

Problem 1

Overview

The overall steps are given in the homework instruction. In this report I will discuss about some detailed manipulations and show the study results.

Data Collection

The data are all downloaded from CSMAR as required by the problem, namely the Daily Market Returns, Daily Stock Price Returns, Index per Share, Statements Release Dates over 2016 to 2022 (some data in 2014 and 2015 are included for window operations).

Step 1 & 2

The data is downloaded and read as DataFrame as required. Noted that there are so many files if we choose .csv files. ‘concat’ function can easily combine these files together in y axis. The columns are renamed accordingly as well.

Step 3

The EPS file is read and converted to DataFrame. Because excluding parent statements is required, all the rows with ‘statement_type’ are deleted.

Claim

On step 3.3, the instruction requires to delete stock names starting with ST and PT. However, some stocks names may coincidentally start with PT/ST (e.g. 836871.BJ PTE). Therefore, I use the Chinese version of stock names, and deleted all stocks starting with ST, PT, *ST, SST, S*ST (once they have such prefix in one period, I delete the company completely).

Then step 3.4 – 3.6 are some time manipulation procedure, with the usage of `pd.to_datetime` & `to_period`.

Warning

On step 3.7 & 3.8, we can NOT directly use ‘`shift()`’, because there exists cases that the downloaded data only include December EPS at some random years. So the data can be completely messy if using ‘`groupby()`’ + ‘`shift()`’ or ‘`diff()`’.

Thus, step 3.7 & 3.8 are rather complex. I first extract EPS values for June and December in the same year, then calculates the difference between December EPS and June EPS, reflecting the EPS change for the second half of the fiscal year. This result replaces the original December EPS

in the data. Noted that the code processes this calculation by grouping the data by stock code and applying these functions to each group, updating the entire dataset accordingly.

Continue on step 3.9, now `‘.shift()’` can be used, because step 3.8 ensures that the data is well structured (all the years with only one entry data will be discarded because of `‘.dropna()’`)

Steps 3.11 – 3.13 are steps for correctly derive the SUE deciles. Noted that `‘.pivot()’` is used for converting the long DataFrame to wide DataFrame.

Claim

On Step 3.9, UE's from the subtraction of this year's EPS with last year's. As 2014h1 & 2014h2 do not have "last year" in the data set, so their UE will be NaN. On Step 3.10, we include the recent three periods' data as the window for s.d. Thus, 2015h1, 2015h2 and 2016h1 will be used to get s.d. for 2016h2, and these three periods will not have their own SUE, instead they are NaN and be discarded. That's why the wide data set of EPS from Step 3.13 may start with 2016h2 instead of 2016h1.

Step 4

Very similar to the process to get wide DataFrame in step 3.13, with an additional step to convert the body data into datetime.

Step 5

All the useful DataFrame we got above are combined into a wide one using `‘pd.merge()’`.

Step 6

Step 6.3 is complex. It uses a loop for iterating on every time period column. The implementation follows the method 2 to get event day in the tutorial slide. For each iteration, the NaN data will be discarded, then the cumulative return of each decile is calculated using `‘.cumsum()’`.

Warning

Directly using the method on tutorial may cause some time disruption due to unexpected trading suspension (without PT/ST labeled, e.g 000716.SZ). Thus, I limited the original trading date range before getting the relative trading date difference. Data out of range will be discarded for data accuracy.

On step 6.4, we get the mean of decile cumulative return of all half-year periods, namely `‘mean_portfolio_car’`. Finally we can get the result figure of this event study:

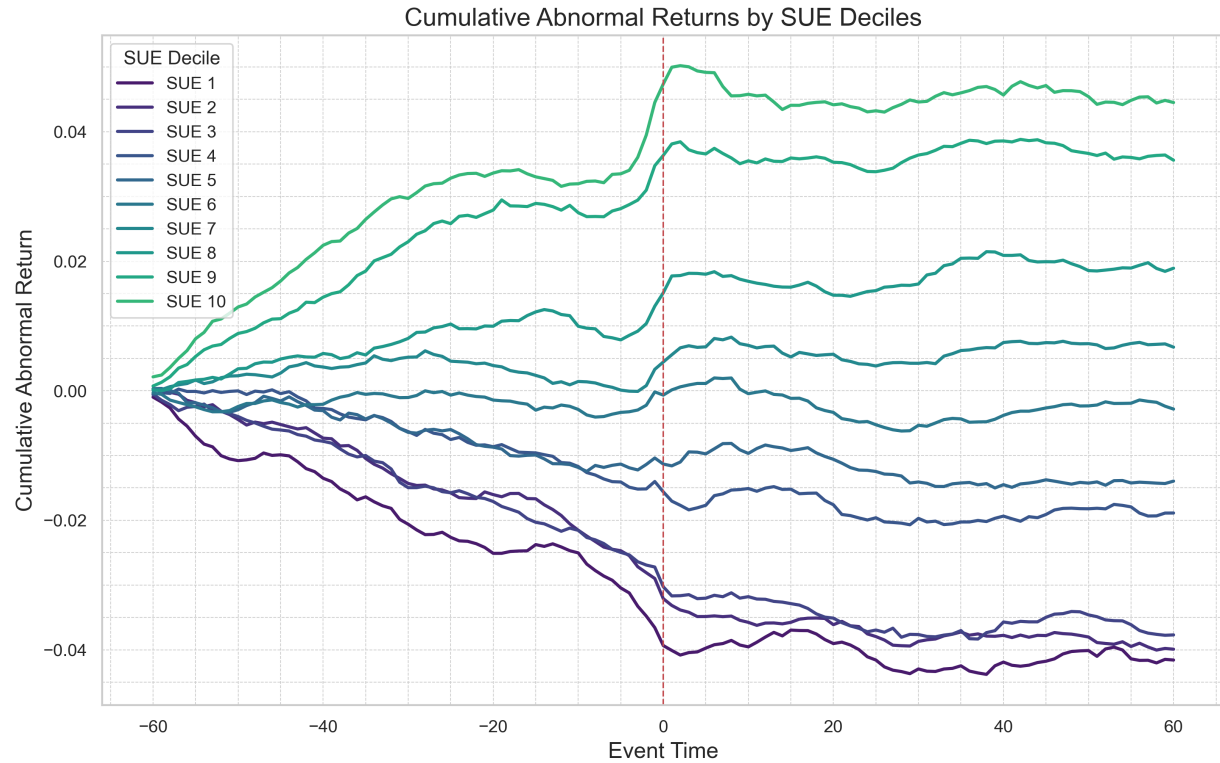


Figure 1: Cumulative Abnormal Returns by SUE Deciles

Results

As shown in the figure, the deciles with higher SUE (SUE 6 – 10) has higher cumulative abnormal returns with increasing trends; the deciles with lower SUE (SUE 1 – 4) has lower (more negative) cumulative abnormal returns with decreasing trends.

In conclusion, Post Earnings Announcement Drift (PEAD) exists in China's A-share markets.

– END –