# ETH

**Swiss Federal Institute of Technology Zurich**

**Department of Mathematics**

Master Thesis                                                   Fall 2019

Tianqi Wang

# An In-Depth Analysis of the Impact of ESG Investing on Returns Using Large-Scale News Data

Submission Date:   August 26th 2019

Adviser          Prof. Peter Bühlmann
Supervisor:   Dr. Ryohei Hisano

# Acknowledgements

I would first like to express my deepest gratitude to my thesis advisor Dr. Ryohei Hisano for introducing me to this interesting topic and providing me with support and guidance throughout this process. I thank Prof. Sornette Didier, Dr. Ke Wu, and Chair of Entrepreneurial Risk for giving me the opportunity to work on this thesis here and providing me with additional data required for the analysis. Furthermore, I thank Prof. Peter Bühlmann for co-supervising this master thesis. Last but not the least, I want to express my gratitude to my parents and boyfriend for their support and encouragement all the time.

# Abstract

By using the Dow Jones Adverse Media Entity dataset that covers 2012 to 2019, this thesis finds that environmental, social, and governance(ESG) risks generate negative short-run stock returns with a negative cumulative abnormal return of -0.32% for a 21-day event window. This discovery demonstrates that the market fails to fully incorporate value-relevant ESG information into stock prices in the short-run, and thus implies potentially profitable investment strategies. An underperformance of controversial firms in long-run with a monthly alpha of -4.37% is also reported. This result is robust to outliers and different risk model specifications, but not to industry or certain combinations of firm characteristics. This shows that the underperformance may stem from underperforming industries and common characteristics of controversial firms rather than ESG incidents. This thesis also examines the financial performance of several ESG investment strategies, including negative and relative screening. The results show that while a equal-weighted portfolio of worst ESG-performing stocks underperform the best ones by 0.54% and a value-weighted portfolio overperforms S&P 500 portfolio by 0.2%, negatively screened portfolios do not systematically overperform or underperform the benchmarks.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Socially responsible investing(SRI) is broadly defined as an investment process that involves identifying companies based on their corporate social responsibility profiles(CSR). ESG investing belongs to the class of SRI and specifically considers integrating environmental, social, and governance factors into investment decisions(Renneboog, Horst, and Zhang 2008). It reflects investors' wish of 'doing well by doing good.' In other words, in addition to the financial utilities, ESG investors also strive for non-financial utilities by holding portfolios that are consistent with social norms and values.

In modern era, SRI became more formalized with the growth of mutual fund industry. As the fund management industry grew, activists started to take SRI as an opportunity for shareholders to influence corporate behaviours. For instance, between the 1970s and 90s, pressures on fund managers to avoid investing in South African companies contributed to the ending of Apartheid, an institutionalized race segregation in South Africa. In the 1980s and beyond, the awareness of climate change and global warming, along with industrial disasters such as Bhopal gas leak and Exxon Valdez oil spill, have brought up concerns regarding environmental issues and drives investors away from these controversial firms.

Nowadays, SRI has grown into a major force in the global market. As revealed in The Global Sustainable Investment Review(GSIR) released by Global Sustainable Investment Alliance, SRI assets in five major markets(Europe, the United States, Japan, Canada, Australia, and New Zealand) approximated \$30 trillions at the beginning of 2018, featuring a 34% growth in two years. They constituted more than 50% of total professionally managed assets in Canada, Austria, and New Zealand, almost half in Europe, 26% in the United States, and 18% in Japan.

Common ESG strategies used by individuals or institutional investors include negative screening, besting in-class-screening, active ownership, ESG integration, and themed in-

vesting. Negative screening, which entails excluding certain companies or sectors from a fund or portfolio based on ESG criteria, is by far the most widely used strategy. It is also unarguably the most extensively studied strategy in the field of finance.

As reported by Global Sustainable Investment Alliance, there are two main motivations for fund managers to apply ESG investing. The first one is to "achieve social and environmental benefit or to fulfill their firms' missions." The second one is to "minimize risk and improve financial performance over time." While the non-financial utility of ESG investing is unquestionable, its financial performance remains unclear. Therefore, the conversations about ESG investing has shifted from the social responsibility issues to questions about the impact of ESG incidents on stock price and the methods for integrating non-financial information into investment theories.

In past decades, many empirical studies have been conducted in ESG investing, but they show inconclusive and mixed results regarding its financial performance. While many studies find a zero-effect(Bauer, Koedijk, and Otten 2005) or a negative effect(Renneboog et al. 2008; Hong and Kacperczyk 2009), some studies find evidence that SRI improves investment performance(Edmans 2011). The source of controversial results could come from many aspects, such as the exact subject of study, sample period, country, and methodologies used.

One important source of uncertainty is the lack of clear definitions and metrics for ESG responsibility, which forms an obstacle to assess the value of the investment. To be more specific, a traditional way of defining low ESG responsibility stocks is by the notion of 'sin stocks'. Sin stocks is defined by the industrial sector of the firm and often includes alcohol, tobacco, gambling, sex-related industries and weapon manufacturers(Hong and Kacperczyk 2009). However, this methods defines ESG responsibility broadly on an industry-level and ignore characteristics of individual firms. Furthermore, it does not allow for researchers to study the changes in stock prices around ESG shocks.

Alternatively, many recent studies start to use ESG ratings as a measure for ESG performance(Glosner 2017), with the emergence of specialized ESG rating agencies. Several examples include RepRisk, Sustainalytics, and MSCI(formerly named KID). Typically, information sources on the predefined ESG issues are screened and then used to calculate the ESG score on a monthly basis. A decrease in the rating of certain amount is usually seen as a negative shock in the ESG issues. However, as the rating is reported on a monthly basis, it introduces time latencies in event reporting.

Different from these two approaches, this thesis takes an event study approach by utilizing a large dataset on negative ESG-related news curated by Dow Jones for professional use. Our event study approach allows us to identify ESG shocks directly and to separate

controversial stocks from well-behaving ones by using the occurrences of negative news as a proxy of bad behavior. Compared with the broad industry classification, we check the ESG issues at an individual firm level, and thus making it easier to measure the impact of negative news in a more direct manner. In contrast to the ESG rating approach, our data allows us to study the short-term effect in a timely manner.

This thesis is consisted of four parts. In the first part, we introduce some theoretical foundations related to returns of ESG investing and review related literature. In the second part, we briefly explain the datasets used throughout the thesis and also provide some basic descriptive analysis of the data. In the third part, we investigate the relationship between companies' ESG performance and stock returns in both the short and long run, using methodologies of event study, Jensen's alpha, and cross-sectionally time-series regression. In the last part, we compare the ESG-based investment strategies, specifically negative screening and relative screening, with passive investment in order to study whether investors can profit from ESG-investing.

The thesis contributes to the literature on ESG investing in the following ways. First of all, by utilizing a large dataset on ESG related news, we perform a comprehensive analysis of the impact of ESG related news on stock returns. As would be detailed in the next section, our ESG related news dataset categorizes news events into 17 categories. These categories enable us to measure the impact as well as confidence intervals of the impact of ESG related news on stock returns. Secondly, we perform an in-depth analysis of the impact of ESG related news events on stock returns paying particular attention to the robustness of the results. We show that negative impact of ESG related news event seems to be robust to long-run impact, outliers, different risk model specification, but not to industry classification and firm characteristics. Thirdly, we also examine the financial performance of several ESG investment strategies, including both negative and relative screening.

# Chapter 2

# Background

In this section, we briefly summarize the theoretical considerations to ESG investing and provide a review of related empirical literature.

## 2.1 Theoretical Considerations of ESG investings

Current research on ESG investing shows both evidence of overperformance and underperformance of stocks with high ESG profiles. In the following, we summarize their main arguments.

Some studies show that ESG investing can improve investment performance relative to passive investing(Edmans 2011). A theoretical explanation is that the stock market undervalues ESG risks. In other words, the value effects of ESG events is not fully captured by the stock market. To further explain, while negative ESG events can almost always destroy shareholder value, as the firms have to pay severe penalties and bear the damages to their intangibles such as reputations, positive events can increase the shareholder value in the opposite way. However, investors may ignore ESG information because they are either obsessed with the short-term financial metric and too myopic and to incorporate long-term fundamentals(Porter 1992), or they fail to recognize the importance of ESG. Moreover, the impact on intangible assets is often underestimated by the market and is only reflected when manifested in some tangible outcomes. Evidence of underreaction to intangibles includes R & D costs, patent citations, advertising, and software development costs.

A second reason for the outperformance of high ESG stocks is that ESG investing has become more popular over time. Therefore, a growing demand for high ESG stocks can push up their prices, and a decreasing demand for low ESG stocks can result in lower stock

prices.

Demand effect is also one reason that some researchers argue that high ESG stocks underperforms relative to low ESG stocks(Hong and Kacperczyk 2009) or the market. To be more specific, when ESG investors ignore certain controversial firms, their stock prices can become undervalued. While these firms have low initial returns, they may yield high subsequent returns relative to high ESG stocks. Another reason is that firms in industries shunned by ESG investors, such as alcohol and tobacco, have incentives for conservative accounting because of more scrutiny from regulators. This will lead to underreaction and thus high subsequent returns.

Other studies show that there is no underperformance or overperformance of high ESG stocks because of the perfectly elastic demand curve(Bauer et al. 2005). To further explain, ESG investing might not be sufficiently widespread to affect prices. Moreover, arbitrageurs can offset any effect of ESG investors by trading in the opposite direction.

## 2.2   Literature Review

A vast body of literature is devoted to studying the financial impact of ESG investing. The literature can be split into different categories according to the source of information they used. A traditional way of screening ESG stocks is to avoid the so-called "sin stocks". This screening method is rooted in Islamic religious principles and has thousands of years of history. However, in modern times, with the formalization of financial reporting procedures and the era of big data, ESG researchers are accessible to more information sources. Rather than sin stocks screening, modern literature approaches can be split into three main categories: i) information voluntarily disclosed by the company, ii) information from a rating agency, and iii) news information.

Compared with (i) and (ii), (iii) is a more neutral and reliable approach. On one hand, (i) can be cooked, which is evident in the Enron Scandal. Enron had fooled regulators for long with fake holding and off-the-books accounting, until the problem was discovered and investigated and the company collapsed. On the other hand, (ii) can be influenced by a conflict of interest. Efing and Hau (2013) study 6500 structured debt ratings produced by Standard and Poor's, Moody's, and Fitch, and found that ratings by agencies are "biased in favor of issuer clients that provide the agencies with more rating business." This phenomenon is evident in the subprime crisis.

Hvidkjaer (2017) provides a nice overview of ESG literature.

### 2.2.1   Sin Stocks

Hong and Kacperczyk (2009) is one of the most cited articles addressing the return effects of investing in sins stocks. They found that a classical Triumvirate of sin stocks(consisted of tobacco, alcohol, and gambling companies) of 156 US stocks outperform their industrial comparable stocks during 1965-2006. To be more specific, the result shows that applying Carhart four-factor model to the portfolio of returns of sin stocks net of their industry comparables yields a monthly alpha of 26 bps at 5% significance level. Moreover, a cross-sectional returns model finds a monthly excess performance of 30 bps at 5% significance level.

Rather than defining sin stocks according to the broad industry classification, Trinks and Scholtens (2017) selects stocks at individual level based on fourteen potentially controversial issues. They show that sin stocks exhibit high returns during 1991-2012 in 94 international markets. Specifically, they show that the value-weighted TotalSin portfolio, which is consisted of over 1,600 stocks, outperforms the global Fama-French four-factor benchmark by 91-104 bps per month, which is significant at 1% level. Also, note that they apply a median regression to address the non-normality of excess returns.

Similar to the idea of sin stocks, Edmans (2011) conducts social screens by selecting firms with high employee satisfaction, which is defined as being on the list of "100 best companies to work for in America". He shows that firms with high employee satisfaction exhibit high future stock returns. As the paper noted, a value-weighted portfolio of the listed firms earned an annual alpha of 3.5% from 1984 to 2009 and it is 2.1% above industrial benchmarks. The results are robust to controls for firm characteristics, different weighted methodologies, and removal of outliers. In addition to the industry-matched returns mentioned in Hong and Kacperczyk (2009), Edmans also uses characteristic-adjusted returns. This controls for interaction effects, which may potentially cause the model to misprice certain stocks. Moreover, Fama-Macbeth cross-sectional regressions confirm the results of the factor regressions.

### 2.2.2   Information Voluntarily Disclosed by the Company

Jacobs, Singhal, and Subramanian (2010) apply event study methodology to study the stock market response to the announcement of various types of corporate environmental initiatives, such as environmental business strategies, eco-friendly products, renewable energy, and recycling. The announcements include both Corporate Environmental Initiatives(CEI), which is the self-reported corporate endeavors to avoid environmental impacts, and Environmental Awards and Certifications (EACs), which is about recognition granted

by a third party. The sample includes 780 announcements between 2004 and 2006. They find mostly insignificant results, except for significantly negative returns for voluntary emission reduction and positive returns for environmental philanthropy and ISO14001 certification.

Fisher-Vanden and Thorburn (2011) use the same methodology to study the abnormal returns around announcements that firms have joined two environmental programs, namely the U.S. Environmental Protection Agency's Climate Leaders program and Ceres. The analysis is based on a sample of 117 announcements by large U.S. firms during the period 1993 to 2008. The results show that the market values decrease by around 1% when firms join the Climate Leaders program, which is statistically significant. By contrast, abnormal returns are insignificant when firms join Ceres.

### 2.2.3   Information from Rating Agencies

Kempf and Osthof (2007) use KLD rating data to construct value-weighted portfolios for the period 1992-2004 based on three strategies, namely negative, positive, and best-in-class screenings. Their first finding is that high ESG-rated portfolio performs better than low-rated portfolio for certain qualitative criteria. Specifically, a long-short strategy yields a positive alpha of up to 8.4% per year using Carhart four-factor model. Furthermore, the maximum alpha is obtained by the best-in-class screening, choosing stocks with extreme ratings and applying several screenings at the same time. However, negative screening alone does not lead to any significant performance effect. Statman and Glushkov (2009) generally confirm the results of Kempf and Osthof (2007) based on 1992-2007 KLD rating data. Both of the studies also show that portfolios formed on community and employee relations criteria result in high excess returns, while those formed on diversity, environment, products and human rights do not have significant results.

Humphrey and Tan (2014) use KLD data to construct portfolios using positive and negative screening strategies, during the sample period 1996-2010. They show that the ESG portfolio returns are lower than, but not significantly different from the benchmarks. They also compare the Sharpe ratios and found that none of the difference is significant.

Kruger (2015) uses event study methodology to study the stock market response to both positive and negative corporate ESG events, identified by KLD data around seven ESG issues. He finds a significantly negative response to negative ESG events and a slightly negative reaction to positive ESG events. To be more specific, the mean 21-day cumulative abnormal return is approximately -111 bps for negative events and -47 bps for positive events.

Halbritter and Dorfleitner (2015) shows that the outperformance of ESG portfolio disappears after the initial sample period in Kempf and Osthof (2007). Moreover, they investigate the returns to ESG ratings of two other agencies, namely ASSET4, which covers a sample period from 2003 to 2012, and Bloomberg, which covers 2006 to 2012. Alphas are generally also insignificant, confirming the KLD results. However, the Fama-Macbeth cross-sectional regression shows a different result: while the excess returns to KLD ratings remain insignificant, those obtained from ASSET4 and Bloomberg are statistically significant.

### 2.2.4    News Information

Flammer (2013) identifies eco-harmful and eco-friendly events based on word search of the Wall Street Journal(WSJ) articles obtained from Factiva, which is one of the major newspaper databases. The search provides 117 eco-friendly and 156 eco-harmful events, ranging from 1980 to 1999. By applying event study methodology, Flammer (2013) finds a negative market reaction to eco-harmful events and a positive reaction to eco-friendly events. She also examines the trend of market reaction over decades and shows that the reward for eco-friendly initiatives has decreased over time while the penalty for eco-harmful events has increased.

# Chapter 3

# Data Description

The primary dataset we used in this data is purchased from Dow Jones. The dataset is called Dow Jones Adverse Media Entity List, and it is said to be created by curating news from about 10,000 sources all over the world in the Factiva dataset. The data covers the negative ESG-related events from January 2012 to January 2019. There are 2,967 firms and 16,380 events involved during this period.

For the robustness study, we collected additional data of company industry, market capacity, dividend yield, and book-to-market ratio for all controversial firms from the Thomson Reuters database. In order to compare the controversial firms with other firms, we also obtained these data for all US public stocks traded on NASDAQ, NYSE, and AMEX, which include around 6,900 stocks, as well as S&P 500 firms. To convert stock symbols from ticker format to RIC format for queries on Thomson Reuter, we applied the inherent function of Eikon. we compared the matched company names and manually checked entries where names differ in order to ensure that the matching is correct. dis

In our dataset, the negative ESG-related news are divided into four main categories and more subcategories, namely competitive/financial(information, anti-competitive, ownership, management, financial, association), environmental/production(service, supply, environmental), social/labour(discrimination, workplace, workforce, human), and regulatory (regulatory, fraud, corruption, sanctions). The distribution of number of negative news per category is as figure 3.1. The most frequent subcategories appeared are product/service, regulatory, and information, and they altogether account for 55% of the incidents. Few observations lie in the subcategory of association, sanction, corruption, human, and financial.
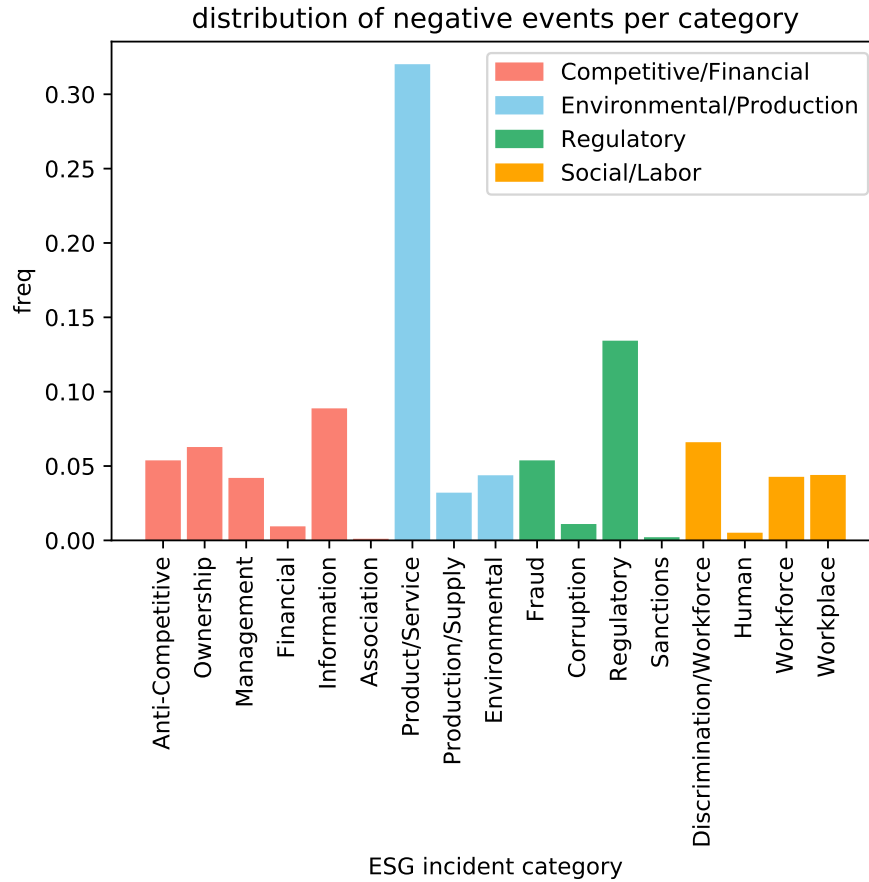
Figure 3.1: Distribution of number of ESG incidents per subcategory

|  | subcategories | | | | | |
|---|---|---|---|---|---|---|
| competitive /financial | info | anti-competitive | ownership | mgmt | financial | assoc |
|  | 843 | 508 | 594 | 395 | 83 | 3 |
| social/labor | discrim | workplace | workforce | human | | |
|  | 625 | 414 | 402 | 42 | | |
| regulatory | regulatory | fraud | corruption | sanction | | |
|  | 1279 | 508 | 98 | 13 | | |
| environmental /production | service | supply | envir | | | |
|  | 3060 | 300 | 412 | | | |

Table 3.1: Number of ESG events per event subcategory

A visualization of the stock returns around event date can help us better understand the impact of ESG incidents. Figure 3.2 shows the stock prices of Celsion, a biopharmaceutical company. The failure of its experimental liver cancer treatment ThermoBox in the Phase 3 heat trial in January 2013 causes a plunge of 80% in its stock price.
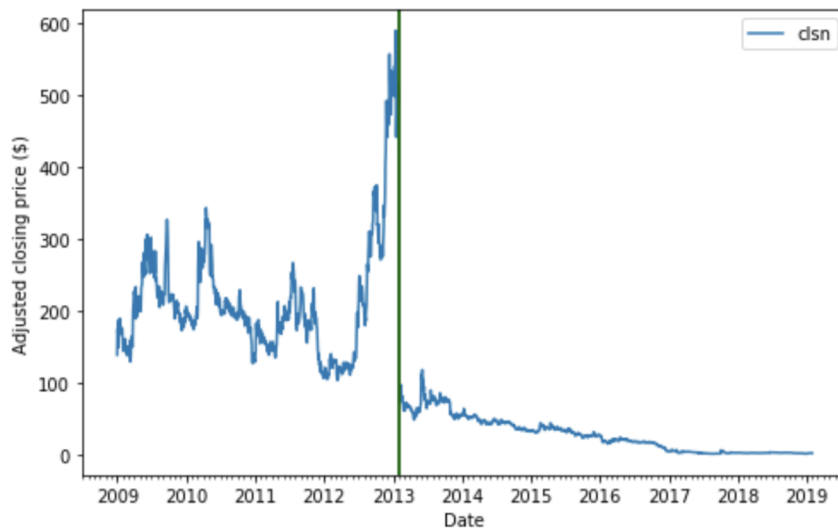
Figure 3.2: Stock prices of Celsion

An inspection of the yearly trend shows that the number of incidents nearly doubles, from 1,225 events in the year 2012 to 2,420 events in 2018. This partly reflects an expanding coverage universe of the dataset and an increasing attention to ESG incidents, and partly indicates that companies become more negligent in their ESG obligations.
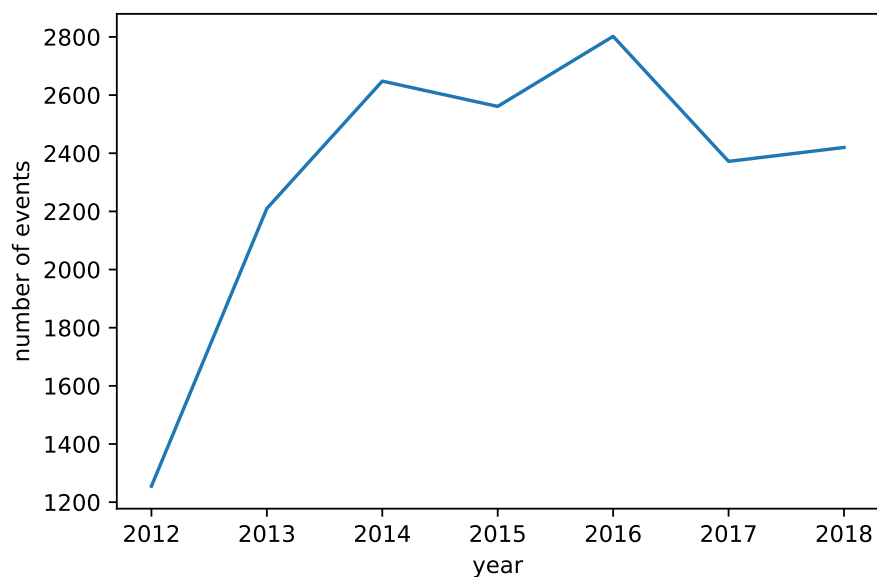


Figure 3.3: Trend of ESG incidents per year from 2012 to 2018

| year | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 |
|---|---|---|---|---|---|---|---|
| total number of incidents | 1255 | 2210 | 2648 | 2561 | 2802 | 2372 | 2420 |
| median market capitalization(bn $) | 1.85 | 2.27 | 2.74 | 2.63 | 2.28 | 2.73 | 3.12 |

Table 3.2: Number of total ESG incidents and median market capitalization by year from 2012 to 2018

The distribution of number of negative news per company is shown in figure 3.4. Around 57% of the companies have recurring negative news, up to February 2019. The median time interval between two events is 52 days. Figure 3.5 reveals that as the number of previous incidents for a company increases, the time interval between incidents decreases. It shows that negative events are persistent, which is another reason why investors should pay attention to ESG incidents.
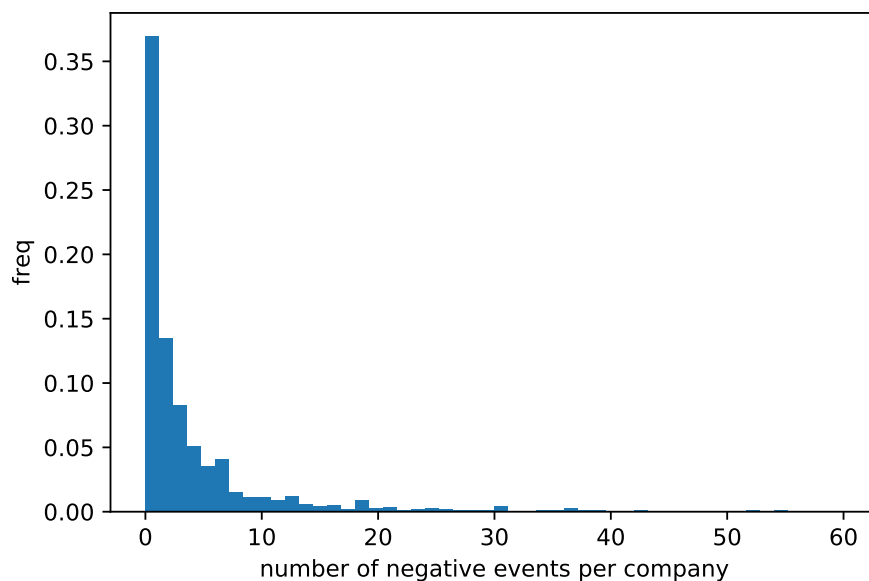
Figure 3.4: Distribution of number of ESG incidents per company

|  | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|
| number of events per company | 5.56 | 14.65 | 1.00 | 1.00 | 2.00 | 4.00 | 297.00 |
| number of days between incidents | 15.20 | 23.88 | 0.00 | 5.00 | 9.00 | 17.00 | 271.50 |

Table 3.3: Summary statistics of number of ESG incidents per company and number of days between ESG incidents
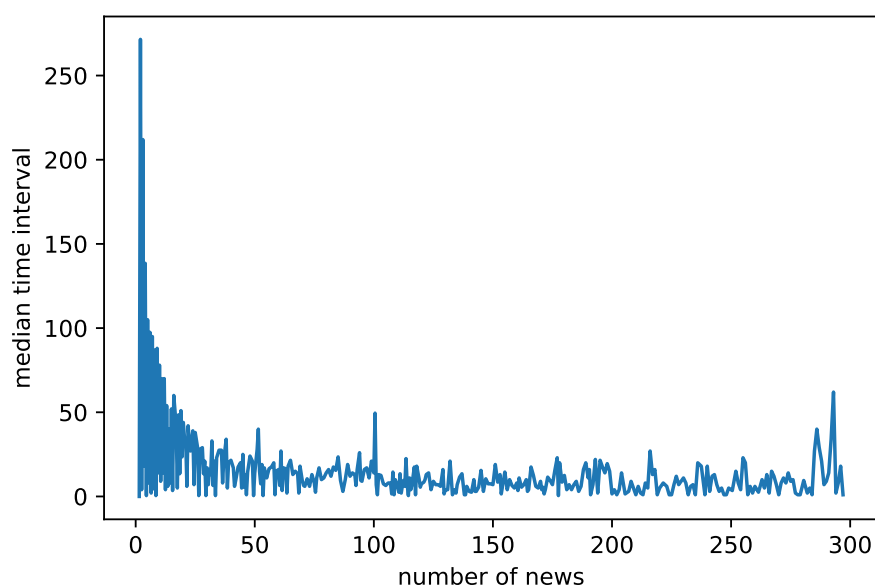


Figure 3.5: Distribution of number of days between ESG incidents

The median market capitalization for controversial companies in our dataset increases from \$1.8 billion as of the year 2012 to \$3.1 billion in 2018. Most of our controversial

firms are small to medium cap. This is a contrast to RepRisk data, which shows a median market cap of around \$49 billion for controversial firms.
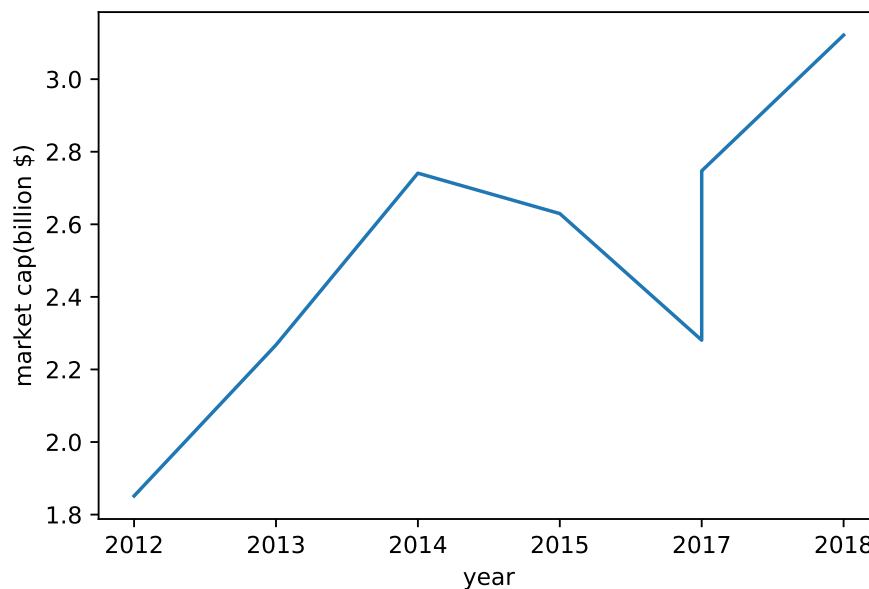


Figure 3.6: Median market capitalization per year from 2012 to 2018

Figure 3.7 is the industry distribution for companies that experiences ESG incidents, compared with the universe of all public US-traded stocks. The controversial firms are mostly concentrated in capital goods, pharmaceutical, and banking industries. Their industry compositions are mostly similar to that of all firms, but more concentrated in retailing, utilities, and material industries, while less concentrated in pharmaceutical and banking industries.
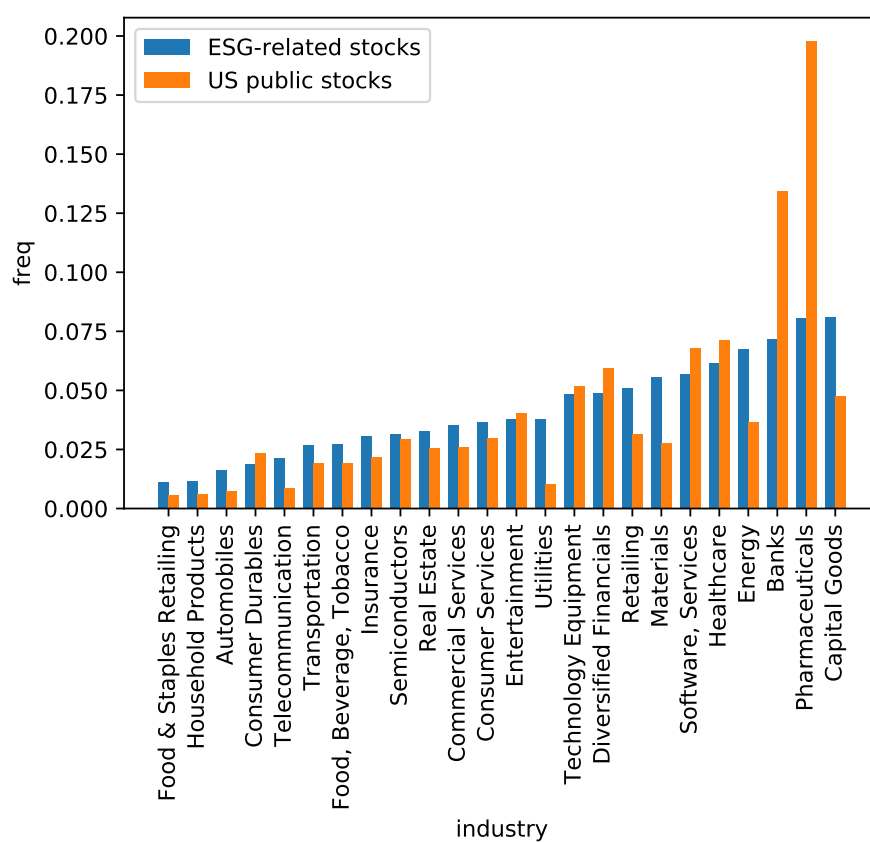
Figure 3.7: Distribution of industries of controversial firms, compared with the universe of all US public traded stocks

|                         | controversial firms | all US traded firms |
|-------------------------|---------------------|---------------------|
| Food, Staples Retailing | 0.011               | 0.006               |
| Household Products      | 0.012               | 0.006               |
| Automobiles             | 0.016               | 0.008               |
| Consumer Durables       | 0.019               | 0.023               |
| Telecommunication       | 0.022               | 0.009               |
| Transportation          | 0.027               | 0.019               |
| Food, Beverage, Tobacco | 0.027               | 0.019               |
| Insurance               | 0.031               | 0.022               |
| Semiconductors          | 0.031               | 0.029               |
| Real Estate             | 0.033               | 0.026               |
| Commercial Services     | 0.035               | 0.026               |
| Consumer Services       | 0.037               | 0.023               |
| Entertainment           | 0.038               | 0.040               |
| Utilities               | 0.038               | 0.010               |
| Technology Equipment    | 0.048               | 0.052               |
| Diversified Financials  | 0.049               | 0.060               |
| Retailing               | 0.051               | 0.032               |
| Materials               | 0.056               | 0.028               |
| Software and Services   | 0.057               | 0.068               |
| Healthcare              | 0.061               | 0.071               |
| Energy                  | 0.067               | 0.037               |
| Banks                   | 0.072               | 0.134               |
| Pharmaceuticals         | 0.080               | 0.198               |
| Capital Goods           | 0.081               | 0.048               |

Table 3.4: Industry composition of controversial firms and all US public traded stocks

# Chapter 4

# Event Study

## 4.1 Methodology

Event study is a statistical method for examining the impact of an event on the value of a firm. It is widely used to investigate the stock market responses to macroeconomic events such as market shock, as well as to corporate events, such as earning announcement, merge and acquisition, and dividend announcement. In this study, we use event study methodology to study the stock market response to ESG-related news in the short-run. This addresses the problem as to how the stock market perceives the value of an ESG shock.
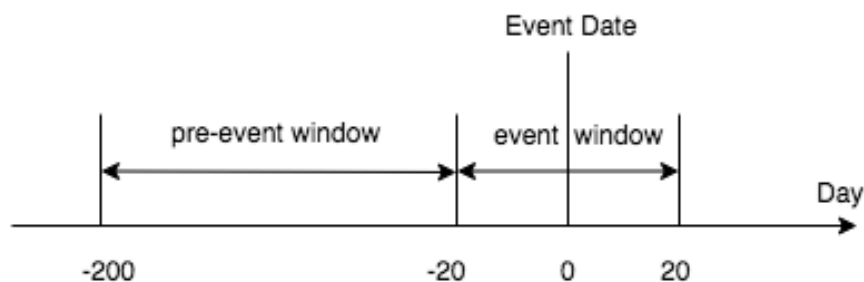


Figure 4.1: Timeline of event study

### 4.1.1 Expected Return Models

We proceed with an event study framework following Mackinlay (1997). For every event, we use a pre-event window to estimate the expected stock returns.

$$R_{it} = E[R_{it}|X_t] + e_{it} \tag{4.1.1.1}$$

19

, where $R_{it}$ is the return for stock i at time t, $X_t$ is the conditioning information, and $e_{it}$ is the abnormal return.

There are a variety of expected return models, such as market model, capital asset pricing model(Treynor 1961, Treynor 1962, Sharpe 1964, Lintner 1965, Mossin 1966), Fama-French three factor model(Fama and R.French 1992), and Carhart four factor model(Carhart 1997). We introduce each of them below. These asset pricing models are also used in the portfolio regressions in the next section.

These estimated coefficients are used to compute the expected returns during the event window as follows.

$$E[R_{it}|X_t] = \alpha_i + \beta_i R_{it} \tag{4.1.1.2}$$

**Market Model**

Market model is also called single index model. It is a common specification that models the return of individual security or portfolio on the market portfolio. It is written as:

$$R_{it} = \alpha_{it} + \beta_{1i} R_{MKT_t} + \epsilon_t \tag{4.1.1.3}$$

, where $R_{MKT}$ is the return of market portfolio. As it is impossible to collect returns of all the stocks, it is a common practice to use S&P 500 index as the market return.

**Capital Asset Pricing Model(CAPM)**

CAPM was introduced by Treynor (1961), Sharpe (1964), Lintner (1965), and Mossin (1966). It builds on the earlier work of Markowitz (1952) on diversification and modern portfolio theory. It assumes that the expected return of a security is based on risk-free-rate plus a risk premium of the market return over the risk-free-rate. The risk-free-rate usually used is the one-month treasury bill rate.

$$R_{it} - Rf_t = \alpha_{it} + \beta_{1i}(R_{MKT_t} - Rf_t) + \epsilon_t \tag{4.1.1.4}$$

, where the additional parameter $Rf_t$ is the risk-free-rate.

**Fama-French Three-factor Model**

Fama and French(Fama and R.French 1992) three-factor model expands on CAPM by adding size and value risk factors. This model considers the fact that value stocks ourperform growth stocks and that small cap stocks outperform big cap stocks on a regular basis. By including these two additional factors, the model adjusts for this outperforming tendency.

$$R_{it} - Rf_t = \alpha_{it} + \beta_{1i}(R_{MKT_t} - Rf_t) + \beta_{2i}SMB_t + \beta_{3i}HML_t + \epsilon_t \quad (4.1.1.5)$$

, where SMB is the size premium, measured by the average return on three small portfolios minus three big portfolios, formed in terms of market capacity. HML is the value premium, computed as the average return on two value portfolio minus two growth portfolios formed according to book-to-value ratio.

**Carhart Four-factor model**

Carhart four-factor model(Carhart 1997) extends Fama and French three-factor model by adding momentum as an additional risk factor, which is defined as a tendency for the stock price to continue rising if it is in an upward trend and to continue declining if it is in a downward trend.

$$R_t - Rf_t = \alpha + \beta_1 MKT_t + \beta_2 HML_t + \beta_3 SMB_t + \beta_4 MOM_t + \epsilon_t \quad (4.1.1.6)$$

, where MOM is the momentum premium, calculated by subtracting the average return on lowest-performing firms to highest-performing firms lagged one month.

### 4.1.2 Abnormal Return and Cumulative Abnormal Return

The abnormal return(AR) is defined as the difference between the actual and expected return during the event window. If AR is systematically different from 0, this suggests that the market misprices the ESG incidents and implies a profitable trading rule. The cumulative abnormal return(CAR) is a computed as time-series aggregation of AR.

$$AR_{it} = R_{it} - E[R_{it}|X_t] \quad (4.1.2.1)$$

$$CAR(t_1, t_2) = \sum_{t=t_1}^{t_2} AR_t \tag{4.1.2.2}$$

The cross-sectional mean abnormal return(AAR) and cumulative abnormal return(CAAR) for any is defined as follows:

$$AAR_t = \frac{1}{n} AR_t \tag{4.1.2.3}$$

$$CAAR(t_1, t_2) = \sum_{t=t_1}^{t_2} AR_t \tag{4.1.2.4}$$

A t-test can be used for testing the following null hypothesis. The null hypothesis of a zero cumulative abnormal return(CAR) is tested against the alternative hypothesis of a nonzero CAR.

$$H_0 : CAR_{it} = 0 \tag{4.1.2.5}$$

$$H_1 : CAR_i \neq 0 \tag{4.1.2.6}$$

The test statistic is computed as follows:

$$s_{AR_i}^2 = \frac{1}{M_i - 2} \sum_{t=t_0}^{t_1} e_{i,t}^2 \tag{4.1.2.7}$$

$$s_{CAR_i}^2 = L S_{AR_i}^2 \tag{4.1.2.8}$$

$$t(CAR) = \frac{CAR_i}{S_{CAR}} \tag{4.1.2.9}$$

, where M is the estimation window and L is the event window.

### 4.1.3 Pros and Cons of Event Study

A strength of event study is that the volatility of returns over a short time window is low, and hence the test power would be high and it would be easier to establish the statistical

relationship. Another advantage is that the expected return component is typically neg-
ligible. Therefore, it rules out much of the probability that the abnormal return can be
caused by some latent variable.

Shedding light to the criticisms of event study, some people argue that t-test might be
misspecified. To further explain, the assumptions for t-test are that the abnormal re-
turns are normally distributed, the variances of abnormal returns are equal across firms,
and there is no cross-correlation in abnormal returns. If abnormal returns exhibit either
heteroskedasticity or cross-sectional dependence, t-test can be inappropriate. However,
J.Brown and B.Warner (1985) rebutted this criticism and conclude from their specifica-
tion test that "although daily excess returns are so highly non-normal, there is evidence
that the mean excess return in a cross-section of securities converges to normality as the
number of sample securities increases. Standard parametric test are well specified."

Another criticism of event study is that other events may occur within the event window,
which could confound the result. For example, a positive ESG event could have been a
response to some negative news about the firm. In that case, even if the ESG event is
value-creating, the stock market may respond negatively because of the previous negative
news. Evidence supporting this theory is found in Kruger (2015): Kruger finds that the
stock market has a larger response to a positive ESG event if KLD has registered an ESG
concern for the company within the last year.

## 4.2   Results

We applied a pre-event window ranging from 200 to 20 trading days prior to the event,
and an event window covering 20 trading days before and after the event.

We chose a simple market model to compute the expected stock return. The first reason
is that small error in risk adjustment does not make much difference in a short window.
Empirically, J.Brown and B.Warner (1985) test the market model in an event study sim-
ulation and find out that it incurs very small type one error. Another reason is that
Fama-French and Carhart models are often used to compute monthly abnormal over the
long-run. Even though they can be adapted to daily data and short-run windows, their
statistical properties as an underlying model for short-term event studies has not been
confirmed.

The resulting cross-sectional mean for the CAR is -0.32%. Given that the median market
capitalization for stocks in our dataset in around $ 2.6 billion, the loss caused by a single
ESG incident is approximated at $8 million. Applying a two-tailed t-test for CAR, we

conclude a nonzero CAR at 1% significance level. This roughly aligns with the result of Glosner (2017) which finds a CAR of 0.4%.

Figure 4.1 reveals an increase in the cumulative abnormal return in the controversial stock, even though most of it happens prior to the reporting date of negative news. The possible reasons could be the following: (i) the incident is reflected in other aspects of the company and the price is partly adjusted beforehand, (ii) lag between the date on which ESG incident occurred and the news reported, (iii) investors react before the news released based on their speculations, (iv) insider trading, (v) confounding event.
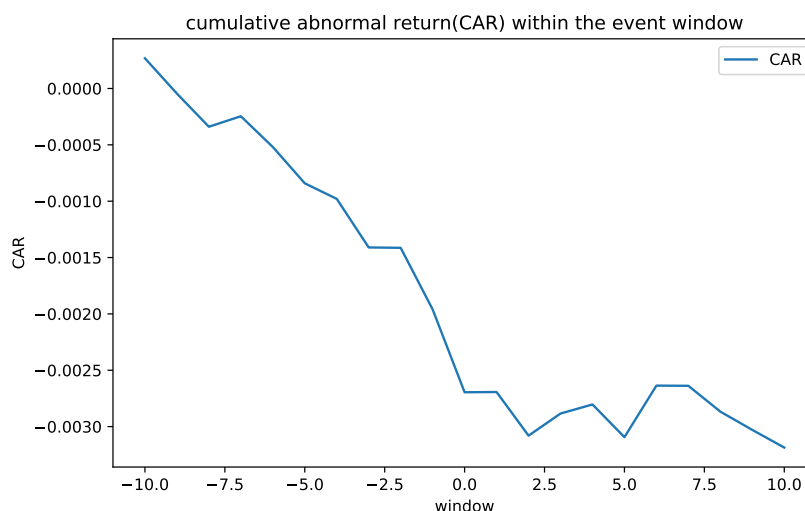


Figure 4.2: Cumulative abnormal return(CAR) within the 21-day event window

The CAR plots by incident subcategory is shown in figure 4.2 and 4.3. The plot shows that many of the event categories have a negative CAR and is significant at 90% confidence level, including product/service, workforce, environmental, and financial. Event categories such as association and corruption have a large increase in the magnitude in CAR, but a conclusion cannot be drawn because of the insufficient amount of data in these categories.

## 4.3   Longevity Study

In addition to the magnitude of abnormal returns, the longevity is also an interesting subject. If excess returns result from mispricing of ESG incidents rather than risk, we would expect the drift to decline over time for two reasons. First, ESG incidents are persistent but not permanent. Put differently, the value of the intangible asset ignored by market is lower overtime, so there should be less mispricing. Secondly, even for firms that persist, the mispricing may be corrected over time as the market slowly learns about their

| day | CAR |
|-----|-----|
| -10 | 0.000268 |
| -9 | -0.000045 |
| -8 | -0.000339 |
| -7 | -0.000246 |
| -6 | -0.000519 |
| -5 | -0.000841 |
| -4 | -0.000979 |
| -3 | -0.001410 |
| -2 | -0.001414 |
| -1 | -0.001960 |
| 0 | -0.002695 |
| 1 | -0.002693 |
| 2 | -0.003080 |
| 3 | -0.002884 |
| 4 | -0.002804 |
| 5 | -0.003094 |
| 6 | -0.002636 |
| 7 | -0.002638 |
| 8 | -0.002867 |
| 9 | -0.003029 |
| 10 | -0.003186 |

Table 4.1: Values of cumulative abnormal return(CAR) within the 21-day event window

value.

We proceed longevity study similar to Edmans (2011). The longevity of underperformance can be calculated with the cumulative abnormal return(CAR) up to time t. The result shows that the magnitude of CARs continue to grow through around day 777, but starts to decline after that. This suggests that it takes around 26 months for the abnormal return to decline. The result is consistent with our hypothesis and implies that the reduction in drift could occur because the market gradually learns how to price intangibles. Note that our event window is around 4 years, so we are only focusing events before the beginning of 2015.
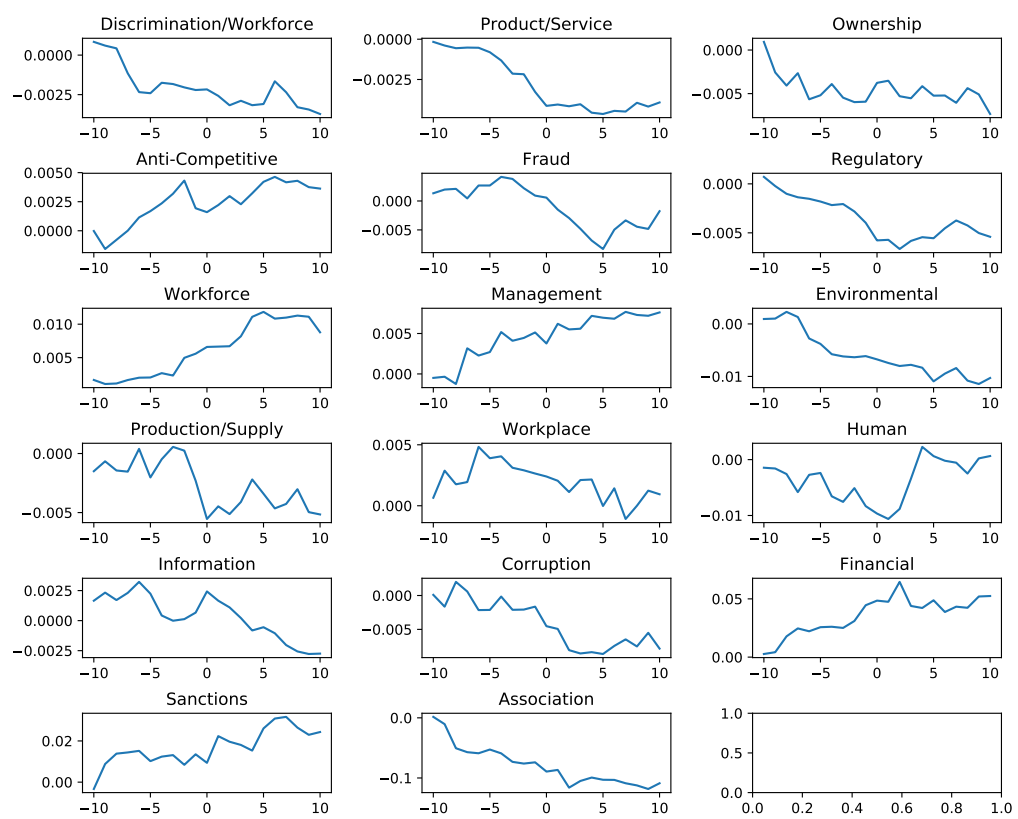
Figure 4.3: Cumulative abnormal return(CAR) for each event subcategory within the 21-day event window
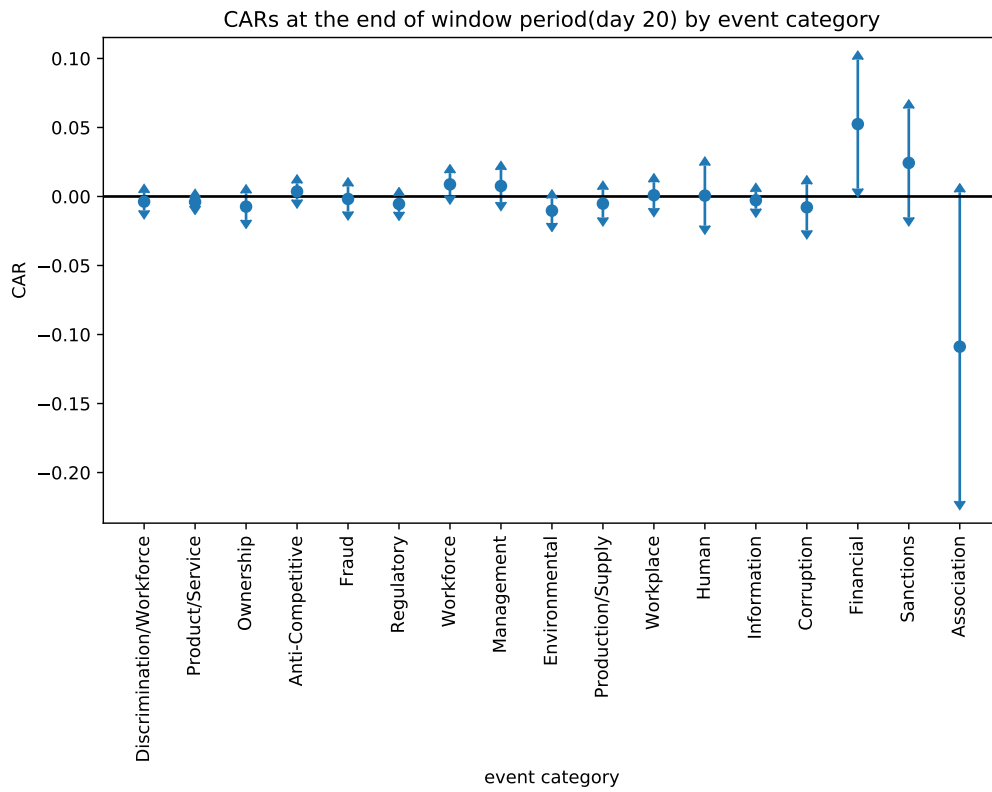
Figure 4.4: Cumulative abnormal return(CAR) and confidence interval for each event subcategory at day 21. The confidence interval is constructed at 90% confidence level.
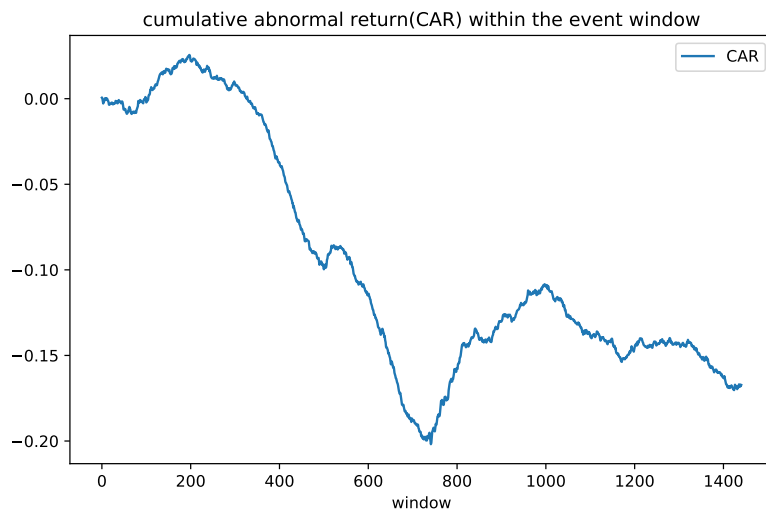


Figure 4.5: Cumulative abnormal return(CAR) plot used for longevity study

We also study the longevity of events by category. The result shows that while drift for categories social/label, environmental/prodcution, competitive/financial start to decrease between month 21 to 26, the cumulative abnormal return continues to grow for regulatory. This implies that regulatory incidents persist longer or it takes longer for the market to learn to price regulatory incidents.

Edmans (2011) continue to confirm this hyptoehsis by showing that the abnormal return die down even in firms for which the event is permanent. Because of the scope of study, we will not further explore it in this paper.
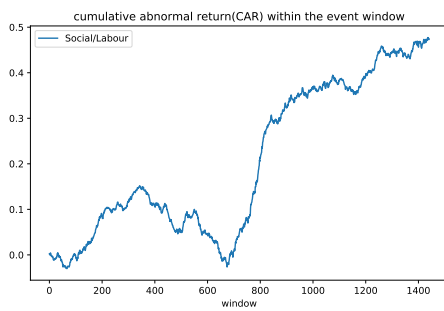
(a) CAR for social/labor
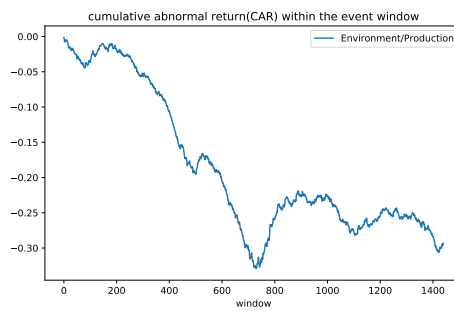


(b) CAR for environmental/preoduction
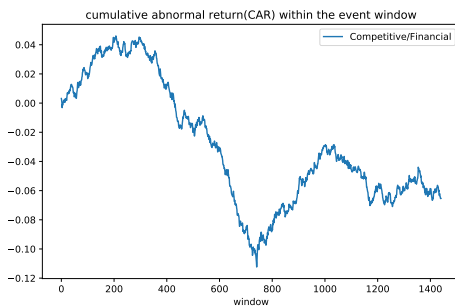


(c) CAR for competitive/financial
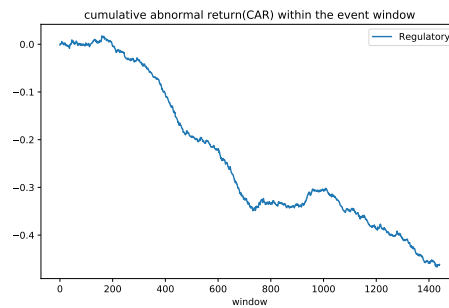


(d) CAR for regulatory

Figure 4.6: Cumulative abnormal return(CAR) plot for each event category, used for longevity study

# Chapter 5

# Jensen's Alpha Approach

## 5.1 Methodology

It is possible that the short-term event study returns substantially underestimate the relationship, as the market fails to fully incorporate intangible price in the short-term. Therefore, we extend the previous study and investigate the impact of ESG incidents on the stock returns in the long run.

For the long-term study, the event study method introduced in the previous section is no longer appropriate. One reason is that the long-horizontal error in risk adjustment accumulates, and thus cannot be neglected anymore as in the short-run. Another reason is that it is unclear which expected return model is correct, so estimates of abnormal returns over long horizons are very sensitive to model choice. Instead, Jensen's alpha approach is often an alternative choice for event study in the long-run.

## 5.2 Introduction to Jensen's Alpha Approach

Jensen's alpha approach, also called calendar-time portfolios abnormal returns, can be summarized with two steps. The first step involves constructing a portfolio that consists of all firms experiencing the event of interest. The second step is to compute the abnormal return of the portfolio. This is done by a time-series regression of actual monthly returns net of the risk-free-rate(or the benchmark return) against some predefined risk factors that explains the systematic risk. Fama-French three-factor model and Carhart four-factor model are often used in this case.

The intercept, also called alpha, provides the estimated average monthly abnormal return of the portfolio of controversial firms over the market. An alpha significantly different

from zero implies that the risk factors fail to fully capture the excess return. R-squared is a statistical measure that represents the percentage of a fund or portfolio's movement that is explained by the factors in the regression. It can be used to assess how specific ESG investments correlated with the movements of the broader market.

In our study, we investigate the stock performance over 1 year following the ESG incident. Given that the sample period starts in January 2012, we create the portfolio for the first time in January 2013. We reform the portfolio every month and all companies experiencing the event within previous 12 months are added to the portfolio. This procedure results in a time-series of monthly portfolio returns from January 2013 to January 2019. The risk model that we choose is the Carhart four-factor model. The factors are obtained from Fama French's homepage.

## 5.3    Improvements and Limitations

The time-series of portfolios returns often suffers autocorrelation and heteroskedasticity in error terms. To address the problem, we estimated the standard error using Newey-West method, the detail of which is further explained in the appendix.

There are several limitations to Jensen's Alpha approach. First, it can only be computed at a portfolio rather than individual firm level, and therefore they do not produce separate measures of abnormal return for each event. As we want to measure it for each event category, we need to group firms into several portfolios according to the event category and then measure a separate alpha for each portfolio. Moreover, there is no consensus on the appropriate kind and number of risk factors.

Another criticism is that Jensen's alpha suffers from a low statistical power(Loughran and Ritter 2000), which implies a high Type 2 error. To further explain, it will more likely to find insignificant results that are consistent with market efficiency. The rationale is that corporate executives exploit mispricing by timing corporate events to coincide with the misvaluation, but the Jensen's alpha approach underweights the timing decisions. As is mentioned in Loughran and Ritter (2000), "if there are time-varying misvaluations that firms capitalize on by taking some action, there will be more events involving larger misvaluations in some periods than in others...... In general, tests that weight firms equally should have more power than tests that weight each time period equally." Since Jensen's alpha approach weights each time period equally, it has lower power to detect abnormal returns if managers time corporate events. To address this problem, a commonly used technique is the weighted least square(WLS) regression, which provides greater weight to months with more firms in portfolio. As a common practice, the weighted vector is the

square root of the number of controversial firms in the relevant month. Since we use news data rather than self-reported corporate events, it is less likely that the events will be subject to behavioral timing considerations. Therefore, we will not further explore this method in this thesis. Kothari and Warner (2004) provides a nice overview of the pros and cons of Jensen's alpha approach.

## 5.4   Results

Our portfolio contains 341 firms on average. The resulting equal-weighted monthly alpha is -4.32% and it is statistically significant at level 1%, which implies an abnormal performance of stock price in the long-term. A value-weighted portfolio return is also computed. It results in a monthly alpha of -4.37% and is also statistically significant. These results provide evidence that an US portfolio of controversial firms underperforms the market by about 4% per month.

|         | equal-weighted | value-weighted |
|---------|----------------|----------------|
| SMB     | 0.0014         | -0.0001        |
|         | (0.002)        | (0.002)        |
| HML     | 0.0071 *       | 0.0060         |
|         | (0.004)        | (0.004)        |
| MKT     | 0.0119 ***     | 0.0026         |
|         | (0.002)        | (0.003)        |
| MOM     | 0.0029         | 0.0038         |
|         | (0.004)        | (0.004)        |
| CONST   | -0.0432 *      | -0.0437 *      |
|         | (0.022)        | (0.023)        |
| $R^2$   | 0.322          | 0.052          |

Table 5.1: Coefficients for portfolio regression, where four factor-model is used

We also examine events by categories and subcategories. The alphas are significant at 10% significance level for all of the event categories, except for those with too few observations that are excluded from our study.

## 5.5   Robustness Checks

This section is intended to provide further evidence that controversial firms are associated with negative risk-adjusted stock returns. We run several robustness tests, aiming to rule out the possibility that the negative alpha stems from outliers, industries, or firm characteristics.

|                          | coeff  | p-value |
|--------------------------|--------|---------|
| Product/Service          | -0.044 | 0.050   |
| Regulatory               | -0.044 | 0.0548  |
| Information              | -0.042 | 0.060   |
| Discrimination/Workforce | -0.042 | 0.073   |
| Anti-Competitive         | -0.044 | 0.048   |
| Fraud                    | -0.044 | 0.046   |
| Ownership                | -0.040 | 0.084   |
| Workplace                | -0.044 | 0.051   |

Table 5.2: Alpha and p-value for each event subcategory

|                        | alpha  | p-value |
|------------------------|--------|---------|
| Environment/Production | -0.044 | 0.046   |
| Competitive/Financial  | -0.043 | 0.053   |
| Regulatory             | -0.043 | 0.051   |
| Social/Labour          | -0.044 | 0.055   |

Table 5.3: Alpha and p-value for each event category

### 5.5.1 Outliers

Our first concern is that outliers drive the results. We test this by winsorizing the stock returns of the controversial firms at 1% and 5% before creating the portfolio. The resulting alphas are still statistically significant.

|        | 1%           | 5%          |
|--------|--------------|-------------|
| SMB    | -0.0004      | -0.0027 *   |
|        | (0.002)      | (0.002)     |
| HML    | 0.0065       | 0.0041      |
|        | (0.004)      | (0.005)     |
| MKT    | 0.0107 ***   | 0.008 ***   |
|        | (0.002)      | (0.003)     |
| MOM    | 0.0036       | 0.0043      |
|        | (0.004)      | (0.004)     |
| CONST  | -0.0424 *    | -0.0409 *   |
|        | (0.023)      | (0.023)     |
| $R^2$  | 0.256        | 0.140       |

Table 5.4: Coefficients for portfolio regression, after removing outliers

### 5.5.2 Misspecified Risk Factors

When estimating abnormal returns, one important concern is a misspecified factor model. We address this problem by additionally estimating the three-factor and five-factor(Fama and R.French 2015) model. The specification for the Fama-French five-factor model is as

follows:

$$R_{it} - Rf_t = \alpha_{it} + \beta_{1i}(R_{MKT_t} - Rf_t) + \beta_{2i}SMB_t + \beta_{3i}HML_t + \beta_{4i}RMW_t + \beta_{5i}CMA_t + \epsilon_t$$

$$(5.5.2.1)$$

, where RMW is the return spread of the most profitable firms minus the least profitable, and CMA is the return spread of firms that invest conservatively minus aggressively.

These two models both result in statistically significant negative alpha.

|         | three-factor model |
|---------|--------------------|
| SMB     | 0.0013             |
|         | (0.002)            |
| HML     | 0.0051**           |
|         | (0.003)            |
| MKT-RF  | 0.0116***          |
|         | (0.002)            |
| CONST   | -0.0421*           |
|         | (0.022)            |
| $R^2$   | 0.326              |

Table 5.5: Coefficients for portfolio regression with three-factor model

|         | five-factor model |
|---------|-------------------|
| SMB     | 0.002             |
|         | (0.002)           |
| HML     | 0.0057            |
|         | (0.005)           |
| MKT-RF  | 0.0114***         |
|         | (0.002)           |
| RMW     | 0.0029            |
|         | (0.004)           |
| CMA     | -0.0023           |
|         | (0.007)           |
| CONST   | -0.0423*          |
|         | (0.022)           |
| $R^2$   | 0.329             |

Table 5.6: Coefficients for portfolio regression with five-factor model

### 5.5.3 Industry

It is possible that the portfolio of controversial firms has a negative alpha because they come from some underperforming industries. We investigate this problem by creating

an industry-matched portfolio, using the 12-industry portfolios from Kenneth French's website.

According to Fama and R.French (1997), each NYSE, AMEX, and NASDAQ stocks are assigned to one of the 12 industries based on the last four digits of company Compustat SIC codes. 12 industrial portfolios are constructed according to this industrial classificiation and value-weighted returns are computed. As we do not have the access to the SIC data, we retrieve the Global Industry Classification Standard(GICS) industrial groups data for each company and map companies from the 24 industry groups to the 12 industries of the Fama-French industrial portfolio.

For each stock in the portfolio, We subtract the value-weighted returns of the corresponding industry portfolio from the stocks' raw returns. Then we calculate the risk-adjusted stock returns of the industry-matched portfolio.

|         | industry-matched |
| ------- | ---------------- |
| SMB     | -0.0004 ***      |
|         | (0.000)          |
| HML     | -0.0002          |
|         | (0.000)          |
| MKT     | -0.0001          |
|         | (0.000)          |
| MOM     | 0.0002           |
|         | (0.000)          |
| CONST   | -0.0002          |
|         | (0.000)          |
| $R^2$   | 0.102            |

Table 5.7: Coefficients for portfolio regression on industry-matched returns

The result shows that an alpha of -0.0002 with p-value of 0.534. which is not statistically significant. In other words, the controversial stocks do not significantly underperform their industry comparable. This suggests that the underperformance of controversial stocks which was discovered previously may stem from the fact that these companies come from underperforming industries. Recall from chapter 2 that we found the controversial firms are more concentrated in energy, utility, materials, telecommunications, automobiles, while they are less concentrated on healthcare, pharmaceutical, banks, and software industries. This coincides with the fact that the former industries underperform the latter ones within the recent 10 years, shown in the figure below. One objection of this conclusion is that many companies fail because of firm-specific events instead of widespread industry features. For example, recall from Chapter 2 that the stock price of Celsion drops sharply because of its unsuccessful drug test, while the average return in pharmaceutical industry is positive at the time. Our explanation is that such firm-specific cases with extreme stock returns

are so rare that they do not make statistically significant impact.
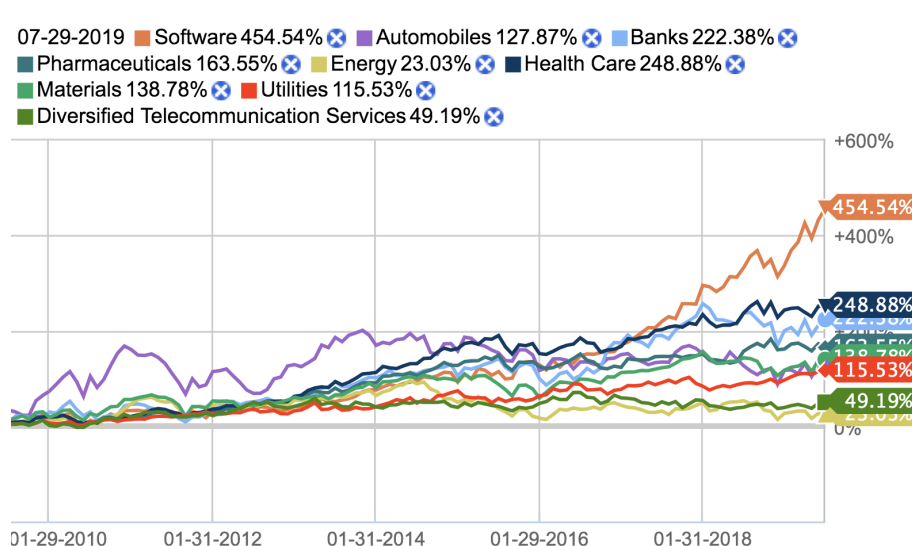


Figure 5.1: Monthly performance of several industries from 2010 to 2019(Fidelity Investment 2019)

We proceed to investigate the excess return for each industry by running seprate industrial portfolio regressions, where the industry benchmark return is subtracted from the raw stock return. The result shown below indicates that the alphas for many industries are statistically significant, including money, non-durables, health, shops, energy, and others.

| industry | alpha | p-value |
|---|---|---|
| Money | -0.004630 | 0.000064 |
| Other | 0.003169 | 0.024338 |
| BusEq | 0.000186 | 0.938074 |
| Manuf | 0.000441 | 0.819017 |
| Hlth | 0.007502 | 0.006547 |
| Shops | 0.004938 | 0.051794 |
| NoDur | -0.002271 | 0.086337 |
| Enrgy | 0.013829 | 0.001568 |
| Utils | -0.001846 | 0.195401 |
| Telcm | -0.005850 | 0.123074 |
| Chems | 0.001402 | 0.730204 |
| Durbl | -0.006831 | 0.149209 |

Table 5.8: Alphas and p-values of portfolio regressions for each industry

### 5.5.4 Firm Characteristics

To rule out the possibility that the negative alpha stems from firm characterisitics, We took the approach from Wermers (2003), which matches each stock to a portfolio of stocks with similar market capitalization, book-to-market ratio, and momentum.

Market capitalization is the firm's market value, computed as the outstanding shares by the price of a single share. It is one of the basic determinants of various characteristics including risk. The smaller the market capitalization, the higher the investment risk.

Book-to-market ratio is defined as the dividend of a firm's book value to market value. Book value can be calculated by subtracting total liabilities, preferred shares, and intangible assets from the total assets of the company. It represents how much a company would have left in assets if it went out of business. If a firm's book-to-market ratio is large, it means that its market value is trading higher than its book value per share and that the company is overvalued.

Momentum here is defined as the sum of 12-months past returns of the stock. It considers the trend that the stock price is likely to continue to rise if it has a high past-year return.

Considering the characteristics-matched portfolio, specifically we first rank all stocks traded on NYSE, NASDAQ, and AMEX stocks by their market capitalization at the end of each June. Quintile portfolios are formed according to the rank, and each quintile portfolio is further divided into book-to-market ratio quintiles, based on their most recently available data. Finally, each of the resulting 25-group portfolios is further divided into quintiles based on the 12-month past returns of stocks. This three-way ranking procedure results in 125-group portfolios, each having different combinations of size, book-to-market ratio, and momentum. The portfolio is turned over every year. Each controversial firm is mapped to one of these portfolios based on their characteristics and the equal-weighted portfolio return is subtracted from the company's raw stock return.
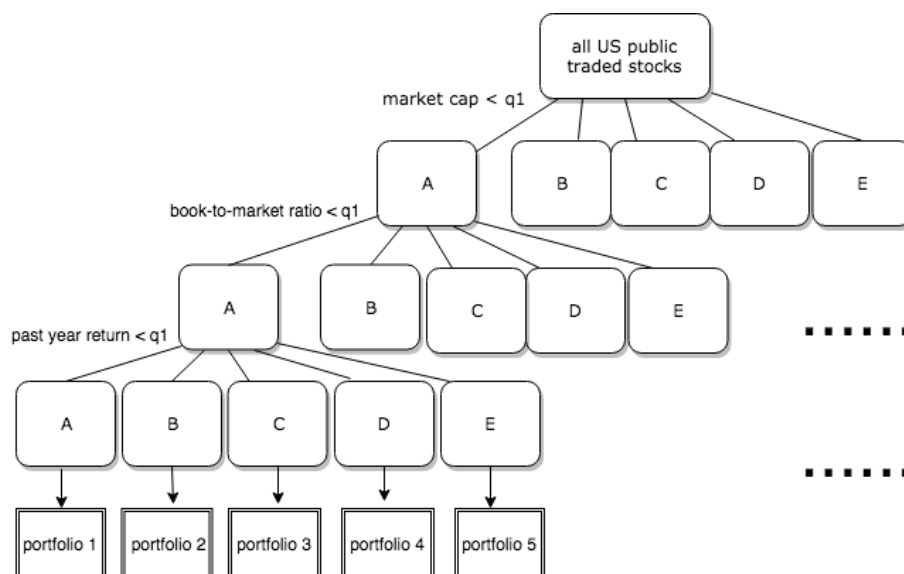


Figure 5.2: Illustration of the procedures to form 125-fractile portfolios

We are left with 42 stocks in each fractile. The result shows an alpha that is not statistically

significant. This suggests firm characteristics, specifically size, book-to-market-ratio, and momentum, is another source for the underperformance of controversial firms.

We proceed to examine which characteristics specifically changes the result and leads to a statistically insignificant alpha. We achieve this by removing one factor from our portfolio-forming procedure at a time. This results in three 25-group portfolios, formed on i) book-to-market ratio and yearly return, ii) market capitalization and yearly return, iii) market capitalization and book-to-market ratio. The result shows that the alpha for portfolio specification ii) remains insignificant, while for i) and iii) are negative and statistically significant. This implies that that the excess return is not robust to the characteristics combination of market capitalization and book-to-market ratio.

|  | mc, bv, yr | bm, yr | mc, yr | mc, bm |
|---|---|---|---|---|
| SMB | -0.0004 *** | -0.0046 *** | -0.0032 *** | -0.0033 *** |
|  | (0.001) | (0.001) | (0.000) | (0.000) |
| HML | 4.374e-05 | -0.0003 | 0.0004 | -0.0003 |
|  | (0.001) | (0.000) | (0.000) | (0.00) |
| MKT | 0.0025 * | 0.0007 | 0.0003 | 0.0002 |
|  | (0.001) | (0.000) | (0.000) | (0.00) |
| MOM | -0.0027 *** | -0.0003 | -0.0012 *** | -0.0013 *** |
|  | (0.0001) | (0.000) | (0.000) | (0.000) |
| CONST | 0.0023 | -0.0037 ** | -0.0024 * | -0.0016 |
|  | (0.005) | (0.002) | (0.001) | (0.001) |
| $R^2$ | 0.063 | 0.532 | 0.556 | 0.536 |

Table 5.9: Coefficients for portfolio regressions with characteristics-matched portfolio returns

# Chapter 6

# Cross-sectional Time-Series Regression

## 6.1 Methodology

### 6.1.1 Model Specification

A concern with the previous approaches is that the underperformance of controversial firms stems from unknown omitted variables rather than the unexplained abnormal return. The characteristics-matched approach is aimed to solve the problem, but as the number of variables increase, the number of stocks in each group decreases, and thus the method does not work anymore. In order to address this problem, in this section we apply the cross-sectional time-series regression to control for a wide range of firm characteristics. The regression model is specified as follows:

$$R_{i,t} = \alpha_0 + \beta_1 X_{i,t-k} + \beta_2 Z_{i,t-k} + \epsilon_{i,t} \tag{6.1.1.1}$$

, where $R_{it}$ is the return of stock i, $X_{i,t-k}$ are the factors that control for firm characteristics, and $Z_{i,t-k}$ is an indicator for ESG incident. We tried three specifications for the indicator: i) a dummy variable, which equals to 1 if there are at least 1 ESG incidents in month i for stock t, and equals to 0 otherwise, ii) a count of number of ESG incidents in month i, and iii) separate counts of number of ESG incidents for each event subcategory. The indicator is lagged 1 month.

Following Brennan, Chordia, and Subrahmanyam (1998) and Edmans (2011), we use firm-characteristics-related factors as control variables, which include the following: "Variable

size" is the logarithms of market capitalization in terms of billions of US dollars. "Variable bm" is the logarithm of book-to-market ratio. "Variable div" is the ratio of dividends paid over the market value. "Variables ret23", "ret46", and "ret712" are the logarithms of compounded returns returns over previous months t-2 through t-3, t-4 through t-6, and t-7 through t-12. "Variable prc" is the logarithm of the reciprocal of the stock price. These risk factors are all lagged 1 month.

### 6.1.2   Fama-MacBeth Regression

We apply the Fama-Macbeth model(Fama and MacBeth 1973) to estimate the betas and risk premia for the risk factors.

To obtain the Fama-MacBeth estimator, we perform T regressions, one for each sample period using all of the entity data. The model is given by:

$$y_{it} = \beta' x_{it} + \epsilon_{it} \tag{6.1.2.1}$$

Denote the estimated model parameters as $\hat{\beta}_t$. The reported estimator is then:

$$\hat{\beta} = T^{-1} \sum_{t=1}^{T} \hat{\beta}_t \tag{6.1.2.2}$$

$$\sigma(\hat{\beta}) = \frac{\sigma(\hat{\beta}_t)}{\sqrt{M}} \tag{6.1.2.3}$$

One criticism of the Fama-MacBeth regression is that it is robust only against time but not against firm effects. Alternatively, we can apply a pooled-OLS panel regression and double-cluster the standard error to account for cross-sectional and serial correlations. The detail of it is explained in appendix A.

## 6.2   Results

|       | mc     | prc    | bm     | div    | ret23  | ret46  | ret712 | ret     |
|-------|--------|--------|--------|--------|--------|--------|--------|---------|
| mean  | 1.915  | -1.301 | -0.829 | 2.372  | 0.002  | 0.005  | 0.009  | 0.010   |
| std   | 1.476  | 0.198  | 0.660  | 1.595  | 0.020  | 0.028  | 0.045  | 0.070   |
| min   | -2.978 | -1.819 | -3.217 | 0.018  | -0.132 | -0.198 | -0.329 | -0.3090 |
| 25%   | 0.872  | -1.441 | -1.268 | 1.263  | -0.008 | -0.009 | -0.014 | -0.031  |
| 50%   | 1.951  | -1.322 | -0.789 | 2.038  | 0.003  | 0.006  | 0.011  | 0.011   |
| 75%   | 3.044  | -1.184 | -0.350 | 3.050  | 0.013  | 0.020  | 0.035  | 0.051   |
| max   | 5.307  | -0.312 | 1.049  | 13.553 | 0.136  | 0.188  | 0.246  | 0.408   |

Table 6.1: Summary statistics of control variables

|        | fama-macbeth    | double-clustered |
|--------|-----------------|------------------|
| count  | 0.000800        | 0.0006           |
|        | (0.000755)      | (0.0009)         |
| mc     | -0.000859       | -0.0009          |
|        | (0.000609)      | (0.0006)         |
| prc    | 0.005494 **     | 0.0108 **        |
|        | (0.002230)      | (0.0043)         |
| bm     | -0.000775       | 8.532e-05        |
|        | (0.000686)      | (0.0007)         |
| div    | -0.000753 **    | -.0004           |
|        | (0.000352)      | (0.0004)         |
| ret23  | 0.008944        | 0.0360           |
|        | (0.034099)      | (0.0916)         |
| ret46  | -0.011331       | 0.0621           |
|        | (0.027928)      | (0.0531)         |
| ret712 | 0.015977        | 0.0436           |
|        | (0.016967)      | (0.0340)         |
| const  | 0.019381 ***    | 0.0257 ***       |
|        | (0.004796)      | (0.0070)         |

Table 6.2: Results of cross-sectional time-series regression, where the event indicator is a count of the number of events

|        | double-clustered | Fama-Macbeth |
|--------|------------------|--------------|
| dummy  | 5.233e-05        | 0.0001       |
|        | (0.0016)         | (0.0013)     |
| mc     | -0.0009          | -0.0008      |
|        | (0.0007)         | (0.0007)     |
| prc    | 0.0151 ***       | 0.0073 ***   |
|        | (0.0057)         | (0.0025)     |
| bm     | 0.0019           | 0.0004       |
|        | (0.0010)         | (0.0009)     |
| div    | -0.0002          | -0.0007      |
|        | (0.0006)         | (0.0005)     |
| ret23  | 0.0569           | 0.0076       |
|        | (0.1161)         | (0.0411)     |
| ret46  | 0.0721           | -0.0170      |
|        | (0.070)          | (0.0339)     |
| ret712 | 0.0480           | 0.0029       |
|        | (0.0455)         | (0.0217)     |
| const  | 0.0322 ***       | 0.0227 ***   |
|        | (0.0089)         | (0.0057)     |

Table 6.3: Results of cross-sectional time-series regression, where the event indicator is a dummy variable

Both results from Fama-MacBeth regression and pooled-OLS with double-clustered standard error show that neither a dummy indicator nor a count of the number of events has a significant effect on the stock return. This result is consistent with our finding from the characteristic-matched portfolio that the excess returns from ESG incidents are not robust to firm characteristics.

| | double-clustered | Fama-Macbeth |
|---|---|---|
| Association | 0.0655 | 0.0018 |
| | (0.0515) | (0.0013) |
| Sanctions | 0.0398 * | 0.0016 |
| | (0.0220) | (0.0013) |
| Financial | -0.0072 | 0.0016 |
| | (0.0135) | (0.0015) |
| Corruption | -0.0067 | 0.0003 |
| | (0.0093) | (0.0030) |
| Information | 0.0009 | -0.0076 * |
| | (0.0051) | (0.0040) |
| Human | -0.0174 | 0.0002 |
| | (0.0194) | (0.0020) |
| Workplace | 0.0008 | -0.0090 ** |
| | (0.0044) | (0.0042) |
| Production/Supply | 0.0013 | -0.0022 |
| | (0.0077) | (0.0052) |
| Environmental | 0.0008 | 0.0040 |
| | (0.0060) | (0.0049) |
| Management | 0.0066 | 0.0055 |
| | (0.0061) | (0.0047) |
| Workforce | -0.0078 | -0.0112 * |
| | (0.0061) | (0.0059) |
| Regulatory | 0.0008 | 0.0048 |
| | (0.0033) | (0.0040) |
| Fraud | -0.0007 | -0.0051 |
| | (0.0041) | (0.0043) |
| Anti-Competitive | -0.0102 | -0.0017 |
| | (0.0063) | (0.0045) |
| Ownership | 0.0054 | -0.0009 |
| | (0.0052) | (0.004780) |
| Product/Service | -0.0016 | -0.0016 |
| | (0.0019) | (0.0019) |
| Discrimination/Workforce | 0.0015 | 0.0036 |
| | (0.0039) | (0.0040) |
| mc | -0.0008 | -0.0007 |
| | (0.0007) | (0.0007) |
| prc | 0.0154 *** | 0.0077 *** |
| | (0.0058) | (0.0025) |
| bm | 0.0018 * | 0.0003 |
| | (0.0010) | (0.0009) |
| div | -0.0002 | -0.0008 |
| | (0.0006) | (0.0005) |
| ret23 | 0.0571 | 0.0003 |
| | (0.1167) | (0.0412) |
| ret46 | 0.0720 | -0.0144 |
| | (0.0706) | (0.0340) |
| ret712 | 0.0471 | 0.0014 |
| | (0.0455) | (0.0218) |
| const | 0.0326 *** | 0.0232 *** |
| | (0.0090) | (0.0057) |

Table 6.4: Results of cross-sectional time-series regression, where the number of events is counted for each subcategory

We proceed to run the regression with our third specification of event indicator, which separately counts the number of events for each subcategory. The pooled OLS result shows that while for most of the categories the coefficients remain insignificant, the coefficients for management, human, and information are statistically significant at 5%, 10%, and 5% level. To be more specific, while other variables remain the same, companies with one additional management incident have 1.12% lower stock returns, companies with one additional human incident have 0.9% lower stock return, and companies with one additional information incident have 0.76% lower stock return for the month.

# Chapter 7

# Portfolio Construction

In order to evaluate whether portfolios with low ESG-profile underperform typical passive benchmarks, we compare and construct a series of portfolios, based on several ESG investing strategies. According to the GSIA by Global Sustainable Investment Alliance (2018), ESG investing strategies are divided into the following main categories:

- **Negative screening**: the exclusion of certain companies or sectors from the portfolio based on ESG criteria.

- **Relative/Best-in-class screening**: investment in companies or sectors based on ESG criteria relative to their industry peers.

- **Norm-based screening**: screening of portfolios against some minimum standards of business practice according to international norms, such as those issued by OECD and UN.

- **ESG inclusion**: systematic and explicit incorporation of ESG factors into financial analysis

- **Themed investing**: investment in themes specifically related to ESG considerations

- **Impact investing**: targeted investments aimed at solving ESG problems

- **Corporate engagement**: utilization of shareholder power to influence corporate behaviors, including direct corporate engagement, filing or co-filing shareholder proposals, and proxy voting that is consistent with ESG guidlines.

While investors using themed investing, impact investing, and norm-based screening strategies are more concerned with social responsibility than financial returns, corporate engagement and ESG inclusion are rarely used by individual investors and hard to evaluate. Therefore, in this section we only focus on examining the financial implication of negative

and positive screenings.

## 7.1   Methodoloy

Firstly, we construct portfolios based on negative and positive screenings. Then we model the difference between monthly portfolio returns and benchmark returns using four-factor model, which is the same as Jensen's Alpha approach. Finally, we do hypothesis testing to check if the alpha is statistically significantly different from 0. If not, this implies that ESG portfolios do not significantly underperform or overperform the benchmarks.

A critique of applying Jensen's Alpha method to investigate the excess returns of positive or negative screening strategy is that this method is primarily concerned with the exposure of market risk and is thus only appropriate for well-diversified portfolios. However, ESG screenings restrict the investment universe. In addition, there is no consensus on the appropriate kind and number of risk factors. Instead, Ledoit and Wolf (2008) suggested comparing the two portfolios by hypothesis testing of Sharpe ratio. Sharpe ratio is defined as the expected return in excess of risk-free-rate per unit of volatility and is widely used to understand the risk-adjusted return potentials for portfolios. We also applied this method and provide the results from both methods below. Details of the hypothesis testing of Sharpe ratio is explained in Appendix B.

## 7.2   Relative Screening

Relative screening refers to the investment in companies selected based on the criteria of ESG performance relative to industry peers.

In our case, for each month we rank the companies by the number of previous ESG incidents as an approximation for ESG score. Based on this ranking, we select the 30 worst stocks with the most number of previous ESG incidents and form an equal-weighted portfolio. This gives a monthly time-series return from April 2012 to February 2019. Similarly, we also construct a portfolio with the 30 best stocks for companies with the least number of previous ESG incidents, which is used as a benchmark. The other two benchmark portfolios are S&P 500 stocks and all publicly traded US stocks on NASDAQ, NYSE, and AMEX.

|       | all US stocks | S&P 500      | best 30 stocks |
|-------|---------------|--------------|----------------|
| MKT   | -0.0003       | -0.0010 **   | -0.0007        |
|       | (0.000)       | (0.000)      | (0.001)        |
| HML   | -0.0010       | -0.0014 *    | -0.0022 ***    |
|       | (0.001)       | (0.001)      | (0.001)        |
| SMB   | -0.0107 ***   | -0.0046 ***  | -0.0078 ***    |
|       | (0.001)       | (0.001)      | (0.001)        |
| MOM   | -0.0025 ***   | -0.0028 ***  | -0.0002        |
|       | (0.001)       | (0.001)      | (0.001)        |
| const | 9.422e-07     | -0.0009      | -0.0054 **     |
|       | (0.002)       | (0.002)      | (0.003)        |
| $R^2$ | 0.677         | 0.345        | 0.414          |

Table 7.1: Coefficients for equal-weighted portfolio consisted of worst 30 stocks, compared with all US stocks, SP500 stocks, and best 30 stocks

|              | worst 30 stocks | best 30 stocks | S&P500 stocks | all US stocks |
|--------------|-----------------|----------------|---------------|---------------|
| Sharpe Ratio | 0.2342239       | 0.3217523      | 0.3264605     | 0.2260067     |
| P value      |                 | 0.3133373      | 0.2337532     | 0.9346131     |

Table 7.2: Results for Sharpe Ratio test, where the worst 30 stocks are compared with best 30, SP500, and all US stocks



Figure 7.1: Profit and loss graph for equal-weighted portfolio of worst 30 stocks, compared with SP500, all US, and best 30 stocks

The result shows that a equal-weighted portfolio of the worst ESG-performing stocks underperforms the best ones with a negative alpha of -0.54% and it is statistically significant at 5% significance level. However, it does not systematically underperform S&P 500 or all US stocks. Moreover, Sharpe ratio test does not show any statistical significance for any

benchmark strategies.

Value-weighted returns are also computed. A value-weighted portfolio of the worst ESG-performing stocks overperforms S&P 500 stocks by 0.29% at 1% significance level, but not systematically overperform all US stocks or underperform best stocks.

|  | all US stocks | S&P 500 | best 30 stocks |
|---|---|---|---|
| MKT | 0.0020*** | 0.0020 *** | 0.0025 * |
|  | (0.000) | (0.000) | (0.001) |
| HML | -0.0012 | -0.0043 *** | -2.127e-05 |
|  | (0.001) | (0.001) | (0.001) |
| SMB | -0.0018 ** | -0.0008 | -0.0048 * |
|  | (0.001) | (0.001) | (0.002) |
| MOM | 0.0007 | 0.0006 | 0.0006 |
|  | (0.001) | (0.001) | (0.002) |
| const | 0.0003 | 0.0029 ** | -0.0052 |
|  | (0.002) | (0.001) | (0.003) |
| $R^2$ | 0.175 | 0.344 | 0.072 |

Table 7.3: Coefficients for value-weighted portfolio consisted of worst 30 stocks, compared with all US stocks, SP500 stocks, and best 30 stocks

|  | worst 30 stocks | best 30 stocks | SP500 stocks | all US stocks |
|---|---|---|---|---|
| Sharpe Ratio | 0.4181311 | 0.3886057 | 0.3915104 | 0.4394484 |
| P value |  | 0.7280544 | 0.3757508 | 0.6802639 |

Table 7.4: Results for Sharpe Ratio test, where the a portfolio of value-weighted worst 30 stocks are compared with best 30, SP500, and all US stocks
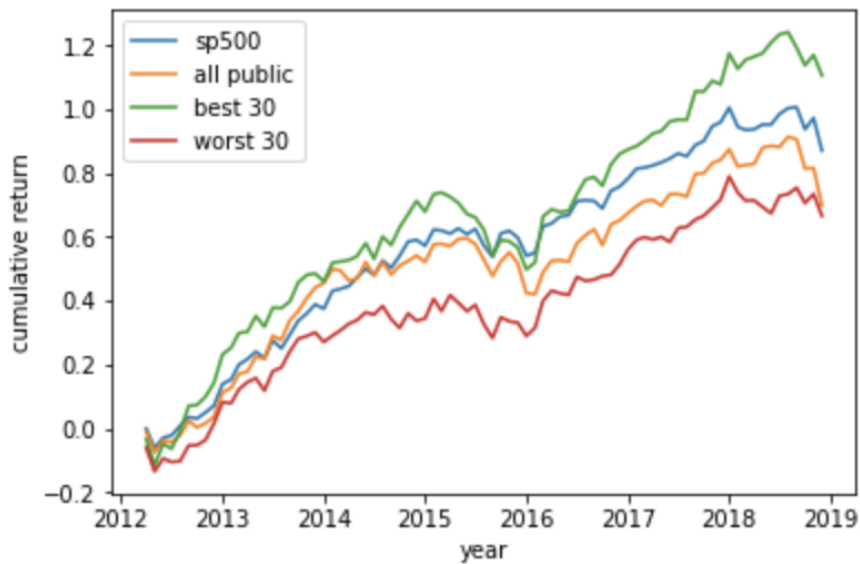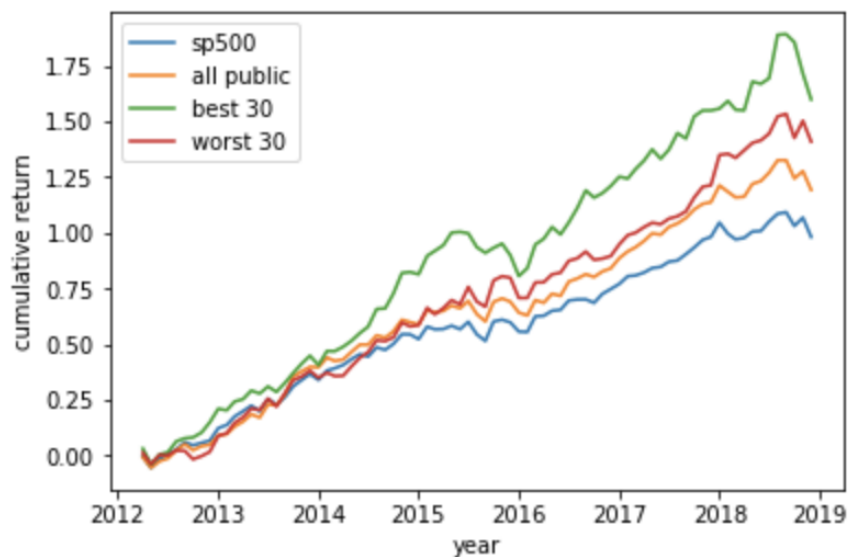


Figure 7.2: profit and loss graph for value-weighted portfolio of worst 30 stocks, compared with SP500, all US, and best 30 stocks

## 7.3 Negative Screening

Negative screening entails the exclusion of certain companies from the portfolio based on some ESG criterion. The benchmark portfolios are constructed using both the S&P 500 stocks and all public traded US stocks on NYSE, NASDAQ, and AMEX. In our dataset, approximately 69% of S&P 500 stocks and 19% of all public traded US stocks are controviersial firms. For every month, we remove the stocks with historical ESG-related news in the past to construct our comparable ESG portfolio.

|        | all US stocks | S&P500        |
|--------|---------------|---------------|
| MKT    | -0.0002 ***   | -0.0004 ***   |
|        | (5.59e-05)    | (0.000)       |
| HML    | -3.809e-05    | -0.0002       |
|        | (0.000)       | (0.000)       |
| SMB    | -7.069e-05    | -0.0002       |
|        | (5.49e-05)    | (0.000)       |
| MOM    | -7.324e-05    | -9.124e-05    |
|        | (9.81e-05)    | (0.000)       |
| CONST  | -0.0003       | 9.811e-05     |
|        | (0.000)       | (0.000)       |
| $R^2$  | 0.173         | 0.164         |

Table 7.5: Coefficients for equal-weighted negatively screened portfolio, compared with benchmarks

|              | all public US stocks | negative screened | SP500 stocks | negative screened |
|--------------|----------------------|-------------------|--------------|-------------------|
| Sharpe Ratio | 0.2447               | 0.2333            | 0.3467       | 0.3463            |
| P value      | 0.0856               |                   | 0.9650       |                   |

Table 7.6: Results for Sharpe Ratio test, where equal-weighted portfolios are compared

|        | all US stocks | S&P500        |
|--------|---------------|---------------|
| MKT    | -0.0007 ***   | -0.0002       |
|        | (0.000)       | (0.000)       |
| HML    | 0.0002        | -4.873e-05    |
|        | (0.000)       | (0.000)       |
| SMB    | -0.0003       | -0.0004 **    |
|        | (0.000)       | (0.000)       |
| MOM    | -0.0007       | -0.0003       |
|        | (0.000)       | (0.000)       |
| CONST  | -0.0005       | -0.0003       |
|        | (0.001)       | (0.000)       |
| $R^2$  | 0.217         | 0.089         |

Table 7.7: Coefficients for value-weighted negatively screened portfolio, compared with benchmarks

|  | all public US stocks | negative screened | SP500 stocks | negative screened |
|---|---|---|---|---|
| Sharpe Ratio | 0.4619 | 0.4335 | 0.4145 | 0.3956 |
| P value | 0.2623 | | 0.3651 | |

Table 7.8: Results for Sharpe Ratio test, where value-weighted portfolios are compared
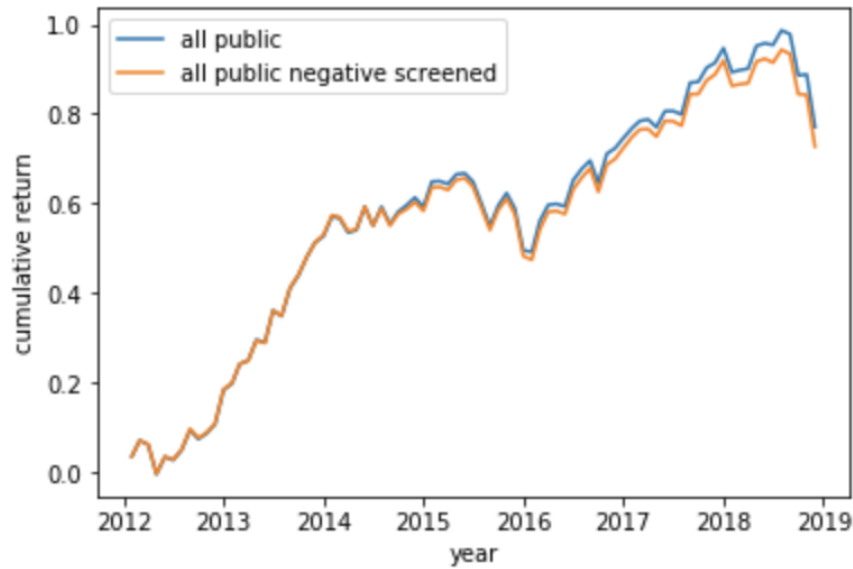


Figure 7.3: Profit and loss graph for equal-weighted negative screened portfolio on all US stocks
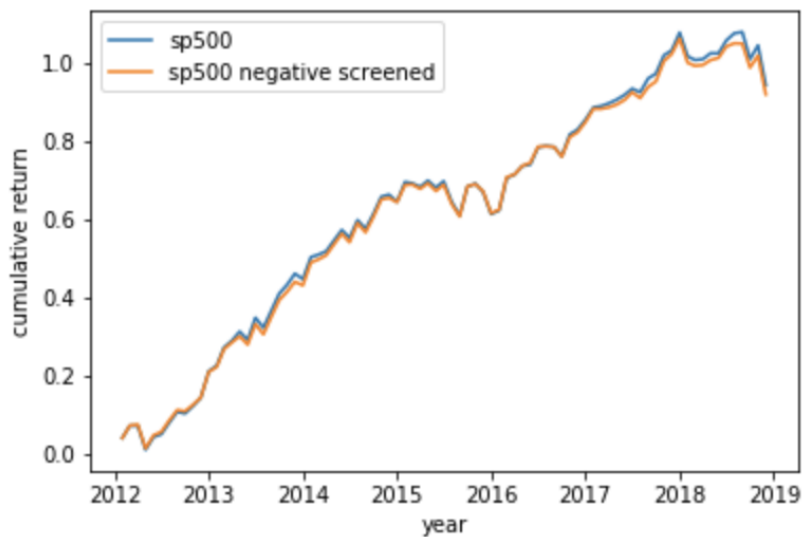


Figure 7.4: Profit and loss graph for equal-weighted negative screened portfolio on S&P 500 stocks

The portfolio regression results show that negatively screened portfolios slightly underper-
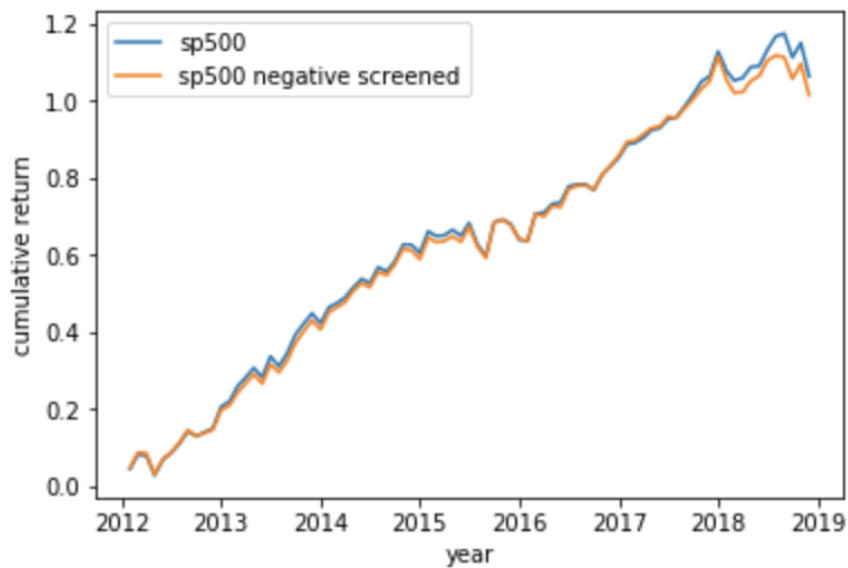
Figure 7.5: Profit and loss graph for value-weighted negative screened portfolio on all US stocks
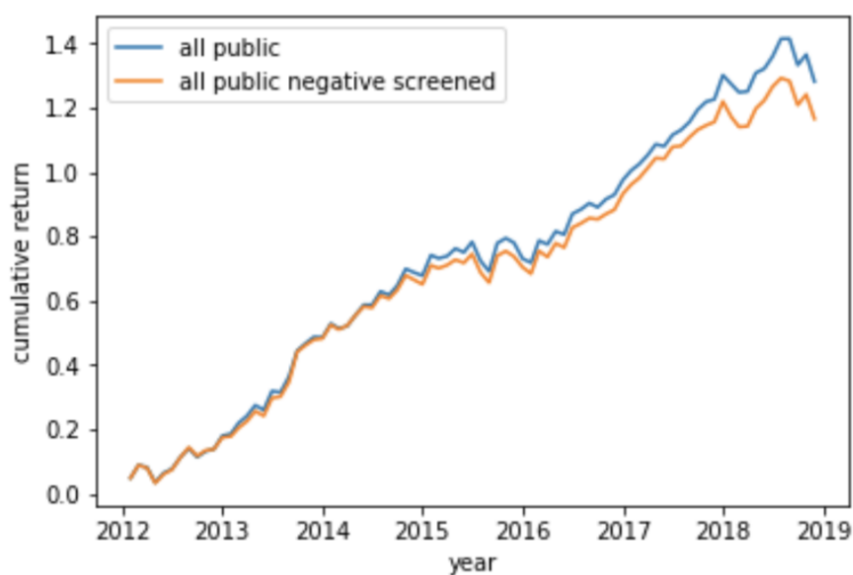


Figure 7.6: Profit and loss graph for value-weighted negative screened portfolio on S&P 500 stocks

form the benchmarks, but none of them is statiatically significant. In other words, both equal-weighted and value-weighted negatively screened portfolios do not systematically underperform or overperform S&P 500 portfolio or all US stocks portfolio. Hypothesis testing of Sharpe ratio gives slightly different results: while the equal-weighted portfolio obtained from negative screening all public US stocks underperform the unscreened

portfolio at 10% significance level, the rests remain insignificant.

The statistically insignificant results might stem from the fact that the dataset used here does not have any severity index for each ESG-related news event. This makes it impossible for us to distinguish the major probles from the minor ones, and thus, ESG screening becomes less effective.

# Chapter 8

# Conclusion and Future Work

There is a long-running debate on the financial implications of ESG investing. It indicates whether investors have to sacrifice financial performance by investing socially responsible. ESG investors can apply different screening strategies to construct a socially responsible portfolio. In this thesis, we analyze two strategies, namely negative and positive screening. After implementing these strategies using controversial firms obtained from Dow Jones Adverse Media Entity dataset that covers 2012 to 2019, we get the following main results:

i) ESG risks generate negative excess stocks returns in the short-run. Specifically, it generates a cumulative abnormal return of -0.32% for a 21-day event window. It demonstrates that the market fails to fully incorporate ESG information in the short-run.

ii) In the long-run, a monthly excess return of -4.37% for portfolios of controversial firms is reported. The result is robust to the removal of outliers and different model specifications. However, it is not robust to industry-matched returns, which implies that the underperformance of controversial stocks can stem from underperforming industries. Within industry information, management, and human, the negative alphas remain statistically significant. Moreover, the negative alpha is not robust to several combinations of firm characteristics, including size, book-to-market ratio, and momentum.

iii) ESG portfolios with negative screening strategy do not significantly overperform or underperform passive investing portfolios, including S&P stocks and all US stocks publicly traded on NASDAQ, NYSE, and AMEX. This implies that for socially responsible investors, negative screening ESG stocks does not harm the financial performance of their portfolios, but neither does it brings extra financial profits.

iv) An equal-weighted portfolio with relative screening strategy, which selects worst ESG-performing stocks, underperform the best-performing ones by -0.54%, while a value-weighted portfolio overperforms S&P 500 portfolio by 0.29%. This suggests shorting or

longing controversial firms with extreme numbers of ESG incidents could bring excess returns.

One major concern of our study is the short sample period, which only covers seven years from 2012 to 2019. The sample size for some event subcategories, such as sanctions and association, is especially small during this sample period. This forms an obstacle for precise inference for two main reasons. Firstly, the asset pricing tests have relatively low statistical power, which implies a lower probability of detecting abnormal returns. Secondly, the results are sensitive to particular developments during the period. For instance, the economy in the US has been in an expansionary stage for most of the sample period and statistical techniques may not be able to fully explain such realizations of underlying factors. To provide evidence that the findings from the portfolio regressions in these seven years are not spurious, we could further run out-of-sample tests with, for instance, European stock data. A second concern is the lack of severity index in our dataset, which is mentioned in chapter 7. This impedes us from distinguishing major events from minor ones, and thus potentially introduce noises with minor events. Taking the significance of each news events by mining the raw text data of each news into account might overcome this issue, and this is left for future work.

A further improvement we could make is to apply buy-and-hold abnormal return(BHAR) method in addition to Jensen's alpha approach. This involves calculating a stock's un-adjusted return by geometrically compounding its monthly returns and subtracting the benchmark returns. Conrad and Kaul argue that BHAR is a better methodology than Jensen's alpha for some statistical reasons.

One interesting subject for further study is the earnings surprises. Our hypothesis is that the intangible consequences of ESG incidents, such as reputational loss, are not immediately capitalized by the market. Instead, it only affects the stock price when it subsequently manifests in tangible outcomes, thus generating lower profits overtime. To confirm the hypothesis, we could firstly examine whether controversial firms have lower operating performance, such as return on equity(ROE) and return on assets(ROA). As weaker operating performance only results in abnormal returns if it is unexpected by the stock market, we could proceed with investigating the earning surprise, which is calculated as the actual earnings per share less the analysts' forecast. Another interesting topic would be studying the values of positive ESG events and the financial implication of the positive screening strategy. This entails a dataset of positive news and is left for future study.

# Bibliography

Andrews, D. and C. Monahan (1992). An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica 60*, 953–966.

Bauer, R., K. Koedijk, and R. Otten (2005). International evidence on ethical mutual fund performance and investment style. *Journal of Banking and Finance 29*, 1751–1767.

Behr, P., A. Guettler, and F. Truebenbach (2012). Using industry momentum to improve portfolio performance. *Journal of Banking and Finance 36*, 1414–1423.

Brennan, M., T. Chordia, and A. Subrahmanyam (1998). Alternative factor specifications, security characteristics, and the cross-section of expected stock returns. *Journal of Financial Economics 49*, 345–373.

Carhart, M. M. (1997). On persistence in mutual fund performance. *The Journal of Finance 52*, 57–82.

Conrad, J. and G. Kaul (1993). Long-term market overreaction or biases in computed returns? *The Journal of Finance 48*, 39–63.

Edmans, A. (2011). Does the stock market fully value intangibles? employee satisfaction and equity prices. *Journal of Financial Economics 101*, 621–640.

Efing, M. and H. Hau (2013). Corrupted credit ratings: Standard & poor's lawsuit and the evidence.

Fama, E. F. and J. D. MacBeth (1973). Risk, return, and equilibrium: Empirical tests. *The Journal of Political Economy 81*, 607–636.

Fama, E. F. and K. R.French (1992). The cross-section of expected stock returns. *The Journal of Finance 47*, 427.

Fama, E. F. and K. R.French (1997). Industry costs of equity. *Journal of Financial Economics 43*, 153–193.

Fama, E. F. and K. R.French (2015). A five-factor asset pricing model. *Journal of Financial Economics 116*, 1–22.

Fidelity Investment (2019). Industry performance. figure retrieved from fidelity investment.

Fisher-Vanden, K. and K. Thorburn (2011). Voluntary corporate environmental initiatives and shareholder wealth. *Journal of Environmental Economics and Management 62*, 430–445.

Flammer, C. (2013). Does corporate social responsibility lead to superior financial performance? a regression discontinuity approach. *Management Science 61*.

Global Sustainable Investment Alliance (2018). 2018 global sustainable investment review. Technical report, Global Sustainable Investment Alliance.

Glosner, S. (2017). Esg risks and the cross-section of stock returns.

Halbritter, G. and G. Dorfleitner (2015). The wages of social responsibility — where are they? a critical review of esg investing. *Review of Financial Economics 26*, 25–35.

Hong, H. and M. Kacperczyk (2009). The price of sin: The effects of social norms on markets. *Journal of Financial Economics 93*, 15–36.

Humphrey, J. E. and D. T. Tan (2014). Corporate goodness and shareholder wealth: Does it really hurt to be responsible? *Journal of Business Ethics 122*, 375–386.

Hvidkjaer, S. (2017). Esg investing: a literature review.

Jacobs, B. W., V. R. Singhal, and R. Subramanian (2010). An empirical investigation of environmental performance and the market value of the firm. *SSRN Electronic Journal 28*, 430–441.

J.Brown, S. and J. B.Warner (1985). Using daily stock returns: The case of event studies. *Journal of Financial Economics 14*, 3–31.

Kempf, A. and P. Osthof (2007). The effect of socially responsible investing on financial performance. *European Financial Management 13*, 908–922.

Kothari, S. and J. B. Warner (2004). The econometrics of event studies. *SSRN*.

Kruger, P. (2015). Corporate goodness and shareholder wealth. *Journal of Financial Economics 115*, 304–329.

Ledoit, O. and M. Wolf (2008). Robust performance hypothesis testing with the sharpe ratio. *Journal of Empirical Finance 15*, 850–859.

Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics 47*, 13–37.

Loughran, T. and J. Ritter (2000). Uniformly least powerful tests of market efficiency. *Journal of Financial Economics 55*, 361–389.

Mackinlay, A. C. (1997). Event studies in economics and finance. *Journal of Economic Literature 35*, 13–39.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance 7*, 77–91.

Mossin, J. (1966). Equilibrium in a capital asset market. *Econometrica 34*, 768–783.

MSCI. Esg-rating agency.

Newey, W. K. and K. D. West (1987). A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica 55*, 703–708.

Politis, D. N. and J. P. Romano (1992). A general resampling scheme for triangular arrays of alpha0-mixing random variables with application to the problem of spectral density estimation. *The Annals of Statistics 20*, 1985–2007.

Porter, M. E. (1992). Capital choices: Changing the way america invests in industry.

Renneboog, L., J. Horst, and C. Zhang (2008). Socially responsible investments: Institutional aspects, performance, and investor behaviour. *Journal of Banking and Finance 32*, 1723–1742.

RepRisk. Esg-rating agency.

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance 19*, 425–442.

Statman, M. and D. Glushkov (2009). The wages of social responsibility. *Financial Analysts Journal 65*, 33–46.

Sustainalytics. Esg-rating agency.

Treynor, J. L. (1961). Market value, time, and risk.

Treynor, J. L. (1962). Toward a theory of market value of risky assets.

Trinks, P. J. and B. Scholtens (2017). The opportunity cost of negative screening in socially responsible investing. *Journal of Business Ethics 40*, 193–208.

Wermers, R. (2003). Is money really 'smart'? new evidence on the relation between mutual fund flows, manager behavior, and performance persistence. *SSRN Electronic Journal*.

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica 48*, 817–838.

# Appendix A

# Heteroskedasticity and Autocorrelation Consistent Standard Errors

## A.1    Ordinary Least Square(OLS)

Linear Regression assumes the following form:

$$Y = \beta X + \epsilon \tag{A.1.0.1}$$

It is normally assumed that $\epsilon_i, ..., \epsilon_j$ are i.i.d. with $E[\epsilon_i] = 0, Var(\epsilon_i) = \sigma^2$

The ordinary least square(OLS) estimator $\hat{\beta}$ is defined as:

$$\hat{\beta} = argmin||Y - X\beta||^2 \tag{A.1.0.2}$$

This gives the solution:

$$
\begin{aligned}
\hat{\beta} &= (X^T X)^{-1} X^T Y \\
&= (X^T X)^{-1} X^T (X\beta + \epsilon) \\
&= \beta + (X^T X)^{-1} X^T \epsilon
\end{aligned} \tag{A.1.0.3}
$$

For the OLS estimators to be unbiased and efficient, the following assumptions need to

hold:

- The errors have mean 0. $E[\epsilon] = 0$.

- The variance of error is constant(homoskedasticity). $VAR(\epsilon_i) = \sigma^2$ for all i.

- The errors are uncorrelated. $COV(\epsilon_i, \epsilon_j) = 0 \forall i \neq j$.

- The regression model is linear in parameters.

- X is independent of e.

- No multi-collinearity.

- The errors are jointly normally distributed(optional).

If assumptions 1-3 holds, Gauss-Markov theorem suggests that OLS is the best linear unbiased estimator(BLUE). Put differently, OLS estimator will have the lowest variance, compared with other unbiased linear estimators.

However, the error terms of time series data are oftentimes serial-correlated(autocorrelated) and heteroskedastic. In such cases, the point estimators are still unbiased, but the estimation for standard errors become inaccurate, and thus the statistical inference and tests provide unreliable results. Plus, OLS is no longer BLUE anymore.

$$
\begin{aligned}
E[\hat{\beta}] &= E[\beta + (X^T X)^{-1} X^T \epsilon] \\
&= \beta + (X^T X)^{-1} X^T E[\epsilon] \\
&= \beta
\end{aligned}
\tag{A.1.0.4}
$$

$$
\begin{aligned}
COV[\hat{\beta}] &= E[(\beta - \hat{\beta})(\beta - \hat{\beta})^T] \\
&= E[(X^T X)^{-1} X^T \epsilon [(X^T X)^{-1} X^T \epsilon]^T] \\
&= E[(X^T X)^{-1} X^T \epsilon \epsilon^T X (X^T X)^{-1}] \\
&= (X^T X)^{-1} X^T E[\epsilon \epsilon^T] X (X^T X)^{-1}
\end{aligned}
\tag{A.1.0.5}
$$

If the assumptions are met,

$$
\begin{aligned}
E[\epsilon \epsilon^T] &= COV[\epsilon] \\
&= \Omega \\
&= \sigma^2 I
\end{aligned}
\tag{A.1.0.6}
$$

$$COV[\hat{\beta}] = (X^TX)^{-1}X^TE[\epsilon\epsilon^T]X(X^TX)^{-1}$$
$$= \sigma^2(X^TX)^{-1}$$

(A.1.0.7)

## A.2 Heteroskedasticity and Serial-Correlation

If the homoskedasticity and non-serial-correlation properties are violated, $\Omega \neq \sigma^2 I$ and thus the above formula for estimating the covariance matrix will be inaccurate. If successive errors are positively correlated, we are underestimating the standard error of parameters and t-statistic will be much too high, and vice versa.

### A.2.1 Correlogram and Residual Plots

Correlogram and residual plot can provide a first inspection on these issues.

**Correlogram**

Correlogram is the standard means of visualization for autocorrelation function, which is calculated by:

$$\rho_{(k)} = \mathrm{Cor}\left(X_{t+k}, X_t\right) = \frac{\mathrm{Cov}\left(X_{t+k}, X_t\right)}{\sqrt{\mathrm{Var}\left(X_{t+k}\right)\mathrm{Var}\left(X_t\right)}}$$

(A.2.1.1)

A plug-in estimation is then:

$$\hat{\rho}(k) = \frac{\sum_{t=1}^{n-k}\left(x_{s+k} - \overline{x}\right)\left(x_s - \overline{x}\right)}{\sum_{t=1}^{n}\left(x_t - \overline{x}\right)^2}, \text{ for } k = 1, \ldots, n-1$$

(A.2.1.2)

Confidence band is also shown in the correlogram. It is computed based on the asymptotic assumption that $\hat{\rho(k)}$ follows a $N(0, \frac{1}{n})$ distribution for long i.i.d. time series. Sample autocorrelation coefficients $\hat{\rho(k)}$ that fall within the confidence band of $\pm\frac{1.96}{\sqrt{n}}$ are considered to be different from 0 only by chance, whereas those outside are considered to be truly different from 0.

I plotted the correlogram for the time-series of excess returns of the portfolio that is consisted of all controversial stocks over the risk-free-rate. This series is used in chapter 4. A rough inspection of the correlogram shows that the autocorrelation exceeds the confidence bound for lags up to 7.
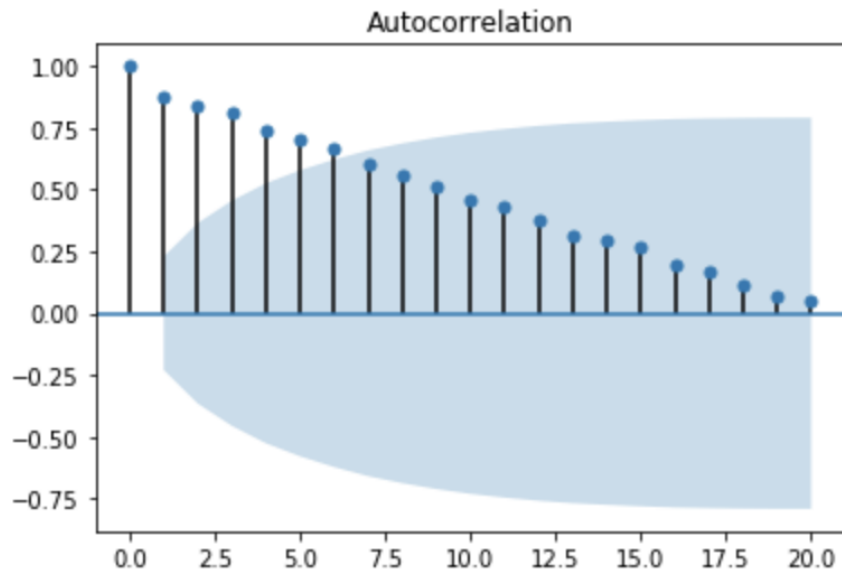
Figure A.1: Correlogram of time-series of portfolio return consisted of controversial firms over the risk-free-rate

**Residual Plot**

The residual plot is used to identify heteroskedasticity. In this case, the variance appears to increase with increased value of fitted excess return. Therefore, there is some slight deviation from the constant variance assumption.
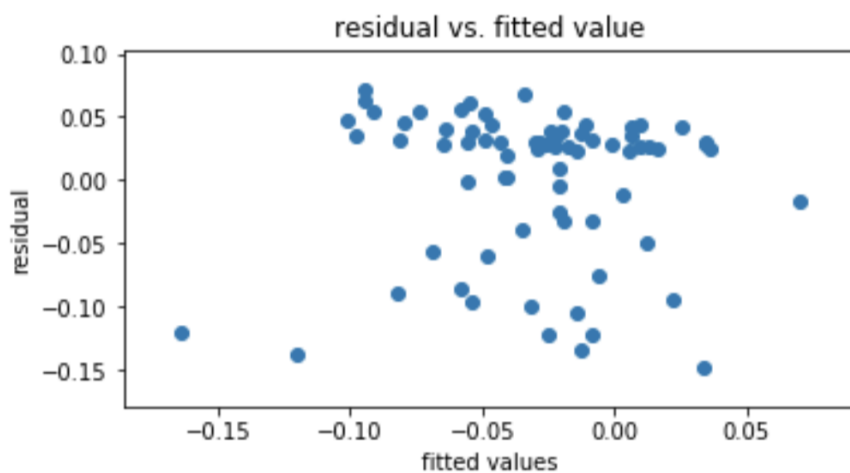


Figure A.2: Residual plot of portfolio return consisted of controversial firms over the risk-free-rate

### A.2.2 Tests for Heteroskedasticity and Serial-Correlation

**White Test for Heteroskedasticity**

White Test is a formal test for constant variance. The null hypothesis is $H_0 : \sigma_i^2 = \sigma^2$ and the alternative hypothesis is $H_1 : \sigma_i^2 \neq \sigma^2$. The squared residual from the original OLS is regressed against the original regressors along with their squares and cross-products. The lagrange multiplier(LM) is then calculated as the product of sample size and the $R^2$ value:

$$LM = nR^2 \tag{A.2.2.1}$$

The LM statistics should follow a $\chi^2$ distribution, with degree of freedom P-1, where P is the number of regressors in the auxiliary regression.

**Durbin-Watson Test for Serial-Correlation**

Assume that the regressors are strictly exogenous, which rule out the lagged dependent variables. Also assume the AR(1) case, where the error follows a first order Markov error process as follows:

$$u_t = \rho u_{t-1} + e_t, |\rho| < 1, e_{\ i.i.d} N(0, c) \tag{A.2.2.2}$$

The null hypothesis is $H_0 : \rho = 0$ and the alternative hypothesis is $H_1 : \rho \neq 0$. The Durbin-Watson test statistics is computed as:

$$DW = \frac{\sum_{t=2}^{n} (v_t - v_{t-1})^2}{\sum_{t=1}^{n} v_t^2} \tag{A.2.2.3}$$

, where v is the residual.

Simple algebra shows that DW statistic is closely related to the estimated $\rho$ from the large sample test:

$$DW \simeq 2(1 - \hat{\rho}) \tag{A.2.2.4}$$

$d = 2$ indicates no autocorrelation.

**Breusch-Godfrey Test for Serial-Correlation**

In presence of more lagged variables, the Breusch-Godfrey test is typically used. It considers the null hypotehesis of non-autocorrelated errors against an alternative that they are AR(P). The equation for AR(P) is as follows:

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \cdots + \rho_p u_{t-p} + \varepsilon_t \tag{A.2.2.5}$$

The simple regression model is first fitted by OLS to obtain a residuals $\hat{u}_t$. Then an auxiliary regression model is fitted as follows:

$$\hat{u}_t = \alpha_0 + \alpha_1 X_{t,1} + \alpha_2 X_{t,2} + \rho_1 \hat{u}_{t-1} + \rho_2 \hat{u}_{t-2} + \cdots + \rho_p \hat{u}_{t-p} + \varepsilon_t \tag{A.2.2.6}$$

The test statistics is computed as follows and it should follow a chi-square distribution:

$$nR^2 \sim \chi_p^2 \tag{A.2.2.7}$$

, where $R^2$ is obtained from the auxiliary regression.

### A.2.3 Estimators Addressing Heteroskedasticity and Autocorrelation

**White Estimator**

In case of heteroskedasticity, the covariance matrix of the OLS takes the form:

$$COV[\hat{\beta}] = \sigma^2 (X^T X)^{-1} X^T \Omega X (X^T X)^{-1} \tag{A.2.3.1}$$

, which can be used for inference once the estimator is available. Based on this consideration, White (1980) proposes the following unbiased estimator:

$$\begin{bmatrix} \hat{u_1^2} & 0 & 0 & . & . & 0 \\ 0 & \hat{u_2^2} & 0 & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & . & . & 0 & \hat{u_{T-1}^2} & 0 \\ 0 & . & . & 0 & 0 & \hat{u_T^2} \end{bmatrix} \tag{A.2.3.2}$$

The choice for $\hat{\Omega}$ leads to the following covariance variance estimator:

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \cdots + \rho_p u_{t-p} + \varepsilon_t \Sigma_\beta^W = \frac{T}{T-k} \left( \mathbf{X'X} \right)^{-1} \left( \sum_{t=1}^T \hat{u}_t^2 \mathbf{X}_t \mathbf{X}_t' \right) \left( \mathbf{X'X} \right)^{-1} \tag{A.2.3.3}$$

**Newey-West Estimator**

Newey and West (1987) proposed a more general covariance estimator that is robust to heteroskedasticity and autocorrelation of the residuals of unknown form. This heteroskedasticity-and-autocorrelation consistent(HAC) is given by:

$$\Sigma_\beta^{NW} = \left( \mathbf{X'X} \right)^{-1} T \hat{\Omega} \left( \mathbf{X'X} \right)^{-1} \tag{A.2.3.4}$$

where $\hat{\Omega}$ is the long-run covariance estimators.

$$\hat{\Omega} = \hat{\Gamma}(0) + \sum_{j=1}^p \left[ 1 - \frac{j}{p+1} \right] [\hat{\Gamma}(j) + \hat{\Gamma}(-j)] \tag{A.2.3.5}$$

$$\hat{\Gamma}(j) = \left( \sum_{t=1}^T \hat{u}_t \hat{u}_{t-j} \mathbf{X}_t \mathbf{X}_{t-j}' \right) \frac{1}{T} \tag{A.2.3.6}$$

In absence of serial-correlation, $\hat{\Omega} = \hat{\Gamma}(0)$ and it goes back to the White estimator. Note that Newey-West estimator must be calculated conditional on a maximum lag. Normally a lag exceeding the periodicity of the data is sufficient. For example, for monthly data a lag of 12 is often chosen.

## A.3    Cluster-robust Inference

In this section, we present the fundamentals of cluster-robust inference. For these basic results, we first assume that the following: i) the model does not include cluster-specific fixed effects, ii) it is clear how to form the clusters, and iii) there are many clusters.

Clustered errors have two main consequences: they reduce the precision of parameter estimation and the standard estimator for the variance of parameters is biased downward from the true variance.

$$\boldsymbol{y}_g = \boldsymbol{X}_g\boldsymbol{\beta} + \boldsymbol{u}_g, \quad g = 1, \ldots, G \tag{A.3.0.1}$$

### A.3.1    One-way Clustering

The key assumption for the clustered-covariance is that errors are uncorrelated across clusters, while they can be correlated within clusters.

$$\mathrm{E}\left[u_{ig}u_{jg'}|\boldsymbol{x}_{ig}, \boldsymbol{x}_{jg'}\right] = 0 \, for \, g \neq g' \tag{A.3.1.1}$$

The OLS estimator is:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} = \left(\sum_{g=1}^{G} \boldsymbol{X}_g'\boldsymbol{X}_g\right)^{-1}\sum_{g=1}^{G}\boldsymbol{X}_g'\boldsymbol{y}_g \tag{A.3.1.2}$$

The estimated covariance matrix is:

$$\mathrm{V}[\widehat{\boldsymbol{\beta}}] = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{B}(\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{A.3.1.3}$$

with

$$\boldsymbol{B} = \boldsymbol{X}'\mathrm{V}[\boldsymbol{u}|\boldsymbol{X}]\boldsymbol{X} \tag{A.3.1.4}$$

Given that the error is independent across clusters, $\mathrm{V}[\boldsymbol{u}|\boldsymbol{X}]$ has a block-diagonal structure, and thus the previous equation simplifies to:

$$B_{\mathrm{clu}} = \sum_{g=1}^{G} X_g'\mathrm{E}\left[\boldsymbol{u}_g\boldsymbol{u}_g'|\boldsymbol{X}_g\right]\boldsymbol{X}_g \tag{A.3.1.5}$$

The cluster-robust estimate of covariance matrix can be written as:

$$\widehat{\nabla}_{\text{clu}}[\widehat{\boldsymbol{\beta}}] = \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\widehat{\boldsymbol{B}}_{\text{clu}}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1} \tag{A.3.1.6}$$

, where

$$\widehat{\boldsymbol{B}}_{\text{clu}} = \sum_{g=1}^{G}\boldsymbol{X}_g'\widehat{\boldsymbol{u}}_g\widehat{\boldsymbol{u}}_g'\boldsymbol{X}_g \tag{A.3.1.7}$$

### A.3.2 Multi-way Clustering

For multi-way clustering, the key assumption becomes:

$$\mathrm{E}\left[u_{ig}u_{jg'}|\boldsymbol{x}_{ig},\boldsymbol{x}_{jg'}\right] = 0\text{unless i and j share any cluster dimension} \tag{A.3.2.1}$$

Then the formula for covariance estimation becomes:

$$\widehat{\boldsymbol{B}} = \sum_{i=1}^{N}\sum_{j=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_j'\widehat{u}_i\widehat{u}_i\mathbf{1}[i,j \text{ share any cluster }] \tag{A.3.2.2}$$

It can be easily shown that the two-way cluster-robust covariance estimation can be rewritten as:

$$\widehat{\mathrm{V}}_{2\text{way}}[\widehat{\boldsymbol{\beta}}] = \widehat{\mathrm{V}}_1[\widehat{\boldsymbol{\beta}}] + \widehat{\mathrm{V}}_2[\widehat{\boldsymbol{\beta}}] - \widehat{\mathrm{V}}_{1\cap2}[\widehat{\boldsymbol{\beta}}] \tag{A.3.2.3}$$

# Appendix B

# Hypothesis Testing with Sharpe Ratio Test

Assume that we have two portfolios: portfolio i with a series of excess returns $r_{i1}, ..., r_{it}$ and portfolio n with excess returns $r_{n1}, ..., r_{nt}$]. Their mean vector $\mu$ and covariance matrix $\Sigma$ is defined as follows:

$$\mu = \begin{bmatrix} \mu_i \\ \mu_n \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_i^2 & \sigma_{in} \\ \sigma_{ni} & \sigma_n^2 \end{bmatrix}$$

The difference between two Sharpe ratios is given by:

$$\begin{aligned}
\delta &= s_i - s_n \\
&= \frac{\mu_i}{\sigma_i} - \frac{\mu_n}{\sigma_n} \\
&= \frac{\mu_i}{\sqrt{\gamma_i - \mu_i^2}} - \frac{\mu_n}{\sqrt{\gamma_n - \mu_n^2}} \\
&= f(\mu_i, \gamma_i, \mu_n, \gamma_n) \\
&= f(v)
\end{aligned} \tag{B.0.0.1}$$

, where $\gamma_i = E[r_i^2]$ and $v = [\mu_i, \gamma_i, \mu_n, \gamma_n]$.

It is assumed that $\sqrt{T}(\hat{v} - v) \to N(0, \Psi)$, where $\Psi$ is an unknown symmetric positive semi-definite matrix. Then this implies that:

$$\sqrt{T}(\hat{\Delta} - \Delta) \xrightarrow{d} N\left(0; \nabla' f(v) \Psi \nabla f(v)\right) \tag{B.0.0.2}$$

Now given a consistent estimator $\hat{\Psi}$, the standard error of the estimated difference can be computed as:

$$s(\hat{\Delta}) = \sqrt{\frac{\nabla' f(\hat{\mathfrak{v}}) \hat{\Psi} \nabla f(\hat{\mathfrak{v}})}{T}} \tag{B.0.0.3}$$

Given the standard error, the p-value can be estimated as:

$$\hat{p} = 2\Phi\left(-\frac{|\hat{\Delta}|}{s(\hat{\Delta})}\right) \tag{B.0.0.4}$$

Bootstrap leads to higher inference accuracy than standard inference procedures under the assumption of asymptotic normal distribution. Therefore, Ledoit and Wolf (2008) propose to test the null hypothesis by constructing a time-series bootstrap confidence interval with nominal level $\alpha$. If zero is not included in the confidence interval, the null hypothesis of equal Sharpe ratios is rejected. The distribution function of the studentized difference is approximated by a distribution from the bootstrap samples.

$$\mathcal{L}\left(\frac{|\hat{\Delta} - \Delta||}{s(\hat{\Delta})}\right) \approx \mathcal{L}\left(\frac{\left|\hat{\Delta}^* - \hat{\Delta}\right|}{s\left(\hat{\Delta}^*\right)}\right) \tag{B.0.0.5}$$

The bootstrap data is generated using the circular block bootstrap method of Politis and Romano (1992), resampling blocks of pairs of excess returns from the observed values with replacement. The estimator $\Psi$ is obtained from HAC kernel inference, specifically, a prewhitened quadratic spectral kernel estimation of Andrews and Monahan (1992). The standard error and p-value are calculated using the same formulas above.

To conduct this bootstrap, we can either follow Ledoit and Wolf (2008) and use the calibration procedure for selecting the required fixed block size, or follow Behr, Guettler, and Truebenbach (2012) that selects the size manually to preserve the autocorrelation structure in the data. We follow the latter one and set the block size to 5. For the number of bookstraps, we set the number to 5000, as this is sufficient for reliable inference suggested by ?Ledoit2008). As to the standard error, we calculated it based on HAC kernel estimation.

# Appendix C

# Code

## C.1 Event Study

```python
import pandas as pd
import numpy as np
from datetime import datetime
import matplotlib.pyplot as plt
import warnings
import datetime as dt
from scipy import stats
import math
import operator

''' read data '''
data_ret = pd.read_pickle('data_ret_new.pkl')
df_negative = pd.read_pickle('df_negative_new_1.pkl')
df_negative = df_negative[df_negative.date > dt.datetime(2012,1,3)
    ]
data_ret = data_ret[data_ret.date > dt.datetime(2012,1,3)]
data_ret.index= data_ret['date']
data_ret = data_ret.drop(columns = ['date'])
data_ret= data_ret.iloc[:,(data_ret.values < 1).all(axis=0) & (
    data_ret.values > -1).all(axis=0)]

stocks = data_ret.columns.unique().tolist()
stocks.remove('^GSPC')

''' parameter initialization '''
window = 10
estimation_period = 200
windows_indexes = range(-window, window + 1)
estimation_indexes = range(-estimation_period- window, - window)

dr_equities_window = pd.DataFrame(index=windows_indexes)
dr_equities_estimation = pd.DataFrame(index=estimation_indexes)
dr_market_window = pd.DataFrame(index=windows_indexes)
dr_market_estimation = pd.DataFrame(index=estimation_indexes)
```

```
33
34   ''' for each stock, compute daily stock and market return during
        the window period'''
35   for symbol in stocks:
36       ''' get the date on which negative event happens '''
37       neg_event_dates = df_negative[df_negative.ticker == symbol].
            date.tolist()
38       for neg_event in neg_event_dates:
39           evt_idx = data_ret.index.get_loc(neg_event)
40           col_name = symbol + ' ' + neg_event.strftime('%Y-%m-%d')
41
42           ''' starting and ending index for the estimation period
                '''
43           start_idx = evt_idx - window - estimation_period
44           end_idx = evt_idx - window
45
46           if start_idx < 0 or evt_idx + window >= data_ret.shape[0]:
47               continue
48
49           ''' compute daily stock and market return during the pre-
                event window '''
50           new_dr_equities_estimation = data_ret[symbol][start_idx:
                end_idx]
51           new_dr_equities_estimation.index = estimation_indexes
52           dr_equities_estimation[col_name] =
                new_dr_equities_estimation
53
54           new_dr_market_estimation = data_ret['^GSPC'][start_idx:
                end_idx]
55           new_dr_market_estimation.index = estimation_indexes
56           dr_market_estimation[col_name] = new_dr_market_estimation
57
58           ''' starting and ending index for the window period '''
59           start_idx = evt_idx - window
60           end_idx = evt_idx + window + 1
61
62           ''' compute daily stock and market return during the
                window period '''
63           new_dr_equities_window = data_ret[symbol][start_idx:
                end_idx]
64           new_dr_equities_window.index = windows_indexes
65           dr_equities_window[col_name] = new_dr_equities_window
66
67           new_dr_market_window = data_ret['^GSPC'][start_idx:end_idx
                ]
68           new_dr_market_window.index = windows_indexes
69           dr_market_window[col_name] = new_dr_market_window
70
71   ''' initalize regression_estimation and expected_return datasets
        '''
72   reg_estimation = pd.DataFrame(index=dr_market_estimation.columns,
        columns=['intercept', 'beta', 'rse'])
73   er = pd.DataFrame(index=dr_market_window.index, columns=
        dr_market_window.columns)
74
```

```
75  ''' compute the expected return of each date using the regression
        '''
76  for col in dr_market_estimation.columns:
77      x = dr_market_estimation[col]
78      y = dr_equities_estimation[col]
79      slope, intercept, _, _, _ = stats.linregress(x, y)
80      reg_estimation['beta'][col] = slope
81      reg_estimation['intercept'][col] = intercept
82      reg_estimation['rse'][col] = sum((y - slope * x - intercept)
            **2)/(len(x)-2)
83      er[col] = intercept + dr_market_window[col] * slope
84
85  er.columns.name = 'Expected return'
86  ''' calculate abnormal return '''
87  ar = dr_equities_window - er
88  ar.columns.name = 'Abnormal return'
89
90  ''' remove AR with NA or all values as 0 '''
91  ar = ar.dropna(axis = 1, how = 'any')
92  ar = ar.loc[:, (ar != 0).any(axis=0)]
93
94  ''' calculate cumulative abnormal return '''
95  car = ar.apply(np.cumsum)
96  car.columns.name = 'Cum Abnormal Return'
97
98  ''' calcualte mean for CAR and AR '''
99  mean_car = car.mean(axis=1)
100 mean_car.name = 'CAR'
101
102 mean_ar = ar.mean(axis=1)
103 mean_ar.name = 'AR'
104
105 ''' compute mean and variance for CAAR '''
106 cols = car.columns.tolist()
107 rse = reg_estimation.loc[reg_estimation['rse'].index.isin(cols),'
        rse']
108 var_car = (window*2+1) * rse
109 mean_val = mean_car[window]
110 print('mean = %.5f'%(mean_val))
111
112 N = len(cols)
113 var_caar = sum(var_car)/(N*N)
114 t_car = mean_val/math.sqrt(var_caar)
115
116 pval = stats.t.sf(np.abs(t_car), N-1)*2
117 print('t-statistic = %6.3f, pvalue = %6.4f'%(t_car, pval))
118
119 ''' plot CAR'''
120 x = mean_car.index.values
121 y = mean_car.values
122 z = mean_ar.values
123 fig, ax = plt.subplots(figsize=(8,5))
124 label = mean_car.name
125 ax.plot(x, y, label=label)
126 ax.set_xlabel('window')
```

```
127  ax.set_ylabel('CAR')
128  plt.title('cumulative abnormal return(CAR) within the event window
         ')
129  plt.legend()
130
131
132  ''' compute CAR by label '''
133  def car_by_label(label):
134      ''' select ticker and date for negative event in this label
             category '''
135      negative_by_label = df_negative[df_negative.label2 == label
             ][['ticker', 'date']]
136      negative_by_label['date_str'] = negative_by_label['date'].
             apply(lambda x: x.strftime('%Y-%m-%d'))
137      negative_by_label['col_names'] = negative_by_label['ticker'] +
             ' ' + negative_by_label['date_str']
138      ''' select these events from CAR '''
139      col_names = negative_by_label['col_names'].values.tolist()
140      col_names = [col_name for col_name in col_names if col_name in
             car.columns.tolist()]
141      car_sub = car[col_names]
142
143      ''' calcualte mean CAR for these events '''
144      mean_car = car_sub.mean(axis=1)
145      ''' calculate variance CAR for all events in this category '''
146      cols = car_sub.columns.tolist()
147      N = len(cols)
148      rse = reg_estimation.loc[reg_estimation['rse'].index.isin(cols
             ),'rse']
149      var_car_i = (window*2+1) * rse
150      std_car = math.sqrt(sum(var_car_i)/N**2)
151      t_car = mean_car[window]/std_car
152      pval = stats.t.sf(np.abs(t_car), N-1)*2
153
154      return mean_car, std_car, pval
155
156  mean_dic = {}
157  pval_dic = {}
158  std_dic = {}
159  mean_cum_val = {}
160  label2 = df_negative['label2'].unique().tolist()
161  for label in label2:
162      print(label)
163      mean_car, std_car, pval = car_by_label(label)
164      print(mean_car)
165      mean_dic[label] = mean_car
166      mean_cum_val[label] = mean_car[window]
167      pval_dic[label] = round(pval,5)
168      std_dic[label] = std_car
169
170  ''' plot CAR for each event subcategory during the event window'''
171  fig, axs = plt.subplots(6,3, figsize=(10, 8), constrained_layout=
         True)
172  axs = axs.ravel()
173  for i,var in enumerate(mean_dic):
```

```
174       x = mean_dic[var].index.values
175       y = mean_dic[var].values
176       yerr = std_dic[var]
177       ymin = y - 1.65*yerr
178       ymax = y + 1.65*yerr
179       axs[i].plot(x, y)
180       axs[i].set_title(var)
181   plt.show()
182
183   ''' plot CAR and confidence interval at the end of event window'''
184   yerror=[std_dic[i]*1.65 for i in std_dic]
185   mean_window_end = {k: mean_dic[k][window] for k in mean_dic}
186   plt.figure(figsize=(9,5))
187   plt.xticks(rotation=90)
188   x = range(len(mean_window_end))
189   y = list(mean_window_end.values())
190   plt.axhline(color='black')
191   plt.errorbar(x, y, yerr=yerror, fmt='o',lolims=True, uplims=True)
192   plt.xticks(x, list(mean_window_end.keys()))
193   plt.title('CARs at the end of window period(day 20) by event
          category')
194   plt.xlabel('event category')
195   plt.ylabel('CAR')
196   plt.show()
```

## C.2   Jensen's Alpha

```
1   import pandas as pd
2   import numpy as np
3   from datetime import datetime
4   import matplotlib.pyplot as plt
5   import matplotlib.dates as mdates
6   import warnings
7   import datetime as dt
8   from scipy import stats
9   import math
10  import operator
11  from statsmodels.graphics.tsaplots import plot_acf
12
13  ''' read and preprocess data'''
14  factors = pd.read_pickle('factors_monthly.pkl')
15  factors = factors[(factors.index >= '2013-1-1') & (factors.index <
          '2019-2-1')]
16
17  factors_5 = pd.read_csv('fama_french_5_factor.csv')
18  factors_5 = factors_5.iloc[12:-5]
19  factors_5 = factors_5[['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA', 'RF
          ']]
20  factors_5['const'] = 1
21
22  data_ret = pd.read_csv('monthly_return.csv', header=1)
23  data_ret = data_ret.iloc[1:,:]
24  data_ret.index = data_ret.ticker
25  data_ret = data_ret.iloc[:,1:]
```

```
26  data_ret /= 100
27  data_ret['date'] = data_ret.index
28  data_ret.index = data_ret.date.apply(lambda x: datetime.strptime(x
        ,'%Y-%m-%d'))
29  data_ret = data_ret.drop('date', axis=1)
30  data_ret = data_ret.dropna(axis=1)
31  data_ret_1 = data_ret.dropna(axis=1)
32  data_ret= data_ret.iloc[:,(data_ret.values < np.percentile(
        data_ret_1,99)).all(axis=0) & (data_ret.values > np.percentile
        (data_ret_1,1)).all(axis=0)]
33
34  df_negative = pd.read_pickle('df_negative_new_1.pkl')
35  stock_tickers = data_ret.columns.tolist()
36
37  ''' equal-weighted portfolio'''
38  event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
39  monthly_rets = []
40  num_firms = []
41  while event_starting_date < pd.Timestamp(year=2018, month=2,day=1)
        :
42      event_ending_date = relativedelta(months=12) +
            event_starting_date
43      portfolio_starting_date = event_ending_date
44      portfolio_ending_date = portfolio_starting_date +
            relativedelta(months=1)
45
46      ''' find stocks with negative news within the last 12 months
            '''
47      df = df_negative[(df_negative.date < event_ending_date)&(
            df_negative.date >= event_starting_date)]
48      tickers = df.ticker.unique().tolist()
49      new_tickers = list(set(tickers) & set(stock_tickers))
50      num_firms.append(len(new_tickers))
51
52      ''' compute monthly return for every stock '''
53      rets = data_ret[new_tickers][portfolio_starting_date:
            portfolio_ending_date]
54      monthly_ret = np.mean(rets, axis=1).values[0]
55      monthly_rets.append(monthly_ret)
56      event_starting_date += relativedelta(months=1)
57
58  '''Fama-French five-factor model'''
59  factors_5['R'] = monthly_rets
60  factors_5['R-RF'] = factors_5['R'] - factors_5['RF']
61  mod = sm.OLS(factors_5['R-RF'], factors_5[['SMB', 'HML', 'Mkt-RF
        ','RMW', 'CMA', 'const']])
62  res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
63  res.summary()
64
65  '''Carhart four-factor model'''
66  factors['R'] = monthly_rets
67  factors['R-RF'] = factors['R'] -factors['RF']
68  factors['const'] = 1
69  factors.columns = ['Mkt-RF', 'SMB', 'HML', 'RF', 'MKT', 'Mom', 'R
        ', 'R-RF', 'const']
```

```
70   mod = sm.OLS(factors['R-RF'], factors[['SMB', 'HML', 'MKT','Mom',
         'const']])
71   res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
72   res.summary()
73
74   '''Fama-French three-factor model'''
75   mod = sm.OLS(factors['R-RF'], factors[['SMB', 'HML', 'Mkt-RF', '
         const']])
76   res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
77   res.summary()
78
79   '''residual and ACF plot'''
80   resids = res.resid.values.tolist()
81   y_fitted = res.fittedvalues
82   plot_acf(resids, lags=20)
83
84   fig, ax = plt.subplots(figsize=(6,3))
85   _ = ax.scatter(y_fitted, resids)
86   plt.xlabel('fitted values')
87   plt.ylabel('residual')
88   plt.title('residual vs. fitted value')
89
90   ''' run the regression by event category'''
91   labels = df_negative.label2.unique().tolist()
92   alphas = []
93   pvals = []
94   num_firms_list = []
95
96   for label in labels:
97       df_sub = df_negative[df_negative.label2 == label]
98       print('==== currently running regression for label: {}'.format
             (label))
99
100      event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
101      monthly_rets = []
102      num_firms = []
103
104      while event_starting_date < pd.Timestamp(year=2018, month=2,
             day=1):
105          event_ending_date = relativedelta(months=12) +
                 event_starting_date
106          portfolio_starting_date = event_ending_date
107          portfolio_ending_date = portfolio_starting_date +
                 relativedelta(months=1)
108
109          df = df_sub[(df_sub.date < event_ending_date)&(df_sub.date
                 >= event_starting_date)]
110          tickers = df.ticker.unique().tolist()
111          new_tickers = list(set(tickers) & set(stock_tickers))
112          num_firms.append(len(new_tickers))
113
114          rets = data_ret[new_tickers][portfolio_starting_date:
                 portfolio_ending_date]
115          monthly_ret = np.mean(rets, axis=1).values[0]
116          monthly_rets.append(monthly_ret)
```

```
117        event_starting_date += relativedelta(months=1)
118     factors['R'] = monthly_rets
119     factors['R-RF'] = factors['R'] -factors['RF']
120     mod = sm.OLS(factors['R-RF'], factors[['SMB', 'HML', 'MKT', '
           Mom', 'const']])
121     res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
122
123     alphas.append(res.params[-1])
124     pvals.append(res.pvalues[-1])
125     num_firms_list.append(np.mean(num_firms))
126
127 d = {'label': labels, 'alpha': alphas, 'p-value': pvals, '
       num_firms': num_firms_list}
128 res = pd.DataFrame(d)
129 res.index = res.label
130 res = res.drop(columns = 'label')
131 res['num_firms'] = res['num_firms'].astype(int)
132 res = res.sort_values('num_firms', ascending=False)
133 res = res[['alpha', 'p-value', 'num_firms']]
134 res
135
136 '''value-weighted portfolios'''
137 market_cap = pd.read_csv('market_cap.csv', header=1)
138 market_cap = market_cap.iloc[1:,:]
139 market_cap.index = market_cap.ticker.apply(lambda x: datetime.
       strptime(x, '%Y-%m-%d'))
140 market_cap = market_cap.drop('ticker', axis=1)
141 market_cap_1 = market_cap.dropna(axis=1)
142 mc_max_cut = np.percentile(market_cap_1,99)
143 mc_min_cut = np.percentile(market_cap_1,1)
144 market_cap_1 = market_cap.iloc[:,(market_cap.values < mc_max_cut).
       all(axis=0) & (market_cap.values > mc_min_cut).all(axis=0)]
145 market_cap = market_cap_1
146
147 event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
148 monthly_rets = []
149 num_firms = []
150 while event_starting_date < pd.Timestamp(year=2018, month=2,day=1)
       :
151     event_ending_date = relativedelta(months=12) +
           event_starting_date
152     portfolio_starting_date = event_ending_date
153     portfolio_ending_date = portfolio_starting_date +
           relativedelta(months=1)
154
155     df = df_negative[(df_negative.date < event_ending_date)&(
           df_negative.date >= event_starting_date)]
156     tickers = df.ticker.unique().tolist()
157     new_tickers = list(set(tickers) & set(stock_tickers) & set(
           market_cap.columns))
158     num_firms.append(len(new_tickers))
159     rets = data_ret[new_tickers][portfolio_starting_date:
           portfolio_ending_date]
160     mc = market_cap[new_tickers][portfolio_starting_date:
           portfolio_ending_date]
```

```
161        rets = rets[sorted(rets.columns)]
162        mc = rets[sorted(mc.columns)]
163        if (sorted(mc.columns) != sorted(rets.columns)) or (len(mc.
              columns) != len(rets.columns)):
164            print('False!')
165        value_weighted_rets = mc*rets/np.sum(mc, axis=1).values[0]
166        monthly_ret = np.mean(value_weighted_rets, axis=1).values[0]
167
168        monthly_rets.append(monthly_ret)
169        event_starting_date += relativedelta(months=1)
170
171  '''value-weighted four-factor model'''
172  factors['R'] = monthly_rets
173  factors['R-RF'] = factors['R'] -factors['RF']
174  mod = sm.OLS(factors['R-RF'], factors[['SMB', 'HML', 'MKT','Mom',
          'const']])
175  res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
176  res.summary()
177
178  '''winsorize stock returns'''
179  q95 = np.percentile(data_ret.values,95)
180  q5 = np.percentile(data_ret.values,5)
181  data_ret_1 = data_ret.iloc[:,(data_ret.values < q95).all(axis=0) &
          (data_ret.values > q5).all(axis=0)]
182
183  stock_tickers = data_ret_1.columns.tolist()
184  event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
185  monthly_rets = []
186  num_firms = []
187  while event_starting_date < pd.Timestamp(year=2018, month=2,day=1)
          :
188        event_ending_date = relativedelta(months=12) +
              event_starting_date
189        portfolio_starting_date = event_ending_date
190        portfolio_ending_date = portfolio_starting_date +
              relativedelta(months=1)
191
192        df = df_negative[(df_negative.date < event_ending_date)&(
              df_negative.date >= event_starting_date)]
193        tickers = df.ticker.unique().tolist()
194        new_tickers = list(set(tickers) & set(stock_tickers))
195        num_firms.append(len(new_tickers))
196
197        rets = data_ret_1[new_tickers][portfolio_starting_date:
              portfolio_ending_date]
198        monthly_ret = np.mean(rets, axis=1).values[0]
199        monthly_rets.append(monthly_ret)
200        event_starting_date += relativedelta(months=1)
201
202  factors['R'] = monthly_rets
203  factors['R-RF'] = factors['R'] -factors['RF']
204  factors['const'] = 1
205  factors.columns = ['Mkt-RF', 'SMB', 'HML', 'RF', 'MKT', 'Mom', 'R
          ', 'R-RF', 'const']
206  mod = sm.OLS(factors['R-RF'], factors[['SMB', 'HML', 'MKT','Mom',
```

```
          'const ']])
207  res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
208  res.summary()
209
210  '''industry-matched portfolio'''
211
212  industry = pd.read_csv('industry.csv')
213  industry = industry[['ticker', 'industry', 'industry_sub']]
214  industry.columns = ['tiker', 'industry1', 'industry2']
215  industry.loc[(industry.industry1 == 'Food, Beverage & Tobacco') |
216              (industry.industry1 == 'Household & Personal Products
                      ') |
217              (industry.industry2 == 'Textiles, Apparel & Luxury
                      Goods')
218              , 'industry'] = 'NoDur'
219
220  industry.loc[(industry.industry2 == 'Household Durables') |
221              (industry.industry2 == 'Leisure Products')
222              , 'industry'] = 'Durbl'
223
224  industry.loc[(industry.industry1 == 'Automobiles & Components') |
225              (industry.industry1 == 'Capital Goods') |
226              (industry.industry2 == 'Paper & Forest Products')
227              , 'industry'] = 'Manuf'
228
229  industry.loc[(industry.industry1 == 'Software & Services') |
230              (industry.industry1 == 'Technology Hardware &
                      Equipment') |
231              (industry.industry1 == 'Semiconductors &
                      Semiconductor Equipment')
232              , 'industry'] = 'BusEq'
233
234  industry.loc[industry.industry1 == 'Energy', 'industry'] = 'Enrgy'
235  industry.loc[industry.industry1 == 'Telecommunication Services', '
        industry'] = 'Telcm'
236  industry.loc[industry.industry1 == 'Utilities', 'industry'] = '
        Utils'
237  industry.loc[industry.industry2 == 'Chemicals', 'industry'] = '
        Chems'
238
239  industry.loc[(industry.industry1 == 'Retailing') |
240              (industry.industry1 == 'Food & Staples Retailing') ,
                      'industry'] ='Shops'
241
242  industry.loc[(industry.industry1 == 'Health Care Equipment &
        Services') |
243              (industry.industry1 == 'Pharmaceuticals,
                      Biotechnology & Life Sciences')
244              , 'industry'] ='Hlth'
245
246  industry.loc[(industry.industry1 == 'Diversified Financials') |
247              (industry.industry1 == 'Insurance') |
248              (industry.industry1 == 'Banks') |
249               (industry.industry1 == 'Real Estate')
250              , 'industry'] ='Money'
```

```
251
252  industry.loc[(industry.industry1 == 'Media & Entertainment') |
253               (industry.industry1 == 'Consumer Services') |
254               (industry.industry1 == 'Commercial  & Professional
                      Services') |
255             (industry.industry1 == 'Transportation') |
256               (industry.industry2 == 'Metals & Mining') |
257               (industry.industry2 == 'Construction Materials') |
258               (industry.industry2 == 'Containers & Packaging')
259               , 'industry'] ='Other'
260
261  industry = industry.dropna(subset = ['industry'])
262
263  '''substract the industry return from the company monthly return
          '''
264  industry = industry.drop(columns = ['Unnamed: 0'])
265  industry_portfolios = pd.read_csv('12_portfolio_equal_weighted.csv
          ')
266  industry_portfolios['Unnamed: 0'] = industry_portfolios['Unnamed:
          0'].apply(lambda x: datetime.strptime(str(x),'%Y%m'))
267  industry_portfolios.columns = ['date', 'NoDur', 'Durbl', 'Manuf',
          'Enrgy', 'Chems', 'BusEq',
268          'Telcm', 'Utils', 'Shops', 'Hlth ', 'Money', 'Other']
269  industry_portfolios.index = industry_portfolios.date
270  industry_portfolios = industry_portfolios.iloc[:,1:]
271  industry_portfolios/=100
272  industry_portfolios['date'] = industry_portfolios.index
273  industry_portfolios = industry_portfolios[(industry_portfolios.
          date >= datetime.strptime('2011-11-01','%Y-%m-%d'))
274                                            & (industry_portfolios.
                                                  date <= datetime.
                                                  strptime
                                                  ('2019-01-01','%Y-%m
                                                  -%d'))]
275  data_ret_1 = data_ret
276  data_ret_1.index = data_ret_1.index - pd.tseries.offsets.MonthEnd
          () + timedelta(days=1)
277
278  industry_portfolios.columns = ['NoDur', 'Durbl', 'Manuf', 'Enrgy',
          'Chems', 'BusEq', 'Telcm', 'Utils',
279          'Shops', 'Hlth', 'Money', 'Other', 'date']
280
281  for t in data_ret_1.columns:
282      indstr = industry[industry.ticker == t].industry.values.tolist
              ()
283      if len(indstr) == 0:
284          data_ret_1.drop(columns = t,inplace=True)
285      else:
286          indstr = indstr[0]
287          data_ret_1[t] -= industry_portfolios[indstr]
288
289  ''' construct the industry-matched portfolio'''
290  stock_tickers = data_ret_1.columns.tolist()
291  event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
292  monthly_rets = []
```

```
293  num_firms = []
294  while event_starting_date < pd.Timestamp(year=2018, month=2,day=1)
         :
295      event_ending_date = relativedelta(months=12) +
             event_starting_date
296      portfolio_starting_date = event_ending_date
297      portfolio_ending_date = portfolio_starting_date +
             relativedelta(months=1)
298      df = df_negative[(df_negative.date < event_ending_date)&(
             df_negative.date >= event_starting_date)]
299      tickers = df.ticker.unique().tolist()
300      new_tickers = list(set(tickers) & set(stock_tickers))
301      num_firms.append(len(new_tickers))
302      rets = data_ret_1[new_tickers][portfolio_starting_date:
             portfolio_ending_date]
303      monthly_ret = np.mean(rets, axis=1).values[0]
304      monthly_rets.append(monthly_ret)
305      event_starting_date += relativedelta(months=1)
306
307  factors['R'] = monthly_rets
308  factors['R-RF'] = factors['R'] -factors['RF']
309  factors['const'] = 1
310  factors.columns = ['Mkt-RF', 'SMB', 'HML', 'RF', 'MKT', 'Mom', 'R
         ', 'R-RF', 'const']
311  mod = sm.OLS(factors['R'], factors[['SMB', 'HML', 'MKT','Mom', '
         const']])
312  res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
313  res.summary()
314
315  '''characteristics matched portfolio '''
316  mc_all_stocks = pd.read_csv('market_cap_all_stocks_1.csv')
317  mc_all_stocks.date = mc_all_stocks.date.apply(lambda x: datetime.
         strptime(x, '%Y-%m-%d'))
318  mc_all_stocks.index = mc_all_stocks.date
319  mc_all_stocks = mc_all_stocks.drop(columns= ['date'])
320
321  bv_all_stocks = pd.read_csv('bv_all_stocks_1.csv')
322  bv_all_stocks.date = bv_all_stocks.date.apply(lambda x: datetime.
         strptime(x, '%Y-%m-%d'))
323  bv_all_stocks.index = bv_all_stocks.date
324  bv_all_stocks = bv_all_stocks.drop(columns= ['date'])
325
326  yearly_ret_all_stocks = pd.read_csv('yearly_return_all_stocks_1.
         csv')
327  yearly_ret_all_stocks.date = yearly_ret_all_stocks.date.apply(
         lambda x: datetime.strptime(x, '%Y-%m-%d'))
328  yearly_ret_all_stocks.index = yearly_ret_all_stocks.date
329  yearly_ret_all_stocks = yearly_ret_all_stocks.drop(columns= ['date
         '])
330
331  mc_all_stocks = mc_all_stocks.dropna(axis=1)
332  bv_all_stocks = bv_all_stocks.dropna(axis=1)
333  yearly_ret_all_stocks = yearly_ret_all_stocks.dropna(axis=1)
334
335  mc_esg_stocks = pd.read_csv('yearly_market_cap_esg_stocks.csv',
```

```
          header =1)
336 | mc_esg_stocks = mc_esg_stocks.iloc[1:,:]
337 | mc_esg_stocks.index = mc_esg_stocks.ticker
338 | mc_esg_stocks = mc_esg_stocks.iloc[:,1:]
339 |
340 | bv_esg_stocks = pd.read_csv('bv_esg_stocks.csv',header=1)
341 | bv_esg_stocks = bv_esg_stocks.iloc[1:,:]
342 | bv_esg_stocks.index = bv_esg_stocks.ticker
343 | bv_esg_stocks = bv_esg_stocks.iloc[:,1:]
344 |
345 | yearly_ret_esg_stocks = pd.read_csv('yearly_return_esg_stocks.csv
          ',header=1)
346 | yearly_ret_esg_stocks = yearly_ret_esg_stocks.iloc[1:,:]
347 | yearly_ret_esg_stocks.index = yearly_ret_esg_stocks.ticker
348 | yearly_ret_esg_stocks = yearly_ret_esg_stocks.iloc[:,1:]
349 |
350 | mc_esg_stocks.columns =[t.split('.', 1)[0].lower() for t in
          mc_esg_stocks.columns.tolist()]
351 | bv_esg_stocks.columns =[t.split('.', 1)[0].lower() for t in
          bv_esg_stocks.columns.tolist()]
352 | yearly_ret_esg_stocks.columns =[t.split('.', 1)[0].lower() for t
          in yearly_ret_esg_stocks.columns.tolist()]
353 | mc_all_stocks.columns =[t.split('.', 1)[0].lower() for t in
          mc_all_stocks.columns.tolist()]
354 | bv_all_stocks.columns =[t.split('.', 1)[0].lower() for t in
          bv_all_stocks.columns.tolist()]
355 | yearly_ret_all_stocks.columns = [t.split('.', 1)[0].lower() for t
          in yearly_ret_all_stocks.columns.tolist()]
356 |
357 | monthly_ret_all_stocks = pd.read_csv('monthly_ret_all_stocks_1.csv
          ')
358 | monthly_ret_all_stocks.date = monthly_ret_all_stocks.date.apply(
          lambda x: datetime.strptime(x, '%Y-%m-%d'))
359 | monthly_ret_all_stocks.index = monthly_ret_all_stocks['date']
360 | monthly_ret_all_stocks = monthly_ret_all_stocks.iloc[:,1:]
361 | monthly_ret_all_stocks /= 100
362 | monthly_ret_all_stocks.columns = [t.split('.', 1)[0].lower() for t
           in monthly_ret_all_stocks.columns.tolist()]
363 | d = [mc_all_stocks.columns, bv_all_stocks.columns,
          yearly_ret_all_stocks.columns, monthly_ret_all_stocks.columns]
364 | tickers = list(set.intersection(*map(set,d)))
365 | tickers = [t for t in tickers if t not in df_negative.ticker.
          unique().tolist()]
366 |
367 | mc_all_stocks = mc_all_stocks[tickers]
368 | bv_all_stocks = bv_all_stocks[tickers]
369 | yearly_ret_all_stocks = yearly_ret_all_stocks[tickers]
370 | monthly_ret_all_stocks = monthly_ret_all_stocks[tickers]
371 |
372 | data_ret_1 = data_ret
373 | data_ret_1 = data_ret_1[(data_ret_1.index >= '2012-01-01') & (
          data_ret_1.index <= '2018-12-31')]
374 |
375 | i, m = 0, 0
376 | event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
```

```
377  monthly_rets = []
378
379  while event_starting_date < pd.Timestamp(year=2018, month=1,day=1)
         :
380
381      event_ending_date = relativedelta(months=12) +
             event_starting_date
382      portfolio_starting_date = event_ending_date
383      portfolio_ending_date = portfolio_starting_date +
             relativedelta(months=1)
384
385      ''' divide the stocks into 125 portfolios, based on market
             capitalization, price to market ratio, and past year
             return '''
386
387      if m % 12 == 0:
388          q1, cutoff1 = pd.qcut(mc_all_stocks.iloc[i], 5, labels=
                 False, retbins=True)
389          q1 = pd.DataFrame(q1)
390          q1.columns = ['group']
391          cutoff1_1 = cutoff1.tolist()
392
393          ''' first level '''
394          q1_1 = q1.groupby('group').groups
395          q1_1_values = list(q1_1.values())
396
397          q1_2_values = []
398          cutoff1_2 = []
399
400          for t in q1_1_values:
401              t = t.tolist()
402              bv_stocks = bv_all_stocks[t]
403              ''' second level '''
404              q1_2, cutoff_2 = pd.qcut(bv_stocks.iloc[i], 5, labels=
                     False, retbins=True)
405              cutoff1_2.append(cutoff_2.tolist())
406              q1_2 = pd.DataFrame(q1_2)
407              q1_2.columns = ['group']
408              q1_2 = q1_2.groupby('group').groups
409              q1_2_values.append(list(q1_2.values()))
410
411          q1_2_values = sum(q1_2_values, [])
412
413          cutoff1_3 = []
414          q1_3_values = []
415
416          for t in q1_2_values:
417              t = t.tolist()
418              yr_stocks = yearly_ret_all_stocks[t]
419              ''' third level '''
420              q1_3, cutoff_3 = pd.qcut(yr_stocks.iloc[i], 5, labels=
                     False, retbins=True)
421              cutoff1_3.append(cutoff_3.tolist())
422              q1_3 = pd.DataFrame(q1_3)
423              q1_3.columns = ['group']
```

```
424            q1_3 = q1_3.groupby('group').groups
425            q1_3_values.append(list(q1_3.values()))
426
427        q1_3_values = sum(q1_3_values, [])
428        q1_3_values = [q1_3_values[i].tolist() for i in range(125)
               ]
429
430
431     '''compute the equal-weighted monthly return for each of the
           125 portfolios '''
432
433        monthly_returns_all_groups = []
434
435        for t in q1_3_values:
436            df = monthly_ret_all_stocks[t]
437            df = df[(df.index < portfolio_starting_date+pd.offsets
                  .DateOffset(years=1))
438                  & (df.index > portfolio_starting_date)]
439            ret = np.mean(df, axis=1).values.tolist()
440            #print(len(ret))
441            monthly_returns_all_groups.append(ret)
442
443        i += 1
444
445    ''' find the stocks with negative ESG news within the previous
           12 months '''
446
447    df = df_negative[(df_negative.date < event_ending_date)&(
          df_negative.date >= event_starting_date)]
448    tickers = df.ticker.unique().tolist()
449    new_tickers = list(set(tickers) & set(mc_esg_stocks.columns) &
450                  set(bv_esg_stocks.columns) & set(
                        yearly_ret_esg_stocks.columns) & set(
                        data_ret_1.columns))
451
452
453    ''' assign them to each of the 125 portolios based on the
           precomputed cutoff '''
454
455    mc_esg = pd.DataFrame(mc_esg_stocks.iloc[0][new_tickers])
456    mc_esg.columns = ['mc']
457    bv_esg = pd.DataFrame(bv_esg_stocks.iloc[0][new_tickers])
458    bv_esg.columns = ['bv']
459    yr_esg = pd.DataFrame(yearly_ret_esg_stocks.iloc[0][
          new_tickers])
460    yr_esg.columns = ['yr']
461    c_esg = mc_esg.join(bv_esg).join(yr_esg)
462
463    c_esg['mc_group'] = pd.cut(c_esg['mc'], bins=cutoff1_1, labels
          =False, include_lowest=True)
464
465    for i, c in enumerate(cutoff1_2):
466        c_esg.loc[c_esg.mc_group == i,'bv_group'] = \
467        pd.cut(c_esg[c_esg.mc_group == i]['bv'], bins=c, labels=
              False, include_lowest=True)
```

```
468
469        for i in range (5):
470            for j in range (5):
471                c_esg.loc [( c_esg.mc_group == i) & (c_esg.bv_group == j
                       ), 'yr_group'] = \
472                pd.cut(c_esg [( c_esg.mc_group == i) &(c_esg.bv_group ==
                       j)]['yr'], bins = cutoff1_3 [i*5+j],
473                    labels=False, include_lowest=True)
474
475        c_esg['group'] = c_esg['mc_group'] * 25 + c_esg['bv_group'] *
               5 + c_esg['yr_group']
476
477        ''' compute the adjusted return for ESG-related stocks '''
478        ''' compute the equal-weighted monthly return '''
479        rets = data_ret_1 [new_tickers][portfolio_starting_date:
               portfolio_ending_date]
480        for t in rets.columns:
481            group = c_esg.loc [c_esg.index == t, 'group'].values [0]
482            if np.isnan(group):
483                rets.drop(columns=t, inplace=True)
484            else:
485                group_monthly_ret = monthly_returns_all_groups[int(
                       group)]
486                rets[t] -= group_monthly_ret [(m % 12)]
487
488        monthly_ret = np.mean(rets, axis=1).values [0]
489        monthly_rets.append(monthly_ret)
490
491        event_starting_date += relativedelta(months=1)
492        m += 1
493
494 factors ['R'] = monthly_rets
495 factors ['const'] = 1
496 factors.columns = ['Mkt-RF', 'SMB', 'HML', 'RF', 'MKT', 'Mom', 'R
        ','const']
497 mod = sm.OLS(factors['R'], factors [['SMB', 'HML', 'MKT','Mom', '
        const']])
498 res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
499 res.summary()
```

## C.3 Panel Regression

```
1  import pandas as pd
2  import numpy as np
3  from datetime import datetime, timedelta
4  from dateutil.relativedelta import relativedelta
5  import matplotlib.pyplot as plt
6  import warnings
7  import datetime as dt
8  from scipy import stats
9  import pandas_datareader.data as web
10 import statsmodels.api as sm
11 import statsmodels.formula.api as smf
12 from datetime import datetime
```

```
13  from linearmodels import PooledOLS
14
15  warnings.filterwarnings("ignore")
16  pd.set_option('display.max_columns', None)
17
18  ''' preprocess'''
19
20  ''' dividend yield '''
21  div = pd.read_csv('div_yield.csv', header = 1)
22  div = div.iloc[1:,:]
23  div.index = div.ticker
24  div = div.iloc[:,1:]
25
26  ''' log of market capitalization, in terms of billion of dollar
        '''
27  mc = pd.read_csv('market_cap.csv', header = 1)
28  mc = mc.iloc[1:,:]
29  mc.index = mc.ticker
30  mc = mc.iloc[:,1:]
31  mc /= 1000000000
32  mc = np.log(mc)
33
34  ''' log of book-to-market ratio '''
35  bm = pd.read_csv('book_to_market_ratio.csv', header=1)
36  bm = bm.iloc[1:,:]
37  bm.index = bm.ticker
38  bm = bm.iloc[:,1:]
39  bm = np.log(bm)
40
41  ''' log of reciprocal of stock price '''
42  price = pd.read_csv('monthly_price.csv', header=1)
43  price = price.iloc[1:,:]
44  price.index = price.ticker
45  price = price.iloc[:,1:]
46  price = np.log(price)
47
48  prc = 1/price
49  prc = np.log(prc)
50
51  ''' log of cumulative returns '''
52  ret23 = np.log((price.shift(2)/price.shift(3)))
53  ret46 = np.log((price.shift(4)/price.shift(6)))
54  ret712 = np.log((price.shift(7)/price.shift(12)))
55
56  div = div.shift(1)
57  mc = mc.shift(1)
58  bm = bm.shift(1)
59  prc = prc.shift(1)
60
61  df_negative = pd.read_pickle('df_negative_new_1.pkl')
62  df_negative = df_negative[['ticker', 'date', 'label1', 'label2']]
63  df_negative.date = df_negative.date + pd.offsets.MonthEnd(0)
64  df_negative.index = df_negative.ticker
65  df_negative = df_negative[['date', 'label1', 'label2']]
66  df_negative_1 = df_negative
```

```python
67
68  ''' create a dummy , which is the count of negative events for the
        last month '''
69  df_negative = pd.get_dummies(df_negative['date']).groupby(
        df_negative.index).sum()
70  df_negative.columns = [i + relativedelta(months=1) + pd.offsets.
        MonthEnd(0) for i in df_negative.columns]
71  df_negative['tickers'] = df_negative.index
72  df_negative = pd.melt(df_negative,
73          id_vars= 'tickers',
74          value_vars=list(df_negative.columns)[:-1],
75          var_name='date',
76          value_name='dummy')
77  df_negative.date = df_negative.date.apply(lambda x: x.strftime('%Y
        -%m-%d'))
78
79  ''' create a set of dummies , which count the negative events for
        the last month per event subcategory '''
80  labels = df_negative_1.label2.unique().tolist()
81  l = labels[0]
82  df_negative = df_negative_1[df_negative_1.label2 == l]
83  df_negative = df_negative[['date']]
84  df_negative = pd.get_dummies(df_negative['date']).groupby(
        df_negative.index).sum()
85  df_negative.columns = [i + relativedelta(months=1) + pd.offsets.
        MonthEnd(0)  for i in df_negative.columns]
86  df_negative['tickers'] = df_negative.index
87  df_negative = pd.melt(df_negative,
88          id_vars= 'tickers',
89          value_vars=list(df_negative.columns)[:-1],
90          var_name='date',
91          value_name='dummy')
92  df_negative.date = df_negative.date.apply(lambda x: x.strftime('%Y
        -%m-%d'))
93  df_negative.columns = ['tickers', 'date', l]
94
95  for l in labels[1:]:
96      df_new = df_negative_1[df_negative_1.label2 == l]
97      df_new = df_new[['date']]
98      df_new = pd.get_dummies(df_new['date']).groupby(df_new.index).
            sum()
99      df_new.columns = [i + relativedelta(months=1) + pd.offsets.
            MonthEnd(0)  for i in df_new.columns]
100     df_new['tickers'] = df_new.index
101     df_new = pd.melt(df_new,
102         id_vars= 'tickers',
103         value_vars=list(df_new.columns)[:-1],
104         var_name='date',
105         value_name='dummy')
106     df_new.date = df_new.date.apply(lambda x: x.strftime('%Y-%m-%d
            '))
107     df_new.columns = df_new.columns[:-1].tolist() + [l]
108     df_negative = df_new.merge(df_negative, on = ['tickers', 'date
            '], how = 'outer')
109 df_negative = df_negative.fillna(0)
```

```
110
111  mc = mc.iloc[2:,:]
112  mc = mc.iloc[:,(mc.values < np.percentile(mc,99)).all(axis=0) & (
         mc.values > np.percentile(mc,1)).all(axis=0)]
113
114  prc = prc.iloc[2:,:]
115  prc = prc.dropna(axis=1)
116  prc = prc.iloc[:,(prc.values < np.percentile(prc,99)).all(axis=0)
         & (prc.values > np.percentile(prc,1)).all(axis=0)]
117
118  div = div.iloc[2:,:]
119  div = div.dropna(axis=1)
120  div = div.iloc[:,(div.values < np.percentile(div,99)).all(axis=0)
         & (div.values > np.percentile(div,1)).all(axis=0)]
121
122  bm = bm.iloc[2:,:]
123  bm = bm.dropna(axis=1)
124  bm = bm.iloc[:,(bm.values < np.percentile(bm,99)).all(axis=0) & (
         bm.values > np.percentile(bm,1)).all(axis=0)]
125
126  ret23 = ret23.iloc[3:,:]
127  ret23 = ret23.dropna(axis=1)
128  ret23 = ret23.iloc[:,(ret23.values < np.percentile(ret23_1,99)).
         all(axis=0) & (ret23.values > np.percentile(ret23_1,1)).all(
         axis=0)]
129
130  ret46 = ret46.iloc[6:,:]
131  ret46 = ret46.dropna(axis=1)
132  ret46 = ret46.iloc[:,(ret46.values < np.percentile(ret46_1,99)).
         all(axis=0) & (ret46.values > np.percentile(ret46_1,1)).all(
         axis=0)]
133
134  ret712 = ret712.iloc[12:,:]
135  ret712 = ret712.dropna(axis=1)
136  ret712 = ret712.iloc[:,(ret712.values < np.percentile(ret712_1,99)
         ).all(axis=0) & (ret712.values > np.percentile(ret712_1,1)).
         all(axis=0)]
137
138  monthly_return = pd.read_csv('monthly_return.csv',header=1)
139  monthly_return = monthly_return.iloc[1:,:]
140  monthly_return.index = monthly_return.ticker
141
142  monthly_return= monthly_return.drop(columns = ['ticker'])
143  monthly_return = monthly_return.dropna(axis=1)
144  mc_max_cut = np.percentile(monthly_return,99)
145  mc_min_cut = np.percentile(monthly_return,1)
146  monthly_return = monthly_return.iloc[:,(monthly_return.values <
         mc_max_cut).all(axis=0) &
147                              (monthly_return.values > mc_min_cut).
                                     all(axis=0)]
148
149  monthly_return['ticker'] = monthly_return.index
150  monthly_return = pd.melt(monthly_return,
151          id_vars= 'ticker',
152          value_vars=list(monthly_return.columns)[:-1],
```

```
153          var_name='tickers',
154          value_name='return')
155  monthly_return.columns = ['date', 'tickers', 'return']
156  monthly_return['return'] = monthly_return['return']/100
157
158  data = df_negative.merge(mc, how = 'inner', on = ['tickers', 'date
         '])\
159  .merge(prc, how = 'inner', on = ['tickers', 'date'])\
160  .merge(bm, how = 'inner', on = ['tickers', 'date']).merge(div, how
          = 'inner', on = ['tickers', 'date'])\
161  .merge(ret23, how = 'inner', on = ['tickers', 'date']).merge(ret46
         , how = 'inner', on = ['tickers', 'date'])\
162  .merge(ret712, how = 'inner', on = ['tickers', 'date']).merge(
         monthly_return, how='inner', on=['tickers','date'])
163  data.columns = df_negative.columns.tolist() + ['mc', 'prc', 'bm',
         'div', 'ret23', 'ret46','ret712', 'ret']
164
165  data = data.replace([np.inf, -np.inf], np.nan)
166  data1 = data.dropna()
167
168  #data1['dummy'] = data1['dummy'].clip(0, 1)
169
170  '''Fama-Macbeth regression'''
171  def ols_coef(x,formula):
172      return smf.ols(formula,data=x).fit().params
173
174  def fm_summary(p):
175      s = p.describe().T
176      s['std_error'] = s['std']/np.sqrt(s['count'])
177      s['tstat'] = s['mean']/s['std_error']
178      s['pval'] = stats.t.sf(np.abs(s['tstat']), s['count']-1)*2
179      return s[['mean','std_error','tstat', 'pval']]
180
181  gamma = data1.groupby('date').apply(ols_coef,'ret ~ 1 + dummy + mc
          + prc + bm + div + ret23 + ret46 + ret712')
182  res = fm_summary(gamma)
183  res.pval = [round(x,3) for x in res.pval.values.tolist()]
184
185  gamma = data1.groupby('date').apply(ols_coef,'ret ~ 1 +Association
         +Sanctions+Financial+Corruption+Information+Human+Workplace+
         Production_Supply+Environmental+Management+Workforce+
         Regulatory+Fraud+Anti_Competitive+Ownership+Product_Service+
         Discrimination_Workforce+mc + prc + bm + div + ret23 + ret46 +
          ret712')
186  res = fm_summary(gamma)
187  res.pval = [round(x,3) for x in res.pval.values.tolist()]
188  res
189
190  '''pooled OLS with double-clustered standard error'''
191  data1.date = data1.date.apply(lambda x: datetime.strptime(x, '%Y-%
         m-%d'))
192  data1 = data1.set_index(['tickers','date'])
193  data1['const'] = 1
194  mod = PooledOLS(data1['ret'], data1[['dummy','mc','prc','bm','div
         ','ret23','ret46','ret712', 'const']])
```

```
195  res = mod.fit(cov_type='clustered', cluster_entity=True,
         cluster_time = True)
```

## C.4   Relative Screening

```
1   import pandas as pd
2   import numpy as np
3   import random
4   from dateutil.relativedelta import relativedelta
5   from datetime import date, datetime
6   import matplotlib.pyplot as plt
7   from scipy import stats
8   import statsmodels.api as sm
9
10  '''read data'''
11  df_negative = pd.read_pickle('df_negative_new_1.pkl')
12  df_negative = df_negative[['ticker', 'date','label2']]
13  df_negative['news_count'] = df_negative.groupby('ticker')['date'].
        rank(ascending=True)
14  df_negative = df_negative.sort_values(['ticker', 'date'])
15
16
17  data_ret = pd.read_csv('monthly_return.csv', header = 1)
18  data_ret = data_ret.iloc[1:,]
19  data_ret.ticker = data_ret.ticker.apply(lambda x: datetime.
        strptime(x, '%Y-%m-%d'))
20  data_ret.index = data_ret['ticker']
21  data_ret = data_ret.iloc[:,1:]
22  data_ret /= 100
23  data_ret = data_ret.dropna(axis='columns')
24
25  stock_tickers = list(set(data_ret.columns) & set(df_negative.
        ticker.unique().tolist()))
26  df_negative = df_negative[df_negative.ticker.isin(stock_tickers)]
27  data_ret = data_ret[stock_tickers]
28
29  monthly_ret_all = pd.read_csv('monthly_ret_all_stocks_1.csv')
30  monthly_ret_all.index = monthly_ret_all['date']
31  monthly_ret_all = monthly_ret_all.iloc[:,1:]
32  monthly_ret_all /= 100
33  monthly_rets_all = monthly_ret_all.mean(axis=1) .values.tolist()
34
35  monthly_ret_sp500 = pd.read_csv('monthly_return_sp500.csv', header
        =1)
36  monthly_ret_sp500 = monthly_ret_sp500.iloc[1:,:]
37  monthly_ret_sp500.ticker = monthly_ret_sp500.ticker.apply(lambda x
        : datetime.strptime(x, '%Y-%m-%d'))
38  monthly_ret_sp500.index = monthly_ret_sp500.ticker
39  monthly_ret_sp500 = monthly_ret_sp500.iloc[:,1:]
40  monthly_ret_sp500/=100
41
42  all_monthly_rets = monthly_ret_all.mean(axis=1).values.tolist()
43  sp500_monthly_rets = monthly_ret_sp500.mean(axis=1).values.tolist
        ()
```

```
44
45  event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
46  event_ending_date = relativedelta(months=1) + event_starting_date
47  portfolio_starting_date = event_ending_date
48  portfolio_ending_date = portfolio_starting_date + relativedelta(
        months=1)
49  df = df_negative[(df_negative.date < event_ending_date)&(
        df_negative.date >= event_starting_date)]
50
51  '''value-weighted portfolio returns for worst 30 stocks'''
52
53  event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
54  event_ending_date = pd.Timestamp(year=2012, month=4, day=1)
55  df = df_negative[(df_negative.date < event_ending_date)&(
        df_negative.date >= event_starting_date)]
56  tickers = ['a' for i in range(30)]
57  monthly_rets = []
58  turn_overs = []
59
60  while event_ending_date < pd.Timestamp(year=2019, month=1,day=1):
61
62      portfolio_starting_date = event_ending_date
63      portfolio_ending_date = portfolio_starting_date +
            relativedelta(months=1)
64
65      df = df_negative[(df_negative.date < event_ending_date)]
66      news_count = pd.DataFrame(df.groupby('ticker')['news_count'].
            max())
67      prev_tickers = tickers
68      tickers = news_count.sort_values('news_count', ascending=False
            )[:30].index.tolist()
69      turnover_rate = 1-len(list(set(prev_tickers) & set(tickers)))
            /30
70      turn_overs.append(turnover_rate)
71      rets = data_ret[tickers][portfolio_starting_date:
            portfolio_ending_date].values[0]
72      mc = mc_all[tickers][portfolio_starting_date:
            portfolio_ending_date]
73      mc_sum = np.sum(mc,axis=1).values[0]
74      monthly_ret = np.sum((rets * mc)/ mc_sum ,axis=1).values[0]
75      monthly_rets.append(monthly_ret)
76      event_ending_date += relativedelta(months=1)
77
78  monthly_rets_worst = monthly_rets
79
80  ''' value-weighted portfolio returns for best 30 stocks '''
81  event_starting_date = pd.Timestamp(year=2012, month=1, day=1)
82  event_ending_date = pd.Timestamp(year=2012, month=4, day=1)
83  df = df_negative[(df_negative.date < event_ending_date)&(
        df_negative.date >= event_starting_date)]
84  tickers = ['a' for i in range(30)]
85  monthly_rets = []
86  turn_overs = []
87
88  while event_ending_date < pd.Timestamp(year=2019, month=1,day=1):
```

```
89
90       portfolio_starting_date = event_ending_date
91       portfolio_ending_date = portfolio_starting_date +
            relativedelta(months=1)
92
93       df = df_negative[(df_negative.date < event_ending_date)]
94       news_count = pd.DataFrame(df.groupby('ticker')['news_count'].
            max())
95       prev_tickers = tickers
96       tickers = news_count.sort_values('news_count', ascending=True)
            [:30].index.tolist()
97       turnover_rate = 1-len(list(set(prev_tickers) & set(tickers)))
            /30
98       turn_overs.append(turnover_rate)
99       rets = data_ret[tickers][portfolio_starting_date:
            portfolio_ending_date].values[0]
100      mc = mc_all[tickers][portfolio_starting_date:
            portfolio_ending_date]
101      mc_sum = np.sum(mc,axis=1).values[0]
102      monthly_ret = np.sum((rets * mc)/ mc_sum ,axis=1).values[0]
103      monthly_rets.append(monthly_ret)
104      event_ending_date += relativedelta(months=1)
105
106  '''value-weighted benchmark returns'''
107  mc_sp500 = pd.read_csv('mc_sp500.csv',header = 1)
108  mc_sp500 = mc_sp500.iloc[1:,:]
109  mc_sp500.ticker = mc_sp500.ticker.apply(lambda x: datetime.
        strptime(x, '%Y-%m-%d'))
110  mc_sp500.index = mc_sp500.ticker
111  mc_sp500 = mc_sp500.iloc[:,1:]
112  mc_all = pd.read_csv('monthly_market_cap_all_stocks_1.csv')
113  mc_all.date = mc_all.date.apply(lambda x: datetime.strptime(x, '%Y
        -%m-%d'))
114  mc_all.index = mc_all.date
115  mc_all = mc_all.iloc[:,1:]
116  all_tickers =  mc_all.columns
117  all_tickers = [t.split('.', 1)[0].lower() for t in all_tickers]
118  mc_all.columns = all_tickers
119  all_tickers =  monthly_ret_all.columns
120  all_tickers = [t.split('.', 1)[0].lower() for t in all_tickers]
121  monthly_ret_all.columns = all_tickers
122  stocks_sp500 = list(set(mc_sp500.columns) & set(monthly_ret_sp500.
        columns))
123  mc_sp500 = mc_sp500[stocks_sp500]
124  monthly_ret_sp500 = monthly_ret_sp500[stocks_sp500]
125  stocks_all = list(set(mc_all.columns) & set(monthly_ret_all.
        columns))
126  mc_all = mc_all[stocks_all]
127  monthly_ret_all = monthly_ret_all[stocks_all]
128  sp500_weighted_returns = (monthly_ret_sp500 * mc_sp500).div(
        mc_sp500.sum(axis=1),axis=0).sum(axis=1).values.tolist()
129  all_weighted_return = (monthly_ret_all * mc_all).div(mc_all.sum(
        axis=1),axis=0).sum(axis=1).values.tolist()
130  tickers = list(set(df_negative.ticker.unique()) & set(mc_all.
        columns))
```

```
131 df_negative = df_negative[df_negative.ticker.isin(tickers)]
132
133 '''compare portfolio returns with the benchmark'''
134 cumsum_all_stocks = np.cumsum(all_monthly_rets[5:-1])
135 cumsum_sp500 = np.cumsum(sp500_monthly_rets[5:-1])
136 cumsum_worst = np.cumsum(monthly_rets_worst)
137 cumsum_best = np.cumsum(monthly_rets_best)
138 date = pd.date_range(start='2012-04-01', end='2018-12-31', freq='
       MS')
139 data = pd.DataFrame(data = {'date': date,'sp500': cumsum_sp500, '
       all public': cumsum_all_stocks,
140                    'best 30': cumsum_best, 'worst 30':
                          cumsum_worst})
141
142 plt.plot('date', 'sp500', data = data)
143 plt.plot('date', 'all public', data = data)
144 plt.plot('date', 'best 30', data = data)
145 plt.plot('date', 'worst 30', data = data)
146 plt.xlabel('year')
147 plt.ylabel('cumulative return')
148 plt.legend()
149
150 factors = pd.read_pickle('factors_monthly.pkl')
151 factors = factors[(factors.index >= '2012-4-1') & (factors.index <
        '2019-1-1')]
152
153 ''' Carhart four-factor model '''
154 factors['R'] = monthly_rets
155 factors['RF'] = all_monthly_rets[5:-1]
156 factors['R-RF'] = factors['R'] -factors['RF']
157 factors['const'] = 1
158 factors.columns = ['Mkt-RF', 'SMB', 'HML', 'RF', 'MKT', 'MOM', 'R
       ', 'R-RF', 'const']
159 mod = sm.OLS(factors['R-RF'], factors[['MKT', 'HML', 'SMB', 'MOM
       ','const']])
160 res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
161 res.summary()
```

## C.5  Negative Screening

```
1 import pandas as pd
2 import numpy as np
3 import random
4 from dateutil.relativedelta import relativedelta
5 from datetime import date, datetime
6 import matplotlib.pyplot as plt
7 from scipy import stats
8 import statsmodels.api as sm
9
10 ''' read  data '''
11 df_negative = pd.read_pickle('df_negative_new_1.pkl')
12 df_negative = df_negative[['ticker', 'date','label2','label1']]
13 df_negative['news_count'] = df_negative.groupby('ticker')['date'].
       rank(ascending=True)
```

```
14  df_negative = df_negative.sort_values(['ticker', 'date'])
15
16  monthly_ret_all = pd.read_csv('monthly_ret_all_stocks_1.csv')
17  monthly_ret_all.date = monthly_ret_all.date.apply(lambda x:
        datetime.strptime(x, '%Y-%m-%d'))
18  monthly_ret_all.index = monthly_ret_all['date']
19  monthly_ret_all = monthly_ret_all.iloc[:,1:]
20  monthly_ret_all /= 100
21
22  all_tickers =  monthly_ret_all.columns
23  all_tickers = [t.split('.', 1)[0].lower() for t in all_tickers]
24  monthly_ret_all.columns = all_tickers
25
26  monthly_ret_sp500 = pd.read_csv('monthly_return_sp500.csv', header
        =1)
27  monthly_ret_sp500 = monthly_ret_sp500.iloc[1:,:]
28  monthly_ret_sp500.ticker = monthly_ret_sp500.ticker.apply(lambda x
        : datetime.strptime(x, '%Y-%m-%d'))
29  monthly_ret_sp500.index = monthly_ret_sp500.ticker
30  monthly_ret_sp500 = monthly_ret_sp500.iloc[:,1:]
31  monthly_ret_sp500/=100
32
33  all_monthly_rets = monthly_ret_all.mean(axis=1).values.tolist()
34  sp500_monthly_rets = monthly_ret_sp500.mean(axis=1).values.tolist
        ()
35
36  mc_sp500 = pd.read_csv('mc_sp500.csv',header = 1)
37  mc_sp500 = mc_sp500.iloc[1:,:]
38  mc_sp500.ticker = mc_sp500.ticker.apply(lambda x: datetime.
        strptime(x, '%Y-%m-%d'))
39  mc_sp500.index = mc_sp500.ticker
40  mc_sp500 = mc_sp500.iloc[:,1:]
41  mc_all = pd.read_csv('monthly_market_cap_all_stocks_1.csv')
42  mc_all.date = mc_all.date.apply(lambda x: datetime.strptime(x, '%Y
        -%m-%d'))
43  mc_all.index = mc_all.date
44  mc_all = mc_all.iloc[:,1:]
45  all_tickers =  mc_all.columns
46  all_tickers = [t.split('.', 1)[0].lower() for t in all_tickers]
47  mc_all.columns = all_tickers
48  stocks_sp500 = list(set(mc_sp500.columns) & set(monthly_ret_sp500.
        columns))
49  mc_sp500 = mc_sp500[stocks_sp500]
50  monthly_ret_sp500 = monthly_ret_sp500[stocks_sp500]
51
52  stocks_all = list(set(mc_all.columns) & set(monthly_ret_all.
        columns))
53  mc_all = mc_all[stocks_all]
54  monthly_ret_all = monthly_ret_all[stocks_all]
55  sp500_weighted_returns = (monthly_ret_sp500 * mc_sp500).div(
        mc_sp500.sum(axis=1),axis=0).sum(axis=1).values.tolist()
56  all_weighted_return = (monthly_ret_all * mc_all).div(mc_all.sum(
        axis=1),axis=0).sum(axis=1).values.tolist()
57
58
```

```
59  '''value-weighted negatively screening portfolios'''
60
61  '''perform negative screening on all public traded US stocks'''
62  threshold = 0
63  event_ending_date = pd.Timestamp(year=2012, month=2, day=1)
64  monthly_rets_all_public = []
65  num_stocks_list = []
66  stock_tickers = monthly_ret_all.columns.tolist()
67
68  while event_ending_date < pd.Timestamp(year=2019, month=1,day=1):
69
70      portfolio_starting_date = event_ending_date
71      portfolio_ending_date = portfolio_starting_date +
            relativedelta(months=1)
72      df = df_negative[(df_negative.date < event_ending_date)]
73      count = pd.DataFrame(df.groupby('ticker')['news_count'].max())
74      negative_tickers = count[count.news_count > threshold].index.
            tolist()
75      tickers = [x for x in stock_tickers if x not in
            negative_tickers]
76      num_stocks = len(tickers)
77      num_stocks_list.append(num_stocks)
78      rets = monthly_ret_all[tickers][portfolio_starting_date:
            portfolio_ending_date]
79      mc = mc_all[tickers][portfolio_starting_date:
            portfolio_ending_date]
80      mc_sum = np.sum(mc,axis=1).values[0]
81      monthly_ret = np.sum((rets * mc)/ mc_sum ,axis=1).values[0]
82      monthly_rets_all_public.append(monthly_ret)
83      event_ending_date += relativedelta(months=1)
84
85  factors['R'] = monthly_rets_all_public
86  factors['RF'] = all_weighted_return[3:-1]
87  factors['R-RF'] = factors['R'] - factors['RF']
88  mod = sm.OLS(factors['R-RF'], factors[['SMB', 'HML', 'MKT', 'MOM',
        'const']])
89  res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
90  res.summary()
91
92  '''perform negative screening on SP500 stocks'''
93
94  threshold = 0
95  event_ending_date = pd.Timestamp(year=2012, month=2, day=1)
96  monthly_rets_sp500 = []
97  num_stocks_list = []
98  stock_tickers = monthly_ret_sp500.columns.tolist()
99
100 while event_ending_date < pd.Timestamp(year=2019, month=1,day=1):
101     portfolio_starting_date = event_ending_date
102     portfolio_ending_date = portfolio_starting_date +
            relativedelta(months=1)
103     df = df_negative[(df_negative.date < event_ending_date)]
104     count = pd.DataFrame(df.groupby('ticker')['news_count'].max())
105     negative_tickers = count[count.news_count > threshold].index.
            tolist()
```

```
106        tickers = [x for x in stock_tickers if x not in
              negative_tickers]
107        num_stocks = len(tickers)
108        num_stocks_list.append(num_stocks)
109        rets = monthly_ret_sp500[tickers][portfolio_starting_date:
              portfolio_ending_date]
110        mc = mc_sp500[tickers][portfolio_starting_date:
              portfolio_ending_date]
111        mc_sum = np.sum(mc,axis=1).values[0]
112        monthly_ret = np.sum((rets * mc)/ mc_sum ,axis=1).values[0]
113        monthly_rets_sp500.append(monthly_ret)
114        event_ending_date += relativedelta(months=1)
115
116  factors['R'] = monthly_rets_sp500
117  factors['RF'] = sp500_weighted_returns[3:-1]
118  factors['R-RF'] = factors['R'] - factors['RF']
119  mod = sm.OLS(factors['R-RF'], factors[['SMB', 'HML', 'MKT', 'MOM',
        'const']])
120  res = mod.fit(cov_type='HAC', cov_kwds={'maxlags': 12})
121  res.summary()
122
123  '''plot cumulative returns'''
124
125  date = pd.date_range(start='2012-02-01', end='2018-12-31', freq='
        MS')
126  cumsum_all_stocks = np.cumsum(all_weighted_return[3:-1])
127  cumsum_sp500 = np.cumsum(sp500_weighted_returns[3:-1])
128  cumsum_all_screened = np.cumsum(monthly_rets_all_public)
129  cumsum_sp500_screened = np.cumsum(monthly_rets_sp500)
130  data = pd.DataFrame(data = {'date': date,'sp500': cumsum_sp500, '
        all public': cumsum_all_stocks ,
131                        'all public negative screened':
                              cumsum_all_screened ,
132                        'sp500 negative screened':
                              cumsum_sp500_screened})
133  plt.plot('date', 'sp500', data = data)
134  plt.plot('date', 'sp500 negative screened', data = data)
135  plt.xlabel('year')
136  plt.ylabel('cumulative return')
137  plt.legend()
```

# Declaration of Originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

---

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor .

**Title of work** (in block letters):

> AN IN-DEPTH ANALYSIS OF THE IMPACT OF ESG
> INVESTING ON RETURNS USING LARGE-SCALE NEWS DATA .

**Authored by** (in block letters):
*For papers written by groups the names of all authors are required.*

| Name(s): | First name(s): |
|---|---|
| TIANQI | WANG |
| | |
| | |
| | |
| | |

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the Citation etiquette information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work .
- I am aware that the work may be screened electronically for plagiarism.
- I have understood and followed the guidelines in the document *Scientific Works in Mathematics*.

**Place, date:**

ZÜRICH, 15.08.2019

**Signature(s):**

*Tianqi Wang.*

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*