# *How Trump and Paris Agreement shock Climate Change Exposure in stock market?*

**Sep 12th, 2024**

## *Main References:*

SAUTNER, Z., VAN LENT, L., VILKOV, G. and ZHANG, R. (2023), **Firm-Level Climate Change Exposure.** J Finance, 78: 1449-1498. https://doi.org/10.1111/jofi.13219

Child, Travers & Massoud, Nadia & Schabus, Mario & Zhou, Yifan. (2020). **Surprise Election for Trump Connections.** Journal of Financial Economics. 140. 10.1016/j.jfineco.2020.12.004.

## *Setting:*

We conduct two event studies to consider how climate change policy shocks affect the performance of public companies. When the U.S. first joined the Paris Agreement in 2015[1], we expect high-polluting firms to underperform, i.e., high-Climate Exposure firms have a negative CAR. The Democratic Party is not pro-ESG, and when Trump comes to power and becomes President of the U.S.[2], we expect high-polluting firms to generate positive performance, i.e., high-Climate Exposure firms have a positive CAR.

我们进行两个事件研究，考虑气候变化政策的冲击如何影响上市公司的表现。当美国在 2015 年第一次加入巴黎协定时，我们预计高污染企业的表现变差，即高 Climate Exposure 企业有负的 CAR。民主党并不支持 ESG，当特朗普上台成为美国总统后，我们预计高污染企业会产生正向的表现，即高 Climate Exposure 企业有正的 CAR。

## *Data:*

### *1) Data Source:*

We use the Center for **Research in Security Prices (CRSP)** files to obtain stock returns, **Standard and Poor's Compustat database** to obtain financial information.

From *text mining* we get the ratio of climate change occur in the transcripts of earnings conference calls. From *Trucost* we get data on carbon emmssion.

[1] The Paris Agreement was adopted on 12 December 2015(Saturday, so we choose **14Dec2015** as event day) at the 21st United Nations Climate Change Conference, signed on 22 April 2016 at the United Nations building in New York, United States of America, and formally implemented from 4 November 2016 onwards.

[2] On **9 November 2016**, at 1:40 a.m. EST, Republican presidential candidate Donald Trump won the presidential election.

### *2) Data*

- Period: The sample spans the period Q1 2014 to Q2 2019.
- Data Sets:
  - Main data sets:
    - Company fundamentals: annual frequency.
    - Stock performance: daily frequency.
    - The index obtained from text mining *Climate_Regulation_Risk*, a third-party database of *carbon emissions* data, was merged with other data sets through **cik**.
  - Other data sets are required:
    - CRSP-Annual Update-Stock / Events-Names(NO period) & Delist(from Q1 2024 to Q2 2019).
    - CRSP-Annual Update-CRSP/Compustat Merged-Compustat CRSP Link.

## *Method: Event Research*

**CAR(Cumulative Abnormal Returns)** mostly used to look at the sum of abnormal returns of a company's stock price over multiple days during the window before and after a certain time occurrence (Often using daily data.[3]). In our research, event occurrence includes **Paris Agreement Announced** and **Trump Wins Election.**

- Abnormal return AR for a single day = daily return on the company's stock on that day - benchmark's market return

- Cumulative abnormal returns CAR = sum of AR for multiple days

## *Tasks:*

1. Using Jupyter to merge CRSP stock return and CompStat Fundamental Information data set by debugging the open source code.

2. Then, merging the former data to CRE and carbon data by column **cik**.

Note:【寻找最邻近日期[4]】一组是股票数据，有日期、代码、收益率等，这个日期和交易日相吻合；一组是自己制作提取的数据 CRE，有日期、代码、CRE 值，但是这个日期是和公司股东大会日期吻合的。使用 STATA 软件，将两组数据匹配，要求将 CRE 数据合并到股票数据中，先在股票数据中寻找 cik，然后看股票数据中的日期和 CRE 数据中日期最接近的 CRE 数据中的那一行，匹配给股票数据.

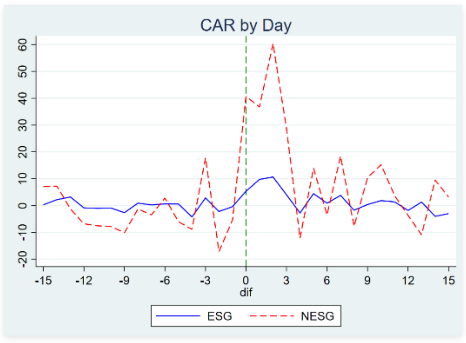
3.Event Study and cross-section regression.

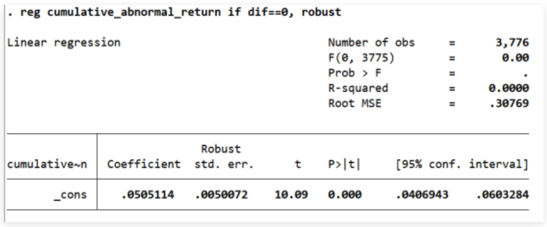
[3] Please DO NOT FORGET to convert date or month to date format in Stata and install the packages needed to run the code.

[4] The nearest date should first satisfy that the stock trading time is **LATER** than the CRE value generation time..

## *Experiment 1* (Sep 14th - 16th ):

1. Trump: event windows=±15 estimation window=[-25,-15]



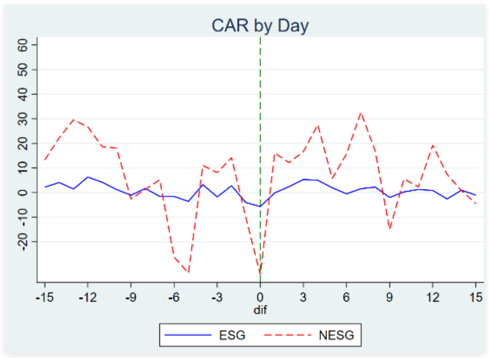


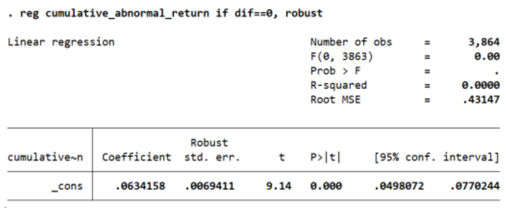
```
. reg cumulative_abnormal_return if dif==0, robust

Linear regression                               Number of obs     =      3,776
                                                F(0, 3775)        =       0.00
                                                Prob > F          =          .
                                                R-squared         =     0.0000
                                                Root MSE          =     .30769

------------------------------------------------------------------------------
             |               Robust
cumulative~n | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _cons |   .0505114   .0050072    10.09   0.000     .0406943    .0603284
------------------------------------------------------------------------------
```

2. Paris Agreement: event windows=±15 estimation window=[-25,-15]



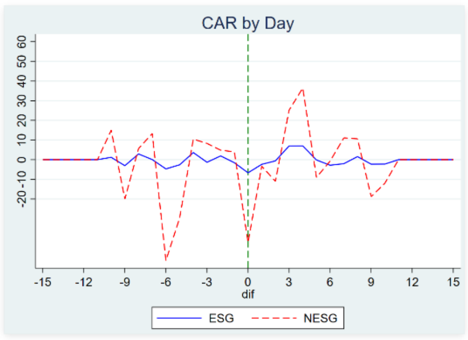


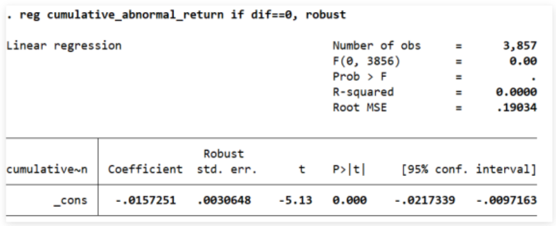
```
. reg cumulative_abnormal_return if dif==0, robust

Linear regression                               Number of obs     =      3,864
                                                F(0, 3863)        =       0.00
                                                Prob > F          =          .
                                                R-squared         =     0.0000
                                                Root MSE          =     .43147

------------------------------------------------------------------------------
             |               Robust
cumulative~n | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _cons |   .0634158   .0069411     9.14   0.000     .0498072    .0770244
------------------------------------------------------------------------------
```

3. Paris Agreement: event windows=±10 estimation window=[-50,-10]



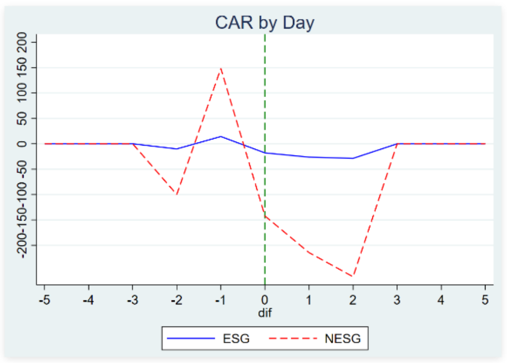


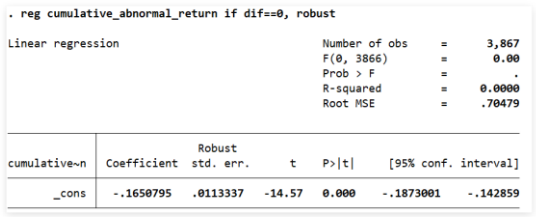
```
. reg cumulative_abnormal_return if dif==0, robust

Linear regression                               Number of obs     =      3,857
                                                F(0, 3856)        =       0.00
                                                Prob > F          =          .
                                                R-squared         =     0.0000
                                                Root MSE          =     .19034

------------------------------------------------------------------------------
             |               Robust
cumulative~n | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _cons |  -.0157251   .0030648    -5.13   0.000    -.0217339   -.0097163
------------------------------------------------------------------------------
```

4. Paris Agreement: event windows=±2 estimation window=[-5,-2]



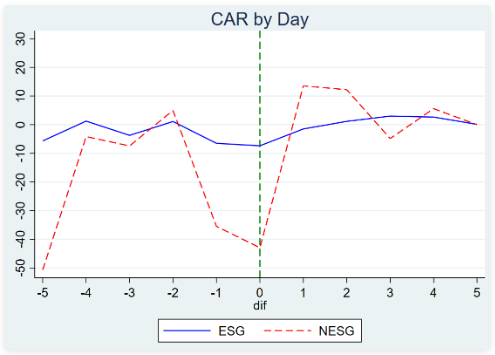


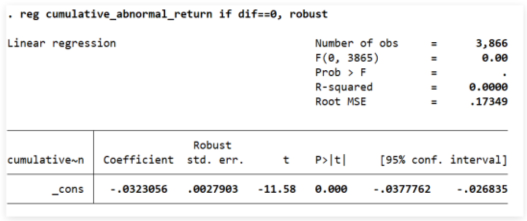
```
. reg cumulative_abnormal_return if dif==0, robust

Linear regression                               Number of obs     =      3,867
                                                F(0, 3866)        =       0.00
                                                Prob > F          =          .
                                                R-squared         =     0.0000
                                                Root MSE          =     .70479

------------------------------------------------------------------------------
             |               Robust
cumulative~n | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _cons |  -.1650795   .0113337   -14.57   0.000    -.1873001    -.142859
------------------------------------------------------------------------------
```

5. Paris Agreement: event windows=±5 estimation window=[-10,-5]



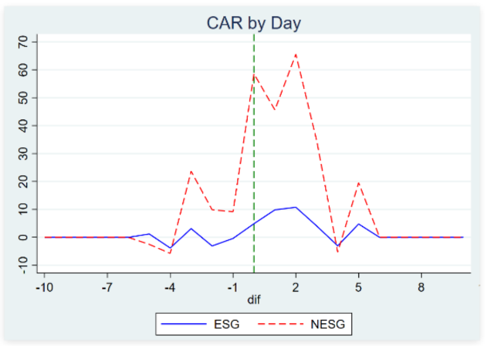


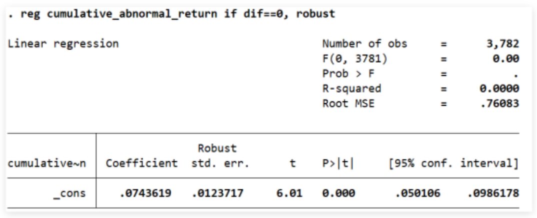
```
. reg cumulative_abnormal_return if dif==0, robust

Linear regression                               Number of obs     =      3,866
                                                F(0, 3865)        =       0.00
                                                Prob > F          =          .
                                                R-squared         =     0.0000
                                                Root MSE          =     .17349

------------------------------------------------------------------------------
             |               Robust
cumulative~n | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _cons |  -.0323056   .0027903   -11.58   0.000    -.0377762    -.026835
------------------------------------------------------------------------------
```

6. Trump: event windows=±5 estimation window=[-10,-15]





```
. reg cumulative_abnormal_return if dif==0, robust

Linear regression                               Number of obs     =      3,782
                                                F(0, 3781)        =       0.00
                                                Prob > F          =          .
                                                R-squared         =     0.0000
                                                Root MSE          =     .76083

------------------------------------------------------------------------------
             |               Robust
cumulative~n | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _cons |   .0743619   .0123717     6.01   0.000      .050106    .0986178
------------------------------------------------------------------------------
```

**All samples**

|  | (1) | (2) | (4) | (3) | (5) |
|---|---|---|---|---|---|
|  | Trump-CAR ±15 | Paris-CAR ±15 | Paris-CAR ±10 | Paris-CAR ±2 | Paris-CAR ±5 |
| climate__risk | -2.1480$^{***}$ | 1.6150$^{***}$ | -1.5137*** | -33.3784$^{***}$ | -1.7476$^{***}$ |
|  | (-15.8412) | (11.5430) | (-25.2203) | (-1.7e+02) | (-25.5624) |
| me | 0.0000 | -0.0000$^{***}$ | 0.0000 | -0.0000$^{***}$ | 0.0000$^{***}$ |
|  | (0.2079) | (-6.8318) | (1.4152) | (-5.9939) | (4.0463) |
| cumretx | -0.0173$^{***}$ | 0.0007$^{***}$ | 0.0004*** | -0.0032$^{***}$ | 0.0002$^{**}$ |
|  | (-15.2795) | (4.4539) | (6.2679) | (-14.7841) | (2.4204) |
| be | -0.0000$^{***}$ | -0.0000$^{***}$ | -0.0000*** | 0.0000$^{***}$ | -0.0000$^{***}$ |
|  | (-7.5926) | (-7.6786) | (-3.3437) | (17.1766) | (-4.3301) |
| vol | 0.0000$^{***}$ | 0.0000$^{***}$ | 0.0000*** | 0.0000 | 0.0000$^{***}$ |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  | (19.9569) | (30.7309) | (22.0421) | (1.5043) | (12.9496) |
| _cons | 0.0514$^{***}$ | 0.0784$^{***}$ | -0.0119*** | -0.0594$^{***}$ | 0.0202$^{***}$ |
|  | (30.0438) | (47.8280) | (-16.9453) | (-25.5688) | (25.1853) |
| *N* | 72983 | 137687 | 137687 | 137687 | 137687 |
| adj. $R^2$ | 0.011 | 0.009 | 0.008 | 0.172 | 0.006 |

**Only Dirty Firms Sample**

|  | (1) Trump-CAR ±15 | (2) Paris-CAR ±15 | (4) Paris-CAR ±10 | (3) Paris-CAR ±2 | (5) Paris-CAR ±5 |
|---|---|---|---|---|---|
| climate__risk | -4.6305 | 1.2507 | -0.5118 | -2.5140 | -1.7476$^{***}$ |
|  | (-0.8877) | (0.4672) | (-0.3830) | (-0.2281) | (-25.5624) |
| me | -0.0000 | -0.0000 | 0.0000 | 0.0000 | 0.0000$^{***}$ |
|  | (-0.6833) | (-0.1205) | (0.0303) | (0.2199) | (4.0463) |
| cumretx | -0.0044 | -0.0005 | -0.0000 | -0.0027 | 0.0002$^{**}$ |
|  | (-0.1008) | (-0.1589) | (-0.0006) | (-0.2272) | (2.4204) |
| be | -0.0000 | 0.0000 | -0.0000 | 0.0000 | -0.0000$^{***}$ |
|  | (-0.8779) | (0.5033) | (-0.5845) | (0.1016) | (-4.3301) |
| vol | 0.0000$^{***}$ | -0.0000$^{**}$ | 0.0000$^{***}$ | -0.0000 | 0.0000$^{***}$ |
|  | (5.9343) | (-2.1795) | (2.7257) | (-0.8553) | (12.9496) |
| _cons | 2.6646$^{***}$ | 2.1833$^{***}$ | -0.2019$^{***}$ | -5.5285$^{***}$ | 0.0202$^{***}$ |
|  | (38.1171) | (64.3275) | (-11.9150) | (-39.5649) | (25.1853) |
| *N* | 61876 | 116214 | 116214 | 116214 | 137687 |
| adj. $R^2$ | 0.000 | -0.000 | 0.000 | -0.000 | 0.006 |

**Other Experiment**

|  | (6) Trump-CAR ±5 All Sample | (6) Trump-CAR ±5 Dirty Sample |
|---|---|---|
| climate__risk | 10.4347$^{***}$ | -6.3708 |
|  | (23.7504) | (-1.1221) |
| me | -0.0000$^{***}$ | -0.0000 |
|  | (-5.8560) | (-0.9099) |
| cumretx | -0.0363$^{***}$ | -0.0029 |
|  | (-9.9050) | (-0.0624) |
| be | 0.0000 | -0.0000 |
|  | (1.4016) | (-1.1012) |
| vol | 0.0000 | 0.0000$^{***}$ |
|  | (0.5216) | (7.5037) |
| _cons | 0.1172$^{***}$ | 3.9426$^{***}$ |
|  | (21.1274) | (51.8109) |
| *N* | 73024 | 61917 |
| adj. $R^2$ | 0.010 | 0.001 |

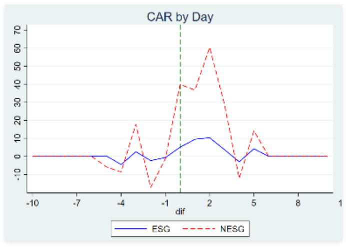
**Problem:**

It is evident that the regression *t-statistic is disproportionately large* in the results. This phenomenon may be attributed to the absence of *clustering in the variance estimation process*. Additionally, the issue of *inference efficiency* remains unaddressed, and the *variance clustering* necessitates adjustment. Cross-sectional regression can be adjusted to the industry level to attempt to identify the optimal window from 0 and different training windows. It is anticipated that Trump will have a positive impact, while Paris is expected to have a negative impact. It is possible that the two events may result in the same significant window.

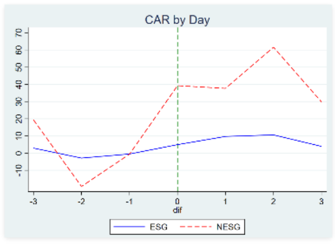
## *Experiment 2* (Sep 17th - 18th ):

**Trump**

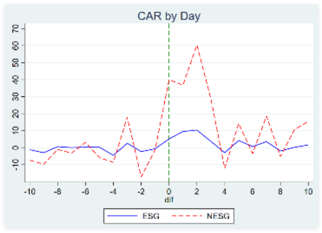
(1) event windows=±5
estimation window=[-25,-10]



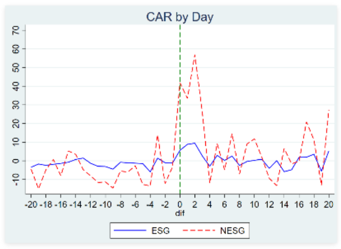
(2) event windows=±3
estimation window=[-25,-3]



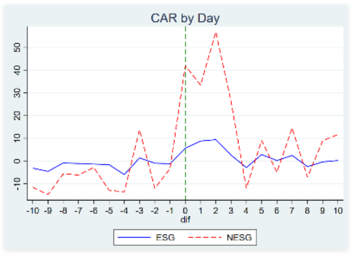
(2) event windows=±10
estimation window=[-25,-10]



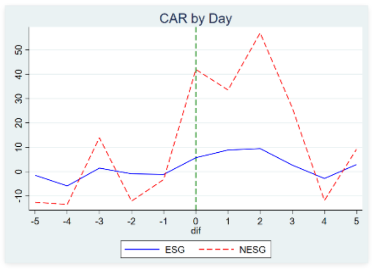
(4) event windows=±20
estimation window=[-255,-46]



(5) event windows=±10
estimation window=[-255,-30]

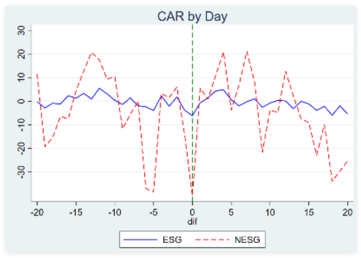


(6) event windows=±5
estimation window=[-255,-10]

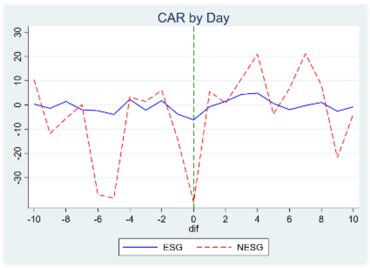


**Paris**

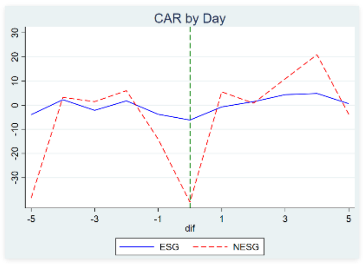
(7) event windows=±20
estimation window=[-255,-10]



(8) event windows=±10
estimation window=[-255,-10]



(9) event windows=±5
estimation window=[-255,-10]



**Empirical Regression (variance clustering industry level, CAR[0,window])**

| | (1) | | (2) | |
|---|---|---|---|---|
| | car_window | nesg_car_window | car_window | nesg_car_window |
| climate__risk | -0.6320 | -4.5926*** | -0.7219 | -5.3018*** |
| | (-1.5016) | (-3.3192) | (-1.4816) | (-3.4553) |
| me | -0.0000*** | -0.0000 | -0.0000*** | -0.0000* |
| | (-3.5954) | (-1.6266) | (-3.6558) | (-1.6527) |
| cumretx | -0.0104** | -0.0014 | -0.0108** | -0.0022 |
| | (-2.1066) | (-0.2347) | (-2.1374) | (-0.3440) |
| be | -0.0000 | -0.0000 | 0.0000 | -0.0000 |
| | (-0.0947) | (-1.5734) | (0.5643) | (-1.6220) |
| vol | 0.0000 | 0.0000*** | 0.0000 | 0.0000*** |
| | (1.2071) | (2.9180) | (1.1837) | (3.2236) |

|  |  |  |  |  |
|---|---|---|---|---|
| _cons | 0.0651$^{***}$ | 2.3738$^{***}$ | 0.0679$^{***}$ | 2.6004$^{***}$ |
|  | (7.1933) | (88.8939) | (5.9576) | (89.6282) |
| $N$ | 72983 | 61876 | 72983 | 61876 |
| adj. $R^2$ | 0.019 | 0.001 | 0.026 | 0.001 |

|  | (3) |  | (4) |  |
|---|---|---|---|---|
|  | car_window | nesg_car_window | car_window | nesg_car_window |
| climate__risk | 0.2887 | -4.2062$^{***}$ | 1.7325$^{***}$ | -3.0698$^{***}$ |
|  | (0.5021) | (-3.4326) | (3.0141) | (-3.0137) |
| me | -0.0000$^{***}$ | -0.0000 | -0.0000$^{***}$ | -0.0000 |
|  | (-3.8906) | (-1.5247) | (-3.5185) | (-1.5243) |
| cumretx | -0.0095$^{**}$ | -0.0012 | -0.0126$^{*}$ | -0.0020 |
|  | (-2.1370) | (-0.2004) | (-1.7980) | (-0.4067) |
| be | -0.0000 | -0.0000 | 0.0000 | -0.0000 |
|  | (-0.5504) | (-1.5492) | (0.5595) | (-1.5207) |
| vol | 0.0000$^{*}$ | 0.0000$^{***}$ | 0.0000 | 0.0000$^{***}$ |
|  | (1.7317) | (2.6092) | (0.6000) | (2.7844) |
| _cons | 0.0702$^{***}$ | 2.6554$^{***}$ | 0.0677$^{***}$ | 1.0822$^{***}$ |
|  | (8.1829) | (98.8920) | (6.3404) | (54.9920) |
| $N$ | 72983 | 61876 | 96548 | 81911 |
| adj. $R^2$ | 0.011 | 0.000 | 0.020 | 0.000 |

|  | (5) |  | (6) |  |
|---|---|---|---|---|
|  | car_window | nesg_car_window | nesg_car_window | car_window |
| climate__risk | 0.2666 | -2.4006$^{***}$ | -2.7507$^{***}$ | -0.7655$^{**}$ |
|  | (0.7255) | (-2.9007) | (-3.1667) | (-2.2960) |
| me | -0.0000$^{***}$ | -0.0000 | -0.0000 | -0.0000$^{***}$ |
|  | (-4.3715) | (-1.2861) | (-1.4031) | (-4.0712) |
| cumretx | -0.0100$^{**}$ | -0.0050 | -0.0053 | -0.0105$^{**}$ |
|  | (-2.1477) | (-1.0275) | (-0.9193) | (-2.0334) |
| be | 0.0000 | -0.0000 | -0.0000 | 0.0000 |
|  | (0.3039) | (-1.4134) | (-1.4563) | (0.3563) |
| vol | -0.0000 | 0.0000$^{**}$ | 0.0000$^{**}$ | -0.0000 |
|  | (-0.5641) | (2.1656) | (2.5265) | (-0.5742) |
| _cons | 0.0649$^{***}$ | 1.2797$^{***}$ | 1.5057$^{***}$ | 0.0630$^{***}$ |
|  | (8.0174) | (73.0671) | (81.0730) | (7.1270) |
| $N$ | 96547 | 81910 | 81910 | 96547 |
| adj. $R^2$ | 0.018 | 0.000 | 0.000 | 0.026 |

|  | (7) |  | (8) |  |
|---|---|---|---|---|
|  | car_window | nesg_car_window | car_window | nesg_car_window |
| climate__risk | -1.6274$^{**}$ | 0.3239 | -0.8076 | -1.2434$^{***}$ |
|  | (-2.4458) | (0.9196) | (-1.5976) | (-3.9896) |

| | | | | |
|---|---|---|---|---|
| me | 0.0000 | 0.0000 | -0.0000 | 0.0000 |
| | (0.3574) | (1.3629) | (-1.3898) | (1.4648) |
| cumretx | -0.0006$^{***}$ | -0.0005$^{***}$ | 0.0000 | 0.0001$^{**}$ |
| | (-3.9667) | (-8.7708) | (0.9123) | (2.4787) |
| be | 0.0000 | -0.0000 | 0.0000 | -0.0000$^{**}$ |
| | (1.6434) | (-0.1190) | (0.4775) | (-2.4762) |
| vol | -0.0000 | -0.0000 | 0.0000 | 0.0000$^{**}$ |
| | (-0.2841) | (-1.2687) | (1.0092) | (2.4353) |
| _cons | -0.0378$^{***}$ | -2.0945$^{***}$ | 0.0078$^{**}$ | -0.9504$^{***}$ |
| | (-3.5972) | (-3.8e+02) | (2.0155) | (-2.1e+02) |
| *N* | 113590 | 95750 | 113589 | 95749 |
| adj. $R^2$ | 0.011 | -0.000 | 0.007 | -0.000 |

| | (9) | |
|---|---|---|
| | car_window | nesg_car_window |
| climate__risk | -2.7222$^{***}$ | -0.7759$^{***}$ |
| | (-7.2262) | (-3.6854) |
| me | -0.0000 | 0.0000 |
| | (-1.2890) | (1.2787) |
| cumretx | 0.0003$^{***}$ | 0.0001$^{**}$ |
| | (9.0995) | (2.0757) |
| be | -0.0000 | -0.0000$^{***}$ |
| | (-0.1545) | (-2.7432) |
| vol | 0.0000 | 0.0000$^{**}$ |
| | (1.3135) | (2.5305) |
| _cons | 0.0113$^{**}$ | -0.5633$^{***}$ |
| | (2.5850) | (-1.9e+02) |
| *N* | 113590 | 95750 |
| adj. $R^2$ | 0.088 | -0.000 |

***Problem:***

A cross-sectional regression corresponds to only one event in Trump. the number of observations should just be the number of stocks in that time period. Each regression here is at 50k at least, and the 10k full sample is only 10w. there should be a mistake somewhere, the 10k full sample MEASURE is 10w, and the time period is 8/9 years, how can there not be 10w observation in one year. The reason is that the full sample was used before, and what is actually needed is to change the panel data regression to cross-section data regression.

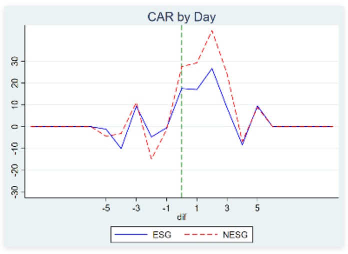
## ***Experiment 3*** **(Sep 19th):**

Change the independent variable climate_risk to climate_regulation_risk.

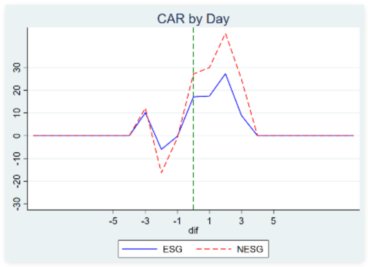
Keep the observations NON-EMPTY in event-day.

**Trump**

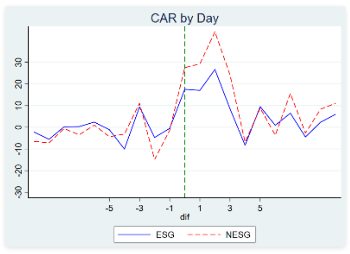
(1) event windows=±5
estimation window=[-25,-10]



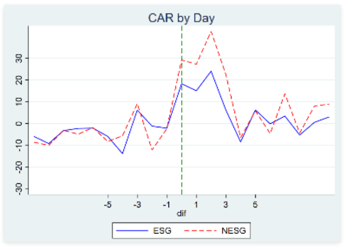
(2) event windows=±3
estimation window=[-25,-3]



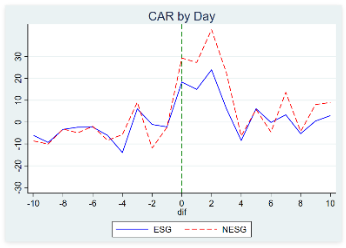
(3) event windows=±10
estimation window=[-25,-10]



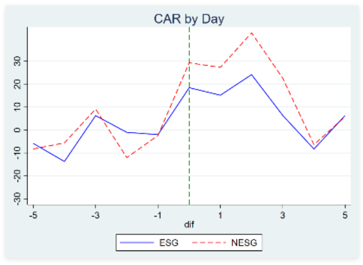
(4) event windows=±20
estimation window=[-255,-46]



(5) event windows=±10
estimation window=[-255,-30]

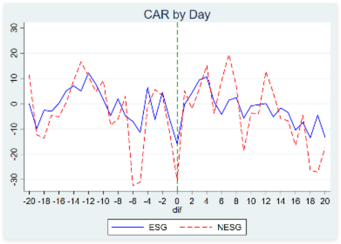


(6) event windows=±5
estimation window=[-255,-10]

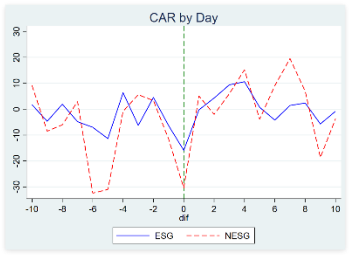


**Paris**

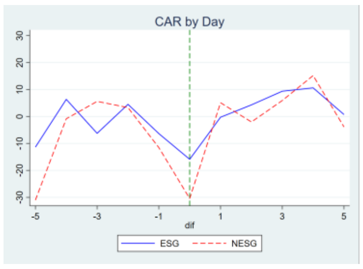
(7) event windows=±20
estimation window=[-255,-10]



(8) event windows=±10
estimation window=[-255,-10]



(9) event windows=±5
estimation window=[-255,-10]



| | (1) | | (2) | |
|---|---|---|---|---|
| | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return |
| climate__risk | -0.5527 | -0.6112 | -0.5821 | -0.6045 |
| | (-1.3835) | (-1.5927) | (-1.2655) | (-1.4715) |
| me | -0.0000*** | -0.0000** | -0.0000 | -0.0000 |

| | | | | |
|---|---|---|---|---|
| | (-2.5999) | (-2.3763) | (-1.1138) | (-0.5572) |
| wt | -0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | (-0.6012) | (0.5116) | (0.0558) | (1.1714) |
| cumretx | -0.0092$^{**}$ | -0.0099$^{*}$ | -0.0095$^{**}$ | -0.0101$^{*}$ |
| | (-2.0113) | (-1.8158) | (-2.0640) | (-1.8685) |
| mebase | 0.0000 | 0.0000$^{**}$ | 0.0000 | 0.0000$^{**}$ |
| | (1.1437) | (2.1321) | (1.5855) | (2.2054) |
| lme | 0.0000 | 0.0000 | 0.0000 | -0.0000 |
| | (1.2357) | (1.0411) | (0.2068) | (-0.3620) |
| dec_me | 0.0000 | -0.0000 | 0.0000 | 0.0000 |
| | (0.4295) | (-0.3628) | (1.0812) | (0.3565) |
| be | -0.0000 | -0.0000 | -0.0000 | -0.0000 |
| | (-0.3384) | (-0.4836) | (-0.1256) | (-0.4866) |
| _cons | 0.0642$^{***}$ | 0.0672$^{***}$ | 0.0665$^{***}$ | 0.0687$^{***}$ |
| | (7.2436) | (7.1019) | (5.9145) | (6.4765) |
| *N* | 1137 | 746 | 1138 | 746 |
| adj. $R^2$ | 0.013 | 0.022 | 0.021 | 0.031 |

| | (3) | | (4) | |
|---|---|---|---|---|
| | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return |
| climate_ _risk | 0.4229 | 0.1712 | 1.7867$^{***}$ | 1.2228$^{**}$ |
| | (0.7391) | (0.2772) | (3.0327) | (1.9844) |
| me | -0.0000$^{***}$ | -0.0000$^{***}$ | -0.0000$^{***}$ | -0.0000$^{***}$ |
| | (-2.6911) | (-2.9799) | (-3.4670) | (-3.1407) |
| wt | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | (0.1071) | (1.1807) | (0.1212) | (0.2608) |
| cumretx | -0.0083$^{**}$ | -0.0112$^{*}$ | -0.0112$^{*}$ | -0.0109$^{*}$ |
| | (-1.9929) | (-1.8309) | (-1.6526) | (-1.6683) |
| mebase | 0.0000 | 0.0000$^{*}$ | 0.0000 | 0.0000 |
| | (1.3462) | (1.6840) | (0.4591) | (0.4180) |
| lme | 0.0000 | 0.0000 | 0.0000$^{*}$ | 0.0000 |
| | (1.0579) | (1.1495) | (1.9406) | (1.6394) |
| dec_me | 0.0000 | -0.0000 | 0.0000 | 0.0000 |
| | (0.2992) | (-0.5180) | (0.6513) | (1.0776) |
| be | -0.0000 | -0.0000 | 0.0000 | -0.0000 |
| | (-0.5515) | (-0.5627) | (0.2144) | (-1.2777) |
| _cons | 0.0694$^{***}$ | 0.0793$^{***}$ | 0.0682$^{***}$ | 0.0877$^{***}$ |
| | (7.9313) | (6.8535) | (6.2941) | (6.5986) |
| *N* | 1138 | 746 | 1134 | 742 |
| adj. $R^2$ | 0.005 | 0.013 | 0.014 | 0.016 |

|  | (5) |  | (6) |  |
|---|---|---|---|---|
|  | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return |
| climate_ _risk | 0.3283 | 0.1655 | -0.7151$^{**}$ | -0.7090$^{**}$ |
|  | (0.8649) | (0.4036) | (-2.1362) | (-2.2380) |
| me | -0.0000$^{*}$ | -0.0000$^{*}$ | -0.0000$^{**}$ | -0.0000 |
|  | (-1.9244) | (-1.6914) | (-2.0225) | (-1.5976) |
| wt | 0.0000 | 0.0000 | -0.0000 | 0.0000 |
|  | (0.4095) | (1.2562) | (-0.0520) | (0.7861) |
| cumretx | -0.0090$^{**}$ | -0.0093$^{*}$ | -0.0096$^{*}$ | -0.0088$^{*}$ |
|  | (-2.0582) | (-1.9655) | (-1.9393) | (-1.8596) |
| mebase | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
|  | (0.9906) | (1.0245) | (1.1105) | (1.5568) |
| lme | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
|  | (0.6693) | (0.2787) | (0.9020) | (0.3474) |
| dec_me | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
|  | (0.8026) | (0.5080) | (0.7718) | (0.4064) |
| be | -0.0000 | -0.0000 | -0.0000 | -0.0000 |
|  | (-0.3041) | (-1.2106) | (-0.2251) | (-0.9663) |
| _cons | 0.0645$^{***}$ | 0.0717$^{***}$ | 0.0625$^{***}$ | 0.0635$^{***}$ |
|  | (8.1041) | (7.2214) | (7.2285) | (7.5228) |
| *N* | 1134 | 742 | 1134 | 742 |
| adj. $R^2$ | 0.013 | 0.020 | 0.022 | 0.030 |

|  | (7) |  | (8) |  |
|---|---|---|---|---|
|  | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return |
| climate_ _risk | -1.2622$^{*}$ | -1.5326$^{**}$ | -0.5164 | -0.5548 |
|  | (-1.9117) | (-2.4381) | (-1.1087) | (-1.1800) |
| me | -0.0000$^{**}$ | -0.0000$^{*}$ | -0.0000$^{*}$ | -0.0000 |
|  | (-2.3206) | (-1.8549) | (-1.6672) | (-1.5053) |
| wt | 0.0000 | -0.0000 | -0.0000 | -0.0000 |
|  | (0.8427) | (-0.0088) | (-1.0859) | (-1.2244) |
| cumretx | -0.0019 | -0.0077 | 0.0007 | -0.0012 |
|  | (-1.3610) | (-1.0510) | (1.0837) | (-0.2740) |
| mebase | -0.0000$^{*}$ | -0.0000$^{**}$ | -0.0000$^{***}$ | -0.0000$^{***}$ |
|  | (-1.9478) | (-2.3383) | (-2.7998) | (-2.7007) |
| lme | -0.0000$^{***}$ | -0.0000$^{**}$ | -0.0000 | -0.0000 |
|  | (-3.4944) | (-2.3726) | (-1.3775) | (-1.0461) |
| dec_me | 0.0000$^{***}$ | 0.0000$^{***}$ | 0.0000$^{***}$ | 0.0000$^{**}$ |
|  | (3.4989) | (3.4983) | (2.7101) | (2.4534) |

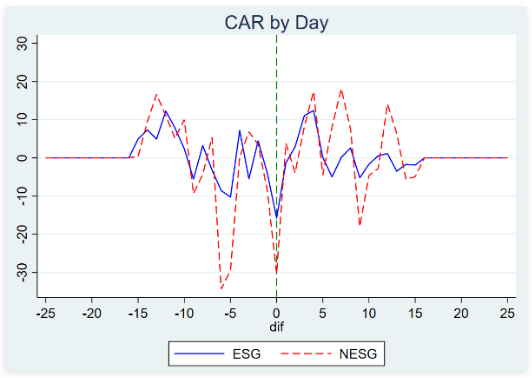
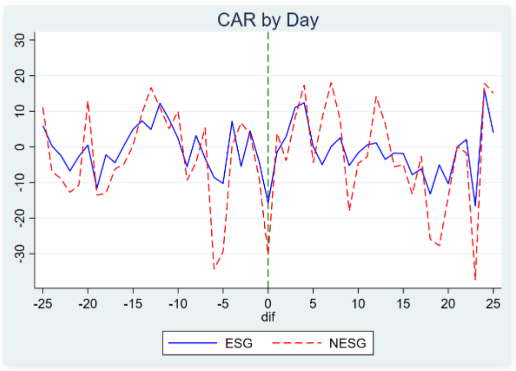
|  |  |  |  |  |
|---|---|---|---|---|
| be | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
|  | (0.9070) | (0.5031) | (0.5921) | (0.5441) |
| _cons | -0.0392*** | -0.0279** | 0.0052 | 0.0074 |
|  | (-4.0941) | (-2.0906) | (1.2867) | (0.8970) |
| *N* | 1283 | 838 | 1283 | 838 |
| adj. $R^2$ | 0.013 | 0.018 | 0.004 | 0.011 |

|  | (9) |  |
|---|---|---|
|  | cumulative_abnormal_return | cumulative_abnormal_return |
| climate__risk | -2.4333*** | -2.4862*** |
|  | (-7.1624) | (-7.6604) |
| me | -0.0000 | -0.0000 |
|  | (-0.8833) | (-0.9908) |
| wt | -0.0000 | -0.0000 |
|  | (-1.2126) | (-1.1797) |
| cumretx | 0.0009* | 0.0011 |
|  | (1.6916) | (0.3151) |
| mebase | -0.0000** | -0.0000** |
|  | (-2.5203) | (-2.3718) |
| lme | 0.0000 | -0.0000 |
|  | (0.2996) | (-0.0059) |
| dec_me | 0.0000 | 0.0000* |
|  | (1.4550) | (1.6889) |
| be | 0.0000 | 0.0000 |
|  | (0.3596) | (0.1986) |
| _cons | 0.0090** | 0.0104 |
|  | (2.2357) | (1.4498) |
| *N* | 1283 | 838 |
| adj. $R^2$ | 0.077 | 0.111 |

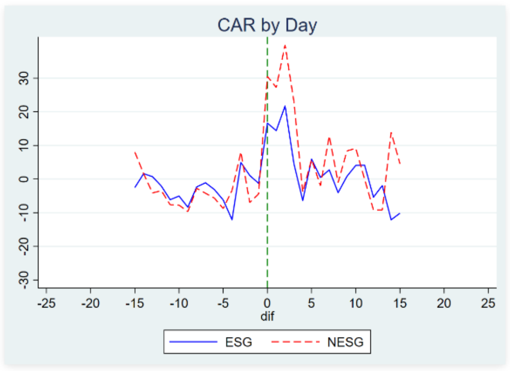
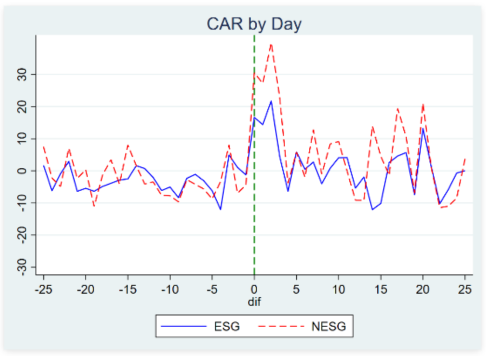
## *Experiment 4*(Sep 20th)*:*

The training sample and the time gap in between are tuned, and the training sample is used for two years, which can be collated for five days according to the grid of the loop.

**Paris**

| (1) event windows=±15<br>estimation window=[-510,-30] | (2) event windows=±25<br>estimation window=[-510,-30] |
|---|---|
|  |  |

**Trump**

| (3) event windows=±15<br>estimation window=[-510,-30] | (4) event windows=±25<br>estimation window=[-510,-30] |
|---|---|
|  |  |

| | (1) | | (2) | |
|---|---|---|---|---|
| | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return |
| climate__risk | 0.5496 | 0.3330 | -0.8006** | -1.0231** |
| | (0.6869) | (0.4023) | (-2.2471) | (-2.4332) |
| me | -0.0000 | -0.0000 | -0.0000 | -0.0000 |
| | (-1.3867) | (-1.1836) | (-1.6322) | (-1.1597) |
| wt | 0.0000* | 0.0000 | 0.0000 | -0.0000 |
| | (1.8251) | (0.7272) | (1.1291) | (-0.3553) |

| | | | | |
|---|---|---|---|---|
| cumretx | -0.0014** | -0.0093** | -0.0012 | -0.0086 |
| | (-2.3127) | (-2.2598) | (-1.1510) | (-1.2244) |
| mebase | -0.0000 | -0.0000 | -0.0000 | -0.0000*** |
| | (-1.3946) | (-1.5836) | (-1.4469) | (-2.8503) |
| lme | -0.0000*** | -0.0000** | -0.0000** | -0.0000 |
| | (-3.2176) | (-2.0473) | (-2.1573) | (-0.7337) |
| dec_me | 0.0000** | 0.0000* | 0.0000** | 0.0000** |
| | (2.0330) | (1.9512) | (2.0490) | (2.1346) |
| be | 0.0000 | -0.0000 | 0.0000 | 0.0000 |
| | (0.1347) | (-0.0001) | (0.0631) | (0.2646) |
| _cons | 0.0046 | 0.0154 | -0.0302*** | -0.0199 |
| | (0.9769) | (1.5701) | (-4.0237) | (-1.4029) |
| *N* | 1286 | 840 | 1286 | 840 |
| adj. $R^2$ | 0.000 | 0.003 | 0.000 | 0.001 |

| | (3) | | (4) | |
|---|---|---|---|---|
| | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return | cumulative_abnormal_return |
| climate_ _risk | 3.5959*** | 3.2318*** | 3.4214*** | 2.9500*** |
| | (6.2968) | (5.0746) | (6.6285) | (5.7277) |
| me | -0.0000*** | -0.0000** | -0.0000*** | -0.0000* |
| | (-2.8061) | (-2.3074) | (-3.1710) | (-1.9191) |
| wt | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | (0.4986) | (0.9274) | (0.6065) | (0.9258) |
| cumretx | -0.0114** | -0.0137** | -0.0085* | -0.0108* |
| | (-2.2114) | (-2.0247) | (-1.7463) | (-1.7948) |
| mebase | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | (1.3573) | (1.0439) | (1.1193) | (0.8205) |
| lme | 0.0000* | 0.0000 | 0.0000 | 0.0000 |
| | (1.7485) | (1.1040) | (1.5113) | (0.1774) |
| dec_me | -0.0000 | -0.0000 | -0.0000 | 0.0000 |
| | (-0.6380) | (-0.0383) | (-0.1497) | (1.2973) |
| be | 0.0000 | -0.0000 | 0.0000 | -0.0000 |
| | (0.8485) | (-0.3149) | (0.7466) | (-1.1134) |
| _cons | 0.0543*** | 0.0704*** | 0.0612*** | 0.0834*** |
| | (6.3002) | (5.8318) | (5.5938) | (6.1045) |
| *N* | 1050 | 692 | 1050 | 692 |
| adj. $R^2$ | 0.054 | 0.068 | 0.036 | 0.054 |

## *Code:*

### *1) To merge CRSP stock return and CompStat Fundamental Information[5](Python version)*

```
################################################
# Daily Stock Return & CompStat Fundamental  #
# Shengjie SONG                             #
# Date: Sep 2024                            #
################################################

import pandas as pd
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
from dateutil.relativedelta import *
from pandas.tseries.offsets import *
from scipy import stats

###################
# Compustat Block #
###################
comp = pd.read_csv('20240912\\Compustat_Fundamental.csv', low_memory=False)
comp.columns = comp.columns.str.lower()
comp['datadate'] = pd.to_datetime(comp['datadate'])
comp['year']=comp['datadate'].dt.year
# create preferrerd stock
comp['ps']=np.where(comp['pstkrv'].isnull(), comp['pstkl'], comp['pstkrv'])
comp['ps']=np.where(comp['ps'].isnull(),comp['pstk'], comp['ps'])
comp['ps']=np.where(comp['ps'].isnull(),0,comp['ps'])
comp['txditc']=comp['txditc'].fillna(0)
# create book equity
comp['be']=comp['seq']+comp['txditc']-comp['ps']
comp['be']=np.where(comp['be']>0, comp['be'], np.nan)
# number of years in Compustat
comp=comp.sort_values(by=['gvkey','datadate'])
comp['count']=comp.groupby(['gvkey']).cumcount()

filtered_data = []
```

[5] Source: This code quote is from Qingyi (Freda) Song Drechsler, rewritten for database merging. (https://www.fredasongdrechsler.com/data-crunching/fama-french)

```
chunksize = 1 * 10000
num = 0
# 可以根据实际情况调整分块大小
for chunk in pd.read_csv('CRSP_Daily_Stock_Return.csv',
chunksize=chunksize, low_memory=False):
    chunk.columns = chunk.columns.str.lower()
    chunk = chunk[['permno', 'permco', 'date', 'ret', 'retx', 'shrout',
'prc']]
#     chunk['datadate'] = pd.to_datetime(comp['datadate'])
    filtered_chunk = chunk[(chunk['date'] >= '2014-06-01') &
(chunk['date'] <= '2014-06-30')]
#     num += 1
#     print(num,filtered_chunk.count)
    # 替换为实际筛选条件
    filtered_data.append(filtered_chunk)
filtered_df = pd.concat(filtered_data)

###################
# CRSP Block      #
###################
# df_a = pd.read_csv('20240912\\CRSP_Daily_Stock_Return.csv',
low_memory=False, nrows=500000)
df_b = pd.read_csv('NAME.csv', low_memory=False)
df_b.columns = df_b.columns.str.lower()
df_b = df_b[['permno','shrcd', 'exchcd']]
# df_a = df_a[(df_a['date'] >= '2014-01-01') & (df_a['date'] <= '2019-
06-30')]
df_c = df_b[(df_b['exchcd'] >= 1) & (df_b['exchcd'] <= 3)]
merged_df = pd.merge(df_a, df_c, on='permno')
crsp_m = merged_df
print(crsp_m.columns)
# crsp_m = merged_df[merged_df['date'] <= merged_df['nameendt']]
# change variable format to int
crsp_m[['permco','permno','shrcd','exchcd']]=crsp_m[['permco','permno',
'shrcd','exchcd']].astype(int)
# Line up date to be end of month
crsp_m['jdate'] = pd.to_datetime(crsp_m['date'])  # 日期对齐到每天
#crsp_m['jdate']=pd.to_datetime(crsp_m['date'])+ MonthEnd(0)

# add delisting return
dlret = pd.read_csv('20240912\\Delisted Stocks.csv', low_memory=False)
dlret.columns = dlret.columns.str.lower()
dlret['dlstdt']=pd.to_datetime(dlret['dlstdt'])
dlret['jdate']=dlret['dlstdt']+MonthEnd(0)
```

```
crsp = pd.merge(crsp_m, dlret, how='left',on=['permno','jdate'])
crsp['dlret']=crsp['dlret_x'].fillna(0)
crsp['ret']=crsp['ret'].fillna(0)
# retadj factors in the delisting returns
crsp['ret'] = pd.to_numeric(crsp['ret'], errors='coerce')
crsp['dlret'] = pd.to_numeric(crsp['dlret'], errors='coerce')
crsp['retadj']=(1+crsp['ret'])*(1+crsp['dlret'])-1
# calculate market equity
crsp['me']=crsp['prc'].abs()*crsp['shrout']
crsp=crsp.drop(['dlret','dlstdt','prc','shrout'], axis=1)
crsp=crsp.sort_values(by=['jdate','permco','me'])

### Aggregate Market Cap ###
# sum of me across different permno belonging to same permco a given date
crsp_summe = crsp.groupby(['jdate','permco'])['me'].sum().reset_index()
# largest mktcap within a permco/date
crsp_maxme = crsp.groupby(['jdate','permco'])['me'].max().reset_index()
# join by jdate/maxme to find the permno
crsp1=pd.merge(crsp, crsp_maxme, how='inner', on=['jdate','permco','me'])
# drop me column and replace with the sum me
crsp1=crsp1.drop(['me'], axis=1)
# join with sum of me to get the correct market cap info
crsp2=pd.merge(crsp1, crsp_summe, how='inner', on=['jdate','permco'])
# sort by permno and date and also drop duplicates
crsp2=crsp2.sort_values(by=['permno','jdate']).drop_duplicates()
# keep December market cap
crsp2['year']=crsp2['jdate'].dt.year
crsp2['month']=crsp2['jdate'].dt.month
# decme=crsp2[crsp2['month']==12]
decme=decme[['permno','date','jdate','me','year']].rename(columns={'me':'de
c_me'})

### July to June dates
crsp2['ffdate']=crsp2['jdate']+MonthEnd(-6)
crsp2['ffyear']=crsp2['ffdate'].dt.year
crsp2['ffmonth']=crsp2['ffdate'].dt.month
crsp2['retx'] = pd.to_numeric(crsp2['retx'], errors='coerce')
crsp2['1+retx']=1+crsp2['retx']
crsp2=crsp2.sort_values(by=['permno','date'])
# cumret by stock
crsp2['cumretx']=crsp2.groupby(['permno','ffyear'])['1+retx'].cumprod()
# lag cumret
crsp2['lcumretx']=crsp2.groupby(['permno'])['cumretx'].shift(1)
# lag market cap
```

```
crsp2['lme']=crsp2.groupby(['permno'])['me'].shift(1)
# if first permno then use me/(1+retx) to replace the missing value
crsp2['count']=crsp2.groupby(['permno']).cumcount()
crsp2['lme']=np.where(crsp2['count']==0, crsp2['me']/crsp2['1+retx'],
crsp2['lme'])
# baseline me
mebase=crsp2[crsp2['ffmonth']==1][['permno','ffyear',
'lme']].rename(columns={'lme':'mebase'})
# merge result back together
crsp3=pd.merge(crsp2, mebase, how='left', on=['permno','ffyear'])
crsp3['wt']=np.where(crsp3['ffmonth']==1, crsp3['lme'],
crsp3['mebase']*crsp3['lcumretx'])
decme['year']=decme['year']+1
decme=decme[['permno','year','dec_me']]
# Info as of June
crsp3_jun = crsp3[crsp3['month']==6]
crsp_jun = pd.merge(crsp3_jun, decme, how='inner', on=['permno','year'])
crsp_jun = crsp_jun[['permno','date', 'jdate',
'shrcd','exchcd','retadj','me','wt','cumretx','mebase','lme','dec_me']]
crsp_jun = crsp_jun.sort_values(by=['permno','jdate']).drop_duplicates()

#######################
# CCM Block           #
#######################
ccm = pd.read_csv('20240912\\CCM_CRSP_Link_Table_CRSP.csv',
low_memory=False)
ccm.columns = ccm.columns.str.lower()
# if linkenddt is missing then set to today date
ccm['linkenddt']=ccm['linkenddt'].fillna(pd.to_datetime('today'))
# ccm1=pd.merge(comp[['gvkey','datadate','be',
'count']],ccm,how='left',on=['gvkey'])
ccm1=pd.merge(comp[['gvkey', 'datadate', 'be', 'count']], ccm, how='left',
on=['gvkey'])
ccm1['yearend']=ccm1['datadate']+YearEnd(0)
ccm1['jdate']=ccm1['yearend']+MonthEnd(6)
# set link date bounds
ccm1['jdate'] = pd.to_datetime(ccm1['jdate'], errors='coerce')
ccm1['linkdt'] = pd.to_datetime(ccm1['linkdt'], errors='coerce')
ccm1['linkenddt'] = pd.to_datetime(ccm1['linkenddt'], errors='coerce')
# Drop rows with NaT values if necessary
ccm1.dropna(subset=['jdate', 'linkdt', 'linkenddt'], inplace=True)
ccm2=ccm1[(ccm1['jdate']>=ccm1['linkdt'])&(ccm1['jdate']<=ccm1['linkenddt']
)]
ccm2['permno'] = ccm2['lpermno']
```

```
ccm2=ccm2[['gvkey','permno','datadate','yearend', 'jdate','be', 'count']]
# link comp and crsp
ccm_jun=pd.merge(crsp_jun, ccm2, how='inner', on=['permno', 'jdate'])
ccm_jun['beme']=ccm_jun['be']*1000/ccm_jun['dec_me']
```

### 2) *Merge the fundamental data to CER and Carbon Emission (Stata version)*

```
cd D:\Users\Desktop\HKUST-RA\0913
* ssc install rangejoin
* ssc install rangestat
```

1. Read stock data and CRE data and convert dates to Stata date format. 读取股票数据和 CRE 数据，并将日期转换为 Stata 日期格式

```
use CRE.dta, replace
gen filedate = substr(file_name, 1, 8)
gen cre_date = date(filedate, "YMD")
format cre_date %td
tempfile cre_data
save CRE_process.dta, replace
```

2. Read the stock data and use the rangejoin command to merge the stock data with the CRE data, matching the nearest neighbouring CRE date. 读取股票数据并使用 rangejoin 命令合并股票数据和 CRE 数据，匹配最邻近的 CRE 日期。

```
use test_stock.dta, clear
gen stock_date = date(date, "YMD")
format stock_date %td
save test_stock_process.dta, replace
```

3. Adding cik values to stock daily frequency data. 给股票日频数据加上 cik 值

```
use link_table.dta, clear
rename lpermno permno
bysort permno: keep if _n == 1
save link_table_process.dta, replace

use test_stock_process.dta,clear
merge m:1 permno using link_table_process.dta
save test_stock_process.dta, replace
drop _merge * Since two merges were performed, a column indicating a 
merge has to be deleted or the merge command will not work.
```

4. Use the joinby command to match all possible combinations and calculate the date difference, keeping the record with the smallest date difference. 使用 joinby 命令匹配所有可能的组合，并计算日期差值，保留日期差值最小的记录。

```
joinby cik using CRE_process.dta
keep if stock_date_stata >= cre_date_stata
gen date_diff = stock_date - cre_date
bysort cik stock_date (date_diff): keep if _n == 1
```

### *3) Compute CER and Cross-sectional regression*[6]

**Data Preparation: Security_id, Market return, Security return**

Before experiment, we have already collect a data set including three main variables in event study: Security_id: permno, event occurrence's date(date), markettype(exchcd). Market return: vwretx. Security return: Value-Weighted Return (excluding dividends) (retx)

In our case, we already have data with a data variable, which we call "stock_date", and a company identifier, which we called "permno".

**Cleaning the data and Calculating the Event and Estimation Windows**

We need to create a variable, **dif**, that will count the number of days from the observation to the event date. Note that when using trading days, Saturday and Sunday are excluded, however dif needs to be a continuous uninterrupted variable.

```
drop if vwretx==.
gen event_date = date("14dec2015", "DMY")
format event_date %td
save test_process.dta,replace  *Adding this line and above if you need.
use test_process.dta,replace
sort cik stock_date
```

For trading days, we first need to create a variable that counts the number of days within each company id.

```
by cik: gen datenum=_n
```

Then we determine which observation occurs on the event date. We create a variable with the event date's day number on all of the observations within that company id.

```
by cik: gen target=datenum if stock_date==event_date
egen td=min(target), by(cik)
drop target
```

Finally, we simply take the difference between the two, creating a variable, dif, that counts the number of days between each individual observation and the event day.

```
gen dif=datenum-td
```

Next, we need to make sure that we have the minimum number of observations before and after the event date, as well as the minimum number of observation: before the event window for the estimation window, Let's say we want 2 days before and after the event date (a total of

[6] Source and Cite: https://mp.weixin.qq.com/s/VXxkQ2jlJFFQlGETfStT0Q

5 days in the event window) and 30 days for the estimation window. (You can of course change these numbers to suit your analysis.)

```
by cik: gen event_window=1 if dif>=-2 & dif<=2
egen count_event_obs=count(event_window) ,by(cik)
by cik: gen estimation_window=1 if dif<-30 & dif>=-60
egen count_est_obs=count(estimation_window), by(cik)
replace event_window=0 if event_window==.
replace estimation_window=0 if estimation_window==.
```

**Estimating Normal Performance**[7]

We will run a seperate regression for each company using the data within the estimation window and save the alphas (the intercept) and betas (the coeffcient of the independent variable).

Before this process, maybe you need to filter data.

```
            drop if dif==.
            local threshold = date("event_date", "YMD")
            bysort id: egen min_date = min(date)
            drop if min_date > `threshold'
            drop min_date
            drop if retx==.b
            replace retx = 0 if retx == .c
```

```
set more off
gen predicted_return=.
egen id=group(cik)
*drop if dif==.
forvalues i=1(1)280{
    count if id == `i' & dif == 0
    if r(N) > 0 {
    l id cik if id==`i' & dif==0
    reg retx vwretx if id==`i' & estimation_window==1
    predict p if id==`i'
    replace predicted_return=p if id==`i' & event_window==1
    drop p
    }
}
```

**Abnormaland Cumulative Abnormal Returns**

[7] Note that it is necessary to first observe whether there are missing variables and whether they need to be excluded. The code here does not include this processing.

The daily abnormal return is computed by subtracting the predicted normal return from the actual return for each day in the event window. The sum of the abnormal returns over the event window is the cumulative abnormal return.

```
sort cik stock_date
gen abnormal_return= retx-predicted_return if event_window==1
by cik:egen cumulative_abnormal_return = total(abnormal_return)
```

**Testing for Significance**

We are going to compute a test statistic, test, to check whether the average abnormal return for each stock is statistically different from zero. Note that the difference between sample standard deviation and the population standard deviation.

```
sort cik stock_date
by cik: egen ar_sd=sd(abnormal_return)
gen test=(1/sqrt(5))*(cumulative_abnormal_return / ar_sd)
list cik cumulative_abnormal_return test if dif==0

export excel cik event_date cumulative_abnormal_return
"stats.xls" if dif==0, firstrow(variables) replace
```

**Testing Across All Events**

```
reg cumulative_abnormal_return if dif==0, robust
```

### *4) Graphing and cross-sectional regression*[8]

```
gen esg=.
replace esg=1 if climate__risk!=0
replace esg=0 if climate__risk==0

* 按 climate_regulation_risk 分组并累加收益
bysort esg dif: egen cum_return = total(abnormal_return)
* 创建两个新的变量，分别存储 climate_regulation_risk 为 1 和 0 的累加收益
gen esg_car = cum_return if esg == 1
gen nesg_car = cum_return if esg == 0

twoway (line esg_car dif, lcolor(blue) lpattern(solid)) ///
       (line nesg_car dif, lcolor(red) lpattern(dash)), ///
       title("CAR by Day") ///
       xlabel(-5(1)5) ///
       ylabel(-2(0.5)2) ///
       legend(order(1 "ESG" 2 "NESG")) ///
```

[8] The regression commands are simple and can be debugged to suit your needs.

```
       xline(0, lcolor(green) lpattern(dash))

reg cumulative_abnormal_return climate__risk company_features
est store m1
esttab m1 using regression.rtf, replace b(%6.4f) t(%6.4f) nogap ar2
star(* 0.1 ** 0.05 *** 0.01)
```