

Mini Project: Linear Regression Models

The `Mini Project` report from NTU102-1 [DMIR](#) course

by NTU [Michael Hsu](#)

如何執行

R cmd:

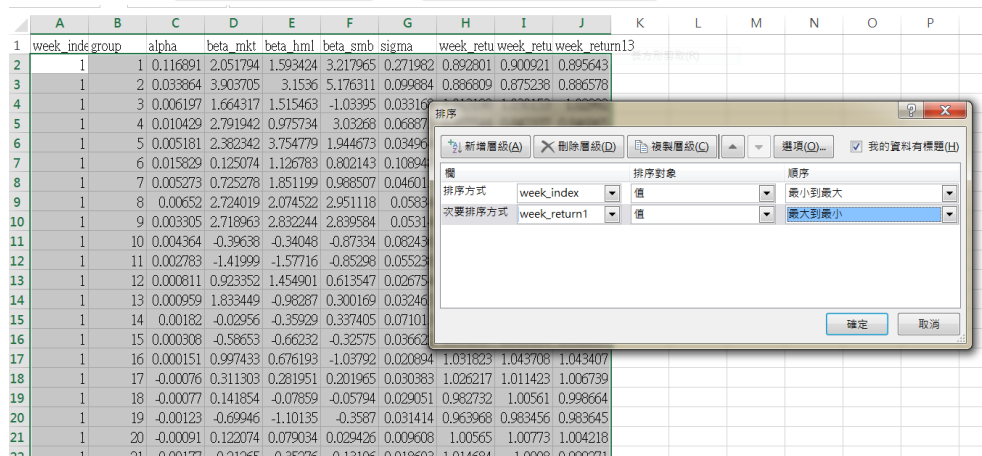
```
> source("/path_to/main.r")
```

example: (可用拖曳方式取得路徑)

```
> source("/Users/michaelhsu/Dropbox/15.\ 碩一上課業/02.\ DMIR\
```

Data Pre-process

1. 新增欄位 `index` : 原本資料的排序 (= sort by `week_index` and `group`) 。
2. 排序與篩選 `week_index` + `week_return1` :



The screenshot shows an Excel spreadsheet with columns A through P. Columns A through G contain data for 'week_index', 'group', 'alpha', 'beta_mkt', 'beta_hml', 'beta_smb', and 'sigma'. Columns H through J contain data for 'week_return1', 'week_return2', and 'week_return3'. A '排序' (Sort) dialog box is open, showing the '排序方式' (Sort by) as 'week_index' and the '次要排序方式' (Secondary sort by) as 'week_return1'. The '排序對象' (Sort range) is set to '值' (Values), and the '順序' (Order) is set to '最小到最大' (Smallest to largest). The '我的資料有標題' (My data has headers) checkbox is checked.

week_index	group	alpha	beta_mkt	beta_hml	beta_smb	sigma	week_return1	week_return2	week_return3
1	1	0.116891	2.051794	1.593424	3.217965	0.271982	0.892801	0.900921	0.895643
2	1	0.033864	3.903705	3.1536	5.176311	0.099884	0.886809	0.875238	0.886578
3	1	0.006197	1.664317	1.515463	-1.03395	0.03316			
4	1	0.010429	2.791942	0.975734	3.03268	0.06887			
5	1	0.005181	2.382342	3.754779	1.944673	0.03496			
6	1	0.015829	0.125074	1.126783	0.802143	0.10894			
7	1	0.005273	0.725278	1.851199	0.988507	0.04601			
8	1	0.00652	2.724019	2.074522	2.951118	0.0583			
9	1	0.003305	2.718963	2.832244	2.839584	0.0531			
10	1	0.004364	-0.39636	-0.34048	-0.87334	0.08243			
11	1	0.002783	-1.41999	-1.57716	-0.85298	0.05523			
12	1	0.000811	0.923352	1.454901	0.613547	0.02675			
13	1	0.000959	1.833449	-0.98287	0.300169	0.03246			
14	1	0.00182	-0.02956	-0.35929	0.337405	0.07101			
15	1	0.000308	-0.58653	-0.66232	-0.32575	0.03662			
16	1	0.000151	0.997433	0.676193	-1.03792	0.020894	1.031823	1.043708	1.043407
17	1	-0.00076	0.311303	0.281951	0.201965	0.030383	1.026217	1.011423	1.006739
18	1	-0.00077	0.141854	-0.07859	-0.05794	0.029051	0.982732	1.00561	0.998664
19	1	-0.00123	-0.69946	-1.10135	-0.3587	0.031414	0.963968	0.983456	0.983645
20	1	-0.00091	0.122074	0.079034	0.029426	0.009608	1.00565	1.00773	1.004218
21	1	0.00177	0.21265	0.35276	0.13105	0.018603	1.014684	1.0008	0.999771

3. 定義分類標籤：

- 1. 新增欄位 `index_sort`：根據上一個步驟後的排序。
- 1. 新增欄位 `index_sort % 30`：`mod(左邊, 30)`
- 1. 給予分類標籤
`class`：`=IF((左邊>0)*(左邊<=6),"1","0")` 前六個為 1，剩下二十四個為 0。

4. 新增欄位 `random_sort`：最後依據這個欄位 `=RAND()` 來做 10-fold classification。

5. 最後整理資料為 `data/ldpa30_train use.csv`

- 剩下 feature
`alpha`、`beta_mkt`、`beta_hml`、`beta_smb`、`sigma`
- 分類的標籤 `class`
- 以及目前的隨機排序依據，作為切割十份用，產生新的
`new_index`。

new_index	alpha	beta_mkt	beta_hml	beta_smb	sigma	class
1	-0.0065066	3.048392	4.07517039	0.9617558	0.01808332	0
2	0.00057031	0.32789225	0.4956953	0.59191614	0.06036168	0
3	0.00236977	1.44659396	-1.835827	2.10289862	0.01848847	0
4	0.00019669	1.51353161	1.08480384	1.94350839	0.01727014	0
5	-0.0042819	1.45120471	1.86734677	0.59260946	0.02066149	0

Evaluation

- use `/data/ldpa30_train.csv`
 - `week_index`：1 ~ 370
- Use 10-fold-validation: 將資料切成十份，輪流當 Training data。
- 最後憑藉 `accuracy`、`rmse` 來挑選適當的 Model。

Model 1: Generative Classification Model

- 執行 `source("generative_classification_model.r")`
- Feature:
`alpha`、`beta_mkt`、`beta_hml`、`beta_smb`、`sigma`
- 依據 Hw3 的 [Generative Classification Models](#) 的結果來看，`Recall` 的結果不是很理想，而且這次 Mini Project 想要的並不是分類的結果，換成

以 Linear Regression Model 來試試看。

- Result: 10-fold-validation 的結果，以及平均。

	accuracy	precision	recall	F-measure
bin1	0.7990991	0.2857143	0.01843318	0.03463203
bin2	0.7918919	0.4090909	0.03964758	0.07228916
bin3	0.7972973	0.4444444	0.01785714	0.03433476
bin4	0.8099099	0.5454545	0.02830189	0.05381166
bin5	0.7873874	0.3571429	0.02155172	0.04065041
bin6	0.8045045	0.2500000	0.01913876	0.03555556
bin7	0.8009009	0.5000000	0.04072398	0.07531381
bin8	0.8045045	0.5000000	0.03686636	0.06866953
bin9	0.7954955	0.5000000	0.04405286	0.08097166
bin10	0.7945946	0.7142857	0.04273504	0.08064516
	accuracy	precision	recall	F-measure
mean	0.79855859	0.4506133	0.03093085	0.05768737
sd	0.006677209	0.1348956	0.01095288	0.01996029

Model 2: Linear Regression Models

- 執行 `source("linear_regression_model.r")`
- Feature1:
 - `alpha` 、 `beta_mkt` 、 `beta_hml` 、 `beta_smb` 、 `sigma`
- Feature2:
 - `alpha` 、 `beta_mkt` 、 `beta_hml` 、 `beta_smb` 、 `sigma` 、 `class`
 - 添加 `class` 的結果希望預期的結果更靠近前幾名的 `week_return1` 的趨勢。
- Result: 10-fold-validation 的結果，可以發現 Feature2 出來預測的結果比 Feature1 還要好。

```
> rmse_matrix
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
lm  0.05058974 0.0471112 0.04578405 0.04909866 0.04181315 0.04537302 0.04837530
glm 0.04444624 0.0404851 0.03973619 0.04292211 0.03478182 0.03955259 0.04131611
      [,8]      [,9]      [,10]
lm  0.04448358 0.04777442 0.04855919
glm 0.03812344 0.04091841 0.04149102
```

Result and Output

- 執行 `source("main.r")`
- 使用 `/data/ldpa30_test_blind.csv` 其中包含 `week_index` : 371 ~ 494。
- 最後挑選 `Model 2: Linear Regression Models`，並且搭配 `Feature2` 來做最後的預測，並且挑選該 `week_index` 區域中 `week_return1` 值最高者，並輸出最後的格式。
- 最後輸出的檔案為 `result.csv`，如下截圖。

	A	B	C
1	week_index	group	
2	371	6	
3	372	7	
4	373	6	
5	374	6	
6	375	5	
7	376	4	
8	377	5	
9	378	3	
10	379	5	
11	380	4	
12	381	5	

Source code

https://github.com/evenchange4/102-1_DMIR_Mini-Project_Linear-Regression-Models

Reference

- [Linear Least Squares Regression](#)
- [Generalized linear models in R](#)
- [11.6.2 glm\(\) 函數](#)
- [Generative Classification Models](#)
- [How to Write CSV in R](#)