表結果



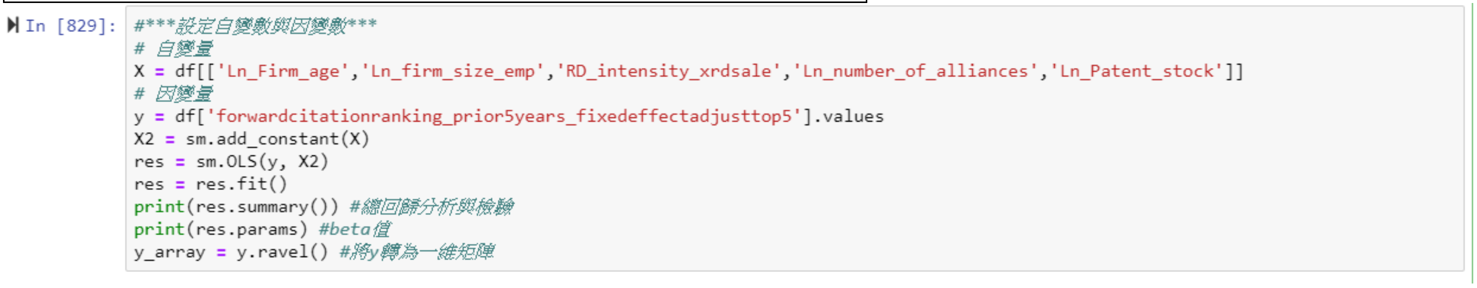
```
In [824]: df

Out[824]:
        Company  Company_noC    Year  foundedyear  Ln_Firm_age  Ln_firm_size_emp  Ln_firm_size_sale  Ln_firm_size_at  RD_intensity_xrdsale  RD_intensity_
    0  C0000001          1.0  2009.0       1958.0     7.580189          0.464363           5.224246         5.266925              0.250892           0.24
    1  C0000001          1.0  2010.0       1958.0     7.580189          0.505009           5.274455         5.362690              0.244771           0.22
    2  C0000003          3.0  1998.0       1958.0     7.580189          0.063913           2.221375         2.061787              0.516910           0.61
    3  C0000003          3.0  1999.0       1958.0     7.580189          0.045929           2.121543         1.923957              0.388072           0.48
    4  C0000003          3.0  2000.0       1958.0     7.580189          0.030529           2.745924         1.536007              0.156389           0.62
    5  C0000005          5.0  1997.0       1958.0     7.580189          1.028905           7.675632         0.944723              0.001087           1.48
    6  C0000005          5.0  1998.0       1958.0     7.580189          0.109751           2.921709         2.901092              0.264667           0.27
    7  C0000005          5.0  1999.0       1958.0     7.580189          0.085260           2.706182         2.838903              0.460922           0.40
    8  C0000005          5.0  2000.0       1958.0     7.580189          0.073250           2.890483         2.462320              0.299788           0.47
    9  C0000005          5.0  2001.0       1958.0     7.580189          0.114221           3.807840         3.145401              0.107575           0.21
   10  C0000005          5.0  2002.0       1958.0     7.580189          0.115113           3.310689         3.085207              0.215876           0.27
   11  C0000005          5.0  2003.0       1958.0     7.580189          0.150142           3.872429         3.790465              0.145604           0.1
```

# SPSS 截圖 給您確認

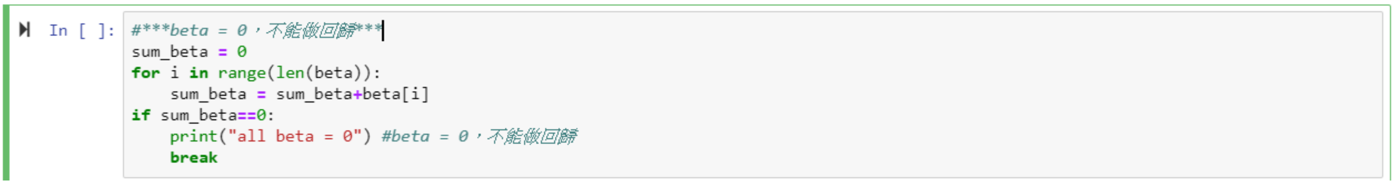


| | Company | Company_noC | Year | foundedyear | Ln_Firm_age | Ln_firm_size_emp | Ln_firm_size_sale | Ln_firm_size_at | RD_intensity_xrdsale | RD_intensity_xrdat |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | C0000001 | 1 | 2009.0 | 1958.0 | 7.580189417944541 | .4643627493556498 | 5.2242455475869340 | 5.2669247338668220 | .2508918856004461 | .2403549442741639 |
| 2 | C0000001 | 1 | 2010.0 | 1958.0 | 7.580189417944541 | .5050087384444258 | 5.2744548580758880 | 5.3626902489682730 | .2447705546916141 | .2240011681692715 |
| 3 | C0000003 | 3 | 1998.0 | 1958.0 | 7.580189417944541 | .0639133257436529 | 2.2213750375685026 | 2.0617866064411150 | .5169099756690997 | .6193877551020407 |
| 4 | C0000003 | 3 | 1999.0 | 1958.0 | 7.580189417944541 | .0459289318883997 | 2.1215427176984716 | 1.9239566388394410 | .3880718954248366 | .4873461012311902 |
| 5 | C0000003 | 3 | 2000.0 | 1958.0 | 7.580189417944541 | .0305292050348228 | 2.7459238535302917 | 1.5360066343482173 | .1563893271143425 | .6253428414701042 |
| 6 | C0000005 | 5 | 1997.0 | 1958.0 | 7.580189417944541 | 1.0289048762432895 | 7.6756324393110730 | .9447226858683891 | .0010867212471356 | 1.4890910247439730 |
| 7 | C0000005 | 5 | 1998.0 | 1958.0 | 7.580189417944541 | .1097508639591193 | 2.9217089132052050 | 2.9010918693998700 | .2646673874694133 | .2705013376759335 |
| 8 | C0000005 | 5 | 1999.0 | 1958.0 | 7.580189417944541 | .0852598439508234 | 2.7061817900421823 | 2.8389030095209720 | .4609218436873748 | .4000745480524321 |
| 9 | C0000005 | 5 | 2000.0 | 1958.0 | 7.580189417944541 | .0732504617395927 | 2.8904828072855087 | 2.4623201511152044 | .2997882778372120 | .4749347745061499 |
| 10 | C0000005 | 5 | 2001.0 | 1958.0 | 7.580189417944541 | .1142211440900229 | 3.8078395523158783 | 3.1454014980726960 | .1075749689595480 | .2131899770569976 |
| 11 | C0000005 | 5 | 2002.0 | 1958.0 | 7.580189417944541 | .1151128071005045 | 3.3106889881406527 | 3.0852072799815940 | .2158763823663082 | .2730931391337678 |
| 12 | C0000005 | 5 | 2003.0 | 1958.0 | 7.580189417944541 | .1501426584297194 | 3.8724294019599120 | 3.7904653548677487 | .1456044570773807 | .1583289044989255 |
| 13 | C0000006 | 6 | 1995.0 | 1958.0 | 7.580189417944541 | 2.2475307715329800 | 3.4878642548727280 | 2.7042429627592560 | .0345882204565519 | .0786774725668794 |
| 14 | C0000006 | 6 | 1996.0 | 1958.0 | 7.580189417944541 | .1275133202989597 | 4.0936276386271160 | 3.6299253112702656 | .0261546550876062 | .0420049032961046 |
| 15 | C0000006 | 6 | 1997.0 | 1958.0 | 7.580189417944541 | .1889660995126232 | 4.5454626329105470 | 4.3332827194349510 | .0312111071543186 | .0386865973348937 |
| 16 | C0000006 | 6 | 1998.0 | 1958.0 | 7.580189417944541 | .4485245975686112 | 4.7497639147210750 | 4.7309831177835950 | .0391333571933623 | .0398818578914125 |
| 17 | C0000006 | 6 | 1999.0 | 1958.0 | 7.580189417944541 | .5247285289349821 | 5.4132830594409675 | 6.1180399276789650 | .0597939740334953 | .0294851360122215 |
| 18 | C0000006 | 6 | 2000.0 | 1958.0 | 7.580189417944541 | .5556077665378838 | 5.6053240108692380 | 5.9508352423551940 | .0620592904345258 | .0438818609459269 |
| 19 | C0000006 | 6 | 2001.0 | 1958.0 | 7.580189417944541 | .6760010217249748 | 5.9017613413048830 | 5.9923891195591170 | .0761295509261396 | .0695169892580815 |
| 20 | C0000006 | 6 | 2002.0 | 1958.0 | 7.580189417944541 | .7738050835773999 | 5.8248164868239420 | 5.9918069884682460 | .0990228080954863 | .0837557029290695 |

## 2. 跑入統計函式庫 (如果全部係數都為 0，則不可以做回歸)

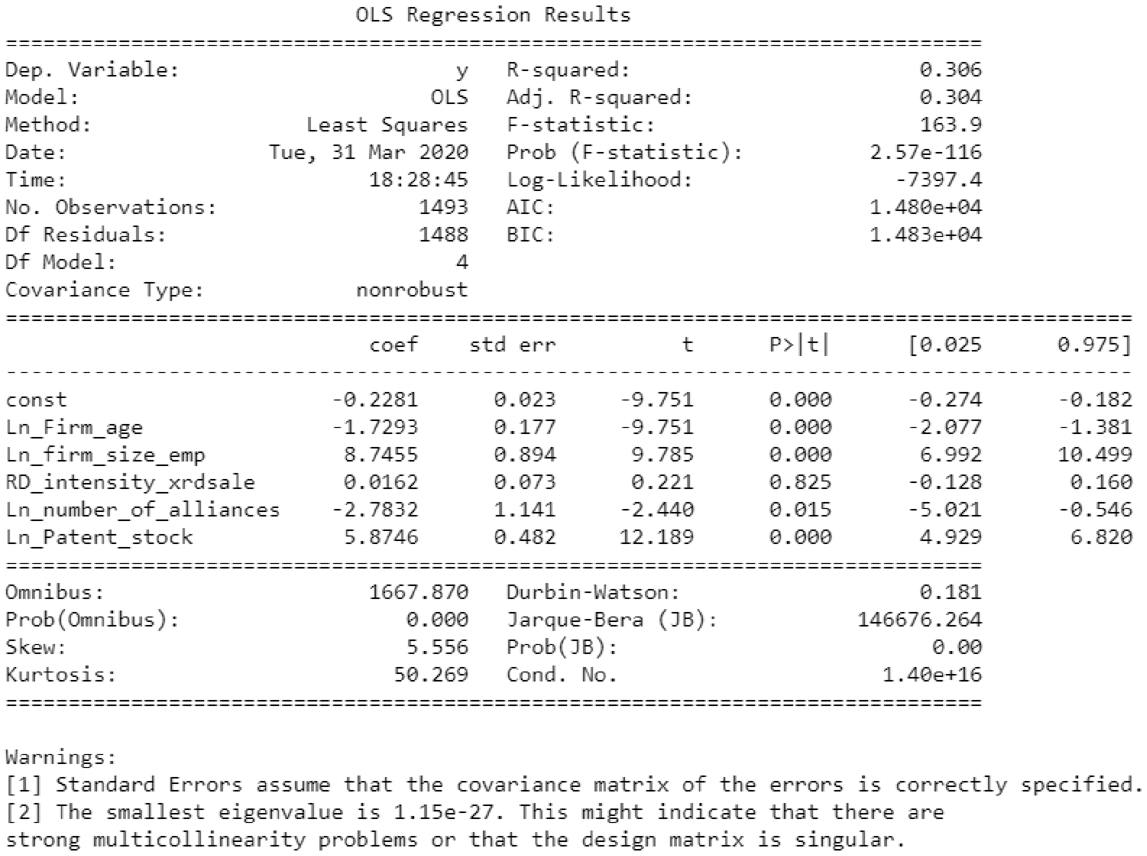


```
In [829]: #***設定自變數與因變數***
          # 自變量
          X = df[['Ln_Firm_age','Ln_firm_size_emp','RD_intensity_xrdsale','Ln_number_of_alliances','Ln_Patent_stock']]
          # 因變量
          y = df['forwardcitationranking_prior5years_fixedeffectadjusttop5'].values
          X2 = sm.add_constant(X)
          res = sm.OLS(y, X2)
          res = res.fit()
          print(res.summary()) #總回歸分析與檢驗
          print(res.params) #beta值
          y_array = y.ravel() #將y轉為一維矩陣
```



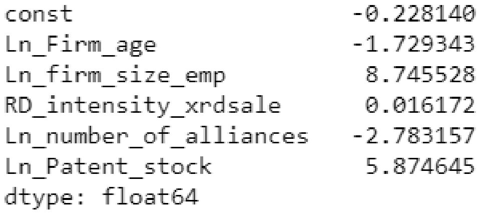
```
In [ ]: #***beta = 0，不能做回歸***
        sum_beta = 0
        for i in range(len(beta)):
            sum_beta = sum_beta+beta[i]
        if sum_beta==0:
            print("all beta = 0") #beta = 0，不能做回歸
            break
```

# python 的回歸結果



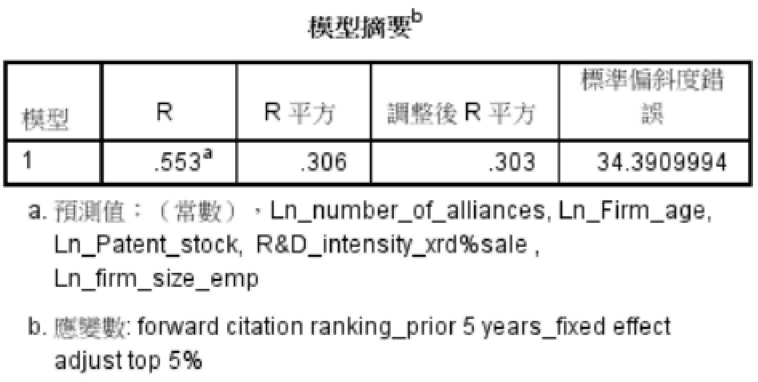
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.306
Model:                            OLS   Adj. R-squared:                  0.304
Method:                 Least Squares   F-statistic:                     163.9
Date:                Tue, 31 Mar 2020   Prob (F-statistic):          2.57e-116
Time:                        18:28:45   Log-Likelihood:                -7397.4
No. Observations:                1493   AIC:                         1.480e+04
Df Residuals:                    1488   BIC:                         1.483e+04
Df Model:                           4
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                     -0.2281      0.023     -9.751      0.000      -0.274      -0.182
Ln_Firm_age               -1.7293      0.177     -9.751      0.000      -2.077      -1.381
Ln_firm_size_emp           8.7455      0.894      9.785      0.000       6.992      10.499
RD_intensity_xrdsale       0.0162      0.073      0.221      0.825      -0.128       0.160
Ln_number_of_alliances    -2.7832      1.141     -2.440      0.015      -5.021      -0.546
Ln_Patent_stock            5.8746      0.482     12.189      0.000       4.929       6.820
==============================================================================
Omnibus:                     1667.870   Durbin-Watson:                   0.181
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           146676.264
Skew:                           5.556   Prob(JB):                         0.00
Kurtosis:                      50.269   Cond. No.                     1.40e+16
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.15e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

# python 的 beta 係數結果



```
const                    -0.228140
Ln_Firm_age              -1.729343
Ln_firm_size_emp          8.745528
RD_intensity_xrdsale      0.016172
Ln_number_of_alliances   -2.783157
Ln_Patent_stock           5.874645
dtype: float64
```

# SPSS 跑完回歸的結果

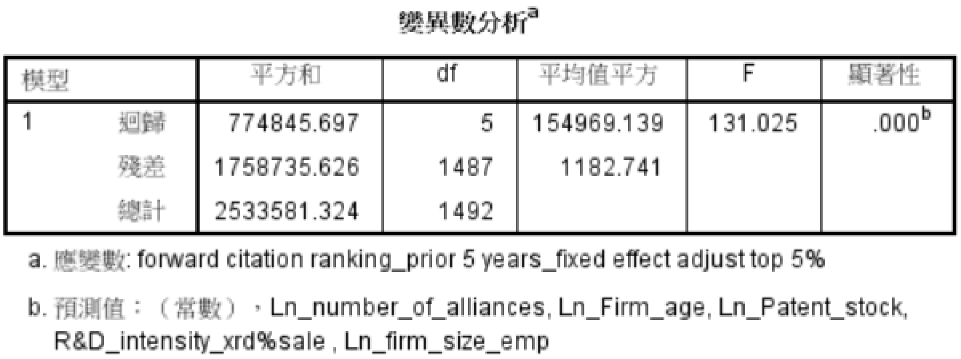


模型摘要[b]

| 模型 | R | R 平方 | 調整後 R 平方 | 標準偏斜度錯誤 |
|---|---|---|---|---|
| 1 | .553[a] | .306 | .303 | 34.3909994 |

a. 預測值：（常數），Ln_number_of_alliances, Ln_Firm_age, Ln_Patent_stock, R&D_intensity_xrd%sale , Ln_firm_size_emp

b. 應變數: forward citation ranking_prior 5 years_fixed effect adjust top 5%

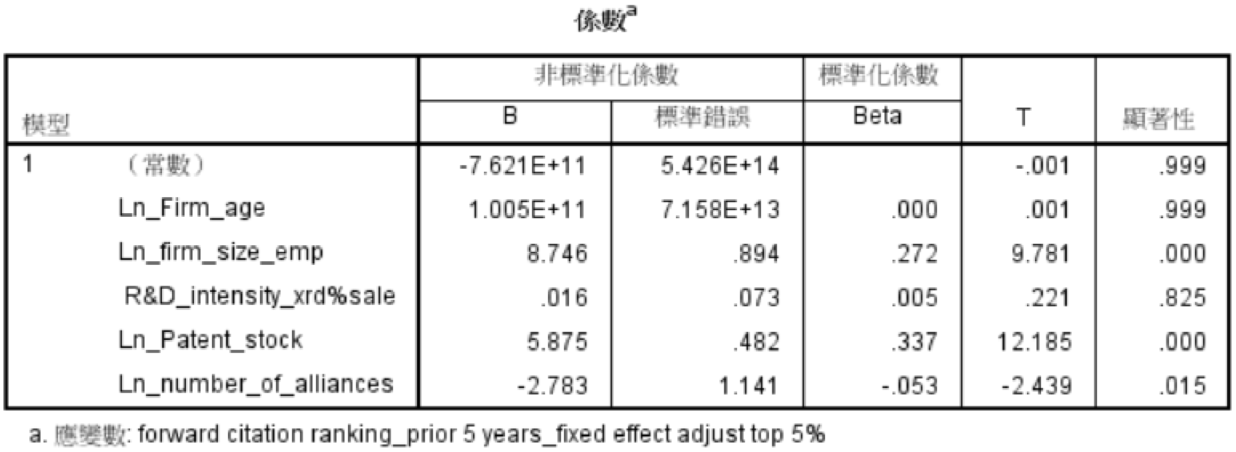


變異數分析[a]

| 模型 | | 平方和 | df | 平均值平方 | F | 顯著性 |
|---|---|---|---|---|---|---|
| 1 | 迴歸 | 774845.697 | 5 | 154969.139 | 131.025 | .000[b] |
| | 殘差 | 1758735.626 | 1487 | 1182.741 | | |
| | 總計 | 2533581.324 | 1492 | | | |

a. 應變數: forward citation ranking_prior 5 years_fixed effect adjust top 5%

b. 預測值：（常數），Ln_number_of_alliances, Ln_Firm_age, Ln_Patent_stock, R&D_intensity_xrd%sale , Ln_firm_size_emp



係數[a]

| 模型 | | 非標準化係數 | | 標準化係數 | T | 顯著性 |
|---|---|---|---|---|---|---|
| | | B | 標準錯誤 | Beta | | |
| 1 | （常數） | -7.621E+11 | 5.426E+14 | | -.001 | .999 |
| | Ln_Firm_age | 1.005E+11 | 7.158E+13 | .000 | .001 | .999 |
| | Ln_firm_size_emp | 8.746 | .894 | .272 | 9.781 | .000 |
| | R&D_intensity_xrd%sale | .016 | .073 | .005 | .221 | .825 |
| | Ln_Patent_stock | 5.875 | .482 | .337 | 12.185 | .000 |
| | Ln_number_of_alliances | -2.783 | 1.141 | -.053 | -2.439 | .015 |

a. 應變數: forward citation ranking_prior 5 years_fixed effect adjust top 5%

## 3. 做標準殘差的計算

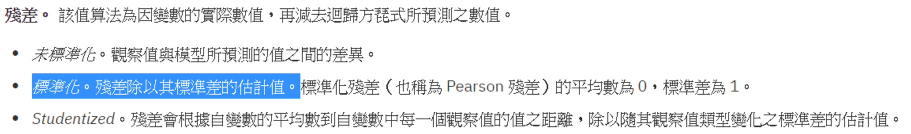
#標準殘差 ＝ 殘差/標準化估計值 (根據 SPSS 官網)

#殘差 ＝ 估計值-原值

#標準差估計值 ＝ ((Σ(y-y')^2)/n)^1/2

# SPSS 官網算法說明 （「標準差的估計值」我是用手算比對數字，推出公式的)

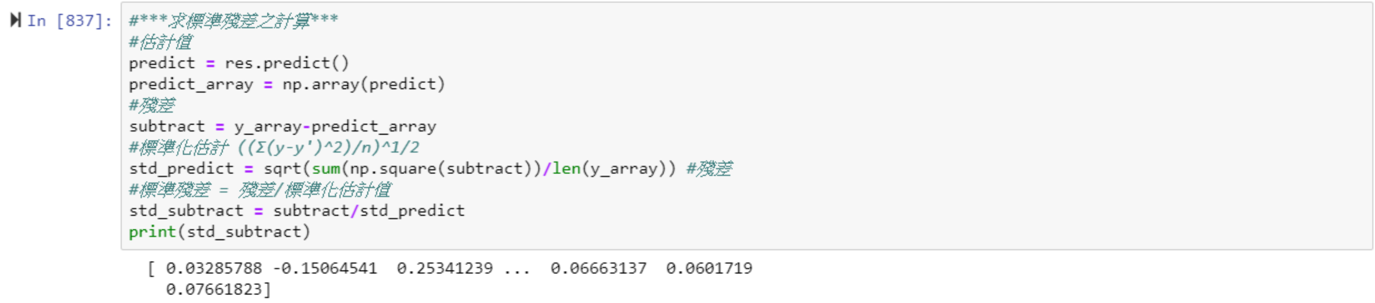
https://www.ibm.com/support/knowledgecenter/zh-tw/SSLVMB_sub/statistics_mainhelp_ddita/spss/base/idh_regr_sav.html



殘差。 該值算法為因變數的實際數值，再減去迴歸方程式所預測之數值。

- *未標準化*。觀察值與模型所預測的值之間的差異。
- *標準化*。殘差除以其標準差的估計值。標準化殘差（也稱為 Pearson 殘差）的平均數為 0，標準差為 1。
- *Studentized*。殘差會根據自變數的平均數到自變數中每一個觀察值的值之距離，除以隨其觀察值類型變化之標準差的估計值。

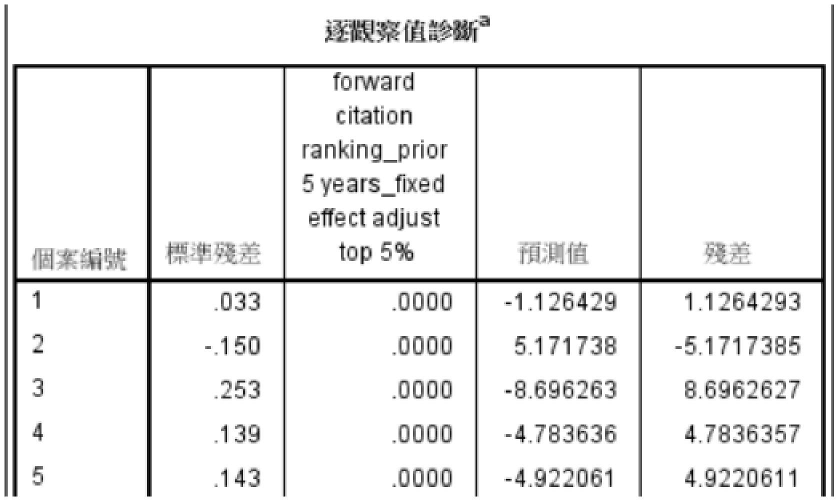
#python 各項標準殘差的結果



```
In [837]:
#***求標準殘差之計算***
#估計值
predict = res.predict()
predict_array = np.array(predict)
#殘差
subtract = y_array-predict_array
#標準化估計 ((Σ(y-y')^2)/n)^1/2
std_predict = sqrt(sum(np.square(subtract))/len(y_array)) #殘差
#標準殘差 = 殘差/標準化估計值
std_subtract = subtract/std_predict
print(std_subtract)
```

```
[ 0.03285788 -0.15064541  0.25341239 ...  0.06663137  0.0601719
  0.07661823]
```

#SPSS 各項標準殘差結果 (跟上面 python 矩陣裡的數字一樣)



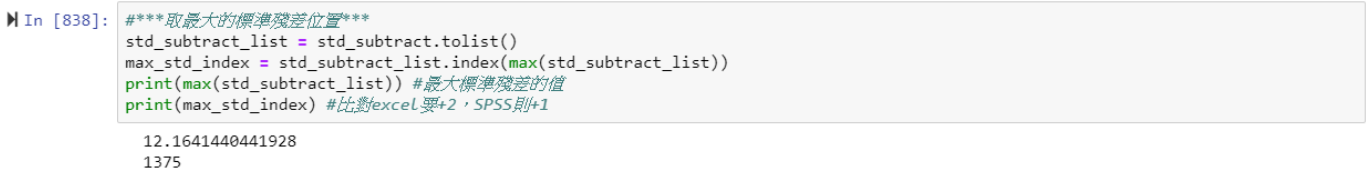
逐觀察值診斷[a]

| 個案編號 | 標準殘差 | forward citation ranking_prior 5 years_fixed effect adjust top 5% | 預測值 | 殘差 |
|---|---|---|---|---|
| 1 | .033 | .0000 | -1.126429 | 1.1264293 |
| 2 | -.150 | .0000 | 5.171738 | -5.1717385 |
| 3 | .253 | .0000 | -8.696263 | 8.6962627 |
| 4 | .139 | .0000 | -4.783636 | 4.7836357 |
| 5 | .143 | .0000 | -4.922061 | 4.9220611 |

## 4. 取標準殘差的最大值在哪一行

(對照 excel 要+2，因為程式中從 0 開始數且沒有標頭；對照 SPSS 則+1，因為 SPSS 本身標頭不算一數。)

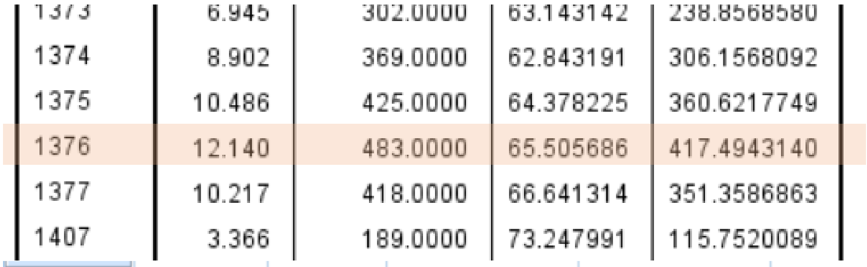
## output 第一次的答案為 1375，編號 379 公司 2015 的值。



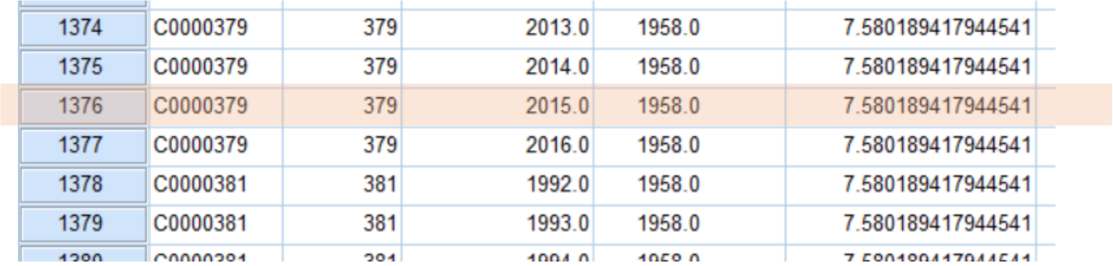
```
In [838]:
#***取最大的標準殘差位置***
std_subtract_list = std_subtract.tolist()
max_std_index = std_subtract_list.index(max(std_subtract_list))
print(max(std_subtract_list)) #最大標準殘差的值
print(max_std_index) #比對excel要+2，SPSS則+1
```

```
12.1641440441928
1375
```

##附上 SPSS 比對圖 (1375+1)

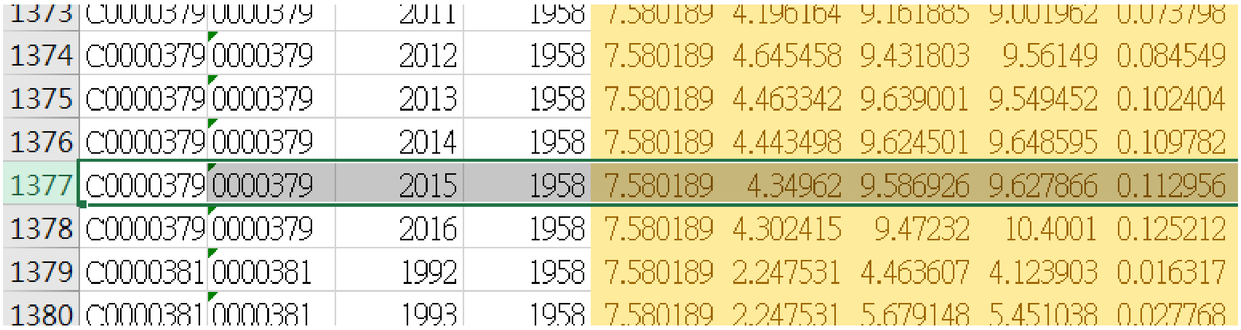


| | | | | |
|---|---|---|---|---|
| 1373 | 6.945 | 302.0000 | 63.143142 | 238.8568580 |
| 1374 | 8.902 | 369.0000 | 62.843191 | 306.1568092 |
| 1375 | 10.486 | 425.0000 | 64.378225 | 360.6217749 |
| 1376 | 12.140 | 483.0000 | 65.505686 | 417.4943140 |
| 1377 | 10.217 | 418.0000 | 66.641314 | 351.3586863 |
| 1407 | 3.366 | 189.0000 | 73.247991 | 115.7520089 |



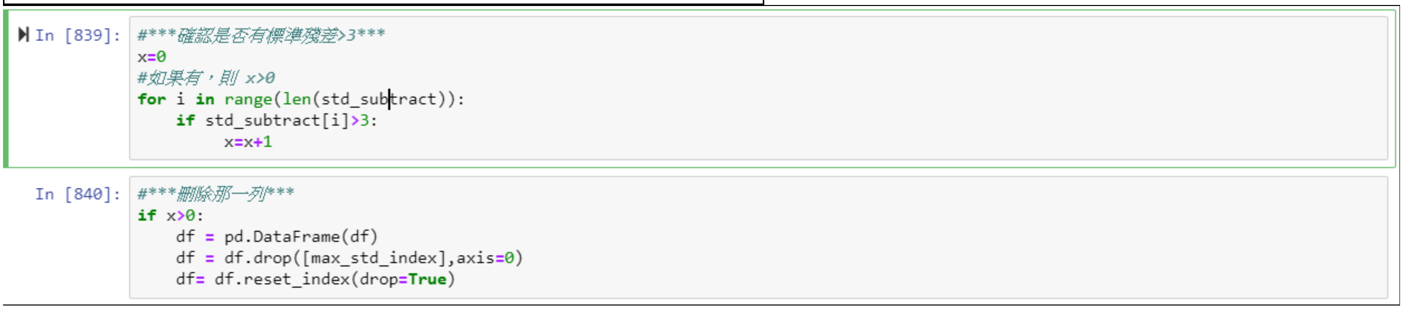
| | | | | | |
|---|---|---|---|---|---|
| 1374 | C0000379 | 379 | 2013.0 | 1958.0 | 7.580189417944541 |
| 1375 | C0000379 | 379 | 2014.0 | 1958.0 | 7.580189417944541 |
| 1376 | C0000379 | 379 | 2015.0 | 1958.0 | 7.580189417944541 |
| 1377 | C0000379 | 379 | 2016.0 | 1958.0 | 7.580189417944541 |
| 1378 | C0000381 | 381 | 1992.0 | 1958.0 | 7.580189417944541 |
| 1379 | C0000381 | 381 | 1993.0 | 1958.0 | 7.580189417944541 |
| 1380 | C0000381 | 381 | 1994.0 | 1958.0 | 7.580189417944541 |

##附上 excel 比對圖 (1375+2)



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1373 | C0000379 | 0000379 | 2011 | 1958 | 7.580189 | 4.196164 | 9.161885 | 9.001962 | 0.073798 |
| 1374 | C0000379 | 0000379 | 2012 | 1958 | 7.580189 | 4.645458 | 9.431803 | 9.56149 | 0.084549 |
| 1375 | C0000379 | 0000379 | 2013 | 1958 | 7.580189 | 4.463342 | 9.639001 | 9.549452 | 0.102404 |
| 1376 | C0000379 | 0000379 | 2014 | 1958 | 7.580189 | 4.443498 | 9.624501 | 9.648595 | 0.109782 |
| 1377 | C0000379 | 0000379 | 2015 | 1958 | 7.580189 | 4.34962 | 9.586926 | 9.627866 | 0.112956 |
| 1378 | C0000379 | 0000379 | 2016 | 1958 | 7.580189 | 4.302415 | 9.47232 | 10.4001 | 0.125212 |
| 1379 | C0000381 | 0000381 | 1992 | 1958 | 7.580189 | 2.247531 | 4.463607 | 4.123903 | 0.016317 |
| 1380 | C0000381 | 0000381 | 1993 | 1958 | 7.580189 | 2.247531 | 5.679148 | 5.451038 | 0.027768 |

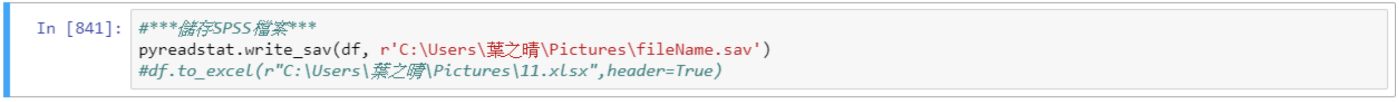
## 5. 判斷是否有標準殘差>3，如果有就刪除最大的那列



```
In [839]: #***確認是否有標準殘差>3***
          x=0
          #如果有，則 x>0
          for i in range(len(std_subtract)):
              if std_subtract[i]>3:
                  x=x+1
```

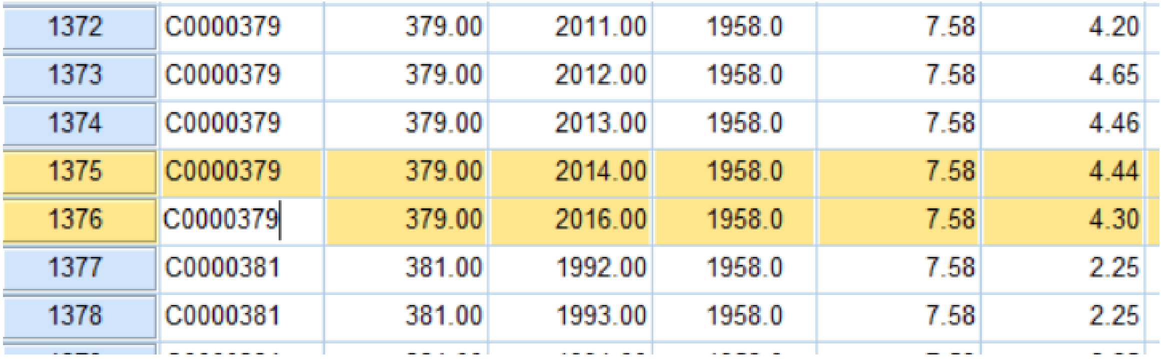
```
In [840]: #***刪除那一列***
          if x>0:
              df = pd.DataFrame(df)
              df = df.drop([max_std_index],axis=0)
              df= df.reset_index(drop=True)
```

## 6. 儲存成新的 SPSS 檔案



```
In [841]: #***儲存SPSS檔案***
          pyreadstat.write_sav(df, r'C:\Users\葉之晴\Pictures\fileName.sav')
          #df.to_excel(r"C:\Users\葉之晴\Pictures\11.xlsx",header=True)
```

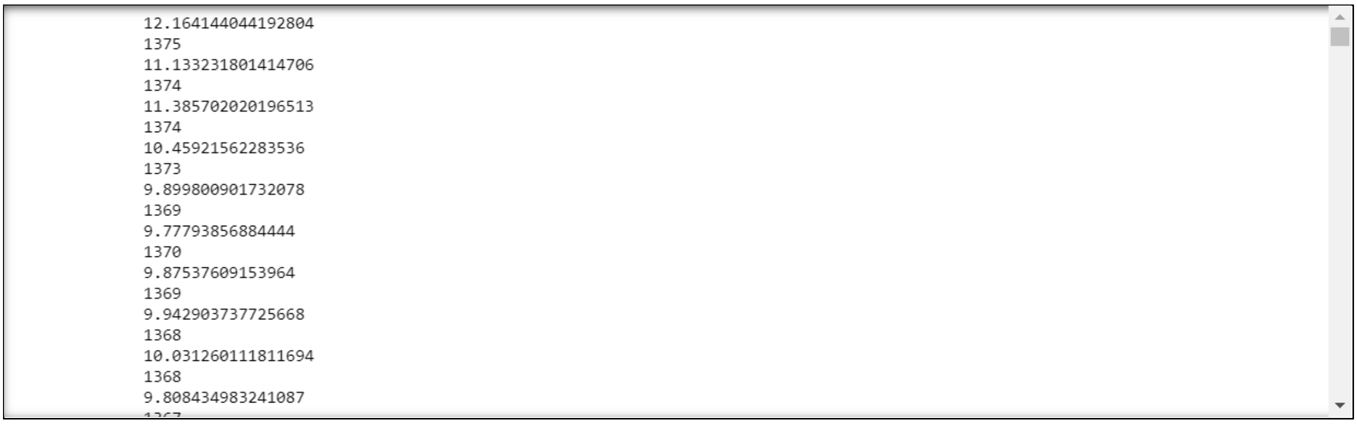
##新的 excel 截圖 (公司編號 379，2015 年已刪除)



| | | | | | | |
|---|---|---|---|---|---|---|
| 1372 | C0000379 | 379.00 | 2011.00 | 1958.0 | 7.58 | 4.20 |
| 1373 | C0000379 | 379.00 | 2012.00 | 1958.0 | 7.58 | 4.65 |
| 1374 | C0000379 | 379.00 | 2013.00 | 1958.0 | 7.58 | 4.46 |
| 1375 | C0000379 | 379.00 | 2014.00 | 1958.0 | 7.58 | 4.44 |
| 1376 | C0000379 | 379.00 | 2016.00 | 1958.0 | 7.58 | 4.30 |
| 1377 | C0000381 | 381.00 | 1992.00 | 1958.0 | 7.58 | 2.25 |
| 1378 | C0000381 | 381.00 | 1993.00 | 1958.0 | 7.58 | 2.25 |

## 7. 我的說明上只有寫作單一次的結果，方便您確認我的計算方式是對的，但實質上我也放入迴圈，讓他自動不斷做這件事情，直到該模型的標準殘差均小於 3 則輸出。

# python 的 while 迴圈結果 (我只截圖前半部)



```
12.164144044192804
1375
11.133231801414706
1374
11.385702020196513
1374
10.45921562283536
1373
9.899800901732078
1369
9.77793856884444
1370
9.87537609153964
1369
9.942903737725668
1368
10.031260111811694
1368
9.808434983241087
1367
```