逐步迴歸(Stepwise Regression)

迴歸分析

- 簡單迴歸(Simple Regression)

$$Y = \beta_o + \beta_1 X + \mu$$

- 複迴歸或多元迴歸(Multiple Regression)

$$Y = \beta_o + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \mu$$

逐步迴歸分析

- 從衆多的自變數中，找出最能預測應變數的因素，大多是出現在以預測為目的之探索性研究。
- 一般作法是投入多個解釋變數後，由各變數的相關高低來決定每一個預測變數是否進入迴歸模型或者淘汰出局，最後得到一個以最少解釋變數解釋最多應變數變異量的最佳迴歸模型。

逐步迴歸分析

- 向前法(Forward)
  - 從所有的自變數中，挑選最顯著的優先進入模型中(低p-value)，當模型已選入第一個自變數，則繼續挑選最顯著的自變數進入模型，以此類推，直到下一個最低p-value值的自變數大於門檻值，立即選擇停止，不再選入任何自變數進入模型中。(不顯著的自變項不挑選
- 向後法(Backward)
  - 首先是將所有的自變數都選至模型中，再一步一步篩選最不顯著的變數，一個一個從模型中挑選出來，直到所有在模型中的自變數都是顯著的。
- 逐步挑選(Stepwise)
  - 整合向前法與向後法兩種策略，首先是依據向前法的原理，將最顯著的自變數挑入迴歸模型中，當繼續挑入第二個最顯著的變數入迴歸模型時，而原本模型中已挑入的自變數中最不顯著者，將會被挑出於模型之外。

利用逐步回歸

- 利用逐步迴歸方法，分析上海證交所掛牌上市公司的平均報酬率，是否與自身風險因子或beta有關
- 變數說明:

RET：上市公司股票平均報酬率

BETA_HML：股票報酬率與風險因子 HML 結合之 beta

BETA_MK：股票報酬率與市場風險結合之 beta，或稱系統性風險

BETA_MOM：股票報酬率動能係數 beta

BETA_SMB：股票報酬率與風險因子 SMB 結合之 beta

FIRMCODE：股票代號

INDUSTRY：公司歸屬之產業代號

LOGV：取對數之交易量

SIGMA：上市公司股票報酬率之波動，以標準差代表

SKEWNESS：上市公司股票報酬率之偏態

建立逐步迴歸-向前法函數

- 建立逐步迴歸-向前法函數

```
import pandas as pd
import statsmodels.api as sm


def forward_regression(X, y,
                       threshold_in,
                       verbose=False):
    initial_list = []
    included = list(initial_list)
    while True:
        changed=False
        excluded = list(set(X.columns)-set(included)) #一個一個變數列入
        new_pval = pd.Series(index=excluded,dtype='float64')
        for new_column in excluded:
            model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included+[new_column]]))).fit()
            new_pval[new_column] = model.pvalues[new_column]  #每一個變數算p-value
        best_pval = new_pval.min()
        if best_pval < threshold_in:
            best_feature = new_pval.idxmin() #找最小值
            included.append(best_feature)
            changed=True
            if verbose:
                print('Add  {:30} with p-value {:.6}'.format(best_feature, best_pval))

        if not changed:
            break

    return included
```

建立逐步迴歸-向後法函數

- 建立逐步迴歸-向後法函數

```
def backward_regression(X, y,
                        threshold_out,
                        verbose=False):
    included=list(X.columns)
    while True:
        changed=False
        model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included]))).fit() #全部變數放入迴歸模型
        # use all coefs except intercept
        pvalues = model.pvalues.iloc[1:] #計算p-value
        worst_pval = pvalues.max() # null if pvalues is empty
        if worst_pval > threshold_out:
            changed=True
            worst_feature = pvalues.idxmax()
            included.remove(worst_feature)
            if verbose:
                print('Drop {:30} with p-value {:.6}'.format(worst_feature, worst_pval))
        if not changed:
            break
    return included
```

讀入資料

- 讀入應變數與自變數資料

```
y= pd.read_csv("ret.csv")
y.head()
```

	ret
0	0.095215
1	0.045228
2	0.073642
3	0.020300
4	0.031626

```
X= pd.read_csv("data.csv")
X.head()
```

	beta_hml	beta_mk	beta_mom	beta_smb	firmcode	industry	logv	sigma	skewness	region
0	0.002111	0.877661	0.087371	-0.005359	600000	17	10.125396	0.416688	-0.106485	17
1	-0.011914	0.607542	0.105205	0.010021	600001	9	9.987034	0.536066	0.992389	30
2	0.004652	1.155926	0.025733	-0.009100	600002	14	8.621410	0.459964	0.492296	6
3	-0.003235	0.742394	0.106752	0.016514	600003	14	8.469752	0.521236	1.552359	26
4	0.000927	0.550775	0.042473	0.002473	600004	9	9.037125	0.458615	-1.585671	4

利用向前法選出變數

門檻值

```
result = forward_regression(X, y,0.01)
print('resulting features:')
print(result)
```

```
resulting features:
['sigma', 'skewness', 'logv']
```

利用向前法選出變數-跑迴歸

```
import statsmodels.api as sm
#應變數ret 自變數是sigma,skewness,logv
pairf=pd.concat([X.sigma,X.skewness,X.logv],axis = 1)
model=sm.OLS(y,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.960
Model:                            OLS   Adj. R-squared:                  0.960
Method:                 Least Squares   F-statistic:                     6804.
Date:                Mon, 03 May 2021   Prob (F-statistic):               0.00
Time:                        01:02:17   Log-Likelihood:                 1227.4
No. Observations:                 850   AIC:                            -2447.
Df Residuals:                     846   BIC:                            -2428.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1460      0.020     -7.356      0.000      -0.185      -0.107
sigma          0.0986      0.001    130.035      0.000       0.097       0.100
skewness       0.0118      0.001     10.346      0.000       0.010       0.014
logv           0.0176      0.002      7.494      0.000       0.013       0.022
==============================================================================
Omnibus:                      122.958   Durbin-Watson:                   1.392
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              319.939
Skew:                           0.759   Prob(JB):                     3.36e-70
Kurtosis:                       5.594   Cond. No.                         87.2
==============================================================================
```

利用向後法選出變數

門檻值

```
result = backward_regression(X, y,0.01)
print('resulting features:')
print(result)
```

```
resulting features:
['beta_mk', 'beta_mom', 'logv', 'sigma', 'skewness']
```

利用向後法選出變數-跑迴歸

```
import statsmodels.api as sm
#應變數ret 自變數是sigma,skewness,logv,beta_mk,beta_mom
pairf=pd.concat([X.sigma,X.skewness,X.logv,X.beta_mk,X.beta_mom],axis = 1)
model=sm.OLS(y,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.961
Model:                            OLS   Adj. R-squared:                  0.961
Method:                 Least Squares   F-statistic:                     4211.
Date:                Mon, 03 May 2021   Prob (F-statistic):               0.00
Time:                        01:02:21   Log-Likelihood:                 1241.1
No. Observations:                 850   AIC:                            -2470.
Df Residuals:                     844   BIC:                            -2442.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1229      0.020     -6.130      0.000      -0.162      -0.084
sigma          0.1069      0.002     59.802      0.000       0.103       0.110
skewness       0.0097      0.001      7.939      0.000       0.007       0.012
logv           0.0183      0.002      7.859      0.000       0.014       0.023
beta_mk       -0.0326      0.007     -4.815      0.000      -0.046      -0.019
beta_mom      -0.0993      0.021     -4.699      0.000      -0.141      -0.058
==============================================================================
Omnibus:                      108.529   Durbin-Watson:                   1.445
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              282.369
Skew:                           0.674   Prob(JB):                     4.83e-62
Kurtosis:                       5.481   Cond. No.                         102.
==============================================================================
```

逐步挑選(Stepwise)-向前與向後混和

```
import pandas as pd
import statsmodels.api as sm


def stepwise_selection(X, y,
                       initial_list=[],
                       threshold_in=0.05,
                       threshold_out=0.05,
                       verbose=True):
    initial_list = []
    included = list(initial_list)
    while True:
        changed=False
        # forward step
        excluded = list(set(X.columns)-set(included))
        new_pval = pd.Series(index=excluded,dtype='float64')
        for new_column in excluded:
            model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included+[new_column]]))).fit()
            new_pval[new_column] = model.pvalues[new_column]
        best_pval = new_pval.min()
        if best_pval < threshold_in:
            best_feature = new_pval.idxmin()
            included.append(best_feature)
            changed=True
            if verbose:
                print('Add  {:30} with p-value {:.6}'.format(best_feature, best_pval))
```

逐步挑選(Stepwise)-向前與向後混和

```
        #backward step
        model = sm.OLS(y, sm.add_constant(pd.DataFrame(X[included]))).fit()
        # use all coefs except intercept
        pvalues = model.pvalues.iloc[1:]
        worst_pval = pvalues.max() # null if pvalues is empty
        if worst_pval > threshold_out:
            changed=True
            worst_feature = pvalues.idxmax()
            included.remove(worst_feature)
            if verbose:
                print('Drop {:30} with p-value {:.6}'.format(worst_feature, worst_pval))
        if not changed:
            break
    return included
```

讀入資料

- 讀入應變數與自變數資料

```
y= pd.read_csv("ret.csv")
y.head()
```

	ret
0	0.095215
1	0.045228
2	0.073642
3	0.020300
4	0.031626

```
X= pd.read_csv("data.csv")
X.head()
```

	beta_hml	beta_mk	beta_mom	beta_smb	firmcode	industry	logv	sigma	skewness	region
0	0.002111	0.877661	0.087371	-0.005359	600000	17	10.125396	0.416688	-0.106485	17
1	-0.011914	0.607542	0.105205	0.010021	600001	9	9.987034	0.536066	0.992389	30
2	0.004652	1.155926	0.025733	-0.009100	600002	14	8.621410	0.459964	0.492296	6
3	-0.003235	0.742394	0.106752	0.016514	600003	14	8.469752	0.521236	1.552359	26
4	0.000927	0.550775	0.042473	0.002473	600004	9	9.037125	0.458615	-1.585671	4

篩選結果

```
result = stepwise_selection(X, y)
print('resulting features:')
print(result)
```

```
Add  sigma                          with p-value 0.0
Add  skewness                       with p-value 4.45653e-20
Add  logv                           with p-value 1.68853e-13
Add  beta_mk                        with p-value 0.0211539
Add  beta_mom                       with p-value 3.0479e-06
Add  industry                       with p-value 0.0356136
Add  beta_smb                       with p-value 0.0363007
Add  beta_hml                       with p-value 0.0127963
resulting features:
['sigma', 'skewness', 'logv', 'beta_mk', 'beta_mom', 'industry', 'beta_smb', 'beta_hml']
```

利用逐步挑選選出變數-跑迴歸

```
import statsmodels.api as sm
#應變數ret 自變數是sigma,skewness,logv,beta_mk,beta_mom,industry,beta_smb,beta_hml]
pairf=pd.concat([X.sigma,X.skewness,X.logv,X.beta_mk,X.beta_mom,X.industry,X.beta_smb,X.beta_hml],axis = 1)
model=sm.OLS(y,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.962
Model:                            OLS   Adj. R-squared:                  0.962
Method:                 Least Squares   F-statistic:                     2672.
Date:                Mon, 03 May 2021   Prob (F-statistic):               0.00
Time:                        01:02:28   Log-Likelihood:                 1248.6
No. Observations:                 850   AIC:                            -2479.
Df Residuals:                     841   BIC:                            -2437.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0910      0.022     -4.176      0.000      -0.134      -0.048
sigma          0.1082      0.002     45.333      0.000       0.103       0.113
skewness       0.0105      0.001      8.421      0.000       0.008       0.013
logv           0.0157      0.002      6.370      0.000       0.011       0.021
beta_mk       -0.0228      0.007     -3.126      0.002      -0.037      -0.008
beta_mom      -0.1125      0.021     -5.238      0.000      -0.155      -0.070
industry      -0.0008      0.000     -2.147      0.032      -0.002   -6.81e-05
beta_smb      -0.4325      0.168     -2.581      0.010      -0.761      -0.104
beta_hml      -0.2445      0.098     -2.495      0.013      -0.437      -0.052
==============================================================================
Omnibus:                       99.288   Durbin-Watson:                   1.426
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              269.994
Skew:                           0.605   Prob(JB):                     2.35e-59
Kurtosis:                       5.481   Cond. No.                     1.30e+03
```