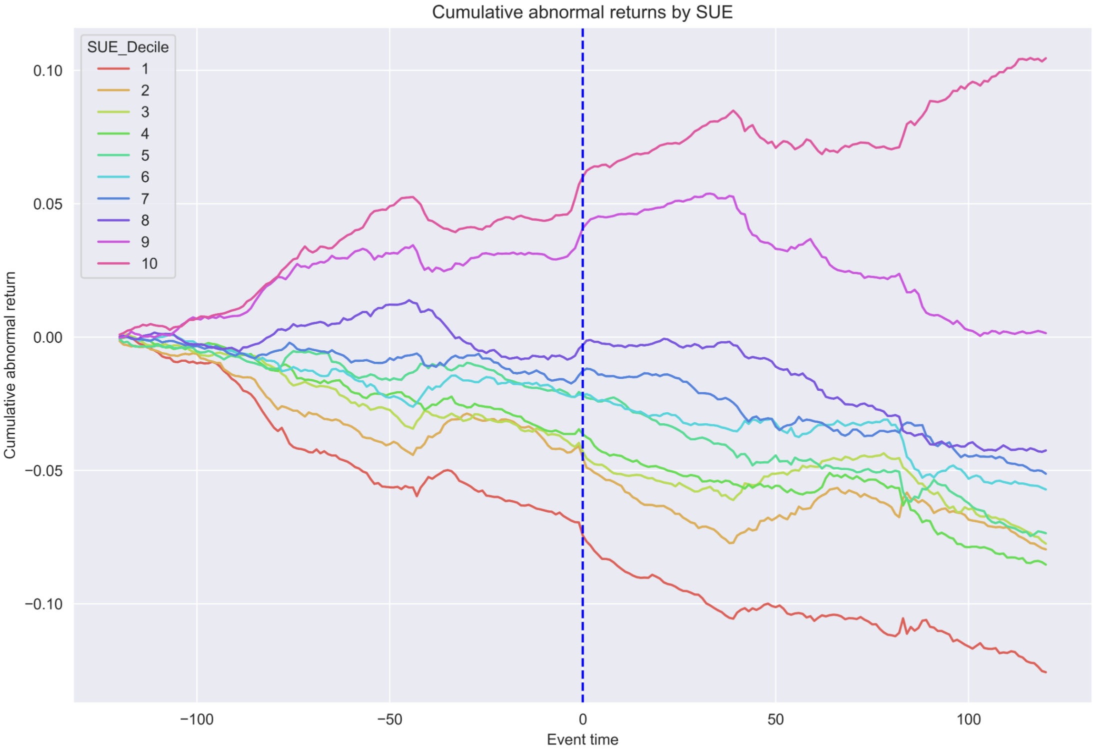
FIN3080_Assignment 5 Report:

Back Testing for Post Earning Announcement Drift (PEAD) in China Stock Market

Weiyuan XU (许炜塬) 120090888

Empirical Replication Result



Analysis:

1. **Theory Analysis:** Post Earning Announcement Drift (PEAD) is one of the most classical phenomena in asset pricing. It describes the inertia for a stock's cumulative abnormal returns to drift in the same direction of an earnings surprise following earnings announcements. Once a firm's current earnings become known, the information content should be quickly digested by investors and incorporated into the efficient market price. However, it has long been known that this is not exactly what happens. For firms that report good news in quarterly earnings, their abnormal security returns tend to drift upwards for at least 60 days following their earnings announcement. Similarly, firms that report bad news in earnings tend to have their abnormal security returns drift downwards for a similar period. This phenomenon is called post-announcement drift.

2. **Observation Analysis:**

   Announcement / Event does not efficiently affect the tendency of the cumulative AR for each group: From the plot, it is obvious that for each group of stocks, it keeps the tendency after the announcement, which means that the announcement / event was not correctly and quickly digested in the Chinese A-Share stock market.

3. **Conclusion Analysis:**

   Market Efficiency Theory may not hold for the Chinese A-Share stock market: As we know, (strong-form) EMH suggests that the **historical, private, public information** should be **correctly and quickly enough** digested and revealed in the stock price. However, after the PEAD empirical analysis for the Chinese A-Share stock market, it is suggested that the information was not correctly and quickly digested in the Chinese A-Share stock market. As a result, Market Efficiency Theory may not hold for the Chinese A-Share stock market.

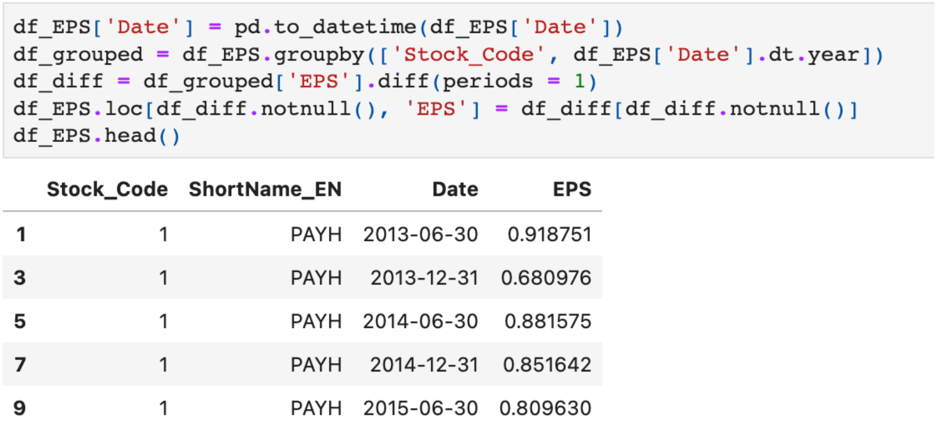
Data Acquisition and Data Processing Highlights

Data Acquisition Highlights

- **EPS**: Set the condition **Statement Type = Consolidated Statements** since:
  - Consolidated statements reflect the financial position and operating results of the entire enterprise group; The parent statement only reflects the business situation of the parent company, and its right to control the entire enterprise group is actually reflected in the long-term asset account as an investment. In the process of merging the statements of the parent company and the subsidiary company, the parent company's investment in the subsidiary company, internal purchase and sale, creditor's rights and debts are offset.
- **Daily Individual Stock Return**: Market Type = 1, 4 (including SME)
- **Daily Market Return:** Set **Market Type = 5 (For A Shares)** in CSMAR
- **Daily Market Return:** Use **Weekly Market Return with Cash Dividend Reinvested (Equal-Value-Weighted)**

Data Processing Highlights

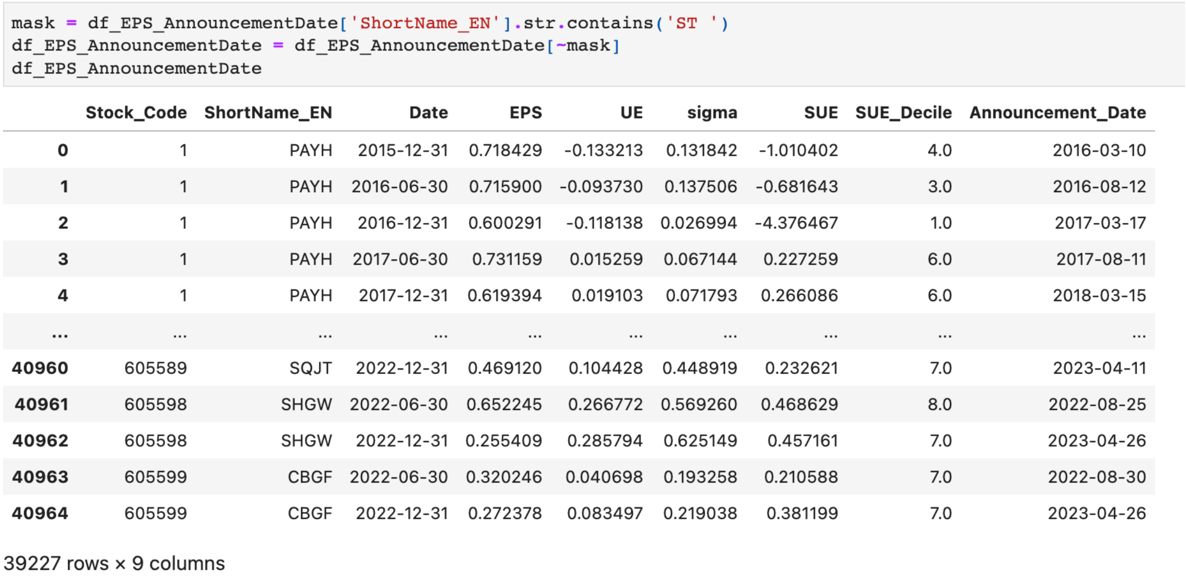
- **Semi-annual EPS Conversion:** Replace EPS over the second half of each year as the difference between EPS as of Q4 and Q2 in that year.



```
df_EPS['Date'] = pd.to_datetime(df_EPS['Date'])
df_grouped = df_EPS.groupby(['Stock_Code', df_EPS['Date'].dt.year])
df_diff = df_grouped['EPS'].diff(periods = 1)
df_EPS.loc[df_diff.notnull(), 'EPS'] = df_diff[df_diff.notnull()]
df_EPS.head()
```

	Stock_Code	ShortName_EN	Date	EPS
1	1	PAYH	2013-06-30	0.918751
3	1	PAYH	2013-12-31	0.680976
5	1	PAYH	2014-06-30	0.881575
7	1	PAYH	2014-12-31	0.851642
9	1	PAYH	2015-06-30	0.809630

- **Exclude firms with "ST" or "PT":** Filter out the stocks with short name beginning from "ST " (Remark: There is an empty space here)

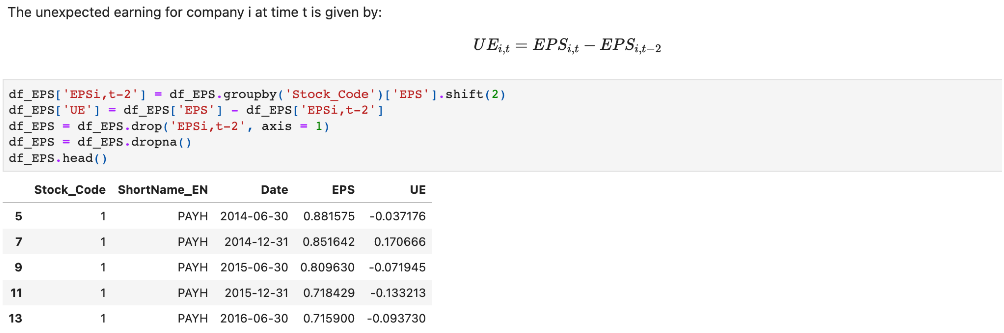


```
mask = df_EPS_AnnouncementDate['ShortName_EN'].str.contains('ST ')
df_EPS_AnnouncementDate = df_EPS_AnnouncementDate[~mask]
df_EPS_AnnouncementDate
```

	Stock_Code	ShortName_EN	Date	EPS	UE	sigma	SUE	SUE_Decile	Announcement_Date
0	1	PAYH	2015-12-31	0.718429	-0.133213	0.131842	-1.010402	4.0	2016-03-10
1	1	PAYH	2016-06-30	0.715900	-0.093730	0.137506	-0.681643	3.0	2016-08-12
2	1	PAYH	2016-12-31	0.600291	-0.118138	0.026994	-4.376467	1.0	2017-03-17
3	1	PAYH	2017-06-30	0.731159	0.015259	0.067144	0.227259	6.0	2017-08-11
4	1	PAYH	2017-12-31	0.619394	0.019103	0.071793	0.266086	6.0	2018-03-15
...	...	...	...	...	...	...	...	...	...
40960	605589	SQJT	2022-12-31	0.469120	0.104428	0.448919	0.232621	7.0	2023-04-11
40961	605598	SHGW	2022-06-30	0.652245	0.266772	0.569260	0.468629	8.0	2022-08-25
40962	605598	SHGW	2022-12-31	0.255409	0.285794	0.625149	0.457161	7.0	2023-04-26
40963	605599	CBGF	2022-06-30	0.320246	0.040698	0.193258	0.210588	7.0	2022-08-30
40964	605599	CBGF	2022-12-31	0.272378	0.083497	0.219038	0.381199	7.0	2023-04-26

39227 rows × 9 columns

- **UE Computation**:



The unexpected earning for company i at time t is given by:

$$UE_{i,t} = EPS_{i,t} - EPS_{i,t-2}$$

```
df_EPS['EPSi,t-2'] = df_EPS.groupby('Stock_Code')['EPS'].shift(2)
df_EPS['UE'] = df_EPS['EPS'] - df_EPS['EPSi,t-2']
df_EPS = df_EPS.drop('EPSi,t-2', axis = 1)
df_EPS = df_EPS.dropna()
df_EPS.head()
```

	Stock_Code	ShortName_EN	Date	EPS	UE
5	1	PAYH	2014-06-30	0.881575	-0.037176
7	1	PAYH	2014-12-31	0.851642	0.170666
9	1	PAYH	2015-06-30	0.809630	-0.071945
11	1	PAYH	2015-12-31	0.718429	-0.133213
13	1	PAYH	2016-06-30	0.715900	-0.093730

- ## SUE Computation:

The standardize unexpected earning for company i at time t is give by:

$$SUE_{i,t} = UE_{i,t}/\sigma_{i,t}$$

in which $\sigma_{i,t}$ is the standard deviation of $\{UE_{i,t-3}, UE_{i,t-2}, UE_{i,t-1}, UE_{i,t}\}$

```
df_EPS['sigma'] = df_EPS.groupby('Stock_Code')['UE'].rolling(window = 4).std().reset_index(0, drop = True)
df_EPS['SUE'] = df_EPS['UE'] / df_EPS['sigma']
df_EPS = df_EPS.dropna()
df_EPS.head(10)
```

	Stock_Code	ShortName_EN	Date	EPS	UE	sigma	SUE
11	1	PAYH	2015-12-31	0.718429	-0.133213	0.131842	-1.010402
13	1	PAYH	2016-06-30	0.715900	-0.093730	0.137506	-0.681643
15	1	PAYH	2016-12-31	0.600291	-0.118138	0.026994	-4.376467
17	1	PAYH	2017-06-30	0.731159	0.015259	0.067144	0.227259
19	1	PAYH	2017-12-31	0.619394	0.019103	0.071793	0.266086
21	1	PAYH	2018-06-30	0.778800	0.047641	0.074157	0.642438
23	1	PAYH	2018-12-31	0.666628	0.047234	0.017540	2.692971
25	1	PAYH	2019-06-30	0.897088	0.118288	0.042312	2.795639
27	1	PAYH	2019-12-31	0.555813	-0.110815	0.096874	-1.143904
29	1	PAYH	2020-06-30	0.704834	-0.192254	0.142285	-1.351192

- ## AR Computation:

The daily abnormal return for firm i at time t is given by:

$$AR_{i,t} = r_{i,t} - r_{m,t}$$

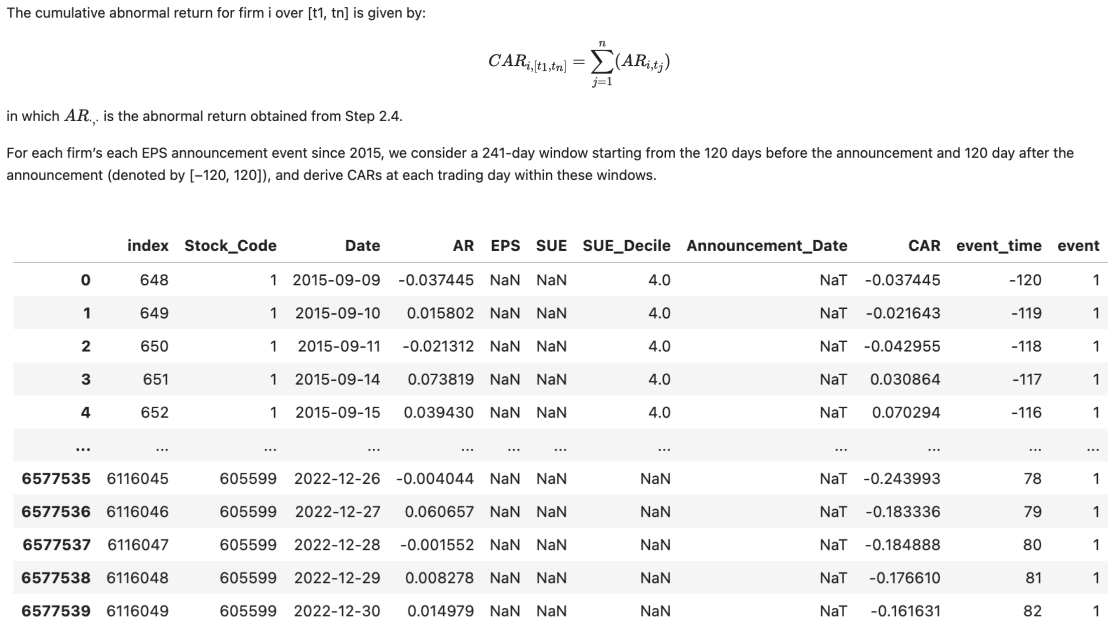
in which $r_{i,t}$ is firm i's stock return at time t and $r_{m,t}$ is the market return at time t.

```
df_Return["AR"] = df_Return["Daily_Individual_Return"] - df_Return["Daily_Market_Return"]
df_Return
```

	Stock_Code	Date	Daily_Individual_Return	Daily_Market_Return	AR
0	1	2013-01-04	-0.001873	-0.003810	0.001937
1	1	2013-01-07	0.019387	0.011894	0.007493
2	1	2013-01-08	-0.018405	0.011362	-0.029767
3	1	2013-01-09	-0.008750	0.001775	-0.010525
4	1	2013-01-10	0.000631	0.006056	-0.005425
...	...	...	...	...	...
6133097	605599	2022-12-26	0.009667	0.013711	-0.004044
6133098	605599	2022-12-27	0.067021	0.006364	0.060657
6133099	605599	2022-12-28	-0.010967	-0.009415	-0.001552
6133100	605599	2022-12-29	0.003024	-0.005254	0.008278
6133101	605599	2022-12-30	0.023116	0.008137	0.014979

6133102 rows × 5 columns

- **CAR Computation**: If the announcement date is not the trading date, window central point **i** will be postponed to the first trading date after announcement date.



The cumulative abnormal return for firm i over [t1, tn] is given by:

$$CAR_{i,[t1,tn]} = \sum_{j=1}^{n}(AR_{i,t_j})$$

in which $AR_{\cdot,\cdot}$ is the abnormal return obtained from Step 2.4.

For each firm's each EPS announcement event since 2015, we consider a 241-day window starting from the 120 days before the announcement and 120 day after the announcement (denoted by [−120, 120]), and derive CARs at each trading day within these windows.

	index	Stock_Code	Date	AR	EPS	SUE	SUE_Decile	Announcement_Date	CAR	event_time	event
0	648	1	2015-09-09	-0.037445	NaN	NaN	4.0	NaT	-0.037445	-120	1
1	649	1	2015-09-10	0.015802	NaN	NaN	4.0	NaT	-0.021643	-119	1
2	650	1	2015-09-11	-0.021312	NaN	NaN	4.0	NaT	-0.042955	-118	1
3	651	1	2015-09-14	0.073819	NaN	NaN	4.0	NaT	0.030864	-117	1
4	652	1	2015-09-15	0.039430	NaN	NaN	4.0	NaT	0.070294	-116	1
...	...	...	...	...	...	...	...	...	...	...	...
6577535	6116045	605599	2022-12-26	-0.004044	NaN	NaN	NaN	NaT	-0.243993	78	1
6577536	6116046	605599	2022-12-27	0.060657	NaN	NaN	NaN	NaT	-0.183336	79	1
6577537	6116047	605599	2022-12-28	-0.001552	NaN	NaN	NaN	NaT	-0.184888	80	1
6577538	6116048	605599	2022-12-29	0.008278	NaN	NaN	NaN	NaT	-0.176610	81	1
6577539	6116049	605599	2022-12-30	0.014979	NaN	NaN	NaN	NaT	-0.161631	82	1

Reference:

- Bernard, Victor L and Jacob K Thomas (1989). "Post-earnings-announcement drift: delayed price response or risk premium?" In: Journal of Accounting research 27, pp. 1–36.

Data Resources:

- Data Service Platform: CSMAR