universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Activist Short-Sellers: An Analysis of a New Generation of Activist Investors“

verfasst von / submitted by

Karel Sarapatka

Angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2020 / Vienna 2020

Studienkennzahl lt. Studienblatt / Degree programme code as it appears on the student record sheet	A 066 974
Studienrichtung lt. Studienblatt / Degree programme as it appears on the student record sheet	Masterstudium Banking and Finance
Betreut von / Supervisor:	Univ.-Prof. Dr. Gyöngyi Lóránth
Mitbetreut von / Co-Supervisor:	

Abstract

A new discipline in short-selling has become increasingly prominent in the last two decades. Unlike passive short-sellers, the new activist short-sellers voluntarily disclose their short position and engage with the public to reshape how the market should think about targeted companies. Using a sample of 239 activist initiated campaigns in the period of 2011 to 2019, I document significantly negative abnormal returns of -4.7% on the day of activist coverage announcement. Negative abnormal returns for targeted companies extend to a period of one year following the announcement, with mean French Fama 3-factor adjusted cumulative returns of -76.8%. I find that target firm characteristics play an important role in the magnitude of negative abnormal returns. Negative returns are more pronounced in the long term for companies manifesting overvaluation, whereas companies shrouded in ambiguity regarding their financials experience comparatively larger negative returns in the short term. In addition to negative abnormal returns, I find that short-seller activists are more likely to target firms exhibiting either of the two characteristics. Determinants that proved to be especially salient in attracting activists were Debt Coverage and Overinvestment of Free Cash Flows in the overvaluation set and Manipulation Score and Bid-Ask Spread in the ambiguity set. My findings suggest a non-negligible impact of activist short-sellers on targeted companies' returns and should be thus of interest for both sophisticated and laymen equity investors.

Kurzfassung

In der letzten zwei Jahrzenten hat eine neue Disziplin der aktivistisch geprägten Leerverkäufe an Popularität gewonnen. Im Gegensatz zu passiven Leerverkäufern, legen die aktivistischen Leerverkäufer ihre Position und Forschung freiwillig offen und treten mit der Öffentlichkeit in Kontakt mit dem Ziel die Wahrnehmung der Firma neu zu formen. Anhand einer Analyse von 239 aktivistischer Angriffe in der Periode von 2011 bis 2019, entdeckte ich, dass die abnormalen Renditen am Tag der Veröffentlichung einer Position signifikant negativ sind und im Durchschnitt -4.7% betragen. Die negativen Erträge erstrecken sich über die ganze beobachtete Periode von einem Jahr, mit einer kumulativen French-Fama 3 Faktor-adjustierten Rendite von -76.8% nach einem Jahr. Überdies wurde im Laufe der Analyse bestätigt, dass Firmencharakteristika eine wichtige Rolle für Höhe der nachfolgenden negativen Renditen spielen. In einem langfristigen Horizont sind die negativen Renditen deutlich mehr geprägt für Firmen die Überbewertung aufweisen, während die Firmen die reich an Unklarheit bezüglich der zugrunde liegenden Finanzdaten sind, vor allem kurzfristig betroffen sind. Darüber hinaus finde ich, dass Firmen, die entweder als überbewertet erscheinen oder Unklarheiten aufweisen, wahrscheinlicher von Aktivisten untersucht werden. Die besonders wichtigen Determinanten einer solchen Untersuchung sind Debt Coverage, Overinvestment of Free Cash Flows, M-score und Bid-Ask Spread. Meine Erkenntnisse deuten auf unvernachlässigbare Renditeeffekte der Aktivisten und sollten daher für die versierten als auch für die unprofessionellen Investoren von Interesse sein.

Acknowledgements

I would hereby like to express my undying gratitude to all of those who provided moral and financial support in the process of completing this thesis.

I am grateful to my parents, who supported me in my endeavors throughout all my study years.

I would also like to thank all my friends, without whom the final version of my thesis would see the light of day much earlier.

Finally, I am indebted to all programmers, who tirelessly fill the remaining gaps in countless programming problems with no expectation of material gains.

Table of Contents

List of Tables in Text

List of Formulas

Introduction

Over the past decade, the institutional asset management industry has seen lackluster returns relative to virtually all major stock indices[1]. As a result, equity investment-based hedge funds, as the main victim, have experienced a long streak of capital withdrawals (as documented by Preqin, in August, 2019[2]). In an environment of easy money-fueled buoyant markets, traditional investment approaches based on fundamentals, as envisaged by Benjamin Graham, appear to be slowly losing their merit. Consequently, active fund managers have found it increasingly more difficult to justify their management and incentive fees, especially in face of success of simple market following passive investment strategies. In these grim times for the hedge fund industry, a new generation of investment managers emerged to deliver the ever-ephemeral alpha to their investors.

These investment managers, and more often non-institutional, "lone-wolf" investors with professional finance background, engage in public campaigns against companies whose shares they deem to be grossly overvalued or which they assert to demonstrate discrepancy between accounting and true economic results. The new breed of activist investors maintains their presence on social media, present their cases and ideas at investment conferences, and appear at popular finance-oriented broadcasters, such as CNBC or Bloomberg. This allows them to capture the attention of wide audience in order to disseminate their investment theses or to substantiate their claims presented therein. Several short-sellers have attained almost celebrity-like status due to their long streak of devastating exposures. As a result, these investors wield considerable power, since their public statements may cause a stampede among retail investors or attract attention of professional investment community and scrutiny of market regulators. A case in point is the announcement of Muddy Waters Research's short position and publication of accompanying investment thesis against Burford Capital, a UK based litigation financing firm. In the course of two business days, Burford lost close to $2 Billion in market capitalization, causing its CFO to be terminated and shareholder class action lawsuit to be filed against it. The outsized influence over stock returns and company's corporate

[1] As seen in Table 1.
[2] https://docs.preqin.com/reports/Preqin-Hedge-Fund-Asset-Flows-Q3-2019.pdf

governance concentrated in the hands of activist short sellers, such as Muddy Waters, merits closer inspection, which I set as a prime objective of this thesis.

Given the novelty of this approach to short-selling and absence of unified database covering activist short-seller campaigns, the literature dealing with the topic remains relatively scarce. Several authors, most notably Wuyang Zhao, Antonis Kartapanis and Alexander Ljungsqvist along with Wenlan Qian have shed some light on the subject, but the topic has arguably a long way ahead before it reaches the notoriety of passive short-selling and traditional shareholder activism.

In the following chapters, I seek to test some of the research results of the above-mentioned authors concerning firm characteristics that may attract activist short sellers. I expand on their work by means of introducing new parameters and adding my own observations to conduct a study of activist short seller campaigns in the period between 2011 and 2019. For this purpose, I have collected over 8000 firm quarters of financial data on 239 US-based companies which were targeted at some point by an activist short seller. In order to build a sample of non-targeted companies allowing for comparison and statistical inferences, close to 60.000 firm-quarters of financial data were collected and evaluated.

Unlike the authors dealing with the topic before me, I have created my own program to extract financial data and stock prices, and developed a simple sorting and statistical algorithm to order and analyze the data set. Given the large variability of research quality and differing background of activist short sellers, I focused exclusively on the most prolific activists with the highest social media following and a demonstrable history of successful investor activism. One of notable implications of this approach is that the identity of short-seller becomes either public knowledge or that it could be uncovered without much effort. Therefore, I have partly marginalized the need to study impact of activist personal recognition among market participants. In so doing, I have concurrently selected for activist who self-moderate the severity of their claims as they could be more easily sued for libel, spreading defamatory information and market manipulation, or could be otherwise personally intimidated.

Since the modus operandi of activist short sellers differs largely from that employed by passive short sellers, Chapter 1 is dedicated to elaborating on activist strategies and how they have evolved since their earlier forms. On the backdrop of the differences between active and passive short selling, I will introduce major players in the institutional short-seller universe and provide an overview of the most notable campaigns carried out in the last decade. Since the often-acrimonious battles between short-sellers and targeted companies constitute a large portion of the constraints to activist short-selling (Lamont, 2012), I shall examine several notable cases from recent history, to illustrate ways in which short-selling may be impeded.

Due to public nature of activist campaigns, the set of risks pertaining to legal and reputational repercussions goes well beyond financial risk of traditional short sellers. While passive short sellers hold onto their research, the information acquired by activists via costly independent investigation is disseminated freely to all market participants. In this manner, the activist can persuade existing shareholders to sell or reduce their stake, rather than quietly wait until the information gets incorporated into firm's share price (Ljungsqvist and Qian, 2016). Furthermore, the activist's research thesis reaches far beyond the principal audience as the information trickles down to traders and other short sellers. Gradual propagation of their findings fuels further price discovery or, if the research is built on murkier grounds, causes the lesser informed market participants to herd once the price starts declining (Tingyu and Lai, 2009). The last part of the first chapter will therefore attempt to shed light on how short-seller activists acquire such impactful information and the techniques they employ to raise awareness among market participants.

The second part of the thesis focuses on firm characteristics that may attract short sellers in the first place. I will examine what set of features might determine the likelihood of activist short-seller involvement and the impact of these determinants on abnormal stock returns of targeted companies both in the short term and long term. I divide the determinants into two groups: valuation-based determinants and ambiguity-based determinants, with the latter comprising features of uncertainty, indicators of poor internal controls and elements of linguistic analysis of annual filings. Prior research (Zhao, 2019; Zhao 2018) has shown that firms exhibiting severe overvaluation and uncertainty regarding their financial position tend to be targeted with higher probability

than their peers. I expand on this research by introducing a parameter akin to textual analysis of the target's financial statements for occurrence of key uncertainty words following Loughran and McDonald's (2011) financial sentiment word list. I introduce a new criterion omitted by previous studies, the measure of inorganic growth of the target company (proxied by overinvestment of free cash flows). Moreover, I combine the above conjectured determinants with features of corporate opacity as part of the ambiguity parameter.

To construct the sample of activists and gather all pertinent financial data of targeted and non-targeted companies, I wrote a series of separate programs in python. These allow collectively for scraping and parsing of SEC Edgar for financial statements and exhibits of interest, extraction of data from third-party API (application programming interface) and analysis of all ordered data.

The sample of activist is by no means exhaustive, as I excluded short-sellers who failed to amass sufficient following, operate solely in pseudo-anonymity or fail to take responsibility for their research. Likewise, those who target exclusively non-US based companies, do not publish their research theses or deliver research theses of questionable quality were also excluded. Consequently, the selection and study presented in this thesis pertains to short seller activists who could be said to have attained a celebrity status in the short seller community.

1 Anatomy of an Activist Short Seller

"Short sellers are the only real time market detectives out there […], the regulators and law enforcement are like financial archeologists, they will tell you ten years after the fact why you lost money. Short sellers are incentivized to actually ferret out the problems in real time. "

- *James Chanos, founder and president of Kynikos Associates Ltd, In interview with Dan Gilbert, December 13, 2017.*

1.1 Passive Short Selling

Passive short-sellers are investors who hold a contrarian view regarding security's future prospects. Unlike long equity investors, passive short-sellers stand to profit from security price decline and lose in case of price increase. To sell a stock short, investors seek out a lender institution, such a brokerage house or a mutual fund, that is willing to lend the shares in the desired volume. In return, the lender negotiates a daily or monthly interest payment (or a rebate payment) for borrowing shares along with, in some cases, a level of liquid asset collateral to be deposited at a broker institution. Once the position is initiated, borrowed shares may return to the lender in three different manners: short-seller decides to cover his position and return shares to the custodian; the lender recalls borrowed shares (if the contract allows for it), or, in the case of adverse development for the short-seller whose collateral no longer covers the agreed-upon maintenance requirement, the position is liquidated and shares are returned to the lender.

1.2 Related Literature on Passive Short Selling

Prior research has established passive short-sellers to be highly skilled market participants who, for example, manifest ability to detect severe accounting irregularities often well in advance of later costly financial statement corrections (Efendi and Swanson, 2009). On a similar note, Karpoff and Lou (2010) demonstrated, that short-sellers are adept at discovering accounting misrepresentation before public revelation, as well as gauge the severity of such misrepresentation, suggesting either exceptional utilization of publicly available information or availability of superior private information channels. Short-sellers were shown to actively and effectively utilize fundamental analysis to determine whether company shares are overvalued and discern the sources of overvaluation to select for companies with lower future returns (Dechow et.al, 2001). Therefore, short-sellers are commonly considered 'smart money' in the equity markets,

with short interest, despite being a publicly observable parameter, functioning as a predictor of lower future returns (as noted by Engelberg, Reed and Ringgenberg, 2015, and further developed by Rapach, Ringgenberg and Zhou, 2016).

Short-sellers have historically held a controversial position in the investment community, being widely regarded by their opponents as vultures, who shamelessly profit from misfortune of others, and blamed for exacerbating market declines, price volatility, price destabilization or distorting managerial decision making (not entirely without merit, as e.g. in Shkilko et. al, 2008, and Shi, Connelly and Cirik, 2018). Short-sellers' ill repute has often made them target of restrictions, as they stood first in line to be curbed once the markets encountered a period of increased volatility, such as in the period preceding World War 1 (as documented in SEC Docket: Volume 11, 1976) or as recently as in 2008, when the SEC introduced a wide ban of short-selling on almost 1000 financial stocks. In some cases, market regulators have gone so far as to propose corporal punishment for anyone involved in short selling (Lamont, 2012, p.5). Fortunately, researchers have come to recognize the value of short-sellers in the proper functioning of markets and facilitating price discovery and no western short-seller has been subjected to caning.

Contrary to unabating negative views about short-sellers held by the general public, a large body of research has documented adverse impact of short-sale constraints on market efficiency. Edward M. Miller (1977) argued in his paper, that short selling constraints (as well as widespread short-selling unwillingness) can induce persistence of speculative excess, since only the views of optimistic investors will be reflected in stock prices. Gross mispricing in the presence of constrained short selling was also evidenced, although in limited scope, by Lamont and Thaler (2003) in their examination of technology carve-outs. They observed how insufficient availability of lendable shares prevented arbitrageurs from taking advantage of overpriced securities and thus effectively hindered the shares from reaching more rational levels.

In addition to overvaluation of individual stocks (Hong and Stein, 2003; Grullon et al. 2014), restrictions on short-selling imposed by regulators and governmental actors were shown to cause number of adverse ramifications, such as increase in volatility and reduction of liquidity (Charoenrook and Daouk, 2005). Prohibition of short-selling leads not only to reduction in activity in the markets, but to demonstrable deterioration in

market quality – widening of quoted and effective spreads and decline in informativeness of prices, as was observed in 2008 (Boehmer, Jones, Zhang, 2008).

Contribution of short-sellers can be thus summarized through two main roles they assume. Firstly, as facilitators of information efficiency who keep market prices from diverging too far from their underlying fundamentals and reduce delay in incorporation of new negative information (Boehmer and Wu, 2013). Secondly, mere presence of short sellers (unconstrained operations) was shown to discipline managers to curb the use of discretionary accruals (Fang, Huang and Karpoff, 2016, Massa, Zhang, Zhang, 2015). Short-seller's role can hence be said to have a character of an external corporate governance body, overseeing managers who might attempt to deliberately obscure or otherwise embellish their financial results.

1.3 Activist short-selling: Origins

The history of short-seller activism reaches as far as the 17th century, when the first recorded short sale of equity shares took place as Isaac Le Maire sought revenge against Dutch East India Company. Feeling wronged for dismissal from the company, he sold short its shares and started spreading rumors to drive the stock price down (David Kestenbaum, 2015). Unethical by then standards and certainly illegal today, short-seller activism has transformed over the centuries into sophisticated investing discipline borrowing from fields of forensic accounting and financial investigation.

The specific activism and its nature examined in this thesis hasn't seen the light of day until the turn of the 21st century. Although some may dispute it, the ascent of short seller activism in today's form can be attributed to Jim Chanos, closely followed by Carl Icahn[3] and their actions against now infamous Enron Corporation and Conseco Inc. in the early 2000s'. Public disclosures of short positions and continuous engagement with the public over their targets can be seen as precursor of today's comprehensive research theses. The true rise of the new breed of activist short-sellers, however, came with the wave of reverse merger-listings (henceforth RM) of Chinese companies in the United States.

[3] Both Chanos and Icahn were preceded by Manuel P. Asensio, who published what can be considered the first activist short-seller report on Diana Corp.in 1996. Nevertheless, the amount of attention his report received hardly makes him the initial spark that motivated later day activists.

During the period of 2001 to 2010, 448 China-domiciled companies entered the U.S. equity market via reverse merger (Chen et al., 2016). As a consequence of American retail investors' enthusiasm for Chinese growth stories and relaxed scrutiny of companies not entering the market via an IPO, stock promoters sensed an opportunity to make easy profit and increased their presence in the market for RM shares (McKenna, 2016). With the benefit of hindsight, it is apparent that these agents, fully aware of the massive cost of double-checking of their reports, had no interest in orderly due diligence beyond analyses of 8-K statements. Once the supply chain of U.S. investors' capital had been established, Chinese companies came out in droves and the US stock market was set to experience an episode of unprecedented siphoning off of US capital.

Chinese RMs serve as an exemplary case of perfect environment for activist short-sellers. Information asymmetry, as conditio sine qua non for activist short-sellers, has been exacerbated by several independent sources throughout the whole period. One of the major drivers can be seen in the conflict between the U.S. regulatory agencies and the Chinese government, exemplified by China's official warning to Deloitte Touché's Chinese offices not to provide audit-related documents to foreign regulators (Stiner and Lynn, 2012). This measure, later amended by restriction of access to companies' SAIC filings[4], effectively prevented the U.S. regulators from verifying adherence to accounting standards and made any crosschecking of filings impossible. To make the issue of intransparency even worse, U.S.-based public accounting firms tasked with auditing financial results were found to be outsourcing most of their tasks to firms or assistants in China without investigating whether PCAOB standards are followed (Templin, 2011). Finally, all of the above was paired with persistently low quality of financial reporting in China[5] (Wang and Wu, 2011), notorious lack of ethical standards in the accounting profession[6] (Peng and Bewley, 2009), and lack of incentives to investigate fraudulent practices of companies listed abroad (Paul Gillis, 2012).

Remarkable degree of opacity and official party protection allowed Chinese companies, such as Rino International Corporation (Dan David, 2016), to present revenue values to US investors that were several times the size of those reported to Chinese

[4] State Administration of Industry and Commerce, responsible for collection and archiving of financial statements.
[5] Wang and Wu (2011) document ubiquitous restatements of financial reports in Chinese companies.
[6] Seen in large scale falsification of financial statements (Peng and Bewley, 2009)

regulators. Ultimately, costly verification of reported operating results and absence of regulatory oversight opened a window of opportunity for on-ground investigations and international cooperation, thus laying foundations of short seller activist research.

1.4 Activist short-Sellers: Modus Operandi

Unlike passive short-sellers, activist short-sellers, once a short position is built, actively endeavor to disseminate their privately acquired information about the presumed state of firm's operations. Rather than waiting for the information to be gradually incorporated into share prices, if at all, activist short-seller discloses a detailed research thesis elaborating on what he purports to be strong arguments against the firm's narrative. Such arguments often concern financial reporting red flags or intellectually fraudulent behavior of the management (failure to disclose related party transactions and management relationships). Activist disclosures, often incorporating elements of forensic accounting, key person screening and wide ranging on-ground investigation, are provided to the public free of charge. The notion of proprietary research being distributed is in stark contradiction to attitudes of traditional asset managers, especially when considering the vast resources required to perform a firm analysis in the scope commonly encountered in activist reports. To illustrate the scope of the activist research; in 2018, Gabrielle Grego of QCM and Nate Anderson of Hindenburg Research have published a research report on Aphria Inc[7], a cannabis producer and distributor. The research stretched across a period of several months, during which the activists untangled a network surrounding their target's executives, traveled to Aphria's facilities in Jamaica and Argentina, and investigated recently acquired subsidiaries in Colombia, only to discover dilapidated offices and abandoned construction sites.

To reach possibly wide audience and convince existing shareholders to re-evaluate beliefs about their holdings, multiple information dissemination channels are employed (discussed below). Although most of the subsequent price correction stems from attrition of long-side investors (as evidenced by Ljungsqvist and Qian, 2016), multitude of hitherto uninvolved traders on the short side were shown to follow these channels and take advantage of short sellers' campaigns. An example of the latter can be found in Gillet and

[7] Published on : https://hindenburgresearch.com/aphria-a-shell-game-with-a-cannabis-business-on-the-side/
Presented personally by Gabrielle Grego at Kase Learning Conference (12 March, 2018)

Renault (2019), who documented a surge in algorithmic trading on information contained in short-seller theses on the day of publication as well as prior to this date. Alas, sophisticated short-sellers trading in response to activists' allegations may not always act in accordance with market regulations. They may attempt to profit from accelerating the price decline with the use of spoofing and layering[8], as was alleged by Burford in the case of Muddy Waters' campaign (McCormick and Fletcher, 2019), further tarnishing short-sellers' troubled reputation.

1.5 Activist Short-Selling: Risks

By assuming an active role in the diffusion of information, activist short sellers face a set of risks that far exceed those of passive actors. In addition to the usual set of risks common in passive short selling, such as adverse development in the stock price itself (while having a capped upside), risk of borrowed shares being recalled, increase in loan fees or, even in the case of favorable price development, arbitrary increases in collateral requirements during crises times (Marc Cohodes, 2017), activist short sellers' public disclosure very often provokes a strong reaction from the target.

Targeted companies have a wide weaponry to choose from when defending against a publicly acting short-seller. The less traditional one would be convincing their shareholders to request delivery of shares from their broker, and thus effectively inducing a short squeeze against the short-seller (Lamont, 2012). Alternatively, companies may choose to fight accusations by issuing an exhaustive public rebuttal of activist's claims, which has so far proven as a mixed-result approach. Less palatable techniques of deterrence target the person of activist himself in order to intimidate him from continuing public engagement. An example thereof seen in the recent past involved interrogation attempts by covert private-intelligence agents and professional ad hominem smear campaigns (seen in GeoInvesting's campaign against AmTrust and Muddy Waters' campaign against Groupe Casino Guichard).

More conventional defense methods and the major source of tangible, financial risk are, however, those of legal nature. Since the source of pressure on stock price can now

8 Spoofing: Placing a sell instruction with no intention for the order to be executed to mislead other market participants about the security order book (Commodity Exchange Act 7 U.S.C. § 6c(a)5 (2014))
Layering: Creating a semblance of market activity in the desired direction by placing opposing orders from multiple accounts (Case SEC vs. Aleksandr Milrud, 2015)

be easily identified, short-seller may face a prospect of legal proceeding for reasons ranging from defamation and libel to disinformation and market manipulation. The most recently concluded case of such proceeding is the defamation lawsuit of the now delisted Yangtze River Port and Logistics against Nathan Anderson of Hindenburg Research[9]. Legal risk concerns even those activists, who operate in pseudonymity, as platforms, through which they published their claims, can be subpoenaed to provide activist's account details with their identity. Activists therefore apply considerable caution when addressing possible frauds due to substantial repercussions in case of misidentification of available evidence.

Ramifications from wrongful accusation of company's management, however, span beyond legal and regulatory action. Activist short-selling is business largely based on credibility, a function of activist's reputation. If the activist's track record were to be blemished by ex post verified erroneous claims leading to a lawsuit defeat, ability to exert pressure on target's stock price could be severely diminished (Kovbasyuk and Pagano, 2015, and Benabou and Laroque, 1992). Despite plenty of publicly known libel and defamation lawsuits (MiMedx vs. Viceroy, GTX vs. Andrew Left, Eros International vs. GeoInvesting), there are only limited case of lost lawsuits, none of which pertains to well-known short-sellers.

Lastly, publicly acting short sellers face the risk of losing the informational advantage over the general market by means of internal information leak and subsequent front-running. A short-seller whose information has reached the market before he had a chance to build his position and release his research, may be completely deprived of any reward for his costly research (analogously to Khan and Lu, 2013, who observed front-running prior to insider sales in companies with poor accounting quality).

1.6 Activist Research Dissemination and Main Actors

The advent of social media and platforms for sharing ideas about developments in public markets marked a radical departure from the usual model of reliance on sponsored sell-side analysts' research, while providing a degree of clarity on the incentives of

[9] Supreme court of the State of New York passed a motion to dismiss the case. Court file available at: https://iapps.courts.state.ny.us/nyscef/ViewDocument?docIndex=mdQ9fGo9d3832BZUZJCKaA==

research authors. Online fora and social news aggregation websites opened up a trove of vast information resources actively used by professionals and retail investors alike.

The pioneer among all investment-oriented platforms facilitating exchange between laymen and professionals alike, was undeniably Seeking Alpha. With almost 17 million active users every month as of November 6, 2019[10] and a broad coverage of various securities classes, it is an unparalleled source of investment ideas and research. Seeking Alpha (SA), meant for investor-to-investor interaction, was later followed by more narrowly focused platforms that aim to collect data rather than to enable idea exchange. Activist Shorts Research, now Activist Insights, and a recent newcomer, BreakOut Point, cater predominantly to an audience interested in activist short selling. These three now form a triumvirate of independent research providers, who brought short-seller activism into prominence.

Seeking Alpha in combination with Twitter has allowed authors with no formal investment background to start disseminating their articles on corporate malfeasance and fraud, gather followers and subsequently generate traction extending beyond to the real world. A fitting example of this development is Andrew Left**.** Andrew Left, now famous character behind Citron Research, who first started publishing research on his own web page, quickly expanded to Seeking Alpha in 2006 and Twitter in 2011. In October 2015, Left released an article questioning business practices of Valeant Pharmaceuticals and accusing it of recording false sales to entities effectively controlled by Valeant. Shortly thereafter, similar broadside by Roddy Boyd of Southern Investigative Reporting Foundation reached SA. The fallout of their revelations that followed pushed Valeant from grace and propelled Left to the forefront of short-seller activism. Whereas the aforementioned Roddy Boyd seeks no personal gain[11], actors such as Andrew Left have enjoyed generous reward for their research (CNBC indicated Left has generated an average annualized return of 89% between 2007 and 2017[12]).

[10] https://seekingalpha.com/page/about_us

[11] R.Boyd has publicly voiced his support for activist short-sellers but described his incentives as being based on moral grounds. His organization, SIRF, depends on public donations and neither he, nor his organization enter into any positions in securities covered in their articles. (R. Boyd, July 9, 2019)

[12] https://www.cnbc.com/2018/10/30/short-seller-andrew-left-to-seek-investor-money-for-his-first-hedge-fund.html

Currently, the short-seller activist scene is dominated by a mere handful of institutional and private investors. The most prominent among those being Carson Block of Muddy Waters and Sahm Adrangi of Kerrisdale Capital, who along with Ben Axler (Spruce Point Capital), have built their reputation on uncovering fraud in US-listed Chinese companies. Although all of the above have grown into sizeable institutional asset managers, Twitter remains one of the main communication media to reach their readers and engage with the public.

Using the number of Twitter followers (as of November 7, 2019) as a proxy for public visibility, the following table shows a summary of the most active and publicly visible short-seller activists.

Table 1 Overview of activist short-sellers

Activist	Twitter Followers	SA Followers	Character	Key Person
Citron Research	120700	5771	institutional	Andrew Left
Muddy Waters	83700	-	institutional	Carson Block
Marc Cohodes	35400	251	private	Marc Cohodes
Kerrisdale Capital	31900	2861	institutional	Sahm Adrangi
Gotham City research	23600	649	private	Daniel Yu
Spruce Point Capital	23600	3024	institutional	Ben Axler
Viceroy	17100	180	private	Fraser Perring
Anonymous analytics	17000	-	private	x
Aurelius Value	15100	931	private	x
Hindenburg Research	14300	2016	private	Nate Anderson
Prescience Point	14300	1120	institutional	Eiad Asbahi
Mox Reports	8900	-	private	Richard Pearson
Unemon Research	7076	584	private	x
Probes Reporter	7515	429	private	John P. Gavin
Quintessential Capital	6151	255	institutional	Gabrielle Grego
Copperfield Research	5793	989	private	x
Wolfpack Research	5467	-	institutional	Dan David
Iceberg Research	5289	-	private	Arnaud Vagner
Blue Orca Capital	4607	-	institutional	Soren Aandahl
Glasshouse Research	4251	-	private	x
Alpha Exposure	4005	2667	private	x
Fuzzy Panda	3500	795	private	x
GMT research	3280	-	private	Gillem Tulloch
Bonitas Research	2876	-	institutional	Matthew Wiechert
J Capital	2650	308	private	Anne Stevenson

The majority of the above activist is comprised of private offices and one-man operations, that invest their own capital and, in some instances, provide equity research services to institutional clientele. It is however reasonable to assume that several seemingly private operations function as fronts for larger institutional asset managers who shun negative stigma surrounding activist short-selling, or alternatively provide research with the added value of acting as a catalyst (as suggested by Joshua Mitts, 2019, who observed abnormal short-selling volume in stock and options prior to releases of a bearish reports, exceeding volumes attributable to private investors).

2 Hypothesis Development

"There's not one specific theme that you would say that's it. If you were shorting just on valuation, over the past few years you're out of business. […] But then there are frauds, fads and failures. Those are the three main themes for any short seller. "

- Andrew Left, founder of Citron Research,
In an interview with Keith McCullough, October 11, 2018

2.1 Campaign Targets and Activist reports (H1 and H2)

As Andrew Left noted in the opening quote, there are no specific themes shared by all short-seller campaigns. Study of activists' reports reveals their claims range from unsustainable debt levels and manufactured revenue accounts (Grego vs. Bio-On SPA) to allegations of insider self-dealing, or undue, undisclosed incentivization of distributors (Hindenburg vs. Predictive Technology, Culper Research vs. AtriCure Inc[13].). Nevertheless, activist as well as passive short-sellers share a common trait of targeting companies whose stock they presume will necessarily decline in the future. Common predictors of stock declines, documented in prior literature, should thus attract attention from both short-seller groups regardless of their strategy.

One of the necessary conditions for downward price correction is a divergence between the firm's underlying value per share and market price for its stock. As such, it should be expected that gross overvaluation of shares will be among the traits sought after by activist short-sellers. In combination with the sizeable body of evidence of overvaluation in markets featuring short-sale restrictions (among others, Hong and Stein, 2000, Chang Cheng and Yu, 2007), my first hypothesis[14] is as follows:

H1: Firms exhibiting overvaluation features are more likely to attract attention of activist short-sellers.

The main premise of activist short-selling is to disclose new or emphasize not-yet incorporated information about the targeted company. Activist short-selling, as a mostly retrospectively oriented endeavor, seldom attempts to predict how a firm will be valued

[13] Culper Research : https://cutt.ly/zrxN2RG,
Hindenburg Research's campaign : https://hindenburgresearch.com/predictive-technology/
[14] All hypotheses are stated in alternative form.

by the market in the future. It rather strives to build a new narrative on how the investors should think about a given company or its management. Therefore, with the novelty and precision of information being paramount for the campaign's ultimate success, companies with low availability of information, wider dispersion of investor opinion and higher uncertainty regarding the veracity of underlying financials should experience higher activist coverage. Following Epstein and Schneider (2005), if investors are confronted with information that compromises quantifiable uncertainty in their prior believes or, alternatively, introduces the notion of Knightian uncertainty[15] their reaction will be ambiguity averse. Although the investors may not fully accept activist's claims, the nascent ambiguity is deterring enough for them to reduce exposure to securities whose uncertainty they no longer trust. Concludingly, I hypothesize the following:

H2: Firms exhibiting ambiguity features are more likely to attract attention of activist short-sellers.

2.2 Impact of Activist Short-Selling (H3)

Activist short sellers may publicly profess their noble intents of uncovering corporate malfeasance for the common good, but it is only sensible to assume, that profit plays a non-negligible role in their motivation. For institutional activist short sellers, the incentives are clearly defined, as they have an obligation to deliver positive returns to allocators. For a campaign to be considered, the expected return must exceed research and investigation expenses, and any transaction costs incurred in the process of placing and maintaining an order. Furthermore, potential return should cover any costs associated with outsized risks assumed by the activist. One of such risks are those related to legal battles and prolonged standoff between the activist and his target, as was extensively documented by Lamont (2012). Lastly, concentrated short positions may result in sizeable losses for the activist in case of takeover announcement and the accompanying spike in stock price, as described in Meneghetti, Williams and Xiao (2019), further increasing minimum required return per campaign.

[15] Alternatively, we can relax the notion of Knightian uncertainty for our purposes to uncertainty that is very hard to estimate.

Passive short-sellers have long shown their prowess in detecting financial misconduct (Karpoff and Lou, 2010), identifying overpriced firms (Dechow et al., 2001) and negative earnings surprises (Christophe, Ferri, and Angel, 2004). Demonstrated skill and informational character of activist campaigns, complemented by analysts' hesitancy to release negative coverage (Scherbina, 2007) should make activist short-seller campaigns a powerful catalyst in correction of mispricing. Increased risk involved and expertise needed for a successful short-seller campaign thus motivate the third hypothesis:

H3: Shares of companies targeted by activist short sellers will deliver significantly negative abnormal returns in the short term (long term).

2.3 Effect of Overvaluation and Ambiguity on Stock Returns (H4 and H5)

Since overvaluation and ambiguity features pertain to functioning of companies on economic and financial level, the expectation of negative ramifications due to abnormal levels in either one of the feature groups appears within reason. Overvaluation, as was observed by Beneish and Nichols (2009) lead to sizeable negative future returns, even if no activist is involved. Ambiguity regarding the underlying information, on the other hand, should entail larger immediate reactions to new negative information, or information perceived as such (Zhang, 2006). Ambiguity characteristics studied here correspond to the well-known Ellsberg paradox, causing shareholders of the targeted firm to react in ambiguity-averse manner upon obtaining a "destabilizing" signal from the activist. Consequently, securities that suddenly appear riskier to hold, should experience attrition of their shareholder base.

Given sufficient novelty of negative information brought to investors' attention and trustworthiness of the source, both characteristics should have a pronounced adverse effect on the stock price following a research publication.

A thorough analysis of firm's operations requires a level of sophistication and research capacity that may not be available to investors or sell-side equity analysts, who remain responsible for the majority of the available equity research[16]. The information that ultimately reaches investors can moreover be influenced by selection bias of analysts, who may choose the amount of research effort according to their prior beliefs about company's prospects (Hayes, 1998), thus applying less vigor at analyzing stocks about which they are pessimistic. Environment of scarce or widely diverging forecasts can furthermore give rise to undue optimism in analyst judgement, as the reputational risk decreases with analysts' increasing uncertainty (Ackert and Athanassakos, 1997).

If analyst coverage is missing entirely or provides a set of discordant inferences, the nature of activist's findings along with the characteristics of the firm can be expected to affect the rate at which the activist's claims are reflected in share prices. The data suggesting overvaluation are comparably easy to collect and verify, unless the overvaluation claims are based on entirely fabricated balance sheets and fictitious assets. In either case, shareholders of the targeted company can test and evaluate short seller's claims and react after due consideration. Short theses targeting ambiguity-heavy companies, on the other hand, strive to change shareholders' perception of the company and, most of all, corrode their assessment of underlying riskiness by means of undermining hitherto accepted information. The process of verification to regain understanding of security's risk will necessarily require more strenuous examination of all pertinent financial data, which in its extent may come close to an audit-like investigation. The scope of such task, paired with corrosion of trust in available data, may be dissuading enough for investors to accept the claims in the short run while the in-depth investigation takes place. Additionally, as suggested by Fox and Tversky (1995) in their comparative ignorance hypothesis, an uncertain prospect of holding onto a security in defiance of potentially better-informed individual (the short-seller) may become less attractive, possibly leading the shareholder to abandon his seemingly inferior judgement.[17]

[16] Considering the impact of sell-side analysts on trading volume activity (Ryan and Taffler, 2001) and their accuracy of prediction in comparison to buy-side analysts (Hobbs and Singh, 2015), they are used throughout this section as a benchmark for sophisticated research expertise.

[17] Fox and Tversky (1995) argue that ambiguity aversion may be driven by contrasting one's position with that of a seemingly more knowledgeable individual, leading to diminishing sense of own's competence in estimating and a decrease in willingness to bet on own assessment of a game.

Concludingly, I hypothesize that in the effort of avoiding being the greater fool by holding onto a misunderstood security in face of ambiguity, investors will choose to accept activist's claim in the short-run. Thus, more pronounced, negative abnormal returns should be observed immediately after the release of activist report:

H4: Ambiguity features will have comparably larger effect on short term returns than overvaluation features.

Conversely, targeted firms that are rich on overvaluation-based features will undergo market-wide scrutiny before activist's discoveries find their way into share prices. Not least due to propensity of individuals to maintain degree of conservatism in revising their information (Doukas and McKnight, 2005), I posit that activist claims targeting overvaluation will be absorbed in a comparably more gradual fashion:

H5: Overvaluation features will have larger effect on long term returns than ambiguity features.

3 Sample Construction

To study the determinants of coming under activists' scrutiny and the effect on subsequent abnormal returns, I have collected data from major activists' own websites and Seeking Alpha. The sample analyzed covers a narrow selection of short-sellers with high public visibility, active on the above platforms in the period between 2011 and 2019.

3.1 Twitter

The main venue for announcing initiation of short-seller coverage is by far Twitter. Founded in 2009, Twitter is one of the major online news and social networking sites. It provides its users space to generate content akin to microblogging and connect with people across the globe. One of Twitter's unique features is a limit imposed on the maximum length of "tweets", messages to be shared with user's followers. Currently at 270 characters, this feature limits usability of Twitter as a medium for exhaustive elaboration on complex content, such as short-selling theses. Nevertheless, the common use of Twitter for spreading news along with the possibility to supplement one's tweet with a video or a screenshot, accustomed its users to treating Twitter feed as an alternative news channel. Consequently, the use of embedded links to own website or employing a continuous flow of short messages with snippets from research theses enabled sharing of content of more demanding nature. Activists with their own websites and newsletter service (such as Spruce Point Capital or Blue Orca Capital) found use for Twitter feed to deliver the most impactful segments of their theses and thus reach an audience, that would otherwise shy away from studying the thesis point by point.

3.2 Seeking Alpha

Having its origin in 2004, Seeking Alpha became the most popular crowd-sourced content service for the financial world. According to Seeking Alpha's homepage, more than seven thousand independent contributors publish ten thousand investing ideas every month to an audience of 17 million users[18]. In order to assure, that Seeking Alpha quality standards are upheld, each article undergoes editorial review prior to publishing. Seeking Alpha provides a unique opportunity for finance professionals to share their views and

[18] https://seekingalpha.com/page/about_us

research with a wider audience. Seeking Alpha complements Twitter in that it allows activist short sellers, who have no website of their own, to refer Twitter followers to a full version of their research while enabling more dynamic interaction with the public on Twitter.

3.3 Other Sources

In addition to Seeking Alpha, Twitter and activist's own websites, short sellers analyzed in this thesis were identified on Preqin and recordings from investment conferences, such as Sohn's and Kase Learning (available online)[19]. Preqin is one of the main financial data providers covering the alternative assets market. Preqin covers wide range of asset classes, among others, long/short equity hedge funds with a short bias such as those managed by celebrity activist short-sellers[20].

3.4 Sample Construction

To construct the initial sample, I have manually searched through Preqin and available recordings of conferences on short-selling. Since short-seller activists are a small community of individuals who know each other either personally, or who are, at minimum, acquainted with undertakings of the others, lists of Twitter followers were utilized to expand the starting selection of activists. Short-sellers were added to our sample if they accumulated at least two thousand followers and have maintained an active status, which I define as a minimum of one campaign in 12 months. This measure filters out two groups of short-sellers, namely opportunistic one-time activists who want to monetize their whistleblowing, and activists with no interest in providing reliable research in the long term. The latter group was documented to switch pseudonyms once their credibility has been lost (Mitts, 2019). To account for short-sellers who do not face the full range of risks (namely risks pertaining to legal recourse and reputation damage), the sample contains only activists whose identities are publicly known, or whose identity can be obtained through a subpoena of the platform on which they operate[21]. Moreover,

[19] https://www.youtube.com/playlist?list=PLAsuvKOcgPWlbekzMXwYcao3Ov7umz9zz
[20] https://www.preqin.com/about/who-we-are/20416
[21] This restriction excludes, for example, the short-seller Anonymous Analytics, who avoids using Seeking Alpha, which requires identity verification and with all likelihood protects her identity by using incognito email service on Twitter.

we exclude short-sellers who do not indicate whether they have acted upon their research (disclosed their short position)[22].

For activist with own website, I built a python spider (full Python code, including the spider can be found in the section Python Code) to crawl through the webpage and collect target company names and research publication dates. For those with no website of their own or no indication of time and date on which their research was published, I collect the relevant data manually from twitter posts. If an activist published several theses covering the same company on multiple occasions, only the earliest research is considered. For reasons ranging from completeness of data and uniformity of accounting standards to ease of direct data extraction, only those campaigns targeting US listed and domiciled companies are considered. For research published after US market closing time or non-trading days, I use the following trading day as the day of campaign initiation.

The sample of 434 campaigns initiated by a total of 19 activist short-sellers was filtered according to the criteria above to ultimately arrive at 239 unique campaigns covering 32 French Fama industry groups in the period between 2011 and 2019. The sample composition and additional description of the nature of sample companies are provided in more detail in Appendix C. In Appendix A we provide an overview of recurring themes in short-seller research.

[22] It is appropriate to provide a caveat regarding voluntary disclosure: Unlike in the EU, US-based short-sellers are not subject to disclosure of significant short positions. As a result, activist's voluntary disclosure carries no indication on position size and must not necessarily reflect any actual position whatsoever.

4 Model Variables and Research Design

In the following chapter, I shall elaborate on the methods employed to support or reject hypotheses presented in section 2. The major part of this chapter addresses the selection of individual determinants included in the overvaluation and ambiguity sets. In case of determinants that are not readily available from financial statements or that are not self-explanatory in their nature, I supplement the description with a brief account of their construction.

4.1 Overvaluation

Whether the underlying value of a company is estimated with a discounted cash flow method or relatively to company's industry peers, price to value mismatches in either direction will inevitably emerge. Mere presence of a mismatch, however, can hardly serve as a reliable indication of impending price correction and even less so as a convincing argument to be used by an activist short-seller. The mismatch becomes important, however, if a substantial and persistent discrepancy between market prices and any reasonable spectra of estimated values per share is observed. In the case of gross overvaluation, market value stands in such disproportion to intrinsic value, that to achieve a performance justifying market expectations would be a matter of sheer luck.

4.1.1 Price to Earnings Ratio

If we were to accept the notion, that valuation ratios proxying for overvaluation tend to oscillate within their historical ranges, and deviations in either direction from the mean are not of long-termed nature, any substantial deviation from the mean should necessarily lead to one of the following scenarios. In the case of PE ratio, the nominator, share price, collapses under the weight of unfulfilled expectations, bringing about a mean reversion as a consequence (Campbell and Schiller, 2001). Alternatively, balance can be restored by means of an increase in the denominator (earnings per share), thus vindicating market expectations. As Campbell and Schiller note in their 1998 study, despite radical changes in the economy, valuation ratios have long remained within fairly well-defined boundaries. Interestingly, this tendency seems to have been gradually disappearing, as neither the Dotcom bubble burst of the early 2000s, nor the Subprime mortgage crisis of 2007 to 2010 brought the average PE ratios of US companies to their historical means (as

shown in Appendix D), leading to an extended growth since 2009, well beyond historical bounds.

In the environment of little regard for sensible valuation ratios, the activist targeting a company on the basis of valuation-related concerns may be seen as leading a futile battle. Looking at overvaluation solely from the perspective of valuation ratios would be, however, missing half of the picture, since the effects frequently co-occurring with overvaluation, rather than the general manifestation through ratios, are of interest to activists. For instance, Jensen (2005) argued that managers face an undue pressure to beat quarterly earnings forecasts lest they want to face disproportionate market value decline and see their incentive pay reduced. Consequently, as Jensen alleges, many bend to the pressure and set on the self-reinforcing path of earnings manipulation. Moreover, such value-destroying managerial behavior is rewarded rather than punished by the market, as only the insiders are informed about the inner deficiencies. As in the case of Valeant Pharmaceuticals, in which the combination of cheap money and weak internal governance allowed for streak of unrestrained acquisitions and unprecedented aggressive accounting (Eavis, 2016), it took a sophisticated outsider to convince the market about its overblown market value. For this reason, the first feature to be considered in the overvaluation set, price to earnings ratio (*PE*), can be understood as a measure of overvaluation as well as possible indication of nefarious activity.

4.1.2 Discretionary Accruals

Overvaluation attributable to a wide divide between intrinsic value and unrealistic market expectations, as suggested above, can be driven by exaggerated earnings management. Although earnings management per se does not necessarily imply malicious intent, practices such as switching between revenue recognition methods and, especially, ample use of discretionary accruals were documented to have a detrimental effect on the extent to which reported earnings reflect operating fundamentals (Yeo et. al, 2002).

Decline in informativeness of earnings and, most of all, association of abnormal accruals with negative future returns (Chan, Chan, Jegadeesh and Lakonishok, 2001) create appropriate conditions for short seller activists. To study whether the accruals or mispricing associated thereto (Xie, 2001) attract activists, I opt for simple measure of discretionary accruals following Collins and Hribar (2000). The estimate of total

discretionary accruals (*TACC*) is based on the difference between Net income before extraordinary items[23] (EBXI) and Operating cash flows (CFO), divided by total assets four quarters prior to the announcement to allow for comparability. Both income and cash flow statement items are recorded at trailing twelve months basis.

Equation 1 Total Accruals

$$TACC_{CF} = \frac{EBXI - CFO_{CF}}{TA_{t-1}} \quad (1)$$

4.1.3 Overinvestment of Free Cash Flows

Management misbehavior can take several forms. In addition to ill-intentioned earnings manipulation, managers may pursue a range of personal goals that are in contradiction to shareholder value maximization. Recurrent critique in short selling theses pertains to one of these pursuits, namely to overinvestment of free cash flows. If a firm has weak internal control mechanisms, managers may engage in wasteful investments with negative net present value (Jensen and Meckling, 1976) or invest beyond optimal levels (Shleifer and Vishny, 1989). Regardless, whether the reasons involve making oneself indispensable or managerial empire building, firm value will be negatively affected as a consequence. The primary assumption for overinvestment, nevertheless, are positive free cash flows. If activists prove to target companies with questionable economic performance, positive free cash flows may not be present at all. In such case, overinvestment of externally sourced funds might not occur due to debtholders' scrutiny (Jensen, 1986) or due to personal costs to the manager if the company were to experience financial distress (e.g. Opler and Titman, 1994). To create an estimate of Free cash flow overinvestment, I build on a simplified version of framework presented by Richardson (2006), who decomposed total investment (I_{Total}) in two components, namely expenditures required to maintain existing assets (amortization and depreciation) and investment in new projects.

[23] Extraordinary items are defined as gains or losses on income statement from events, which are unusual with regards to normal operations of the firm or infrequent in nature.

Equation 2 Overinvestments of Free Cash Flows

$$I_{Total,t} = CAPEX_t + Acquisitions_t + RD_t - SalePPE_t$$

$$I_{Total,t} = I_{Maintenance,t} + I_{New,t}$$

$$I_{New,t} = I^*_{New,t} + I_{OverInv,t} \quad (2)$$

$$FCF_t = CFO_t + RD_t - I_{Maintenance,t} - I^*_{New,t}$$

$$I_{OverInv,t} = FCF_t - \Delta Equity_t - \Delta Debt_t - \Delta Financial\ Asset_t - OtherInv_t - Other_t$$ [24]

To assure comparability across all companies regardless of size, overinvestment (*IOver*) is divided by total assets in the announcement quarter. By employing the above estimation approach, we account for excessive acquisition activity, a theme of enduring popularity among short-seller activists, as well as occurrence of insufficient investment in case of negative values.

4.1.4 Altman's Z-Score

Given that activist short-sellers profit from decline in market value of their targets, selection of especially vulnerable or potentially distressed company appears as a sensible strategy to achieve positive return on invested capital. A target manifesting internal weaknesses and inefficiencies in such extent that operating and financial distress become imminent could thus be seen as low hanging fruit for a short-seller. For this reason, the fourth overvaluation determinant focuses on predictors of Chapter 11 bankruptcy in a compact form of Z-Score, conceived by Altman (2000) as an index of 5 distinct business ratios weighted by their respective discriminant coefficients.

Equation 3 Altman Z-Score

$$Z - Score = 1.2X_1 + 1.4X_2 + 3.3X_3 + 0.6X_4 + 1.0X_5 \quad (3)$$

Where X_1 = working capital/total assets,
X_2 = retained earnings/total assets
X_3 = earnings before interest and taxes/total assets,
X_4 = market value equity/book value of total liabilities,
X_5 = sales/total assets

[24] Formulas for individual components of I_{Over} are provided in Appendix F.

If the resulting Index score lies below the cutoff of 1.81, the firm faces a realistic prospect of bankruptcy in future periods. Although Z-Score was not initially designed to predict delisting, we presume bankruptcy and delisting to carry similarly positive implications for a short-seller. Delisting should be preceded by financial difficulties comparable in nature (not necessarily in scope) to those of bankruptcy, and thus we deem Z-score to be of comparable relevance in our case.

4.1.5 Momentum and Financial Ratios

Sizeable body of research has documented degree of predictability of stock returns on an individual level by observing past price developments. For example, Jagadeesh (1990), observed such regularity and documented the extent of stock returns holding sway over returns in the following month. A portfolio built on the basis of his findings would deliver significantly positive returns in a short to medium term. On a similar note, Doukas and McKnight (2005) along with Jegadeesh and Titman (2011), present evidence for the European and U.S. stock market, respectively, suggesting that stock returns momentum is consistent with slow diffusion of information and belated updating of prior beliefs, ultimately leading to continuation of previously observed trend (similarly to Hong, Lim and Stein, 2000). On the grounds of their observations and our assumption, that activists hold their position for periods of shorter duration[25], the fifth parameter in Overvaluation set is *Momentum*. *Momentum* assumes a value of 1, if the company delivered a negative overall return for the past quarter preceding the announcement and 0 otherwise.

Lastly, the three final features included in the first set pertain to simple, well-established firm evaluation metrics. The first to be applied is the ratio of Book value to market value of equity, as studied by French and Fama (1995) (FF) and used in its more common form as Price to Book (*P/B*) to gauge the potential of future returns. Firms with high readings of BE/ME (low P/B), FF suggested, tend to manifest distress and should be expected to deliver lower returns than their low BE/ME (high P/B) counterparts. The second ratio, Operating Cash Flow to Total Assets (*cROA*), indicates the profitability of targeted company's assets regardless of its revenue and expense recognition policies. By

[25] Our assumption is based on the following arguments: price decrease paired with increased publicity necessarily attracts investors who might attempt to short-squeeze short-sellers (as seen in Ackman's Herbalife campaign); risks associated with short-selling increase with time, such as takeovers, ETF-additions and mutual fund portfolio inclusions are more likely to appear in a period of several quarters rather than in a week or a month; frequency of campaigns paired with capital constraints would necessitate reduced investment size per campaign that might not ultimately earn enough to cover all research expenses.

virtue of focusing on actual cash flows, the ratio should provide a more reliable picture of company's operations and diminish any effects of possible earnings manipulation (Barua and Saha, 2015). The last parameter, Debt Coverage (*DebtC*) aims at measuring firm's ability to honor its obligations to creditors from existing cashflows, in that it relates trailing twelve months Cash flow from operations to total debt (Zeller and Stanko, 1994).

If activists do in fact target companies exhibiting overvaluation, I posit target's *P/B* to be in the lower quintile of its respective industry's P/Bs. Similarly, *cROA* and *DebtC* should also be in the lower strata of observed industry values, as low readings suggest weakness in generating sufficient cash flows.

4.2 Information Ambiguity

As I conjectured in the second hypothesis, I expect activists to target a company, if a degree of uncertainty regarding its financial and operating health is present. If convincing claims are laid against such targets, the ensuing ambiguity about hitherto accepted information may influence investors to reduce their holdings in favor of less uncertain investments. Consequently, targets with characteristics suggesting increased uncertainty about the veracity of underlying financials[26] should be easier to drag down upon revelation of new, negative information and thus be an attractive choice for a profit maximizing activist.

4.2.1 Corporate Opacity Set

For the ambiguity feature set to be a relevant selection determinant, companies should manifest increased opacity with respect to company-specific information. To gauge opacity, I follow approach presented by Uyghur (2017) and consider a set of 4 distinct features. Although my use of the model differs in the argument I attempt to substantiate[27], the objectives of my corroboration are well aligned with those of Uyghur. The features to be included in the opacity set are modified versions of the original parameters: trading volume, bid-ask spread, number of analysts covering the company and analyst's disagreement.

[26] The uncertainty in our definition pertains to the uncertainty itself. Upon introduction of new uncertain information, the short seller brings about destabilization of investors' prior understanding of quantifiable uncertainty. The conception is thus akin to Knightian ambiguity, hence the feature set name.
[27] Uyghur's research focuses on CEOs striving to obscure their managerial deficiencies to retain their position.

Trading volume (*lnVol*), calculated as natural logarithm of average daily dollar volume of the previous fiscal year, reflects security's liquidity and was documented to correspond inversely to presence of information asymmetries (Leuz and Verrecchia, 2000). The bid-ask spread, as suggested by Copeland and Galai (1983) widens in proportion to information asymmetry surrounding a security, as market makers attempt to maximize their profits vis-à-vis a mixture of informed and uninformed traders. Differently to method chosen by Uyghur, bid-ask spread (*spread*) is calculated according to monthly corrected version presented by Abdi and Ronaldo (2017):

Equation 4 Bid-Ask Spread

$$\widehat{s_t} = \sqrt{\max\{4(c_t - \eta_t)(c_t - \eta_{t+1}), 0\}}$$

$$\widehat{Spread}_{Qtr} = \frac{1}{N_{Qtr}} \sum_{t \in Qtr} \hat{s}_t \qquad (4)$$

Where s_t = Spread estimated over two days

c_t = daily close log-price

η_t = daily mid-range, (average of daily high and low log-prices)

N = number of day-pairs in a trading quarter

The spread obtained through the above formula is calculated on daily basis and averaged over the quarter preceding the activist campaign announcement.

The number of analysts following a particular company is added in line with evidence presented by Chang, Dasgupta and Hilary (2007) in their study of financing decisions, and further supported by Hong, Lim and Stein (2000). Who argued that stocks with higher analyst coverage experience less information asymmetry and their prices are generally more informative, leading to a reduction in momentum effects and more favorable financing possibilities for covered companies.

The last feature in the opacity set, analyst disagreement (analysts' forecast error in Uyghur's work) was adjusted to cover a different aspect of analysts' forecasting disagreement. By focusing on this variable, we put less emphasis on forecasted earnings and rather prioritize disagreement in views concerning overall future performance of

covered companies[28]. Analysts' opinion dispersion for individual stock was derived from their buy, sell and hold ratings (and variations thereof), to which I assigned values ranging from -2 to 2. As indicated below, negative total score suggests pessimism about future performance and vice versa for positive scores.

Table 2 Analyst recommendation values

Analyst recommendation:	Strong Buy	Buy	Overweight	Hold	Underweight	Sell	Strong Sell
Value assigned	2	1	0.5	0	-0.5	-1	-2

Table 3 Variations in analyst recommendations

Variations in ratings		
Buy	Hold	Sell
Outperform	Neutral	Underperform
Positive	Market -Perform	Negative
	Equal-weight	
	In-Line	

Equation 5 Dispersion of Analyst Ratings

$$AnnDisp_t = \begin{cases} \sqrt{\frac{\sum_{i=1}^{N}\left(x_{i,t} - \mu_{x,t}\right)^2}{N}}, & if\ \ N > 1 \\ q_{IndAnnDisp-}, & if\ \ N = 1\ \wedge\ x_{i,t} \leq 0 \\ 0 & otherwise \end{cases} \quad (5)$$

Where N = Number of analyst ratings for given stock

$x_{i,t}$ = Value of given rating for the current quarter

$\mu_{x,t}$ = Mean of all analyst ratings

$q_{IndAnnDisp-}$ = first quintile cutoff value of firm's respective industry Analyst Dispersions

[28] This adjustment has two motivations: Firstly, neither Uyghur (2017) nor Zhao (2018) find statistically significant support for the analyst disagreement to be related to CEO ability, and to attract activists in the latter case. Secondly, due to data extraction approach selected for the purposes of this thesis, I encountered budgetary constraints limiting the scope of accessible information.

The formula for *Analyst Dispersion* proposed above is a simple standard deviation of analyst ratings with explicit penalty for negatively rated, underfollowed companies but not for companies with no coverage or one single positive rating. The reason behind this choice, relates in limited extent to arguments put forth by Hayes (1998), who argued that in expectation of positive future performance analysts may provide more precise earnings estimates. Single positive rating can therefore be expected to achieve higher precision than a single negative rating.

4.2.2 Auditor Firm

To ameliorate uncertainty regarding the reliability of data in financial statements, 10-K filings submitted to the SEC are accompanied by an independent auditor's report. Selection of reputable auditor company should assure the investing public, that all accounting standards were upheld and that financial statements are free of material errors. Although the profitability of audit firms is largely a function of reputation and public trust, the four largest companies in the industry have been time and again caught amidst monumental accounting scandals. Nevertheless, despite numerous cases of unqualified audit opinions about companies whose accounting practices were questionable at best[29], the Big 4 audit firms still enjoy a high degree of trust among the investors (Boone, Khurana, Raman, 2010). For this reason, I have collected data from over 24.000 annual 10-K filings (Exhibits 23.1 of 10-K) and recorded whether the auditor belongs to the Big 4. The variable, *Big4*, assumes a value of 1 if the auditor is not in the Big 4 and zero otherwise.

4.2.3 Manipulation Score

The following feature, originally proposed by Beneish (1999) to identify companies that likely committed accounting manipulation in violation of US GAAP, and later tested by Beneish, Lee and Nichols (2013), addresses the prospect of intentionally distorted accounting in our sample firms. In line with our hypothesis, the possibility of corroding investors prior beliefs by revealing evidence of earnings manipulation should present an attractive opportunity for activist short-seller. Furthermore, even if the company is not

[29] Arguably, the biggest offender among the big auditing firms with the highest count of unqualified audit opinions on firms which were later proven to falsify, manipulate and otherwise illegally distort their accounting, was the now dissolved Arthur Andersen, with clients such as Worldcom and Enron.

actually proven to have engaged in fraudulent earnings manipulation, subsequent stock returns of those ranking high on Beneish's manipulation score have been shown to be comparably lower than those with low score (Beneish, Lee, Nichols, 2013). In accordance with Beneish's approach (1999), the formula applied takes on the following form:

Equation 6 Beneish Manipulation Score

$$MSCOR = -4.84 + .920*DSR + .528*GMI + .404*AQI + .892*SGI + .115*DEPI - .172*SGAI + 4.679*TATA - .327*LEVI \quad (6)$$

The variables in the above formula stand for the following (Appendix F provides calculation of individual MSCOR variables): Days Sales Receivable (DSR), Gross Margin Index (GMI), Asset Quality Index (AQI), Sales Growth Index (SGI), Sales General & Administrative Expenses Index (SGAI), Total Accruals to Total Assets (TATA) and Leverage Index (LEVI). The resulting value of Beneish M-Score is of importance if the company achieved a total score greater than -2.22, in which case the company is likely to be a manipulator.

4.2.4 CEO Duality

As was mentioned on numerous occasions, weakness of internal controls can be seen as facilitating managerial decisions that are driven by motives in conflict with firm's shareholders. One of such internal control weaknesses is absence of functioning opposition and oversight of CEO's undertakings. I therefore include CEO duality in the model as an appropriate proxy for ineffective control mechanism. Lack of management oversight due to CEO's concurrent function as the chair of board of directors gives rise to a multitude of agency problems (Jensen, 1993) as it substantially increases the level of power associated with the person of CEO. Terminations of CEO-Chairman in the event of underperformance have been shown to be rather scarce (Goyal, 2001) and, more importantly, the link between independent audit committee and earnings quality (in terms of discretionary accruals) appears to weaken in the presence of CEO duality (Kamarudin, Samsudin and Ismail, 2012[30]). The variable, *duality*, takes on value of 1 if the company

[30] It should be noted, that their observation concerned a study of companies listed on the Malaysian stock exchange. P.Dunn (2004) arrived, however, to a similar, and possibly more extreme conclusion, that occurrence of fraudulent financial statements is likely, when power is concentrated in the hand of insiders.

CEO concurrently holds a position of board chairman prior to being targeted, and 0 otherwise.

4.2.5 Tonal Uncertainty in Financial Statements

The last feature in the ambiguity set relates to a growing area of research on textual analysis and sentiment analysis of written text, such as financial news stories and company filings (e.g. Tetlock, 2007, Tetlock, Saar-Tsechansky and MacsKassy, 2008). Our attention is oriented specifically to a set of 289 tonal words associated with uncertainty, proposed by Loughran and McDonald (2011)[31] (henceforth LM). Following the example of LM, to evaluate tone of financial reports for occurrence of uncertainty words, I parse the last available 10-K filing prior to activist announcement and compute the frequency of occurrence of each uncertainty term. Unlike previous research and approaches suggested by Ashraf (2017), I employ natural language processing methods to filter out non-language elements and to significantly improve code efficiency. Once collected, I apply a weighting function on each term, as in LM, to establish term's relative frequency and importance across all studied firms.

Equation 7 Tonal Uncertainty

$$w_{i,j} = \begin{cases} \frac{\left(1+\log\left(tf_{i,j}\right)\right)}{\left(1+\log\left(a_j\right)\right)} log\frac{N}{df_i} & if\ tf_{i,j} \geq 1 \\ 0 \quad otherwise \end{cases} \tag{7}$$

Where N stands for the total number of 10-Ks collected, df_i the number of 10-Ks with at least one instance of the i_{th} word, $tf_{i,j}$ is the total number of occurrences of the word in the j_{th} document and a_j in our model, unlike in LM's approach, represents the average number of all uncertainty words across all 10-Ks of respective industry in the given year. Uncertainty words are subsequently multiplied with their corresponding weights and summed into single value per company-year. Companies in the last quintile in the studied year receive a value of 1, indicating high tonal uncertainty, and 0 otherwise. Transformed values are then applied in each subsequent quarter until new annual report is submitted and publicly available.

[31] The whole list of words can be found at:
https://afajof.org/wp-content/uploads/files/supplements/Word_lists_for_22When_Is_a.xlsx

4.3 Overview of Model Variables

Table 4 Overview of model variables

Overvaluation	*P/E*	*TACC*	*IOver*	*Zscor*	*Momentum*	*P/B*	*cROA*	*DebtC*
Ambiguity	*lnVol*	*Spread*	*AnnDisp*	*LnAn*	*Big4*	*Mscor*	*Duality*	*Tone*
Controls	*lnSize*	*D/A*	*dayStd*					

Commonly used set of controls was included in the model, comprising natural logarithm of firm size in terms of Total assets (*lnSize*), Total debt to Total assets (*D/A*) and standard deviation of daily returns over the past quarter (*dayStd*).

All continuous variables were transformed into binary values, such that all determinants having a generally negative interpretation associated with elevated levels assume a value of one if they are in the top quintile and zero otherwise. Conversely, *Zscor, P/B, cROA, DebtC, lnVol* and *LnAn* take on a value of 1 if they are within the first quintile of firm-quarter values and zero otherwise.

Panel A of Table 3 presents descriptive statistics of model determinants prior to conversion to binary values (Panel B). The observations comprise 54,053 non targeted and 228 activist-targeted firm-quarters, approximately 0.4% of the total. In terms of assets (*lnSize*), targeted companies are larger on average (at 2.2%) than non-targeted firms and experience lower standard deviation of returns (*dayStd*). Both of these differences are statistically significant, although we ascribe the latter to the broader variety of non-targeted firms, that included numerous highly volatile titles. With respect to overvaluation features, targeted firm-quarters manifested higher values for price to earnings (*P/E*) and, contrary to our expectations, statistically significant, higher price to book ratios (*P/B*), albeit with sizeable standard deviation. Operating cashflows to debt (*DebtC*) of targeted companies are considerably lower and statistically significant, as non-targeted companies generated substantially more cash from operations on average than targeted firms. Total accrual adjustments (*TACC)* were significantly lower than those of non-targeted firm quarters, although the explanation thereof is not restraint of targeted firms with respect to discretionary accruals, but rather the presence of negative earnings across the target group. Overinvestment of free cash flows (*Iover*) is significantly higher for targeted firm-quarters as is the variable *Momentum*, indicating higher positive returns in prior quarter across the targeted firm-quarters.

In the ambiguity feature set, M-Score, *Spread* and tonal uncertainty (*Tone*) were all significantly higher in targeted firms than in non-targeted. Likewise, average daily volume (*lnVol*) readings were significantly higher in targeted firm-quarters. As in the case of standard deviation of returns, we attribute the differences in volume mainly to comparably lower variability in the character of firms included in the targeted firm sample.

To inspect the data for occurrence of commonalities across industries and/or time, silhouette analysis[32] on K-Means clustering of data was applied (as described in Rousseeuw, 1987). Varying number of firm-quarter clusters were tested, ranging from 2 to 60 (with 32 approximately corresponding to French Fama industry groups represented in the dataset) for the optimal number of clusters. The data analysis revealed no discernable improvement in silhouette coefficients beyond 2 to 3 clusters, built from the whole unlabeled dataset across the studied period (Appendix G). Accuracy of predictions of the dependent variable based on K-Means clusters declined dramatically above 2 clusters (average silhouette score of 0.55860 for 2 clusters, 0.45083 for 5 clusters and 0.33557 for 30 clusters). Moreover, beyond 3 clusters, cluster width distribution increased in variability, ultimately suggesting a declining match of objects to its assigned cluster as the k-number of clusters increase. Lastly, we applied the Elbow clustering method for graphical representation of within-cluster sum of squares, commonly used to aid the selection of optimal number of clusters for data partitioning. Albeit rather heuristic in nature, the elbow method confirmed there is no improvement beyond 2 to 3 clusters [33](corresponding python code can be found in the Python code section)

[32] K-Means partitions the data into clusters which are then compared on similarities among own cluster (cohesion) and dissimilarities to neighboring clusters. Neighboring clusters are those for which minimum distances in terms of data similarity are observed. Silhouette score (coefficient) can be defined as the difference between average dissimilarity of objects in their respective cluster and a minimum of average dissimilarity of cluster objects to another cluster, divided by the larger one of those values.

[33] Two possible numbers of clusters are mentioned due to variations in results depending on the random choice of the initial starting points.

5 The Model

5.1 Market Reaction to Short-Seller Campaigns

In order to examine whether the announcement of activist campaigns leads to statistically significant negative abnormal returns in the subsequent periods, I estimated the expected returns, adjusted by French-Fama 3 Factor model portfolio returns (henceforth FF3), by means of an ordinary least square regression for a period of 120 days prior to 3 days preceding the activist campaign announcement. The OLS coefficients were estimated using FF3[34] according to:

Equation 8 French-Fama 3 Factor Model

$$r_i = R_f + \beta_{i,Mkt}\left(R_{mkt} - R_f\right) + \beta_{i,size}(SMB) + \beta_{i,value}(HML) + \varepsilon \qquad (8)$$

The expected returns in the observed event window (-3, 3), with 0 representing the day of campaign announcement, were subtracted from the actual observed returns to deliver estimates of abnormal returns for the period.[35] Two distinct statistical python libraries were applied to estimate the coefficients, namely Scikit-learn and Statsmodels. No differences in their outputs that would merit further discussion were observed. To gauge the impact of short seller activism in both relative (FF3-adjusted) and absolute terms, short and long run (1 fiscal year) cumulative abnormal returns along with observed returns to the lowest point following the campaign announcement were measured.

5.2 Determinants of Activist Involvement

Given the dichotomous nature of the studied event, that has only two possible states, logistic regression and its logit transformation appears to be the most appropriate method to test the hypotheses (H1 and H2) about determinants of activist involvement. In the same fashion as Zhao (2018), the model is estimated at the firm-quarter level with data corresponding to the last quarterly filing. The selection process is more comprehensively described in the Appendix E.

[34] All FF3 data are available at:
http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html#Research
[35] The whole procedure is documented in the section Python Code, OLS-Regression.

Equation 9 Logit Model

$$Strike_{i,t} = f(\sum \beta_d Determinant_{d,i,t} + \alpha_0 + \alpha_1 LnSize_{i,t} + \alpha_2 DA_{i,t} + \alpha_3 dayStd_{i,t} + Cluster_{c,i,t}) \quad (9)$$

Where f (.) stands for Maximum Likelihood estimation of determinants' and controls' coefficients (β and α, respectively). The dependent variable, *Strike,* takes on a value of 1 according to whether the given company-quarter was targeted by an activist and 0, otherwise. The probabilities of activist involvement at means of determinant variables and any marginal effects stemming from variations thereof are then estimated via a logistic sigmoid function. To control for cluster-specific effects, cluster dummies are introduced, assuming a value of 1 in accordance to firm-quarter's assigned K-means cluster (3 clusters in total). All t-statistics are based on standard errors clustered at firm level.

5.3 Predictors of Abnormal Returns

In order to determine whether the overvaluation and ambiguity features hold sway over the magnitude of abnormal returns following an activist campaign and to test which period windows are affected more by either of the feature set (H4 and H5), the following two models were applied:

Equation 10 Model of Abnormal Returns I

$$\begin{aligned} AR(0,T)_t = \; & \beta_0 + \beta_1 Tacc_{i,t} + \beta_2 IOver_{i,t} + \beta_3 Zscor_{i,t} \\ & + \beta_4 Momentum_{i,t} + \beta_5 PB_{i,t} + \beta_6 cROA_{i,t} + \beta_7 Spread_{i,t} \\ & + \beta_8 Big4_{i,t} + \beta_9 Mscor_{i,t} + \beta_{10} Duality_{i,t} \\ & + \beta_{11} Tone_{i,t} + \beta_{12} DA_{i,t} + \beta_{13} dayStd_{i,t} \end{aligned} \quad (10)$$

The first model contains all variables that were applied in the logit model. Unlike the previous model, the coefficients are estimated for variables in their original form (not binary). The dependent variable, *AR(0,T)* stands for abnormal and cumulative abnormal returns, adjusted with 3-factor Fama French model for periods (T) of: 1-day (within announcement day, *AR(0)* and following day *(CAR(0,1),* 5 days, *CAR(0,5),* 1 trading month *CAR(Month),* quarter *CAR(qtr.)* and one full year, *CAR(year).*

Equation 11 Model of Abnormal Returns II

$$AR(0,T)_t = \beta_0 + \beta_1 Overvaluation_{i,t} + \beta_2 Ambiguity_{i,t} + \beta_3 lnVol_{i,t} + \beta_4 DA_{i,t} + \beta_5 dayStd_{i,t} \quad (11)$$

The second model is estimated on the basis of binarized variables that are grouped together into two groups, Overvaluation and Ambiguity. These are constructed as averages of feature set determinants that were iteratively tested and selected to achieve the best fit of the model. Overvaluation constituents that survived this iterative process are Overinvestment, momentum, debt coverage and Altman Z-score. *Ambiguity* variable comprises spread, big4, Duality and Tonal uncertainty. In addition to control variables in previous models, we expand the control set by including average daily volume (formerly part of the ambiguity feature set).

6 Empirical Findings

6.1 Negative Market Reaction

As presented in Table 4**,** targeted companies experience a substantial decline in value on the day of announcement. The targets delivered a 3-Factor Fama French adjusted mean abnormal return of -0.0474, statistically significant at 1% level (p-value <0.01) with a t-statistic of -15.974, and a median return of -0.0386. Target's industry peers, manifested slightly negative mean abnormal returns of -0.0014 and -0.0008 median abnormal returns, of which the former was not significantly different from 0. These observations allow us to conclude, that activist short-selling indeed leads to substantial negative market reaction in the short term, thus providing support for H3 in the short term.

Negative mean abnormal returns of activist targets persisted throughout the observed period of 1 trading month (22 trading days), with a mean *CAR(0,1)* of -0.05088, *CAR(0,2)* of -0.05627, both significant at the 1% level (p-value <0.01), and a *CAR(0,22)* at -0.1203 (t-stat. of -8.903, p <0.01). The cumulative abnormal returns in the period of 6 trading months and one full year after the announcement were significantly negative at -0.471 and – 0.8173, respectively (p-value <0.01), lending further support to H3 with respect to long term returns impact.

In addition to abnormal returns, I investigated the extent of the initial price impact in terms of absolute return to the lowest point of the announcement day. Targeted companies lost 9.96 percent of market value on average on the day of announcement (median decline of -8.13 percent), whereas targets' industry peers experienced mean maximum decline of -2.03 percent, further highlighting the immediate and sizeable impact of activist short seller campaigns on their targets.

6.2 Attractivity of Overvaluation

Some of the critique laid out against targeted companies concerns unreliable financial statements and intentionally omitted data to inflate margins[36]. Financial statement adjustments to make oneself appear better will adversely reflect in our results as these are factors that cannot be accounted for without in-depth study of off-balance sheet items and activities of each individual company.

Regression results presented in Table 5 lend partial support to H1 with respect to 4 of the 9 overvaluation features. These are namely *P/E*, *TACC*, *IOver* and *DebtC*, while the coefficient of *Momentum* has the expected positive sign, we could not establish its statistical significance. Contrary to our expectations, the determinants Z-Score, Price-to-Book and Cash return on assets *(cROA)* demonstrate the opposite effect on activist's interest, although only *cROA* is statistically significant. One of possible reasons for the lack of statistical significance of Z-Score and Price-to-Book can be seen in overwhelmingly dissimilar values attained by targeted firms. Wide dispersion of these determinants thus makes them hardly a reliable predictor in our sample. More importantly, as can be seen in Table 3, Summary statistics, targeted firms in our sample might not generally face prospects of financial distress captured by Altman Z-score.

The coefficients of statistically significant determinants range between 0.313 for Total accruals ratio (*TACC*) and 0.986 for Debt Coverage. The attractivity of P/E ratio (coefficient of 0.851) is not an especially surprising result, given abundant mentions of price to earnings mismatch across activist reports and the ratio's straightforward implications. The highest value of 0.986 was attained by the coefficient of Debt coverage *(DebtC)*, confirming the observation of targeted firms having consistently lower operating cash flows while maintaining debt-heavier capital structure.

In terms of marginal effects, if all variables are held at their mean, a shift from the first four quintiles into the last quintile of determinant's values increases the probability of coming under short-sellers' scrutiny from 0.278% to 0.514% for P/E ratio and by 0.627% to 1.539% for Overinvestment. All other marginal effects of statistically significant coefficients are within this range. Results for coefficients of control variables

[36] Such as in Spruce Point vs. Tootsie Roll Industries. https://www.sprucepointcap.com/tootsie-roll-industries/

suggest that activists tend to target larger, more volatile and more leveraged firms, although only the last variable (*D/A*) proved to be statistically significant across all determinants.

6.3 Attractivity of Ambiguity

In Table 6, we present the regression results for Ambiguity features. In a similar fashion to previous regression results, support for H2 is not clear cut, as the impact of only 3 of the 6 determinants is statistically significant and in the hypothesized direction. The most impactful determinant in the set is the M-Score, with a coefficient of 1.455, followed by Spread (1.261) and CEO Duality at 0.289. Contrary to our expectations, lower volume seems to have the opposite effect on likelihood of activist involvement, as the coefficient of *lnVol* is negative (-1.1404) and statistically significant. Coefficient of Tonal uncertainty (*Tone*), although non-negligible at 0.239 and in the expected direction, has not proven to be statistically significant.

With respect to marginal effects, if all other variables are held at their means, the probability of becoming a campaign target increases by 0.31% to 0.53% for M-Score and from 0.35% to 0.45% for Duality. If firm-quarter Spread values shift from the first four quintiles into the last one, the likelihood of being targeted increase from 0.32% to 0.58%. In comparison to overvaluation features, marginal effects of ambiguity were lower on average, suggesting possibly lower increases in probability of activist involvement on the basis of selected ambiguity parameters. However, interaction between ambiguity variables was observed, as our test for combined effects suggests a cumulative increase in targeting probability from 0.246% to 1.67% when all of the six selected ambiguity variables shift to the top quintile at once. This represents substantially higher increase than in the case of similar combined effect of overvaluation determinants, in which the interaction appears to be acting in the opposite direction (increase from 0.04% to 0.41%).

Lastly, the two determinants describing analyst coverage and dispersion of analyst recommendations were analyzed on a reduced sample size of 9,378 observations. Neither coefficient of the two variables was in line with our expectations as the coefficient of *AnnDisp*, at -2.010, was negative and statistically significant. Similarly, number of analysts covering the stock was inversely related to increase in probability of activist coverage. Possible explanation of this outcome is that activists rarely target stocks with

non-existent or low analyst coverage and select companies with overall positive coverage for maximum surprise effect. Nevertheless, more rigorous test of this conjecture remains out of scope of this thesis. The coefficients of control variables confirm previous results, as Leverage (D/A) remains the most significant determinant.

6.4 Effect of Overvaluation and Ambiguity on Returns

As shown in the first section of this chapter, targeted companies generate significantly negative abnormal returns throughout a period of one year following the activist campaign announcement. Panel A of Table 8 reiterates the findings of outsized negative market reactions and, moreover, separates abnormal returns into quintiles for each period window. Panel B and C present regression results for Ambiguity and Overvaluation feature sets on individual determinant basis and grouped into two artificial groups, allowing for comparison of the two sets.

In Panel B, we evaluate determinants on individual basis. The only statistically significant determinants in the ambiguity set with negative coefficients in the short-term (announcement day to 1 month after the announcement) are *Spread* and *Big4*, with the former ranging from -0.0705 to -0.1377. Big 4 (indicator of 0 or 1) proved to be especially economically significant, as the coefficients ranged from -0.06 to -0.14 at the 1% level. Coefficients of Price to Book variable remained significantly positive at the 5% level throughout the short term, supporting our supposition of decreasing P/B ratios being in line with decreasing returns. The changes in abnormal returns attributable to changes in P/B are, nevertheless, small in magnitude as the coefficients range between 0.0051 and 0.0072 on the day of announcement and after one month, respectively.

In Panel C, we present the results for the feature sets combined into two variables. These variables are composed as averages of their binary constituents taken from each observed quarter. Consequently, the values of their parts present a relative placement of targeted firm-quarters in either the first four or the top quintile of the determinant distribution in the respective quarter. The coefficient on the Ambiguity variable is significantly negative, with statistical significance declining from the 1% level on the second day after the announcement to the 10% level for a quarter since the report publication. Coefficient values increase from -0,091 for AR(0) to -0.65 for CAR(qtr.), after which the coefficient loses its significance. The coefficient on Overvaluation gains

significance on the fifth day, CAR(0,5), at a value of -0.098 and continues to increase in significance and magnitude until the last observed period.

The observations in Panel C lend a broad support for hypotheses 4 and 5. The impact of activist campaigns for firms scoring high on the ambiguity scale decreases in comparison to those scoring high on overvaluation scale throughout the observed period and vice versa for overvaluation set. On the day of announcement, a firm with full ambiguity score (*Ambiguity*= 1, i.e. last quintiles of each ambiguity parameter) would return 0.091 less than its counterpart with no ambiguity. Whereas a firm with full overvaluation would increase by 0.15% on average. Coefficient on Ambiguity loses its primacy in terms of magnitude of effect over Overvaluation coefficient in one month after the announcement (ambiguity coefficient at -0.27 and Overvaluation coefficient at -0.51). Since one month after the report, the divide between these coefficients increases gradually until the last observation period of one year, in which the overvaluation coefficient assumes a value 2.53 times lower than the ambiguity coefficient. Concludingly, the regression results provide support for H5 regarding overvaluation features' increasing effect on abnormal returns relative to ambiguity features as the period window following the announcement extends further. Conversely, the observations support the notion of H4, that ambiguity features have comparably larger effect on abnormal returns in the short term compared to overvaluation features.

7 Conclusion and Suggestion for Future Research

This thesis attempts to expand on existing literature on short-selling with active approach, a short-selling discipline that has seen a noticeable increase in popularity in the last decade. Economic impact of activist short-seller campaigns and a range of determinants that might attract activists were examined by means of analyzing a sample of 239 activist campaigns in the period of 2011 to 2019. By focusing on activists that have a proven history of successful campaigns and demonstrated interest in continuing activity, I implicitly selected for activists with considerable reputation at stake and clearly defined incentives. I show that activist campaigns indeed lead to significantly negative market reactions in the short term as well as in the long term when compared to industry peers. The severity of negative abnormal returns in the short term is most pronounced for companies scoring high on the ambiguity scale. The economic effect of overvaluation becomes more important as the period after the campaign announcement extends beyond one trading month, while the significance of ambiguity decreases. Increased importance of ambiguity in the short term is in accordance with Fox and Tversky's (1995) comparative ignorance hypothesis, which presumes higher ambiguity aversion in face of more knowledgeable individual. In our case, a reputable activist acts as the more knowledgeable party with express intent to corrode investors' understanding of underlying uncertainty, thus possibly causing attrition among company's shareholders.

Moreover, I find partial support for the hypothesized determinants of activist involvement. Companies exhibiting overvaluation and ambiguity are more likely to be targeted by short-seller activists, especially when scoring high on multiple ambiguity parameters at once. With respect to overvaluation parameters, price to earnings and debt coverage proved to be the most attractive determinants for a short-seller. In the ambiguity set, the two most important determinants were information asymmetry captured by bid-ask spread and manipulation score, as defined by Beneish (1999).

Given our approach to data collection and extraction, the thesis is limited to financial data that is available from annual and quarterly filings or made available by online market data providers. Due to the size of our targeted and non-targeted sample, in-depth analysis of off-balance sheet items and inconsistencies in management outlook

remained out of scope. Likewise, no extensive analysis of the overall tone and claims presented in short-seller reports were included in this thesis.

With respect to our limitations, several possible areas for future research of determinants of activist involvement and relevant factors in subsequent market reaction emerge. One of such pertains to proximity between managers and board of directors or other members of the executive leadership, an extension of management duality presented in this thesis. Alternatively, the determinants set studied in this thesis could be enhanced by detailed examination of CEO's past activities, such as the number of companies the CEO managed that ended up filing for bankruptcy, or CEO's past criminal convictions. Lastly, the research of activist short-sellers' modus operandi could be expanded by a study of identifiers of an impending campaign. Number of FOIA requests, analysis of Google search trends (such as patents and company-specific operations), changes in open interest and put option volume appear as especially interesting candidates for future research.

References

Abdi, F., & Ranaldo, A. (2017). A simple estimation of bid-ask spreads from daily close, high, and low prices. *The Review of Financial Studies*, *30*(12), 4437-4480.

Ackert, L. F., & Athanassakos, G. (1997). Prior uncertainty, analyst bias, and subsequent abnormal returns. *Journal of Financial Research*, *20*(2), 263-273.

Altman, E. I. (2013). Predicting financial distress of companies: revisiting the z-score and Zeta® models. 2000. *Stern School of Business, New York: official site. URL: http://pages. stern. nyu. edu/~ ealtman/Zscores. pdf*.

Andrew Left, October 11, 2018, interview with Keith McCullough , transcript available at: https://app.hedgeye.com/insights/70841-icymi-andrew-left-on-the-art-of-short-selling-frauds-fads-fail?type=macro

Ashraf, R. (2017). Scraping EDGAR with python. *Journal of Education for Business*, *92*(4), 179-185.

Barua, S., & Saha, A. K. (2015). Traditional Ratios vs. Cash Flow based Ratios: Which One is Better Performance Indicator?. *Advances in Economics and Business*, *3*(6), 232-251.

Benabou, R., & Laroque, G. (1992). Using privileged information to manipulate markets: Insiders, gurus, and credibility. The Quarterly Journal of Economics, 107(3), 921-958.

Beneish, M. D. (1999). The detection of earnings manipulation. *Financial Analysts Journal*, *55*(5), 24-36.

Beneish, M. D., & Nichols, C. (2009). Identifying overvalued equity. *Johnson School Research Paper Series*, (09-09).

Beneish, M. D., Lee, C. M., & Nichols, D. C. (2013). Earnings manipulation and expected returns. *Financial Analysts Journal*, *69*(2), 57-82.

Boehmer, E., & Wu, J. (2012). Short selling and the price discovery process. The Review of Financial Studies, 26(2), 287-322.

Boehmer, E., Jones, C. M., & Zhang, X. (2013). Shackling short sellers: The 2008 shorting ban. The Review of Financial Studies, 26(6), 1363-1400.

Boone, J. P., Khurana, I. K., & Raman, K. K. (2010). Do the Big 4 and the second-tier firms provide audits of similar quality?. *Journal of accounting and public policy*, *29*(4), 330-352.

Boyd, R. July, 9 2019, interview with QTR, available at https://quoththeraven.podbean.com/e/quoth-the-raven-129-roddy-boyd/

Campbell, Y.J., & Schiller, R. J. (1998). Valuation Ratios and the Long-Run Stock Market Outlook. The Journal of Portfolio Management Jan 1998, 24 (2) 11-26

Campbell, J. Y., & Shiller, R. J. (2001). *Valuation ratios and the long-run stock market outlook: An update* (No. w8221). National bureau of economic research.

Chan, K., Chan, L. K., Jegadeesh, N., & Lakonishok, J. (2001). *Earnings quality and stock returns* (No. w8308). National bureau of economic research.

Chang, E. C., Cheng, J. W., & Yu, Y. (2007). Short-sales constraints and price discovery: Evidence from the Hong Kong market. *The Journal of Finance*, *62*(5), 2097-2121.

Chang, X., Dasgupta, S., & Hilary, G. (2006). Analyst coverage and financing decisions. *The Journal of Finance*, *61*(6), 3009-3048.

Charoenrook, A., & Daouk, H. (2009). A study of market-wide short-selling restrictions (No. 642-2016-44312).

Chen, K. C., Cheng, Q., Lin, Y. C., Lin, Y. C., & Xiao, X. (2016). Financial reporting quality of Chinese reverse merger firms: The reverse merger effect or the weak country effect?. The Accounting Review, 91(5), 1363-1390.

Christophe, S. E., Ferri, M. G., & Angel, J. J. (2004). Short-selling prior to earnings announcements. *The Journal of Finance*, *59*(4), 1845-1876.

Copeland, T. E., & Galai, D. (1983). Information effects on the bid-ask spread. *the Journal of Finance*, *38*(5), 1457-1469.

Dan David, March 2016, China Hustle Briefing Book, available at: http://stopthechinahustle.org/PDFs/China_Hustle_Briefing-Book_website.pdf

David Kestenbaum, January 29, 2015, interview with Robert Siegel, Available at https://www.npr.org/2015/01/29/382463501/the-spicy-history-of-short-selling-stocks?t=1573515797081

Dechow, P. M., Hutton, A. P., Meulbroek, L., & Sloan, R. G. (2001). Short-sellers, fundamental analysis, and stock returns. Journal of financial Economics, 61(1), 77-106.

Doukas, J. A., & McKnight, P. J. (2005). European momentum strategies, information diffusion, and investor conservatism. *European Financial Management*, *11*(3), 313-338.

Dunn, P. (2004). The impact of insider power on fraudulent financial reporting. *Journal of management*, *30*(3), 397-412.

Eavis, P (March 28, 2016). Valeant's accounting error a warning sign of bigger problems. The New York Times (available at : https://cutt.ly/trve0vs)

Efendi, J., & Swanson, E. P. (2009). Short seller trading in companies with a severe accounting irregularity. Available at SSRN 1465156.

Engelberg, J. E., Reed, A. V., & Ringgenberg, M. C. (2018). Short-selling risk. *The Journal of Finance*, *73*(2), 755-786.

Epstein, L. G., & Schneider, M. (2008). Ambiguity, information quality, and asset pricing. *The Journal of Finance*, *63*(1), 197-228.

Fama, E. F., & French, K. R. (1995). Size and book-to-market factors in earnings and returns. *The journal of finance*, *50*(1), 131-155.

Fang, V. W., Huang, A. H., & Karpoff, J. M. (2016). Short selling and earnings management: A controlled experiment. The Journal of Finance, 71(3), 1251-1294.

Fox, C. R., & Tversky, A. (1995). Ambiguity aversion and comparative ignorance. *The quarterly journal of economics*, *110*(3), 585-603.

Gillet, R., & Renault, T. (2019). When machines read the Web: market efficiency and costly information acquisition at the intraday level. Finance, 40(2), 7-49.

Gillis, P. Dec 13, 2012, The theory of holes, available at: https://www.chinaaccountingblog.com/weblog/the-theory-of-holes.html

Goyal, V. K., & Park, C. W. (2002). Board leadership structure and CEO turnover. *Journal of Corporate Finance*, *8*(1), 49-66.

Groysberg, B., Healy, P. M., & Maber, D. A. (2011). What drives sell-side analyst compensation at high-status investment banks?. *Journal of Accounting Research*, *49*(4), 969-1000.

Grullon, G., Michenaud, S., & Weston, J. P. (2015). The real effects of short-selling constraints. The Review of Financial Studies, 28(6), 1737-1767.

Hayes, R. M. (1998). The impact of trading commission incentives on analysts' stock coverage decisions and earnings forecasts. *Journal of Accounting Research*, *36*(2), 299-320.

Hobbs, J., & Singh, V. (2015). A comparison of buy-side and sell-side analysts. *Review of Financial Economics*, *24*, 42-51

Hong, H., and J. Stein. 2003. Differences of Opinion, Short-Sales Constraints, and Market Crashes. Review of Financial Studies 16:487–525

Hong, H., Lim, T., & Stein, J. C. (2000). Bad news travels slowly: Size, analyst coverage, and the profitability of momentum strategies. *The Journal of Finance*, *55*(1), 265-295.

Hribar, P., & Collins, D. W. (2002). Errors in estimating accruals: Implications for empirical research. *Journal of Accounting research*, *40*(1), 105-134.

Jegadeesh, N. (1990). Evidence of predictable behavior of security returns. *The Journal of finance*, *45*(3), 881-898.

Jegadeesh, N., & Titman, S. (2011). Momentum. *Annu. Rev. Financ. Econ.*, *3*(1), 493-509.

Jensen, M. C. (1986). Agency costs of free cash flow, corporate finance, and takeovers. *The American economic review*, *76*(2), 323-329.

Jensen, M. C. (1993). The modern industrial revolution, exit, and the failure of internal control systems. *the Journal of Finance*, *48*(3), 831-880.

Jensen, M. C. (2005). Agency costs of overvalued equity. *Financial management*, *34*(1), 5-19.

Jensen, M. C., & Meckling, W. H. (1976). Theory of the firm: Managerial behavior, agency costs and ownership structure. *Journal of financial economics*, *3*(4), 305-360.

Kamarudin, K. A., Ismail, W. A. W., & Samsuddin, M. E. (2012). The influence of CEO duality on the relationship between audit committee independence and earnings quality. *Procedia-Social and Behavioral Sciences*, *65*, 919-924.

Karpoff, J. M., & Lou, X. (2010). Short sellers and financial misconduct. The Journal of Finance, 65(5), 1879-1913.

Khan, M., & Lu, H. (2013). Do short sellers front-run insider sales?. *The Accounting Review*, *88*(5), 1743-1768.

Kovbasyuk, S., and M. Pagano. 2014. Advertising arbitrage. Working paper, University of Naples.

Lamont, O. A. (2012). Go down fighting: Short sellers vs. firms. The Review of Asset Pricing Studies, 2(1), 1-30.

Lamont, O. A., & Thaler, R. H. (2003). Can the market add and subtract? Mispricing in tech stock carve-outs. Journal of Political Economy, 111(2), 227-268.

Leuz, C., & Verrecchia, R. E. (2000). The economic consequences of increased disclosure. *Journal of accounting research*, 91-124.

Ljungqvist, A., & Qian, W. (2016). How constraining are limits to arbitrage?. The Review of Financial Studies, 29(8), 1975-2028.

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. The Journal of Finance, 66(1), 35-65.

Massa, M., Zhang, B., & Zhang, H. (2015). The invisible hand of short selling: Does short selling discipline earnings management?. The Review of Financial Studies, 28(6), 1701-1736.

McCormick, M and Fletcher, L, in Financial Times August 2019, Burford alleges market manipulation in trading of its shares. Available at https://www.ft.com/content/c9f80ed8-bcc7-11e9-b350-db00d509634e

McKenna, F. 2016, Market Watch, After China fraud boom, Nasdaq steps up scrutiny of shady listings, Available at https://www.marketwatch.com/story/after-china-fraud-boom-nasdaq-steps-up-scrutiny-of-shady-listings-2016-06-20

Meneghetti, C., Williams, R., & Xiao, S. C. Efficient Governance, Inefficient Markets: Short Selling with Takeover Risk.

Miller, E. M. (1977). Risk, Uncertainty, and Divergence of Opinion. The Journal of Finance, 32(4), 1151. doi:10.2307/2326520

Mitts, J. (2019). Short and distort. *Columbia Law and Economics Working Paper*, (592).

Opler, T. C., & Titman, S. (1994). Financial distress and corporate performance. *The Journal of finance*, *49*(3), 1015-1040.

Peng, S. S., & Bewley, K. (2009). Adaptability of fair value accounting in China: Assessment of an emerging economy converging with IFRS. In CAAA annual conference.

Rapach, D. E., Ringgenberg, M. C., & Zhou, G. (2016). Short interest and aggregate stock returns. *Journal of Financial Economics, 121*(1), 46-65.

Richardson, S. (2006). Over-investment of free cash flow. *Review of accounting studies, 11*(2-3), 159-189.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53-65.

Ryan, P., & Taffler, R. J. (2001). What is news? What firm-specific information releases drive market prices?., , Paper read at the INQUIRE UK Autumn Seminar 23-25 September 2001

Scherbina, A. (2007). Suppressed negative information and future underperformance. *Review of Finance*, *12*(3), 533-565.

Shi, W., Connelly, B. L., & Cirik, K. (2018). Short seller influence on firm growth: A threat rigidity perspective. Academy of Management Journal, 61(5), 1892-1919

Shkilko, A., Van Ness, B., & Van Ness, R. (2008). Price-destabilizing short selling. In AFA 2008 New Orleans Meetings Paper.

Shleifer, A., & Vishny, R. W. (1989). Management entrenchment: The case of manager-specific investments. *Journal of financial economics, 25*(1), 123-139.

Stiner, F. M., & Lynn, S. A. (2012). Auditing Issues with Chinese Reverse Merger Companies Traded in the United States. International Journal of Accounting and Financial Reporting, 2(2), 76

Templin, B. A. (2011). Chinese reverse mergers, accounting regimes, and the rule of law in China. T. Jefferson L. Rev., 34, 119.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, *62*(3), 1139-1168.

Tetlock, P. C., Saar-Tsechansky, M., & Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, *63*(3), 1437-1467.

Uygur, O. (2018). CEO ability and corporate opacity. *Global Finance Journal*, *35*, 72-81.

Wang, X., & Wu, M. (2011). The quality of financial reporting in China: An examination from an accounting restatement perspective. China Journal of Accounting Research, 4(4), 167-196.

Williams, G. 2017 on Real Vision, Grant Williams in conversation with Marc Cohodes. Available at https://www.realvision.com/tv/shows/grant-williams/videos/grant-williams-in-conversation-with-marc-cohodes

Xie, H. (2001). The mispricing of abnormal accruals. *The accounting review*, *76*(3), 357-373.

Yeo, G. H., Tan, P. M., Ho, K. W., & Chen, S. S. (2002). Corporate ownership structure and the informativeness of earnings. *Journal of Business Finance & Accounting*, *29*(7-8), 1023-1046.

Zeller, T. L., & Stanko, B. B. (1994). Operating cash flow ratios measure a retail firms ability to pay. *Journal of Applied Business Research (JABR), 10*(4), 51-59.

Zhang, X. F. (2006). Information uncertainty and stock returns. *The Journal of Finance*, *61*(1), 105-137.

Zhao, W. (2018). Activist short-selling (Doctoral dissertation).

Zhao, W. (2019). Activist short-selling and corporate opacity. Available at SSRN 2852041.

Zhou, R. T., & Lai, R. N. (2009). Herding and information based trading. Journal of Empirical Finance, 16(3), 388-393.

Appendices

Appendix A: Recurring Themes in Short-Seller Research Reports

This appendix showcases summaries of campaigns launched against companies allegedly involved in corporate malfeasance or fraud. Given the heterogeneity of short-selling campaigns, these examples are by no means exhaustive in encompassing all points of critique. They, nevertheless, provide an overview of reoccurring themes in short-seller reports.

Panel A: Excerpt from Muddy Waters Research on Sino Forrest

Strong allegations including fraud and fictitious revenue. One of early Chinese RM shorts.

Company:
Sino-Forest Corporation
(TRE.TO, OTC: SNOFF)

Industry:
Forestry

Recommendation:
Strong Sell

Estimated Value:
< $1.00

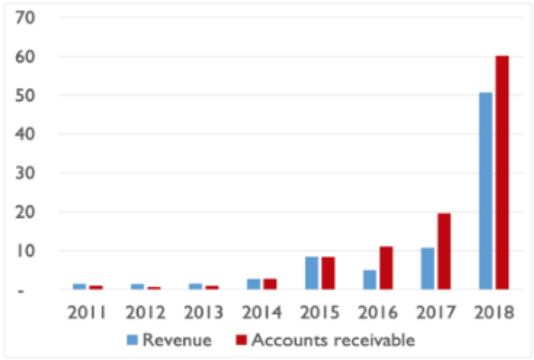
Report Date:
June 2, 2011

- Like Madoff, TRE is one of the rare frauds that is committed by an established institution. In TRE's case, its early start as an RTO fraud, luck, and deft navigation enabled it to grow into an institution whose "quality management" consistently delivered on earnings growth.
- TRE, which was probably conceived as another short-lived Canadian-listed resources pump and dump, was aggressively committing fraud since its RTO in 1995.
- The foundation of TRE's fraud is its convoluted structure whereby it runs most of its revenues through "authorized intermediaries" ("AI"). AIs supposedly process TRE's tax payments, which ensures that TRE leaves its auditors far less of a paper trail.
- On the other side of its books, TRE massively exaggerates its assets. We present smoking gun evidence that TRE overstated its Yunnan timber investments by approximately $900 million.

Figure 1 Excerpt from Short Selling Thesis by Muddy Waters, June 2, 2011[37]

Panel B: Quintessential Capital Management on Bio-On S.A.

One of the most impactful campaigns of 2019, leading to arrest of the CEO and bankruptcy of the company. Allegations of fabricated revenues and fraudulent accounting schemes.



The accounting audit we carried out also shows violations of Articles 2343 and 2343 of the Italian Civil Code regarding the sale and awarding of licenses. In fact, Bio-on should have followed a certain procedure in assigning the value of the licenses sold to its affiliates, which instead appear to have received an arbitrary value.

The value of Bio-on's "sales" and "credits" soared in 2018, as several new shell companies were formed that "acquired" Bio-on licenses at an exponential rate (as usual, such sales were not followed by payments).

Figure 2 Excerpt from Short Selling Thesis by Quintessential Capital, July 19, 2019[38].

[37] The thesis can be found under the following link: https://cutt.ly/9rEDYQf
[38] Full report available at https://cutt.ly/jrEDQxh

Panel C: Blue Orca Capital on Samsonite

Highly impactful research thesis delivered by Soren Aandahl of Blue Orca Capital.

Allegations of overly aggressive acquisitions to inflate revenues, unlawful use of doctoral title by the CEO and vast undisclosed related party transactions with companies under either direct control of the CEO or closely affiliated persons.

Web of Related Party Transactions. Samsonite engages in a number of related party transactions with entities owned and controlled by its CEO and his family members. Such transactions are unbecoming of a global, professional organization and raise a number of troubling questions about Samsonite's corporate governance structure and the sufficiency of its internal controls.

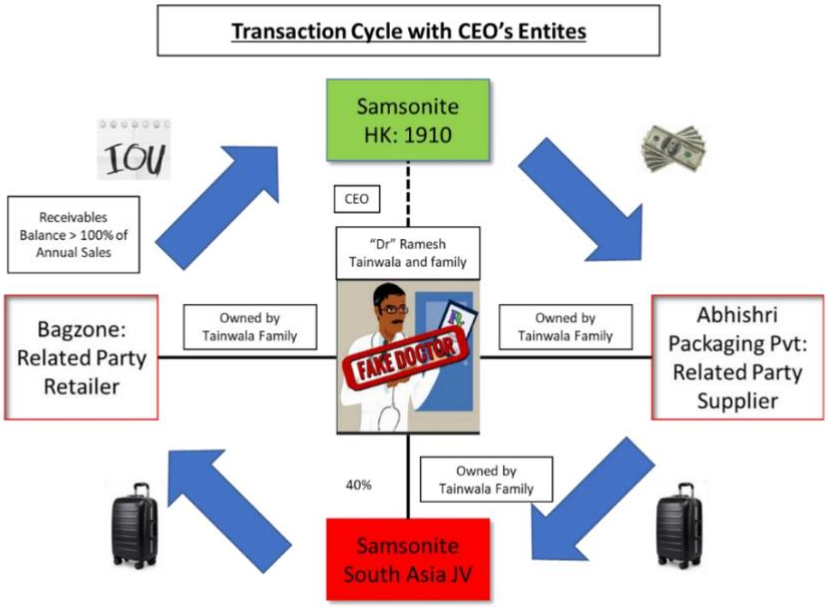


Figure 3 Excerpt from Short Selling Thesis by Blue Orca Capital, May 24, 2018[39]

[39] Research thesis available at (Blue Orca), https://cutt.ly/urEDc6L

Appendix B: Short Sellers included in the sample

Short Seller	Total Number of campaigns	Active period	Median Target Market Capitalization (mil $)
Muddy Waters Capital	28	2010 to 2019	1906,5
Spruce Point Capital	55	2010 to 2019	1277,2
Citron Research	61	2010 to 2019	2118,2
Kerrisdale Capital Partners	29	2010 to 2019	937,2
Prescience Point Capital	17	2015 to 2019	758,2
Friendly Bear/Orso	19	2017 to 2019	1415,4
Hindenburg Research	17	2011 to 2019	81,9
Blue Orca Capital	7	2011 to 2014	1673,5
Glaucus Research	6	2015 to 2019	1373,0
Marcus Aurelius	13	2014 to 2019	731,7
Glasshouse Research	6	2016 to 2018	3494,8
White Diamond	40	2017 to 2018	89,7
Viceroy Research	6	2011 to 2018	2399,5
Mox Reports	61	2012 to 2018	365,5
Copperfield Research	14	2011 to 2018	553,1
Fuzzy Panda Research	5	2018 to 2018	270,4
Gotham City Research	6	2013 to 2018	1709,2
Unemon	9	2017 to 2018	330,6
Total	399		
Of which selected for sample study:	239		

Year	Number of campaigns	Distribution
2011	6	2,5%
2012	20	8,4%
2013	27	11,3%
2014	18	7,5%
2015	27	11,3%
2016	39	16,3%
2017	45	18,8%
2018	49	20,5%
2019	8	3,3%
Total:	239[40]	100,0%

To have a full trading year of data following a campaign announcement, only January was considered for 2019.

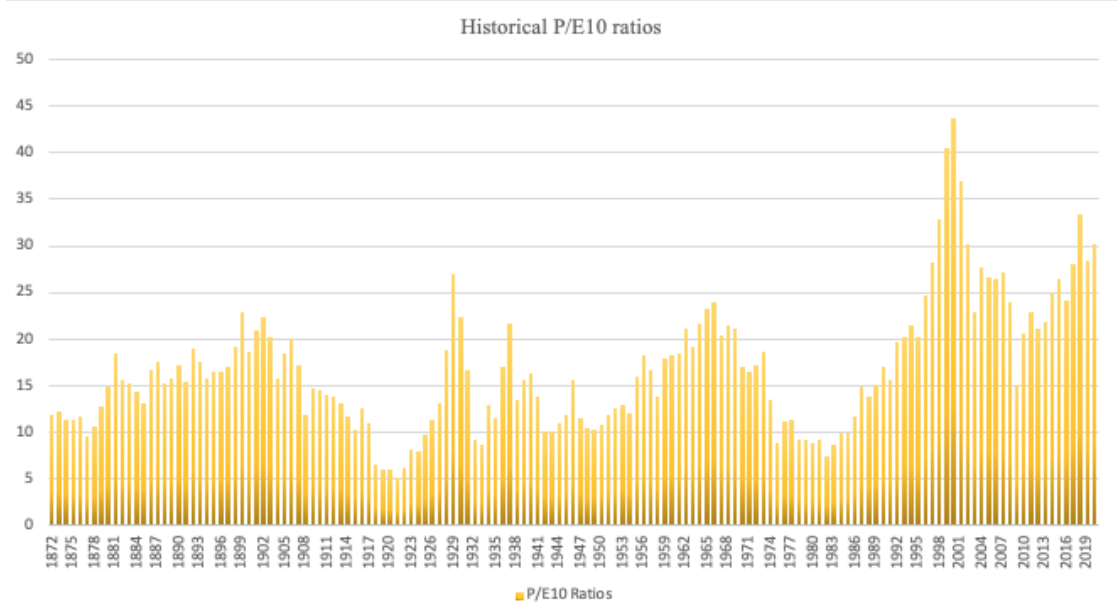
[40] The total number of campaigns was reduced from the original amount to control for multiple campaigns against a single firm. Only the earliest campaign is considered, regardless of activist's prominence.

Appendix C: Overview of targeted industries

French Fama 48 Industry	Targets	Avg. Market Capitalization (in $ mil.)	Industry Peers (#)
1 Agriculture	1	1 372	23
2 Food Products	8	3 841	45
3 Candy & Soda	1	29 554	8
4 Beer & Liquor	1	214	18
5 Tobacco Products	1	335	13
6 Toys Recreation	1	263	23
7 Entertainment	4	24 066	30
9 Consumer Goods	4	426	58
11 Healthcare	4	2 504	26
12 Medical Equipment	19	3 272	175
13 Drugs Pharmaceutical Products	39	6 968	242
14 Chemicals	9	306	97
17 Construction Materials	1	1 223	46
18 Construction	2	4 683	20
19 Steel Works etc.	6	668	37
21 Machinery	3	317	104
22 Electrical Equipment	2	6 153	49
23 Automobiles and Trucks	3	2 871	69
24 Aircraft	3	5 136	17
26 Guns Defense	2	1 539	10
27 Precious Metals	1	82	34
28 Non-Metallic, Industrial Metal Mining	1	235	88
30 Petroleum and Natural Gas	4	107	158
32 Communication	6	5 162	73
33 Personal Services	4	316	57
34 Business Services	40	2 696	370
35 Hardware Computers	3	2 469	53
36 Computer Software	17	7 806	199
37 Electronic Equipment	3	331	78
38 Measuring and Control Equipment	1	44	16
39 Business Supplies	1	1 112	4
40 Shipping Containers	4	3 846	37
41 Transportation	5	4 319	117
42 Wholesale	12	5 757	105
43 Retail	4	398	61
44 Restaurants, Hotels, Motels	4	2 321	249
45 Banking	5	1 152	106
46 Insurance	3	1 941	69
47 Real Estate	7	7 401	170
Total:	239	Total number of peers:	3154
Average Market Capitalization:		4,284.17 M	
Median Market Capitalization:		729.18 M	

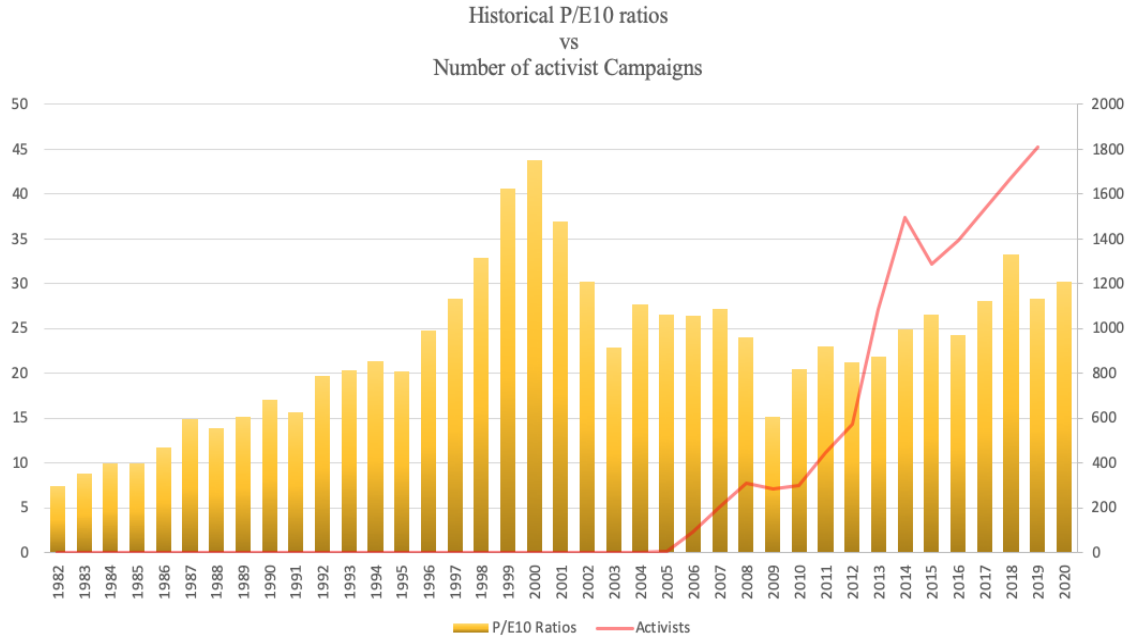
Appendix D: Historical P/E ratios

Figure 4, Charts corresponding to Cyclically Adjusted Schiller-PE ratios based on average inflation-adjusted earnings from the previous 10 years for the period of 1872 to 2019.



Source: https://www.multpl.com/shiller-pe

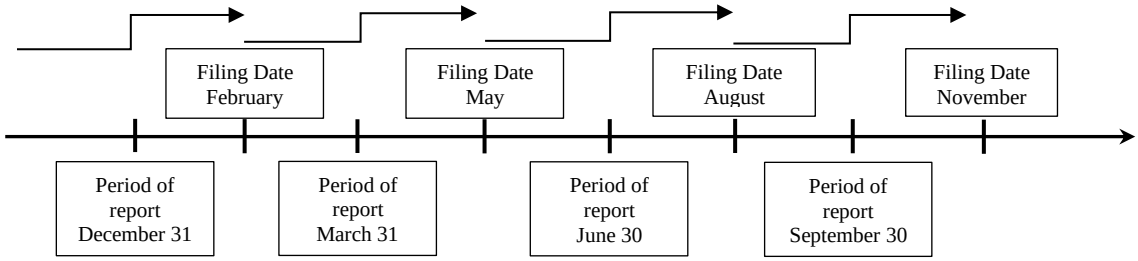
Figure 5, PE ratios compared to number of observed activist campaigns in 2005 to 2015. Number of campaigns beyond 2015 are estimates based on extrapolation from historical trend.



Source of P/E data. https://www.multpl.com/shiller-pe, Source of activist campaign numbers Zhao (2018)

Appendix E: Selection of time period for financial statement data

Due to differences in reporting periods chosen by US-listed companies included in our sample, it is required that we organize reporting dates of studied financial statement data in such manner, that 10-Q and 10-K filings are available at the time of activist campaign announcement. Given our different approach to collecting data, each datapoint we have extracted was recorded for its actual reporting period chosen by the reporting company. As a consequence of discrepancies between the reporting period and the filing date, 45 market days (approximately 2 months) were added to the reporting date as a threshold for data to be considered. For illustration, if an activist campaign was announced in the period between March 31 (10-Q) and the end of May, data considered in the model will be taken from company financial statements filed in February (10-K), and if the short-seller research is published in July, data from May filings will be considered. In this fashion, it is assured that the activist had all studied data at his disposal at the time of campaign announcement.



Unfortunately, this approach still remains a mere approximation, as the completion of an activist thesis oftentimes spans across a period of several months and may thus not be initially motivated by the most recent quarterly report.

Appendix F: Overview of model variables

Model variables not described to their full extent in Chapter 4.

Panel A: Beneish MSCORE (1999):

*MSCORE= -4.84 + .920*DSR + .528*GMI + .404*AQI + .892*SGI + .115*DEPI- .172*SGAI +4.679*TATA - .327*LEVI*

Days Sales Receivable (DSR) $= \frac{Receivables_t}{Sales_t} / \frac{Receivables_{t-1}}{Sales_{t-1}}$
$Gross\ Margin\ Index\ (GMI) = 1 - \frac{COGS_t}{Sales_t} / 1 - \frac{COGS_{t-1}}{Sales_{t-1}}$
$Asset\ Quality\ Index\ (AQI) = 1 - \frac{PPE_t + CurrentAssets_t}{TotalAssets_t} / 1 - \frac{PPE_{t-1} + CurrentAssets_{t-1}}{TotalAssets_{t-1}}$
$Sales\ Growth\ Index\ (SGI) = \frac{Sales_t}{Sales_{t-1}}$
$Depreciation\ Index\ (DEPI) = \frac{Depr_{t-1}}{Depr_{t-1} + PPE_{t-1}} / \frac{Depr_t}{Depr_t + PPE_t}$
$SG\&A\ Expenses\ Index\ (SGAI) = \frac{SGA_t}{Sales_t} / \frac{SGA_{t-1}}{Sales_{t-1}}$
$Accruals\ to\ Assets\ (TATA) = \frac{EBXI_t - CFO_t}{TotalAssets_t}$
$Leverage\ Index\ (LEVI) = \frac{Debt_t}{TotalAssets_t} / \frac{Debt_{t-1}}{TotalAssets_{t-1}}$

Where *t-1 stands* for the same quarter of previous fiscal year (t – 4 quarters), income statement and cash flow statement accounts are trailing last four quarters.

Panel B: Overinvestment of Free Cash Flow (Richardson, 2006), contin.:

$$I_{OverInv,t} = FCF_t - \Delta Equity_t - \Delta Debt_t - \Delta Financial\ Asset_t - OtherInv_t - Other_t$$

$$\Delta Equity_t = Purchase\ of\ Com\&pref.stock + Cash\ Dividends - Sales\ of\ Com\&pref.stock$$

$$\Delta Debt = LT\ Debt\ reduction - LT\ Debt\ Issuance - Changes\ in\ Current\ Debt$$

$$\Delta Fin.Asset_t = Increase\ in\ Cash\&Equiv - Change\ in\ ST\ investments$$

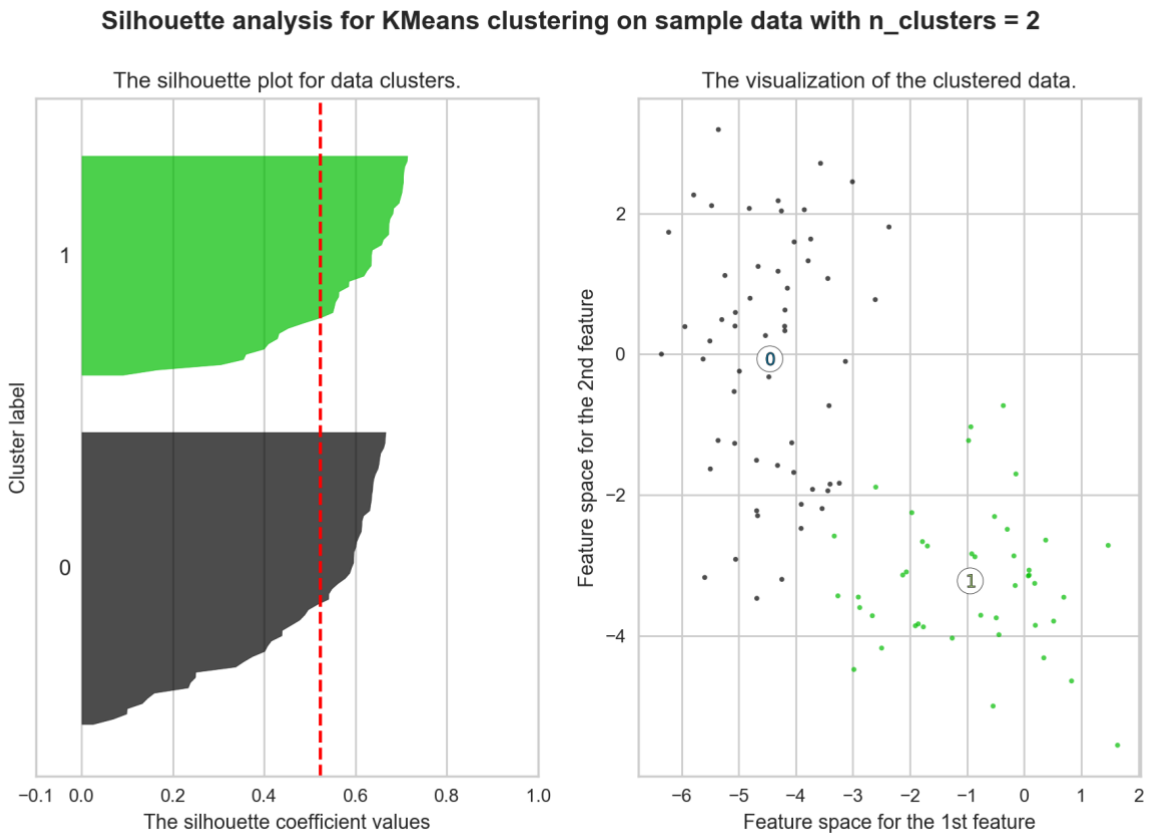
$$OtherInv_t = Increase\ in\ Investments - Sale\ of\ Investments$$

$$Other_t = -Other\ inv.activities - Other\ Financ.activities - Exchange\ Rate\ Effect$$

Appendix G: Data Clustering

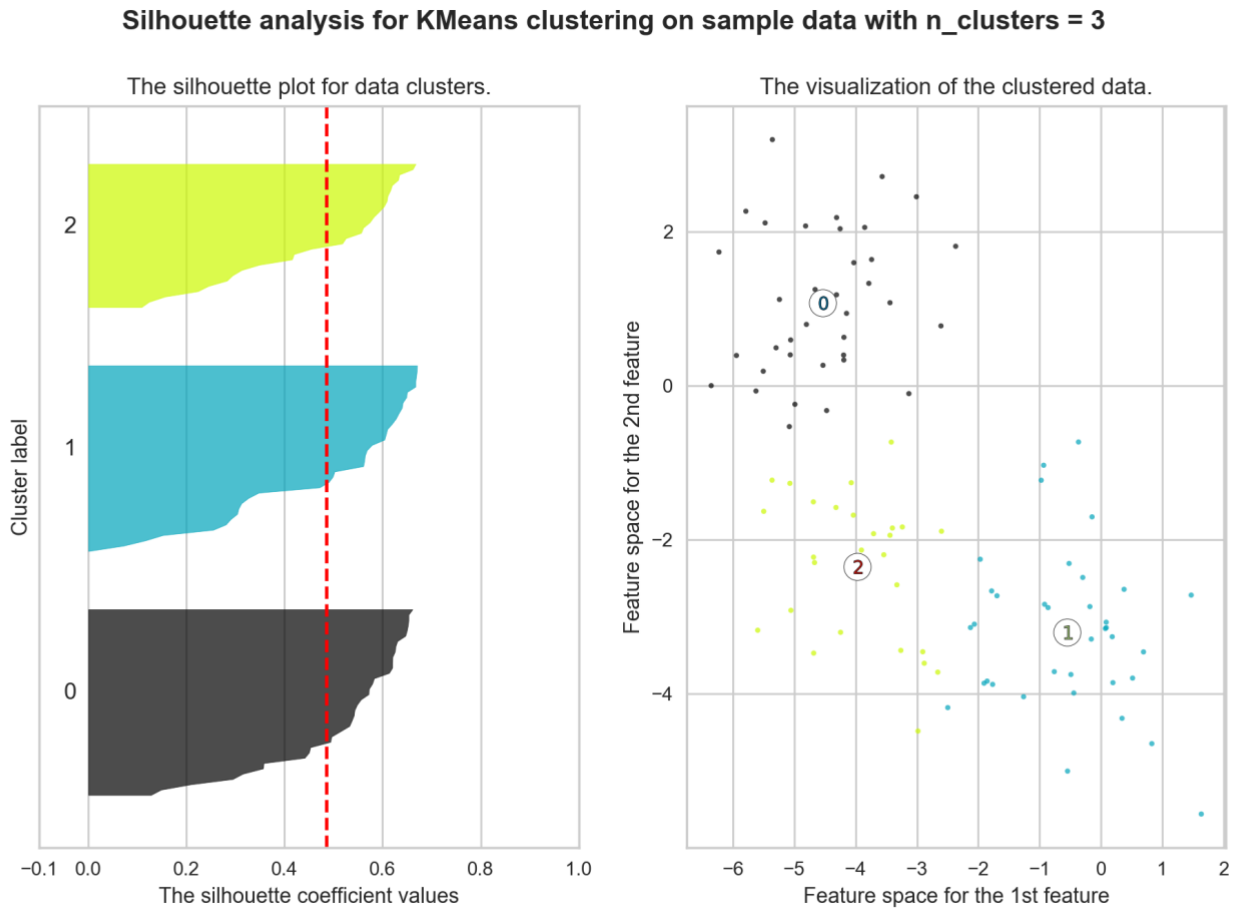
Panel A: Cluster Analysis of dataset with Silhouette method (k=2)

K-Means clustering applied iteratively to the entire dataset for k=2 to k=60



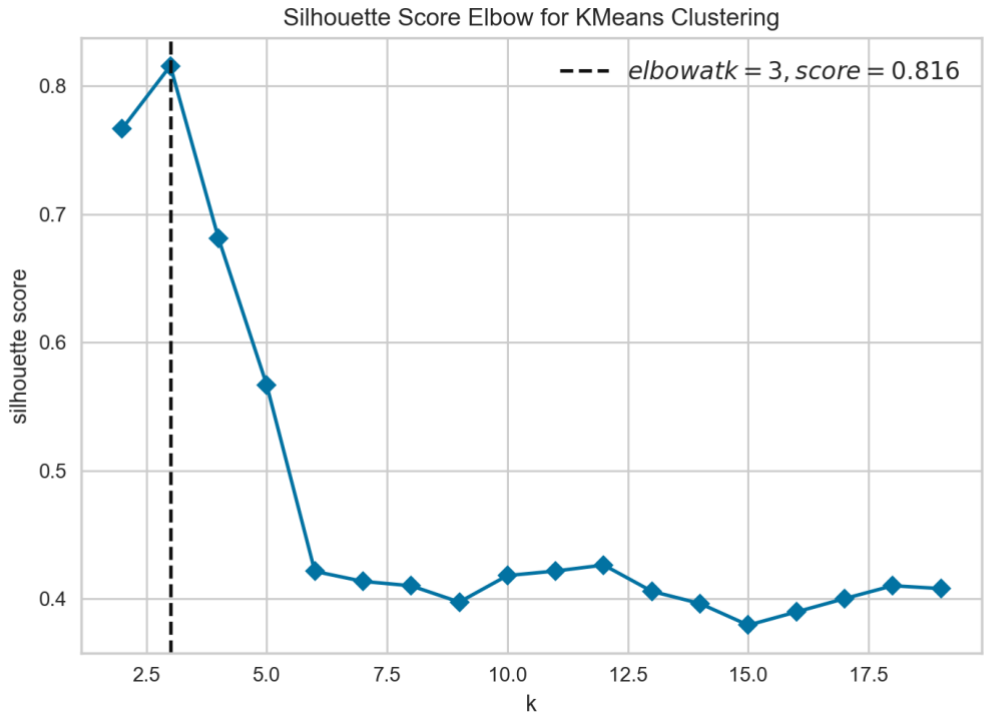
Panel B: Cluster Analysis of dataset with Silhouette method (k=3)

Sizes of the clusters are as follows: Cluster 0: 22,595, Cluster 1: 11,493, Cluster 2: 20,202



Panel C: Visualization of optimal number of clusters

The same unlabeled [41] dataset as in the previous graphics was used to plot silhouette score. Below is one of the iterations of the program[42].



[41] Unlabeled dataset, as in dataset with no indication on whether the firm-quarter was targeted by activist short-sellers or not, to avoid skewing group partitions due to characters other than those contained in financial statements.

[42] The clustering program was run several times to assure consistency with respect to optimal number of clusters. Each iteration returns slightly different result with respect to silhouette score, the inferences, nevertheless, remain unchanged at 3 cluster representing an optimal number of partitions.

Main Tables

Table 1: Comparison between hedge fund returns and major stock indices

Years, in %	2019	2018	2017	2016	2015	2014	2013	2012
EHFI251[43]	8.58	-3.98	8.62	4.76	2.38	5.15	9.12	7.40
HFRI Fund weighted composite index return[44][45]	10.35	-4.49	8.59	5.44	-1.12	2.98	9.13	6.36
S&P500	28.88	-6.24	19.42	9.54	-0.73	11.39	29.6	13.41
NASDAQ	35.23	-3.88	28.24	7.50	5.73	13.40	38.32	15.91
DJIA	22.34	-5.63	25.58	13.42	-2.23	7.52	26.50	7.26

Table 2: Language Processing of Annual Filings

Panel A: Occurrence of uncertainty terms in 10-K filings
A total of 24,110 filings were parsed for existence and frequency of 297 tonal words, suggesting elevated levels of uncertainty.

Filings period	2008 to 2018		
Total 10-K filings analyzed (N):	24 110		
Total Uncertainty words extracted:	14 277 877		
Mean occurrence per 10-K, 0 excluded:	578,14		
Occurrence	Most Frequent	Least Frequent	
Uncertainty Words	"may"	"abeyances", "arbitrariness"	
Total Occurrences	3 589 264	0	
No. of 10-K filings w/ the word	24 110	0	
	Mean	Median	Standard Deviation
Weights of uncertainty words	0,13517	0,07982	0,17128

[43] http://www.eurekahedge.com/Indices/IndexView//473/Eurekahedge_Hedge_Fund_Index
[44] https://www.hedgefundresearch.com/family-indices/hfri
[45] https://etfgi.com/news/press-releases/2019/10/etfgi-reports-assets-invested-global-etfs-industry-extended-lead-over

Panel B: Auditor company and CEO Duality.

Occurrence of one of the below terms in firm's 10-K filing results in the respective variable (*Big4, Duality*) assuming a value of 1, and 0 for Big4, respectively.

Auditors	CEO Duality
KPMG	Chief Executive Officer and Chairman
Pricewaterhouse	Chairman and Chief Executive Officer
Ernst Young	Chairman & CEO
Deloitte	Chairman & CEO
Waterhouse	Chairman of the Board, President and Chief Executive Officer
PWC	Chairman, Chief Executive Officer
Ernst&Young	Chairman of the Board and Chief Executive Officer
ernstyoung	of the Board and Chief Executive Officer
TOUCHE	Board and Chief Executive Officer
DELOITTE	Chief Executive Officer and Chairman of the Board of Directors
kpmg	Chairman of the Board and Chief Executive Officer
PricewaterhouseCoopers	Chairman and Chief Executive
pricewaterhouseCoopers	Chairperson and Chief Executive
ricewaterhousecoopers	Chairman, President and Chief Executive Officer
ErnstYoung	Chairman of the Board, Chief Executive Officer and President
Ernstyoung	Chairman of the Board, Chief Executive Officer
ERNST	Chairman, President, Chief Executive Officer
	Chief Executive Officer and Chairman of the Board
	Chairman of the Board, President and Chief Executive Officer
	Chief Executive Officer, President and Chairman

Table 3: Summary Statistics

Panel A: Summary of descriptive statistics for targeted and non-targeted firm quarters prior to variable binarization.

Variables	Fiscal Quarters, non-targets				Fiscal Quarters, targets				Diff. in means
	Obs.	Mean	Std.dev.	Median	Obs.	Mean	Std.dev.	Median	
lnSize	54 063	19,4390	3,2137	19,9104	228	19,8706	2,0441	19,7446	0,4315***
D/A	54 063	0,1888	0,2031	0,1275	228	0,2094	0,2538	0,1089	0,0206
dayStd	54 063	0,0431	0,0569	0,0243	228	0,0369	0,0304	0,0297	-0,0062***
P/E	54 063	11,1579	19,0195	12,1802	228	23,9161	92,8322	10,8264	12,7582**
P/B	54 063	2,3856	2,2827	1,6815	228	2,8668	59,1026	3,1301	0,4812
cROA	54 063	0,0685	0,3331	0,1353	228	-0,0653	0,3345	0,0498	-0,1338**
DebtC	54 063	0,0719	1,1122	0,1636	228	-0,9435	6,3899	0,1440	-1,0154**
TACC	54 063	-0,0764	0,1289	-0,0426	228	-0,1896	1,1277	-0,0543	-0,1133*
Momentum	54 063	0,1033	2,9166	0,0170	228	0,1538	0,5431	0,0491	0,0505*
Iover	54 063	0,0189	0,0913	0,0017	228	0,0622	0,1764	0,0067	0,0433***
Zscor	54 063	3,2018	3,4877	2,8862	228	7,7883	16,4311	3,4689	4,5866***
lnVol	54 063	14,2093	3,4885	14,7942	228	15,9681	1,9406	15,9397	1,7588***
Spread	54 063	0,3575	0,1778	0,3288	228	0,5141	0,4345	0,4239	0,1565***
Big4	54 063	0.51166	0,4982	1	228	0,5171	0,5003	1	0.00542
Mscor	54 063	-2,8733	1,5976	-2,6794	228	-1,5866	8,6225	-2,6124	1,2866**
Duality	54 063	0,3631	0,4809	0,0000	228	0,4018	0,4914	0,0000	0,0387
Tone	54 063	0,1976	0,3982	0,0000	228	0,2408	0,4287	0,0000	0,0432*
AnnDisp	9338	0,3465	0,2519	0,4421	40	0,3764	0,2588	0,4497	0,0299
LnAnn[46]	9338	1,4687	0,8582	1,6094	40	1,5475	1,0328	1,7918	0,0788

[46] The data for evaluating number of analysts covering a stock as well as dispersion of analyst ratings for given security were available only to a limited extent. Any analysis of their effect on returns/cumulative returns and their respective ramifications on firm's attractivity to short-seller activists will be limited to a period and companies for which the data is available.

Panel B: Summary statistics for targeted and non-targeted firm quarters after transformation of continuous variables to binary.

Variables of binary nature already provided in panel A are excluded, as are the control variables. Statistical significance of mean difference was tested with both t-test and Pearson Chi-square, with the latter confirming the outputs of the former (code for Chi-squared test in the Python code section). Statistical significance in presented tables follows conventional notation (*,**,*** for p<0.1, 0.05 , 0.01, respectively)

Variables	Fiscal Quarters, non-targets		Fiscal Quarters, targets		
	Obs.	Mean	Obs.	Mean	Mean Difference
P/E	54 063	0,1081	228	0,2929	0,1848***
P/B	54 063	0,1092	228	0,1297	0,0205
cROA	54 063	0,1393	228	0,2385	0,0992***
DebtC	54 063	0,1122	228	0,1799	0,0677***
TACC	54 063	0,1266	228	0,1757	0,0491**
Momentum	54 063	0,4532	228	0,4236	-0,0297
Iover	54 063	0,1386	228	0,3013	0,1627***
Zscor	54 063	0,0805	228	0,1255	0,0451**
lnVol	54 063	0,1444	228	0,0377	-0,1067***
Spread	54 063	0,1229	228	0,3556	0,2327***
Mscor	54 063	0,1142	228	0,2552	0,1411***
AnnDisp	9338	0,0218	40	0,0293	0,0075
LnAnn	9338	0,0263	40	0,0502	0,0239**

Panel C: Pearson correlation test for variables used for determining activist involvement

Bold font of characters in the below table indicates statistical significance of correlation coefficient at the 0.05 level.

	lnSize	*D/A*	*daySTD*	*P/E*	*P/B*	*cROA*	*DebtC*	*TACC*	*Momentum*	*IOver*	*Zscor*	*lnVol*	*Spread*	*Big4*	*Mscor*	*Duality*	*Tone*	*AnDisp*
D/A	**0,249**																	
daySTD	**-0,587**	**-0,077**																
P/E	**0,271**	**0,057**	**-0,220**															
P/B	**0,012**	**0,028**	**-0,066**	**0,051**														
cROA	**0,571**	**0,106**	**-0,412**	**0,300**	0,004													
DebtC	**0,373**	**-0,154**	**-0,255**	**0,221**	**0,022**	**0,537**												
TACC	**0,332**	**0,008**	**-0,272**	**0,221**	**-0,047**	**0,296**	**0,141**											
Momentum	**-0,039**	-0,006	0,006	-0,003	-0,008	**-0,021**	-0,006	0,004										
IOver	**0,104**	**0,064**	**-0,097**	**0,099**	**0,054**	**0,074**	**0,054**	**0,041**	-0,001									
Zscor	**0,071**	**-0,330**	**-0,198**	**0,214**	**0,116**	**0,257**	**0,285**	**0,210**	-0,001	**0,072**								
lnVol	**0,765**	**0,226**	**-0,524**	**0,185**	**0,143**	**0,320**	**0,223**	**0,129**	**-0,044**	**0,088**	**0,113**							
Spread	**0,310**	**0,092**	**-0,133**	**0,222**	**0,173**	**0,162**	**0,120**	**0,081**	-0,007	**0,103**	**0,209**	**0,460**						
Big4	**-0,575**	**-0,190**	**0,352**	**-0,164**	**-0,141**	**-0,291**	**-0,225**	**-0,162**	**0,020**	**-0,079**	**-0,098**	**-0,587**	**-0,294**					
Mscor	**0,140**	**-0,036**	**-0,126**	**0,087**	**0,012**	**0,139**	**0,075**	**0,367**	**0,012**	**0,026**	**0,134**	**0,097**	**0,043**	**-0,084**				
Duality	**0,058**	**0,072**	0,007	**0,012**	-0,002	0,008	**0,011**	-0,003	-0,005	**-0,022**	**-0,034**	**0,055**	0,027	**-0,016**	0,008			
Tone	**0,318**	**0,031**	**-0,123**	**-0,055**	**-0,032**	**0,060**	**0,042**	**0,039**	**-0,012**	-0,004	**-0,081**	**0,267**	**0,045**	**-0,173**	0,007	**0,023**		
AnDisp	**0,257**	**0,052**	**-0,085**	**0,059**	0,020	**0,172**	**0,087**	**-0,022**	0,015	**-0,048**	-0,022	**0,334**	**0,096**	**-0,081**	-0,016	**0,058**	**0,035**	
LnAnn	**0,408**	**0,028**	**-0,067**	**0,050**	0,013	**0,175**	**0,118**	**-0,043**	**0,033**	**-0,070**	**-0,039**	**0,544**	**0,156**	**-0,075**	**-0,038**	**0,073**	**0,098**	**0,528**

Table 4: Abnormal returns as a Reaction to Campaign Announcement

Panel A: Market reaction to short seller activist campaigns
The results presented in the table below test H3 that firms targeted by short seller activist will return significantly negative abnormal and cumulative abnormal returns in the short term (long term) compared to their peers as well as well as 0-abnormal return.

Market days	**-3**	**-2**	**-1**	**0** Announcement	**1**	**2**	**3**
A: Market reaction to activist campaign announcement, French Fama 3 factor adjusted (N = 239), t-stat at t0 significant at 1% p-value							
Mean Abnormal Returns	-0,0124	-0,0040	-0,0106	**-0,0485**	-0,0072	-0,0049	-0,0040
Median Abnormal Returns	-0,0087	-0,0093	-0,0096	-0,0392	-0,0034	-0,0025	-0,0036
t- stat	-3,452	-1,090	-2,582	-15,974	-1,176	-1,818	-0,941
B: Abnormal returns of French Fama 48 industry peers, within the same market capitalization quintile (N = 1454)							
Mean Abnormal Returns	-0,0010	0,0001	0,0003	-0,0014	-0,0007	-0,0011	0,0002
Median Abnormal Returns	-0,0005	-0,0004	-0,0004	-0,0008	-0,0011	-0,0010	-0,0006
t- stat	-1,2870	0,1039	0,3174	-1,7945	-0,8297	-1,4099	0,2645
Market days after the event	**0**	**1**	**2**	**3**	**5**	**10**	**22**
C: Absolute Return to daily lows (N = 238, period: pre-event close to t)							
Mean max decline	-0,0996	-0,0870	-0,0877	-0,0848	-0,0808	-0,0795	-0,0822
Median max decline	-0,0813	-0,0727	-0,0750	-0,0665	-0,0662	-0,0659	-0,0716
Mean maximum declines (peers)	-0,0203	-0,0216	-0,0230	-0,0223	-0,0196	-0,0194	-0,0163
D: Cumulative abnormal returns following the campaign initiation (N=238)							
	10D	**Month**	**QTR**	**HY**	**FY**		
Mean CAR (0, t)	-0,0896	-0,1203	-0,2335	-0,4709	-0,8173		
t- stat	-9,484	-8,903	-8.63	-12.06	-14.836		

Long term CARs ranging from 10 trading days to 1 trading year (FY) after the campaign announcement were significantly different from 0 abnormal returns in all periods ($p<0.05$).

Panel B: Cumulative Abnormal Returns

Following a period from 9 days prior to campaign announcement to full market trading year after the event.

Market days	Quarter	HY	Year
CAR (0, t)	-0,2335	-0,4709	-0,8173
t-stat	-8.63	-12.064	-14.836

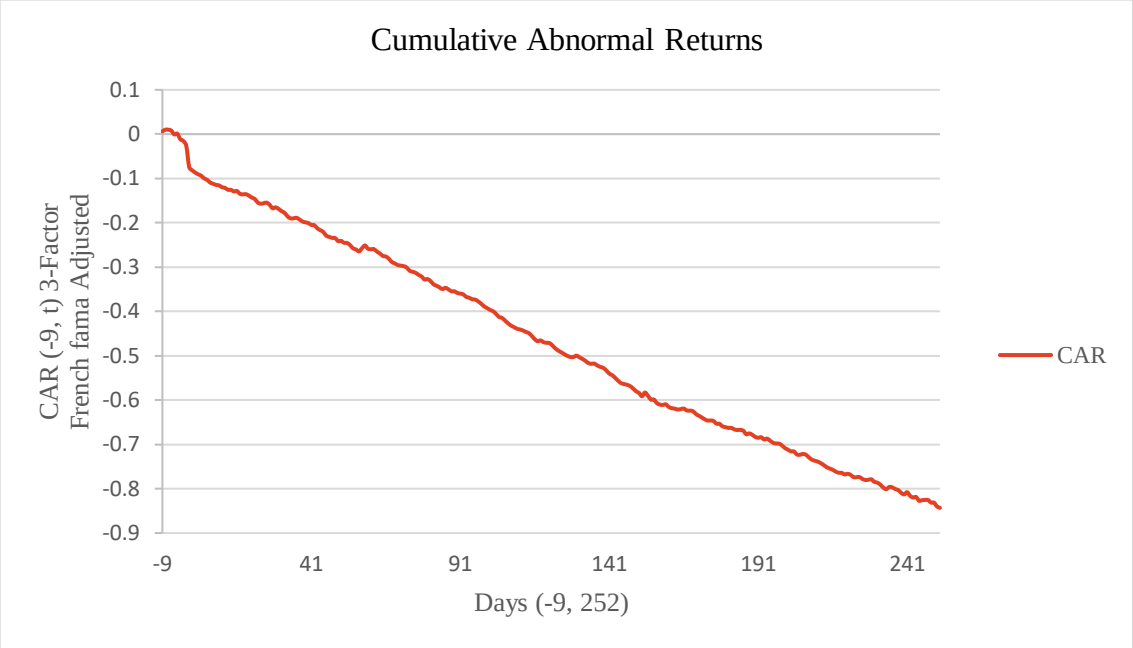


Table 5: Overvaluation Features as determinants of Short-Sellers' Interest

The below table presents results of logit regression to test H1 that activist short-sellers will be attracted by firms exhibiting overvaluation features as defined in Chapter 3. All continuous variables were binarized to reflect their relative quintile position, thus taking a value of one if these were in the last quintile and zero otherwise. Variables with overall positive implications for high reading were multiplied by minus one, such that the top quintile contains range of values found to be least attractive when observed in isolation. All variables are defined in Chapter 4. Standard errors for t-statistics are clustered at firm level. Statistical significance follows the traditional notation (*, ** and *** for two-tailed p<0.1, 0.05 and 0.01, respectively)

Overvaluation								
	P/E	*TACC*	*IOver*	*Zscor*	*Momentum*	*P/B*	*cROA*	*DebtC*
Overvaluation param.	*0,851***	0,3128***	0,6940***	-0,0609	0,2294	-0,1391	-0,3092**	*0,986****
	(5,274)	(7,2571)	(6,0156)	(-0,1817)	(0,7729)	(-0,3599)	(-2,0117)	(6,6211)
lnSize	0,015	0,0661	0,0567	0,0751	0,0775	0,0770	0,0507	0,086
	(0,5613)	(0,9607)	(0,8468)	(1,0458)	(1,1291)	(1,1)	(0,6455)	(3,4391)
dayStd	2,705**	1,2859	1,5746	1,2089	0,9331	1,2798	1,5741	0,584
	(1,9643)	(0,4588)	(0,5373)	(0,4417)	(0,3017)	(0,442)	(0,5737)	(0,3618)
D/A	2,626***	2,0485***	2,0144***	2,0130***	1,9713***	1,9696***	1,9595***	2,425***
	(7,3655)	(6,8094)	(7,0429)	(12,558)	(7,2402)	(5,9939)	(7,2149)	(7,3018)
Constant	-6,779***	-6,2698***	-6,2449***	-6,3484***	-6,4532***	-6,3697***	-5,7814***	-8,221***
	(-12,8464)	(-4,5268)	(-4,6765)	(-4,4937)	(-4,6473)	(-4,5556)	(-3,6862)	(-15,719)
Observations	54,291	54,291	54,291	54,291	54,291	54,291	54,291	54,291
Pseudo-R	*0.04305*	*0.03170*	*0.03787*	*0.03300*	*0.03254*	*0.03484*	*0.03174*	*0.04680*

Table 6: Ambiguity Features as Determinants of Short-Sellers' Interest

This table displays logit regression results testing whether short-seller activists are attracted by Ambiguity features described in Chapter 4 (H2). All variables are defined in Chapter 4. Standard errors for t-statistics are clustered at firm level. Statistical significance follows the traditional notation (*, ** and *** for two-tailed p<0.1, 0.05 and 0.01, respectively).

Ambiguity	*lnVol*	*Spread*	*Big4*	*Tone*	*Mscor*	*Duality*	*AnnDisp*	*LnAn*
Ambiguity parameters	-1,1404***	*1,261***	0,010	0,239	*1,455***	0,2809***	-2,010***	-0,588***
	(-3,4945)	(7,6038)	(0,0539)	(1,2492)	(9,3977)	(2,5874)	(-12,237)	(-3,567)
lnSize	0,0381	0,009	0,049*	0,040	0,045	0,045*	-0,0414	0,0227
	(0,5344)	(0,3102)	(1,7808)	(1,5057)	(1,5149)	(1,7314)	(-1,3195)	(0,3251)
dayStd	3,0136	1,975	1,600	1,568	2,902*	1,578	1,2405	0,4479
	(1,0454)	(1,3229)	(1,07)	(1,0381)	(1,888)	(1,051)	(0,3525)	(0,1505)
D/A	1,9738***	2,584***	2,497***	2,501***	2,681***	2,478***	2,1337***	2,1134***
	(7,269)	(7,3068)	(7,273)	(7,3901)	(7,6982)	(7,2152)	(7,504)	(6,343)
Constant	-5,5528***	-6,697***	-7,056***	-6,929***	-7,636***	-7,058***	-2,2367***	-6,253***
	(-3,796)	(-12,1919)	(-12,092)	(-13,1351)	(-12,7028)	(-13,532)	(-4,334)	(-11,741)
Observations	54,291	54,291	54,291	54,291	54,291	54,291	9378	9378
Pseudo-R	0.04203	0.05657	0.03169	0.03236	0.06723	0.03257	0.09848	0.03655

Table 7: Overvaluation and Ambiguity Features Combined

In this table the overvaluation and ambiguity parameters are tested as part of their respective feature set and combined together into one set of variables. Determinants pertaining to Analyst ratings and dispersion thereof have considerably less data available. Consequently, these cannot be included in a combined regression, unless we wanted to sacrifice most of the data for other determinants. Significance and t-statistics notation follows the notation in the previous tables.

Determinants	Overvaluation Features	Ambiguity Features	Overvaluation & Ambiguity
P/E	0,9592***		0,8181***
	(5,8009)		(6,4046)
TACC	-0,2455***		-0,4747***
	(-4,7705)		(-14,1160)
IOver	0,5762**		0,5224***
	(4,3017)		(4,7579)
Zscor	-0,2557		-0,3728**
	(-0,9604)		(-2,0405)
Momentum	0,3942		0,2855
	(1,5145)		(1,0167)
cROA	-0,4266***		-0,2636***
	(-6,5855)		(-2,9548)
DebtC	1,4110***		1,3536***
	(3,3811)		(3,4867)
lnVol		-1,3511***	-1,3018***
		(-4,1911)	(-4,9848)
Spread		1,0937***	1,0461***
		(4,7999)	(5,2136)
Big4		0,3984**	0,4366**
		(2,5728)	(2,4630)
Tone		0,2447*	0,3039**
		(1,9342)	(2,2297)
Mscor		1,4137	1,4082***
		8,7027	(6,4283)
Duality		0,1295	0,1402
		(1,0811)	(1,2197)
D/A	2,2390***	2,4262***	2,6896***
	(17,824)	(11,292)	(17,12)
Const.	-8,3311***	-6,4654***	-8,3783***
	(-5,399)	(-5,611)	(-5,761)
Observations	54,291	54,291	54,291
Pseudo R	0.06235	0.08363	0.1226

Table 8: Reaction to Activist Campaign Announcement

The following tables display the overview of targets' (cumulative) abnormal returns following an activist campaign announcement in a period ranging from one day to one full trading year (254 market days).[47] Panels B and C report determinants of abnormal returns, with t-statistics in parentheses and their respective significance described as *,**,*** for p<0.1, <0.05 and <0.01, respectively.

Panel A: Summary of Market reactions and overview of descriptive statistics of model variables. Overvaluation and Ambiguity feature grouping described in Panel C.

	Obs.	Mean	Std.	1st Quintile	4th Quintile
AR_0	228	-0,047	0,118	-0,0939	0,0083
CAR_1	228	-0,055	0,139	-0,1168	0,0061
CAR_5	228	-0,067	0,175	-0,1473	0,0088
CAR_22	228	-0,112	0,550	-0,2020	0,0612
CAR_qtr	228	-0,222	1,481	-0,3115	0,0690
CAR_year	228	-0,768	5,546	-1,0245	0,1912
Overvaluation	228	0,263	0,218	0	0,5
Ambiguity	228	0,360	0,250	0,25	0,5
lnSize	228	19,994	1,971	18,435	21,573
Debt to Assets	228	0,208	0,251	0	0,4144
Daily Std. of returns	228	0,036	0,030	0,0171	0,0476
Average daily volume	228	0,041	0,200	0	0

[47] Magnitudes of abnormal returns may differ from those in table 1, since OLS regression required only those companies with no missing data to be.

Panel B: Regression results with all determinants included (non-binarized values).

To avoid collinearity between variables, *lnSize and lnVol* were removed from the dataset. Number of analysts covering a stock and dispersion of analyst rating were excluded from the variable set due to insufficiency of data. T-statistics are provided in parentheses, standard errors are clustered at firm level.

	AR(0)	*CAR(0,1)*	*CAR(0,5)*	*CAR(Month)*	*CAR(qtr.)*	*CAR(year)*
Tacc	-0,0745	0,0054	0,1482**	0,1031	0,6197*	1,6937**
	(-1,0458)	(0,0918)	(2,0478)	(0,5315)	(1,759)	(1,992)
Iover	0,1067	0,0762	0,0708	-0,0007	0,0421	0,1409
	(1,4917)	(1,0187)	(0,9089)	(-0,0050)	(0,1669)	(0,2279)
Zscor	0,0000	-0,0005	-0,0004	-0,0006	-0,0073	-0,0115
	(0,0066)	(-0,3699)	(-0,2963)	(-0,2415)	(-1,4354)	(-0,7563)
Momentum	0,0234	-0,0115	0,0285	-0,0023	0,1911**	-0,0404
	(0,8419)	(-0,4219)	(1,0688)	(-0,0558)	(2,2521)	(-0,2406)
P/B	0,0051**	0,0060**	0,0052*	0,0072**	0,0094*	0,0068
	(2,2669)	(2,2070)	(1,9549)	(2,0474)	(1,6246)	(0,4618)
cROA	0,0171	0,0126	0,0180*	0,0438**	0,0926	0,1483
	(1,3009)	(0,8252)	(1,6702)	(2,0447)	(1,4270)	(1,2442)
Spread	-0,1086**	-0,1377**	-0,0925	-0,0705	-0,0494	-0,0225
	(-1,8675)	(-1,923)	(-1,3950)	(-0,9025)	(-0,3475)	(-0,0632)
Big4	-0,0609***	-0,0671***	-0,0783**	-0,1498**	-0,2081**	-0,3997**
	(-2,653)	(-2,59)	(-2,53)	(-2,89)	(-2,39)	(-2,291)
Mscor	0,0010	0,0008	0,0040	-0,0029	-0,0087	0,0105
	(0,3865)	(0,2966)	(0,9780)	(-0,3588)	(-0,5794)	(0,2374)
Duality	-0,0022	0,0068	0,0028	0,0031	0,0368	0,0425
	(-0,1067)	(0,2918)	(0,1036)	(0,0666)	(0,4750)	(0,2381)
Tone	-0,0097	-0,0095	0,0064	0,0588	0,2015**	0,5614**
	(-0,4514)	(-0,4231)	(0,2279)	(1,1760)	(2,1968)	(2,4367)
dayStd	-0,5750*	-0,3505	-0,8942**	-1,694***	-4,2093***	-7,8899**
	(-1,606)	(-0,9958)	(-2,081)	(-2,595)	(-2,69)	(-2,412)
D/A	0,0671	0,0299	0,0661	0,0292	-0,0667	-0,3343
	(0,9399)	(0,3301)	(0,7490)	(0,2721)	(-0,3280)	(-0,5433)
const	-0,0087	0,0174	0,0079	0,0197	0,0192	0,1287
	(-0,2647)	(0,4731)	(0,1949)	(0,3544)	(0,1686)	(0,3414)
Obs.	228	228	228	228	228	228
Adj. R-sq.	0.139	0.141	0.134	0.157	0.190	0.187

Panel C: Regression results with overvaluation and ambiguity as groups

Both feature sets are divided into two groups comprising of the averages of binary values of targeted firms in corresponding firm quarter according to their respective quintile position. Components of the groups were selected according to statistical significance (p<0.1). For Overvaluation these are: Overinvestment, Momentum, Debt coverage, Z-score. For Ambiguity, Spread, Big4, Duality of CEO and Tonal uncertainty. Standard errors clustered by firm.

	AR(0)	*CAR(0,1)*	*CAR(0,5)*	*CAR(0,22)*	*CAR(qtr)*	*CAR(year)*
D/A	-0,0068	-0,0371	-0,0332	0,0289	0,3172	1,5613
	(-0,213)	(-0,986)	(-0,709)	(0,195)	(0,793)	(1,047)
dayStd	-0,4720*	-0,5648*	-0,9282**	-1,2314	-2,4141	-7,0760
	(-1,767)	(-1,789)	(-2,359)	(-0,990)	(-0,720)	(-0,566)
Avg.daily Vol	0,0485	0,0501	0,0662	0,1329	0,0753	0,3428
	(1,220)	(1,068)	(1,131)	(0,718)	(0,151)	(0,184)
Overvaluation	0,0015	-0,0100	-0,0982*	-0,5102***	-1,4742***	-6,0210***
	(0,042)	(-0,231)	(-1,813)	(-2,979)	(-3,191)	(-3,495)
Ambiguity	-0,0912***	-0,1007***	-0,1077**	-0,2719*	-0,6516*	-2,3732
	(-2,875)	(-2,685)	(-2,305)	(-1,840)	(-1,635)	(-1,597)
Const.	0,0021	0,0101	0,0355	0,1534*	0,4178*	1,5842*
	(0,107)	(0,444)	(1,249)	(1,708)	(1,725)	(1,754)
Obs.	228	228	228	228	228	228
Adj. R-sq.	0.04	0.038	0.056	0.044	0.042	0.05

Coefficient Comparison – Overvaluation / Ambiguity

	AR(0)	*CAR(0,1)*	*CAR(0,5)*	*CAR(0,22)*	*CAR(qtr)*	*CAR(year)*
Overvaluation	0,0015	-0,0100	-0,0982	-0,5102	-1,4742	-6,0210
Ambiguity	-0,0912	-0,1007	-0,1077	-0,2719	-0,6516	-2,3732
Overval./ Ambig.	-0,0167	0,0995	0,9117	1,8768	2,2625	2,5371

Python Code

For the sake of completeness, the entirety of the Python Code employed is provided, including the data extraction itself as well as data treatment and ordering, calculations and statistical operations.[48]

Sections of python code presented herein should be replicable, given the use of the same IDE, PyCharms (as there are minor differences to other IDEs, such as for example Jupyter), and given a valid API access code for data extraction. All data manipulations outside of the boundaries of the code are explicitly stated.

Webpage Spider

Example of spider used for scraping of campaign announcement dates:
Spiders had to be adjusted individually for each activist (groups thereof), as webpage architecture differs from activist to activist. In some cases, use of automated robots was completely prohibited and manual collection was thus required. The spider below is not universal and is not resistant to updates which would alter the original html-structure.

```
# -*- coding: utf-8 -*-

import scrapy

from ..items import CitronItem

#CitronItem is initiated in terminal.


class CitronusSpider(scrapy.Spider):

    name = 'Citronus'

    page_number = 2

    start_urls = [

        'https://citronresearch.com/citron-report-archives/']


    def parse(self, response):

        items = CitronItem()


        research_name = response.css('.entry-title a::text').extract()

        research_date = response.css('span.av-structured-data::atr(datetime)').get()
```

[48] Disclosure: I do not have any programming background nor did I have any experience with Python prior to writing this thesis. The code, therefore, may not comply with pythonic ways of programming and will necessarily be overly verbose at times.

```
        items['research_name'] = research_name
        items['research_date'] = research_date
        yield items

        next_page = "https://citronresearch.com/citron-report-archives/page/"
+str(CitronusSpider.page_number)+ "/"
        if CitronusSpider.page_number < 30:
            CitronusSpider.page_number += 1
            yield response.follow(next_page, callback=self.parse)
```

Extracting Company Sic Codes

```
from __future__ import print_function
import time
import intrinio_sdk
from intrinio_sdk.rest import ApiException
from pprint import pprint
import pandas as pd
import numpy as np
import os

df_x = pd.read_excel(os.path.join(os.path.dirname(__file__),
                          "/Users/karelsarapatka1/Desktop/companies.xlsx"))
identifiers = df_x['companies']

intrinio_sdk.ApiClient().configuration.api_key['api_key'] = '*****'
security_api = intrinio_sdk.SecurityApi()

tag = 'sic'
data = []

for identifier in identifiers:
    company_data = {}
    pass
    company_data['company'] = []
    company_data['sic'] = []
    try:
        api_response = security_api.get_security_data_point_number(identifier, tag)
        #pprint(api_response)
        sic = (api_response)
        company_data['company'].append(identifier)
        company_data['sic'].append(sic)

    except ApiException as e:
        print(
```

```
        "Exception when calling SecurityApi->get_security_data_point_number: %s\r\n" % e)


    data.append(company_data)
#print(data)
df5 = pd.DataFrame(data)
df7 = df5['company'].apply(pd.Series).rename(columns =lambda x : 'Company')
df6 = df5['sic'].apply(pd.Series).rename(columns =lambda x : 'SIC_Code')


df0 = [df7, df6]
df = pd.concat(df0, axis = 1, sort = True)
print(df)
df.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/sample_df.xlsx"))
```

Obtaining SIC codes associated with selected French Fama 48 groups:

```
import pandas as pd
import numpy as np
import os
import time
from datetime import timedelta
import collections

source = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/desktop/sample_marketcap.xlsx"))
ff = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/fama_french/ff.xlsx"))

#extract all sic codes of respective ff48 group
intstep = []
for z in range(len(source)):
    ff48 = {}
    pass
    ff48['FF48'] = []
    ff48['sic'] = []
    for x in range(len(ff)):
        if source.loc[z, 'FF48'] == ff.loc[x, 'FF']:
            ff48['FF48'].append(ff.loc[x, 'FF'])
            ff48['sic'].append(ff.loc[x, 'sic1'])
            ff48['sic'].append(ff.loc[x, 'sic2'])

    intstep.append(ff48)
frama = pd.DataFrame(intstep)

# change FF48 and sic into tuples and FF48 to a single integer
def fct(x):
    return tuple(dict.fromkeys(x))
```

```
frama['ff48'] = ''
for i in range(len(frama)):
    frama.loc[i, 'FF48'] = fct(frama.loc[i, 'FF48'])
    frama.loc[i, 'sic'] = fct(frama.loc[i, 'sic'])
    frama.loc[i,'ff48'] = int(frama.loc[i, 'FF48'][0])

#finalize new dataframe with single value, no duplicates
frama = frama.drop_duplicates(keep='first', inplace = False)
frama = frama.assign(FF48 = frama['ff48'])
frama = frama.drop(columns = ['ff48'])
print(frama)
```

Construction of the sample of companies

Get all companies according to SIC-codes in respective French-Fama 48 groups (two versions of this code were used; the version below is simplified as it uses explicit sic code input and not a loop through list of sic codes from a separate data frame with French Fama 48 sic codes)
The outputs are all companies associated with given French Fama 48 group, which are subsequently assigned to a separate file with all groups/companies for later use. Total collected prior to filtering: 3154 unique companies.

```
from __future__ import print_function
import time
import intrinio_sdk
from intrinio_sdk.rest import ApiException
from pprint import pprint
import pandas as pd
import os
import re


intrinio_sdk.ApiClient().configuration.api_key['api_key'] = '*****'
security_api = intrinio_sdk.SecurityApi()


adj_ff = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/adjusted_ff.xlsx"))
bench = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/bench_comp.xlsx"))


i = (6200,6211, 6299, 6700, 6710, 6719, 6720, 6722, 6723, 6724, 6725, 6726, 6730)
datas = []
paraf = []
for x in i:
    clause1 = intrinio_sdk.SecurityScreenClause(
        field='sic',
        operator='eq',
        value= x
    )
    clause2 = intrinio_sdk.SecurityScreenClause(
```

```
    field='marketcap',
    operator='gt',
    value=1000
)
logic = intrinio_sdk.SecurityScreenGroup(operator="AND",
                                         clauses=[clause1, clause2])
order_column = 'name'
order_direction = 'asc'
primary_only = False
page_size = 100

try:
    api_response = security_api.screen_securities(
        logic=logic,
        order_direction=order_direction,
        primary_only=primary_only,
        page_size=page_size,
    )
    #pprint(api_response)
    #data = pd.DataFrame(api_response.Screen_Securities_dict)
    datas.append(api_response)

    dataa = pd.DataFrame.from_dict(api_response)
    if dataa.empty:
        continue
    #print(dataa)
    for x in range(len(dataa)):
        ticker = str(dataa.loc[x, 0])[-8:]
        pattern = [r'\w+']
        for p in pattern:
            match = re.findall(p, ticker)
            paraf.append(match)

except ApiException as e:
```

```
        print("Exception when calling SecurityApi->screen_securities: %s\r\n" % e)
df = pd.DataFrame(datas)
#print(df)
datapts = pd.DataFrame(paraf)
print(datapts)
datapts.to_excel(os.path.join(os.path.dirname(__file__), "/Users/karelsarapatka1/Desktop/ff47.xlsx"))
```

Market Capitalization

Average market capitalization in a period of 4 months preceding the activist campaign for the purpose of obtaining market capitalization quintiles in which targets are found.

```
from __future__ import print_function

import time

import intrinio_sdk

from intrinio_sdk.rest import ApiException

from pprint import pprint

import pandas as pd

import numpy as np

import os

from datetime import timedelta

from dateutil.relativedelta import *


start_time = time.time()

intrinio_sdk.ApiClient().configuration.api_key['api_key'] = '*****'

security_api = intrinio_sdk.SecurityApi()


df_x = pd.read_excel(os.path.join(os.path.dirname(__file__),

                        "/Users/karelsarapatka1/Desktop/target_master.xlsx"))


identifiers = df_x['Company']

df_x['date-2'] = df_x['date1'] + timedelta(days=-150)

df_x['date-2'] = df_x['date-2'].dt.date

df_x['date+2'] = df_x['date1'] + timedelta(days=-30)

df_x['date+2'] = df_x['date+2'].dt.date

df_x['marketcap'] = ""


tags = 'marketcap'

df_x = df_x.drop(['Unnamed: 0'], axis = 1)

print(df_x)


frequency = 'monthly'
```

```
type = ''
start_date = df_x['date-2']
end_date = start_date + timedelta(days = +4)
sort_order = 'desc'
page_size = 100
next_page = ''
dataf = []
means = []
for z in range(len(df_x)):
    identity = df_x.loc[z, 'Company']
    start = df_x.loc[z, 'date-2'].isoformat()
    end = df_x.loc[z, 'date+2'].isoformat()
    api_response = security_api.get_security_historical_data(identity, tags, frequency=frequency, type=type,
start_date=start, end_date=end, sort_order=sort_order, page_size=page_size, next_page=next_page)
    data = pd.DataFrame(api_response.historical_data_dict)
    data.rename(columns={'value': tags}, inplace=True)
    data.insert(0, 'Company', identity)

    if data.empty:
        continue
    meano = data[tags].mean()
    df_x.loc[z, 'marketcap'] = data[tags].mean()

    dataf.append(data)

print(df_x)
datas = pd.DataFrame(dataf)
df_x.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/market_capss.xlsx"))

print("--- %s seconds ---" % (time.time() - start_time))
```

Market capitalization of industry peers, at latest 30 days before activist campaigns were initiated (average of 4 month market cap prior to campaign used). These will be filtered out to build a benchmark against which performances are compared.

1st step: Creating data frames for each FF48 group of companies with dates of campaign initiation.

```
import pandas as pd
import numpy as np
import os
from datetime import timedelta
import datetime as dt


target = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/sample_marketcap.xlsx"))
bench = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/bench_comp.xlsx"))


columns = list(bench.columns.values)


d= {}
for ff in columns:
    d[ff] = pd.DataFrame(bench[ff]).dropna()
    d[ff] = d[ff].set_index(ff).transpose()
    d[ff].insert(loc = 0, column = 'dates', value = bench[ff])
    #d[ff].to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/companies/bench/marketcap/" + str(ff) + ".xlsx"))


for f in columns:
    dates = []
    bench_dates = {}
    for i in range(len(target)):
        if f == target.loc[i, 'FF48']:
            dates.append(target.loc[i, 'date1'])
    bench_dates[f] = pd.DataFrame(dates)
    d[f] = pd.read_excel(os.path.join(os.path.dirname(__file__),
```

```
"/Users/karelsarapatka1/Desktop/companies/bench/marketcap/" + str(f) + ".xlsx"))
    d[f]['dates'] = bench_dates[f]
    d[f] = d[f].drop(['Unnamed: 0'], axis=1)
    d[f] = d[f].dropna(subset = ['dates'])

#d[f].to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/companies/bench/m
cap/" + str(f) + ".xlsx"))
    #print(d[f])
```

Step 2. Obtaining Market capitalization of targets' peers.
Average of monthly market capitalization in 6 months prior to activist campaign.
Sleep introduced to control for API-provider's download rate limit.

```
from __future__ import print_function
import time
import intrinio_sdk
from intrinio_sdk.rest import ApiException
from pprint import pprint
import pandas as pd
import numpy as np
import os
from datetime import timedelta
from dateutil.relativedelta import *
import glob

start_time = time.time()
intrinio_sdk.ApiClient().configuration.api_key['api_key'] = '*****'
security_api = intrinio_sdk.SecurityApi()

tags = 'marketcap'
frequency = 'monthly'
type = ''
sort_order = 'desc'
page_size = 100
next_page = ''
```

```
#comp_list = glob.glob("/Users/karelsarapatka1/Desktop/companies/bench/mcap/*")
comp_list = [os.path.basename(x) for x in
glob.glob("/Users/karelsarapatka1/Desktop/companies/bench/mcap/*")]
count = 0
for x in comp_list:
    df1 =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/companies/bench/
mcap/" + x))
    df1['date-2'] = df1['dates'] + timedelta(days=-180)
    df1['date-2'] = df1['date-2'].dt.date
    df1['dates'] = df1['dates'].dt.date
    df1 = df1.drop(['Unnamed: 0'], axis = 1)
    past = df1['date-2']
    df1.drop(labels = ['date-2'], axis = 1, inplace = True)
    df1.insert(0, 'date-2', past)

    for col in df1.columns:
        mcap = []
        if col != 'dates' and col !='date-2':
            for i in range(len(df1)):
                try:
                    end = str(df1.loc[i, 'dates'])
                    start = str(df1.loc[i, 'date-2'])
                    api_response = security_api.get_security_historical_data(col, tags, frequency=frequency,
type=type, start_date=start, end_date=end, sort_order=sort_order,page_size=page_size,
next_page=next_page)
                    data = pd.DataFrame(api_response.historical_data_dict)
                    data.rename(columns={'value': tags}, inplace=True)
                    data.rename(columns={'date': col}, inplace=True)
                    print(data)
                    print('__'*20)

                    if data.empty:
                        continue
```

```
                cap = data[tags].mean()
                df1.loc[i, col] = cap
                time.sleep(4)
                df1.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/companies/bench/bench_cap/group" + x))
            except ApiException as e:
                time.sleep(20)
                continue

print("--- %s seconds ---" % (time.time() - start_time))
```

Separation of peer market capitalizations into quintiles in order to select quintile of interest

```
import pandas as pd
import os
import glob
import numpy as np

targets = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/cap_peers.xlsx")).drop(['Unnamed: 0'], axis = 1)

ff_group= [os.path.basename(x) for x in
sorted(glob.glob("/Users/karelsarapatka1/Desktop/companies/bench/bench_cap/*"))]

for F in ff_group[:-1]:
    df1 = pd.read_excel(os.path.join(os.path.dirname(__file__),
                                "/Users/karelsarapatka1/Desktop/companies/bench/bench_cap/" +
F)).drop(['Unnamed: 0'], axis=1)
    gruppe = pd.DataFrame(columns = df1.columns).drop(['date-2'], axis=1)
    gruppe['dates'] = df1['dates']
    for y in range(len(df1)):
        q = pd.qcut(df1.iloc[y, 2:], 5).dropna()
        quint = q.loc[:].unique()
        quint = quint.remove_categories(pd.np.NaN)
```

```
        #print(quint.value_counts())
        for i in quint:
            for x in df1.columns:
                if x!= 'date-2' and x!= 'dates' and df1.loc[y, x] <= i.right and df1.loc[y, x]>= i.left:


                    gruppe.loc[y, x] = i
#gruppe.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/new_bm/qt_"+F))
```

Selection of French Fama 48 industry peers according to market capitalization quintile of the targeted firm.

```
import pandas as pd
import numpy as np
import os
import glob


targets = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/cap_peers.xlsx")).drop(['Unnamed: 0'], axis = 1)


for x in np.unique(targets.FF48):
    df = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/new_bm/qt_group"+ str(x)+ ".xlsx")).drop(['Unnamed: 0'], axis=1)
    comps = df.columns.values.tolist(); del comps[0:1]
    comp_list = targets.loc[targets['FF48']==x].Company


    for ele in enumerate(comp_list):
        peer_list = []
        ranger = str(df.loc[ele])
        y,z = ele
        for G in df.columns:
            if df.loc[y,G ]== ranger:
                peer = df.loc[y,G ]
                peer_list.append(G)
        dates = str(df.loc[y,'dates'])
        df2 = pd.DataFrame(peer_list,columns = [z])
```

```
        df2['dates'] = ''
        days = df2['dates']; df2.drop(labels=['dates'], axis=1, inplace = True)
        df2.insert(0, 'dates', days)
        df2.loc[:, 'dates'] = df.loc[y, 'dates']
        #print('FF48 group:',x, 'target name:',z ,df2)
        df2.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/all_p/group"+str(x)+"_"+z+".xlsx"))
```

Stock returns prior and post activist campaign announcement

Getting EOD close prices for targeted companies prior to activist attack, on the day of campaign announcement, 1, 2, 5 business days, 1 month (22 business days), two quarters (125 business days) and one full year (250 business days) after campaign initiation.

```
from __future__ import print_function
import time
import intrinio_sdk
from intrinio_sdk.rest import ApiException
import pandas as pd
import os
from pandas.tseries.offsets import BDay
start_time = time.time()
intrinio_sdk.ApiClient().configuration.api_key['api_key'] = '*****'
security_api = intrinio_sdk.SecurityApi()
sample_mc =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/tgt_days.xlsx"))

datelist = ['date', 'date_t1', 'date_t2', 'date_t5', 'date_tM', 'date_t6M', 'date_t1Y']
closelist = ['t0_close', 't1_close', 't2_close', 't5_close', 'tM_close', 't6M_close', 't1Y_close']

for z in range(0, len(datelist)):
    sample_mc[datelist[z]] = ''
    sample_mc[closelist[z]] = ''

for x in range(len(sample_mc)):
    sample_mc.loc[x,'date_t1'] = sample_mc.loc[x,'date'] + BDay(1)
    sample_mc.loc[x,'date_t2'] = sample_mc.loc[x,'date'] + BDay(2)
    sample_mc.loc[x,'date_t5'] = sample_mc.loc[x,'date'] + BDay(5)
    sample_mc.loc[x,'date_tM'] = sample_mc.loc[x,'date'] + BDay(22)
    sample_mc.loc[x,'date_t6M'] = sample_mc.loc[x,'date'] + BDay(125)
    sample_mc.loc[x,'date_t1Y'] = sample_mc.loc[x,'date'] + BDay(250)

sample_mc = sample_mc.drop(['Unnamed: 0'], axis=1)
```

```
frequency = 'daily'
page_size = 100
next_page = ''

for i in range(0,len(datelist)):
    for x in range(len(sample_mc)):
        try:
            identifiers = sample_mc.loc[x, 'Company']
            start_date = sample_mc.loc[x, datelist[i]]
            api_response = security_api.get_security_stock_prices(identifiers, start_date=start_date,
end_date=start_date, frequency=frequency, page_size=page_size, next_page=next_page)

            time.sleep(1.1)
            output = pd.DataFrame(api_response.stock_prices_dict)
            if output.empty:
                continue
            deimos = output.loc[0, 'close']
            #print("Printing Deimos",deimos, "__"*10)
            sample_mc.loc[x, closelist[i]] = deimos
            print(sample_mc.loc[x, closelist[i]])
            print('__'*20)
            sample_mc.to_excel(os.path.join(os.path.dirname(__file__),
                                    "/Users/karelsarapatka1/Desktop/tgt_days3.xlsx"))
        except ApiException as e:
            time.sleep(20)
            continue
        print(x)
print("--- %s seconds ---" % (time.time() - start_time))
#sample_mc.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/tgt_days3.xl
sx"))
```

Getting prices for all companies (peers and targets):

Approach 1:

125 business days prior to activist campaign to calculate normal expected returns according to French Fama 3 factor model and calculate abnormal returns prior and after

the campaign. In addition to prices, EOD lows and highs are obtained, for the purpose of bid-ask spread estimation in later stages of data analysis.

Targets and their abnormal returns can be understood as a portfolio of 1 asset (as is often the case with activist short-sellers), whereas target's peers can be either observed individually, similarly to target, or combined into equally weighted portfolio. The former was chosen.

```
from __future__ import print_function
import time
import intrinio_sdk
from intrinio_sdk.rest import ApiException
import pandas as pd
import os
from pandas.tseries.offsets import BDay
import numpy as np
import glob
from datetime import timedelta
from pandas.tseries.offsets import BDay
import datetime


start_time = time.time()


intrinio_sdk.ApiClient().configuration.api_key[
    'api_key'] = '*****'
security_api = intrinio_sdk.SecurityApi()


mkt_hd =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/mkt_holidays.xlsx"))
.drop(['Unnamed: 0'], axis=1)
holiday = mkt_hd.loc[:, 'holidays']
frequency = 'daily'
next_page = ''


tgt_peer = [os.path.basename(x) for x in
```

```
        sorted(glob.glob("/Users/karelsarapatka1/Desktop/all_p/*"))]


for i in tgt_peer:
    df = pd.read_excel(os.path.join(os.path.dirname(__file__),
                                    "/Users/karelsarapatka1/Desktop/all_p/" + i)).drop(['Unnamed: 0'], axis=1)
    start_date = df.loc[0, 'dates'] - BDay(125)
    end_date = df.loc[0, 'dates'] + BDay(22)
    dif = end_date - start_date
    page_size = dif.days
    coll = df.iloc[:, 1].values.tolist()

    days = df['dates'].tolist()
    df2 = pd.DataFrame(data=None, index=days, columns=coll)

    data = []
    data_h = []
    data_l = []

    for identifiers in coll:
        try:
            if identifiers != 'dates':
                api_response =
security_api.get_security_stock_prices(identifiers,start_date=start_date,end_date=end_date,frequency=fre
quency,page_size=page_size,next_page=next_page)
                out = pd.DataFrame(api_response.stock_prices_dict)
                time.sleep(1.5)

                if out.empty:
                    time.sleep(1.5)
                    continue

                output = pd.DataFrame(
                    {'date': out['date'], identifiers: out['adj_close']})
                output_h = pd.DataFrame(
```

```
                {'date': out['date'], identifiers: out['adj_high']})
            output_l = pd.DataFrame(
                {'date': out['date'], identifiers: out['adj_low']})

            if len(output['date'])>= 10:
                data.append(output)
                data_h.append(output_h)
                data_l.append(output_l)
        except ApiException as e:
            time.sleep(5)
            continue

    data = pd.concat(data,axis= 1, join = 'outer')
    data_h = pd.concat(data_h,axis= 1, join = 'outer')
    data_l = pd.concat(data_l,axis= 1, join = 'outer')

    data.to_excel(os.path.join(os.path.dirname(__file__),
                               "/Users/karelsarapatka1/Desktop/pp_test/" + i))
    data_h.to_excel(os.path.join(os.path.dirname(__file__),
                               "/Users/karelsarapatka1/Desktop/bid_ask/high/" + i))
    data_l.to_excel(os.path.join(os.path.dirname(__file__),
                               "/Users/karelsarapatka1/Desktop/bid_ask/low/"+ i))
    print(data)
print("--- %s seconds ---" % (time.time() - start_time))
```

Approach 2:

In this code, prices are collected for the whole study period of 2009 to 2019. Together with EOD closing prices, daily lows, highs and volume are extracted for estimation of spreads and average daily volumes. Unlike the above code, the collected data is combined into a single excel sheet (for each parameter group), which in retrospect proved to be less time efficient for later processing, given its dimensions of 3500x2600 cells.

```
from __future__ import print_function
import time
import intrinio_sdk
```

```
from intrinio_sdk.rest import ApiException
import pandas as pd
import os
from pandas.tseries.offsets import BDay
import numpy as np
import glob
from datetime import timedelta
from pandas.tseries.offsets import BDay
import datetime
from pprint import pprint

start_time = time.time()
all_comp =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/bench_comp.xlsx"))
targets =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/cap_peers.xlsx"))
targ = targets.loc[:,'Company']

dff = []
for x in all_comp.columns:
    company = pd.Series(all_comp.loc[:,x])
    company.dropna(inplace = True)
    dff.append(company)

dff.append(targ)
combined = pd.concat(dff, ignore_index = True)
combined.drop_duplicates()
combined.dropna(inplace = True)
print(combined)

vict = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/tgt_date_ordered.xlsx")).drop(['Unnamed: 0'], axis = 1)
t_0 = vict.loc[0, 'GNI']
```

```
start_date = t_0 - BDay(255)
end_date = start_date + BDay(2610)
dif = (end_date - start_date).days

intrinio_sdk.ApiClient().configuration.api_key[
    'api_key'] = '*****'
security_api = intrinio_sdk.SecurityApi()

frequency = 'daily'
next_page = ''
page_size = dif

close = []
high = []
low = []
volume = []
renegades = []
for identifiers in combined:
    try:
        if identifiers != 'dates':
            api_response =
security_api.get_security_stock_prices(identifiers,start_date=start_date,end_date=end_date,frequency=fre
quency,page_size=page_size,next_page=next_page)
            out = pd.DataFrame(api_response.stock_prices_dict)
            #pprint(api_response)

            time.sleep(1.5)

            if out.empty:
                time.sleep(1.5)
                continue

            output = pd.DataFrame(
                {'date': out['date'], identifiers: out['adj_close']})
```

```
            output_h = pd.DataFrame(
                {'date': out['date'], identifiers: out['adj_high']})
            output_l = pd.DataFrame(
                {'date': out['date'], identifiers: out['adj_low']})
            output_v = pd.DataFrame(
                {'date': out['date'], identifiers: out['volume']})

            if len(output['date'])>= 100:

                close.append(output)
                high.append(output_h)
                low.append(output_l)
                volume.append(output_v)

                print('Printing Volume',volume)

        except ApiException as e:
            time.sleep(5)
            renegades.append(identifiers)
            continue

close = pd.concat(close,axis= 1, join = 'outer')
high = pd.concat(high,axis= 1, join = 'outer')
low = pd.concat(low,axis= 1, join = 'outer')
volume = pd.concat(volume,axis= 1, join = 'outer')

volume.to_excel(os.path.join(os.path.dirname(__file__),
                             "/Users/karelsarapatka1/Desktop/fundamentals/volume.xlsx"))

"""close.to_excel(os.path.join(os.path.dirname(__file__),
                             "/Users/karelsarapatka1/Desktop/bulk_price/eod.xlsx"))
high.to_excel(os.path.join(os.path.dirname(__file__),
                             "/Users/karelsarapatka1/Desktop/bulk_price/high.xlsx"))
low.to_excel(os.path.join(os.path.dirname(__file__),
```

```
                    "/Users/karelsarapatka1/Desktop/bulk_price/low.xlsx"))
print(close)"""


print("--- %s seconds ---" % (time.time() - start_time))
```

OLS Regression of Returns

The following lines of code utilize the extracted data to calculate daily returns of each stock contained in the sample and transform French-Fama 3-factor daily returns [49]into a usable format. Subsequently, it regresses individual daily stock returns against French Fama factors to obtain coefficients. Output factors are then used to project returns over the studied event period, 3 business days prior to short seller campaign announcement and 3 business days following the announcement.

Two approaches to calculating abnormal daily returns in the event window were chosen. Firstly, abnormal returns were calculated "manually" as a difference between the actual returns and French Fama adjusted expected returns. The second approach used a python library to calculate residuals of returns regression for each stock. Both outputs coincide.

Program outputs were saved in excel supported format (.xlsx) throughout the process to allow sanity tests of results and comparisons with expected outcomes. The "Try" wrapper of the code collects companies which were targeted on non-trading day (11 companies), these dates were adjusted manually to first trading day following the activist attack.

Lastly, the expected as well as the actual returns of the targeted companies in the event period were extracted to a single data frame, facilitating statistical testing for difference in later stages and enabling further operations on the output data while avoiding the rather long run time of the code below (approximately 480 seconds).

```
import os
import pandas as pd
import numpy as np
import re
import glob
import time
from sklearn.linear_model import LinearRegression
import datetime
from datetime import datetime as dt
from pandas.tseries.offsets import BDay
import statsmodels.formula.api as sm
from scipy import stats


start_time = time.time()


pattern = re.compile('[\W_]+')
```

[49] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html#Research

```
targets = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/cap_peers.xlsx")).drop(['Unnamed: 0'], axis = 1)
vict = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/tgt_date_ordered.xlsx")).drop(['Unnamed: 0'], axis = 1)
fr_fa = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/ff_3f.xlsx"),index_col=0)
fr_fa.index = pd.to_datetime(fr_fa.index, format = '%Y%m%d')
fr_fa = fr_fa.apply(lambda x: x/100)

# L = change.apply(lambda x: len(x.dropna())).max()
tgt_peer = [os.path.basename(x) for x in sorted(glob.glob("/Users/karelsarapatka1/Desktop/pp_test/*"))]

actual_targets = []
expected_targets = []
renegades = []
comps= []

for filename in os.listdir("/Users/karelsarapatka1/Desktop/pp_test/"):
    tgt = re.split('_|, |\.|\n', filename)
    comps.append(tgt[1])

for peer in tgt_peer:
    df = pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/pp_test/" +
peer)).drop(['Unnamed: 0'], axis=1)
    target = re.split('_|, |\.|\n', peer)
    tgts = target[1]

    col_list = df.columns
    col_ex = col_list[~col_list[:].str.contains('date')]
    col_incl = col_list[col_list[:].str.contains('date')]

    change = []
    for i in col_ex:
```

```
    prc_change = df[i].pct_change(-1)
    change.append(prc_change)
change = pd.concat(change, axis=1, join='outer')
change.insert(0, 'date', df['date'])

change.index = change['date']; change = change.drop(['date'], axis=1)
sample = pd.merge(pd.DataFrame(change), fr_fa, how='inner', left_index=True, right_index=True)
sample.rename(columns={'Mkt-RF': 'mkt_excess'}, inplace=True)
sample.reset_index(inplace = True)
sample.rename(columns={'index': 'date'}, inplace=True)
sample = sample.sort_values(by = ['date'],ascending = False)
sample.reset_index(inplace=True); sample = sample.drop(['index'], axis = 1)

date_list = sample['date']
date = vict.loc[0, tgts]

col_list = sample.columns[:-4]
col_ex = col_list[~col_list[:].str.contains('date')]

try:
    if sample[sample['date'] == date].index.values.astype(int)[0] != 0:
        day = sample[sample['date'] == date].index.values.astype(int)[0]
        #sample.insert(1,'dummy', '') ; sample.loc[day, 'dummy'] = 1
        row = sample.loc[sample['date'] == date]
        #print(row); print(peer); print(day)
        cutoff = day + 4
        d_m3 = day + 4
        d_m2 = day + 3
        d_m1 = day + 2
        d_p1 = day - 2
        d_p2 = day - 3
        d_p3 = day - 4

        pre_event = pd.DataFrame(sample.iloc[cutoff:, :])
```

```
event = pd.DataFrame(sample.iloc[:d_m3, :])
act_returns = sample[sample.columns[1:-4]]

beta_list = []
residuals = []
for x in col_ex:
    x = pattern.sub('', x)
    formula = x + "~ mkt_excess + SMB + HML"
    FF3 = sm.ols(formula, data=sample).fit()
    beta_list.append(FF3.params)
    residuals.append(FF3.resid)

residual = pd.DataFrame(residuals, index = col_ex).transpose()
betas = pd.DataFrame(beta_list, index=col_ex)

std_residual = {}
for bet in residual.columns.values:
    stds = residual.loc[:,bet].std()
    std_residual[bet] = stds

std_resid = pd.DataFrame(std_residual, index= range(len(vict)))
print('stds of residuals:');print(std_resid)

factors = sample[sample.columns[-4:-1]]
factors.insert(0, 'date', sample['date'])
#print('French Fama factors'); print(factors)

expected_returns = pd.DataFrame(data=None, index=range(len(factors)),
                        columns=col_ex)
abnormal_returns = pd.DataFrame(data=None, index=range(len(factors)),
                        columns=col_ex)
tgts_only = []

for y in range(len(betas.index)):
```

```
            intr = betas.iloc[y, 0]
            mkt = betas.iloc[y, 1]
            SMB = betas.iloc[y, 2]
            HML = betas.iloc[y, 3]
            company = betas.index[y]
            for alef in range(len(factors)):
                mkt_exc = mkt * factors.loc[alef, 'mkt_excess']
                smb = SMB * factors.loc[alef, 'SMB']
                hml = HML * factors.loc[alef, 'HML']
                exp_ret = intr + mkt_exc + smb + hml
                expected_returns.loc[alef, company] = exp_ret

            abnormal_returns[company] = act_returns[company] - expected_returns[company]

        peers = col_list[~col_list[:].str.contains(tgts)]
        abnormal_returns['mean_peer'] = abnormal_returns.drop(tgts, axis=1).apply(
            lambda x: x.mean(), axis=1)
        abnormal_returns['median_peer'] = abnormal_returns.drop(tgts, axis=1).apply(
            lambda x: np.median(x), axis=1)

        expected_returns.insert(0, 'date', sample['date'])
        abnormal_returns.insert(0, 'date', sample['date'])
        abnormal_returns.insert(1, 'dummy', ''); abnormal_returns.loc[d_p2:d_m2, 'dummy'] = 1

        abn_t = pd.DataFrame(abnormal_returns)
        for ent in residual.columns:
            s_e = std_resid.loc[0, ent]
            abn_t[ent] = abn_t[ent].apply(lambda x : x/s_e)

        expected_targets.append(expected_returns.loc[d_p2:d_m2, tgts])
        actual_targets.append(act_returns.loc[d_p2:d_m2, tgts])

        #abnormal_returns.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/regression/abnormal/abn_" + tgts + ".xlsx"))
```

```
#expected_returns.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/regres
sion/expected/exp_" + tgts + ".xlsx"))


    except IndexError:
        print('printing errors:', peer)
        renegades.append(peer)


print(renegades)
print("--- %s seconds ---" % (time.time() - start_time))


expected_targets= pd.concat(expected_targets,axis= 1, join = 'outer')
actual_targets= pd.concat(actual_targets,axis= 1, join = 'outer')


print('expected returns: ', expected_targets)
#expected_targets.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/regression/expected_targets.xlsx"))
actual_targets.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/regression/actual_targets.xlsx"))
```

Alternative approach to calculating abnormal returns with scikit-learn along with statsmodels using more comprehensive learning period of 1 year prior to activist campaign and estimating abnormal returns 1 fiscal year from the announcement.

```
import pandas as pd
import numpy as np
import os
import re
import glob
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import datatable as dt
import time
```

```
import datetime


start_time = time.time()


#all_p =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/fundamentals/bulk_
price/eod.xlsx" ))


FF3 = pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/ff_3f.xlsx" ))
FF3['date'] = pd.to_datetime(FF3['date'], format = '%Y%m%d')
FF3 = FF3.apply(lambda x: x/100 if x.name in ['Mkt-RF', 'SMB', 'HML', 'RF'] else x)



targets =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/tgt_date_ordered.xl
sx" )).drop(['Unnamed: 0'], axis = 1)
ebit = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/fundamentals/financials/ebit.xlsx"))
targ_p =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/target_prices.xlsx"))
.drop(['Unnamed: 0'], axis = 1)
dates = targets.columns


"""missing = []
prices = []
for i in targets.columns:
    try:
        col_ind = all_p.columns.get_loc(i)
        target = all_p.loc[:, i]
        dates = all_p.iloc[:, (col_ind - 1)]
        targ = pd.DataFrame({'date':dates, i:target})
        if i not in ebit.columns:
            continue


        print(targ)
```

```
        prices.append(targ)

    except:
        missing.append(i)

print(missing)

target_prices= pd.concat(prices,axis= 1, join = 'outer')
#target_prices.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/target_pric
es.xlsx"))"""

col_list = targ_p.columns
col_ex = col_list[~col_list[:].str.contains('date')]
targ = []
for i in col_ex:
    prc_change = targ_p[i].pct_change(-1)
    ind = targ_p.columns.get_loc(i)
    dates = targ_p.iloc[:-1, (ind - 1)]
    target = pd.DataFrame({'date':dates, i:prc_change})
    targ.append(target)

target_returns = pd.concat(targ, axis = 1, join = 'outer')
print(target_returns)

ind_val = FF3.loc[:,'date']
erores = []
for x in targets.columns:
    if x != 'GALE' and x!= 'HIIT':
        try:
            date = targets.loc[0, x]
            indx = target_returns.columns.get_loc(x)
            #print('printing indx:',indx)
            target_dates = target_returns.iloc[:,(indx-1)]
            row = target_dates[target_dates == date].index[0]
```

```
starting_row = row + 350
latest_row = row + 4
end_row = row - 220
target_data = target_returns.iloc[latest_row:starting_row, (indx - 1):indx + 1]
target_data.reset_index(inplace= True)
target_data = target_data.dropna()
target_date = target_data.drop(['index'], axis=1)
last_target = target_data['date'].iloc[-1]

target_data2 = target_returns.iloc[end_row:latest_row,(indx - 1):indx + 1]
target_data2.reset_index(inplace=True)
target_data2 = target_data2.dropna()
target_date2 = target_data2.drop(['index'], axis=1)
last_target2 = target_data2['date'].iloc[-1]

#print('last date to be used',last_target)
#print(target_data)

date = targets.loc[0, x]
ff_row = ind_val[ind_val == date].index[0]

ff_last = ind_val[ind_val == last_target].index[0]
ff_last2 = ind_val[ind_val == last_target2].index[0]
#print('last value for FF_last', ff_last)

ff_start = ff_row + 350
ff_latest = ff_row + 4
ff_end = ff_row - 220
ff_data = FF3.iloc[ff_latest:ff_last+1, 0:4]
ff_data.reset_index(inplace=True)
ff_data = ff_data.drop(['index'], axis=1)

ff_data2 = FF3.iloc[ff_end:ff_last2+1, 0:4]
ff_data2.reset_index(inplace=True)
ff_data2 = ff_data2.drop(['index'], axis=1)
```

```
differences = len(ff_data) - len(target_data)
df3 = pd.concat([target_data2, ff_data2], axis=1, join = 'inner').drop(['index'], axis = 1)



df2 = pd.concat([target_data, ff_data], axis=1, join = 'inner').drop(['index'], axis = 1)
df2 = df2.fillna(method = 'ffill')
#print(df2.head(20))

X = df2[['Mkt-RF', 'SMB', 'HML']]
y = df2[x]
beta_list = []
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size= 0.1, random_state = 0)
regressor = LinearRegression()
modell = regressor.fit(X_train, y_train)
coeff_df = pd.DataFrame(modell.coef_, X.columns, columns = ['Coefficient'])
intercept = modell.intercept_
#print('R-squared',modell.score(X,y))
#print(coeff_df)

print('__'*20)
X = sm.add_constant(X)
model = sm.OLS(y,X).fit()
predictions = model.predict(X)
#print(model.summary())
print('Predictions: ',predictions[0:220])
beta_list.append(model.params)
betas = pd.DataFrame(beta_list)
print(betas)
const = betas.loc[0, 'const']; Mkt = betas.loc[0, 'Mkt-RF']; SMB = betas.loc[0, 'SMB']
HML = betas.loc[0, 'HML']
df3['abnormal'] = df3[x] - (Mkt*df3['Mkt-RF'] + SMB*df3['SMB'] + HML*df3['HML'] + const)

predict = pd.DataFrame({'predicted':predictions[0:len(df3)]})
print(predict)
```

```
            print('__' * 20)
            print(df3)

        except:
            erores.append(x)

print('Printing missed values',erores)
print("--- %s seconds ---" % (time.time() - start_time))
```

Abnormal Returns Statistics

The following section of code depicts one of the approaches to calculate mean and median abnormal returns within the event window for targeted companies and benchmark companies. Once the abnormal returns are ordered, variances of individual abnormal returns prior to the announcement are calculated, using an approach to account for missing values in the dataset. Following the above, T-statistics and respective p-values for both the targeted companies and their peers are computed. The last section of the code counts unique companies in the peer sample across the whole observed period (2010 to 2019).

(the code below deals with targets' peers, the same code with few adjustments was applied on target companies)

```
import os
import pandas as pd
import numpy as np
import re
import glob
import time
from sklearn.linear_model import LinearRegression
import datetime
from datetime import datetime as dt
from pandas.tseries.offsets import BDay
import statsmodels.formula.api as sm
from scipy import stats


vict = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/tgt_date_ordered.xlsx")).drop(['Unnamed: 0'], axis = 1)
tgt_peer = [os.path.basename(x) for x in
            sorted(glob.glob("/Users/karelsarapatka1/Desktop/regression/actual/*"))]
df=
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/regression/abnorma
l_targets.xlsx")).drop(['Unnamed: 0'], axis = 1)


peer_variances = []
abnormal_r=[]
for x in df.columns:
    dff =
```

```
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/regression/abnorma
l/abn_"+x+".xlsx")).drop(['Unnamed: 0'], axis = 1)
    date = vict.loc[0, x]
    day = dff[dff['date'] == date].index.values.astype(int)[0]
    day_m = day + 3
    day_p = day - 3
    for y in dff.columns:
        if y != x and y!= 'date' and y!= 'dummy' and y!= 'mean_peer' and y!= 'median_peer':

            company = dff.loc[day_m+1:,y]
            variances = np.nanvar(company)
            peer_variances.append(variances)

            abnormals = pd.Series(dff.loc[day_p:day_m,y])
            abnormals.dropna(inplace=True); abnormals = abnormals.reset_index(drop = True)
            abnormal_r.append(abnormals)

abnormal_r = pd.concat(abnormal_r,axis= 1, join = 'outer')
median_ab = abnormal_r.median(axis= 1); mean_abn = abnormal_r.mean(axis= 1)

print(abnormal_r)
print('Median of abnormal returns', median_ab); print('Mean of abnormal returns', mean_abn)

obs = len(peer_variances)
print(obs)

mean_vr = np.sum(peer_variances)/(obs**2)
mean_std = np.sqrt(mean_vr)

t_stat = mean_abn / mean_std; print('tstat with sqrt mean var', t_stat)

pval =(stats.t.sf(np.abs(t_stat), obs - 2) * 2); pv2 = pd.DataFrame(pval);
print('P-values for t-statistic',pv2)
```

```
pp = (1 - stats.t.cdf(abs(t_stat),obs - 2))*2
pval3 = pd.DataFrame(pp); print(pval3)



column_val = abnormal_r.columns.values.tolist()
cmp = pd.DataFrame(column_val)
oneCol = []
for k in cmp.columns:
    oneCol.append(cmp[k])
oneCol = pd.concat(oneCol, ignore_index = True).transpose()
combo = pd.DataFrame(oneCol)
counter = combo.apply(lambda x: len(x.unique()))
print('Count of unique companies in the peer group',counter)
```

Absolute returns and Returns to Daily Lows

To calculate absolute returns to the lowest point (declines) in the observed period and cumulative returns from day 0 until day 22 (full market trading month). The code provided below aims at organizing price data (close and daily low) according to observed events pertaining to respective industry groups. Locally stored excel files from previous operations are used and means(medians) of price declines are calculated.

```
import os
import pandas as pd
import numpy as np
import re
import glob


vict = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/tgt_date_ordered.xlsx")).drop(['Unnamed: 0'], axis = 1)
lows = [os.path.basename(x) for x in sorted(glob.glob("/Users/karelsarapatka1/Desktop/bid_ask/low/*"))]
tgt_peer = [os.path.basename(x) for x in sorted(glob.glob("/Users/karelsarapatka1/Desktop/pp_test/*"))]


tgt_col = vict.columns.tolist()
peer_comps = []
whole_sample = []


for i in tgt_peer:
    target = re.split('_|, |\.|\n', i)[1]
    df = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/pp_test/"+i)).drop(['Unnamed: 0'], axis = 1)
    df2 = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/bid_ask/low/"+i)).drop(['Unnamed: 0'], axis = 1)


    date = vict.loc[0, target]
    index = df[df['date']==date].index.values.astype(int)[0]


    peers = df.loc[index+1, df.columns != target]
    peers = peers.drop(df.columns[df.columns[:].str.contains('date')])
    peer_close = pd.DataFrame(peers).transpose()
```

```
    peer_close = peer_close.reset_index(drop = True)

    peer_lows = df2.loc[:index, df2.columns != target]
    peer_lows.drop(list(df2.filter(regex='date')), axis=1, inplace=True)

    data = []
    for x in peer_lows.columns:
        try:
            column = pd.Series(peer_lows.loc[:, x])
            column.dropna(inplace= True)
            close_p = pd.Series(peer_close.loc[0,x])
            column = column.append(close_p)
            column = column.reset_index(drop=True)
            data.append(column)
        except:
            print('companies not found:', x)

    data = pd.concat(data, axis=1, join='outer')
    datas = data.iloc[::-1]
    datas = datas.reset_index(drop=True)
    print(datas)

    for y in datas.columns:
        col = pd.Series(datas.loc[:, y])
        col.dropna(inplace=True)
        col = col.reset_index(drop=True)
        whole_sample.append(col)

whole_sample = pd.concat(whole_sample, axis=1, join='outer')
whole_sample.columns = np.arange(len(whole_sample.columns))

daily_r = []
absolute = {}
for x in whole_sample.columns:
```

```
    prc_change = whole_sample[x].pct_change(-1)
    daily_r.append(prc_change)

    absolute[x] = []
    for s in range(len(whole_sample[x])):
        abs_changes = (whole_sample.loc[s, x] / whole_sample.loc[0, x]) - 1
        absolute[x].append(abs_changes)

daily_r = pd.concat(daily_r, axis=1, join='outer')
print('daily returns:', daily_r)
abs = pd.DataFrame(absolute)
print('absolute returns to lowest', abs)

# abs.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/regression/r_tolow/absolute_changes.xlsx"))
means = abs.mean(axis=1)
print('targets post announcement mean returns to lowest', means)
medians = abs.median(axis=1)
print('targets post ann. median returns to lowest:', medians)
```

Tonal Uncertainty

The following lines of code describe the approach for extracting tonal words from 10-K filings directly from the SEC database. This approach circumvents the need to first download the whole document from SEC's Edgar, thus saving considerable amount of memory (previously downloaded .txt format filings require around 10MB of memory per file, which in case of our target companies only translates to roughly 21,14 GB). Although Ashraf (2017) proposed a method for extraction of Loughran and McDonald's (2011) tonal words, his approach was largely dysfunctional due to deprecation of applied libraries.

To search for companies' filings on Edgar, it is necessary gather respective CIK codes, accepted as company identifier input on Edgar.
The first step preceding the tonal words extraction is thus assignment of CIK number to each company and storing this CIK-company list locally for later use.

```
from __future__ import print_function

import time

import intrinio_sdk

from intrinio_sdk.rest import ApiException

from pprint import pprint

import os

import pandas as pd


targets =

pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/cap_peers.xlsx"))

targ = targets.loc[:,'Company']

all_comp =

pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/bench_comp.xlsx"))

vict = pd.read_excel(os.path.join(os.path.dirname(__file__),

"/Users/karelsarapatka1/Desktop/tgt_date_ordered.xlsx")).drop(['Unnamed: 0'], axis = 1)

t_0 = vict.loc[0, 'GNI']


dff = []

for x in all_comp.columns:

    company = pd.Series(all_comp.loc[:,x])

    company.dropna(inplace = True)

    dff.append(company)
```

```
dff.append(targ)
combined = pd.concat(dff, ignore_index = True)
combined = combined.drop_duplicates()
combined.dropna(inplace = True)

master = []

dicts = {}
dicts['company'] = combined
dicts['cik'] = ''
print(dicts['company'])

intrinio_sdk.ApiClient().configuration.api_key['api_key'] = '*****'

security_api = intrinio_sdk.SecurityApi()

for i in combined:
    identifier = i
    dicts = {}
    try:
      api_response = security_api.get_security_by_id(identifier)
      cik_num = api_response.cik

      dicts['company'] = i
      dicts['cik'] = cik_num

      master.append(dicts)

      time.sleep(0.5)

      print(i,cik_num)
      print(master)
```

```
    cik_df = pd.DataFrame(master)
    cik_df.to_excel(os.path.join(os.path.dirname(__file__),
                                 "/Users/karelsarapatka1/Desktop/cik_num.xlsx"))


  except ApiException as e:
    print("Exception when calling SecurityApi->get_security_by_id: %s\r\n" % e)
```

Once CIK-codes have been assigned, the code for the actual extraction of tonal uncertainty words can be applied. In brief, the code below turns extracts 10-K filings for each available year directly from SEC, reads it whole in html format and converts it into text. The resulting text is converted to list of word with a library for natural language processing, which assures exclusion of html tags and other non-language features. Consequently, the list is matched with the dictionary of tonal words provided by Loughran and McDonald. Occurrences of searched words are counted, ordered firm-wise according to corresponding year of 10K filing and stored locally in xlsx format for later processing.

```
import requests
import pandas as pd
from bs4 import BeautifulSoup
import re
import urllib.request
import os
import urllib3
from datetime import datetime
from dateutil import parser
from collections import Counter
from itertools import chain
import time
import requests
import string
from lxml.html.soupparser import fromstring
import nltk
from nltk.tokenize import RegexpTokenizer
import numpy as np


tokenizer = RegexpTokenizer('\w+')
```

```
def countOccurences(str, word):
    a = str.split(" ")
    count = 0
    for i in range(0, len(a)):
        if (word == a[i]):
            count = count + 1
    return count

def counterFct(str, word):
    count = 0
    for i in range(0, len(str)):
        if(word == str[i]):
            count = count + 1
    return count

def countings(strg, word):
    c = Counter(strg)
    cnt = c[word]
    return cnt

#base URL for the SED Edgar browser
endpoint = r"https://www.sec.gov/cgi-bin/browse-edgar"

#cik_num = 3545
filing_type = "10-k"
#, 2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009]

cik_num = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/cik_num1.xlsx"))
ciks = cik_num['cik']
erors = []
word = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/tonal_unc.xlsx"))
```

```
words = word['words'].tolist()

#define parameters dictionary, dateb = date before - set to anything before january the first 2019,
output=''=> kept at the standard html now
# count = no. of results you want to see with your request, if not set = max = 40
for cik in ciks:
    start_time = time.time()
    word_df = pd.DataFrame({'words': words})
    param_dict = {'action': 'getcompany',
                  'CIK': cik,
                  'type': '10-k',
                  'dateb' : '20190201',
                  'owner': 'exclude',
                  'start': '',
                  'output': '',
                  'count': '20'}

    #request the url, and then parse the response
    response = requests.get(url = endpoint, params=param_dict)
    soup = BeautifulSoup(response.content,'html.parser')

    # find the document table with our data, class argument = looking for a specific object
    doc_table = soup.findAll('table', class_='tableFile2')

    # define a base url that will be used for link building.
    base_url_sec = r"https://www.sec.gov"

    master_list = []


    # loop through each row in the table. (all the filings are displayed in tabular form - col0 are filing types,
etc.)
    for row in doc_table[0].find_all('tr'):
```

```
        # find all the columns
        cols = row.find_all('td')

        # if there are no columns move on to the next row.
        if len(cols) != 0:
            fil_date = cols[3].text.strip()
            fil_yr = parser.parse(fil_date).year
            fil_type = cols[0].text.strip()
            if fil_yr >= 2008:
                if fil_type == '10-K':
                    # grab the text, strip removes characters both from left and right based on the arguments
(strip anything not useful here), filing numbers have only a tag.
                    filing_type = cols[0].text.strip()
                    filing_date = cols[3].text.strip()

                    filing_num = cols[4].text.strip()

                    # find the links - according to tags, can use selector gadget
                    filing_doc_href = cols[1].find('a', {'href': True, 'id':'documentsbutton'})
                    filing_int_href = cols[1].find('a', {'href': True, 'id': 'interactiveDataBtn'})
                    filing_num_href = cols[4].find('a')

                    # grab the first href, some may not have link
                    if filing_doc_href != None:
                        filing_doc_link = base_url_sec + filing_doc_href['href']
                    else:
                        filing_doc_link = 'no link'

                    # grab the second href
                    if filing_int_href != None:
                        filing_int_link = base_url_sec + filing_int_href['href']
                    else:
                        filing_int_link = 'no link'
                    # grab the third href
```

```
            if filing_num_href != None:
                filing_num_link = base_url_sec + filing_num_href['href']
            else:
                filing_num_link = 'no link'
            # create and store data in the dictionary
            # problems with file_type : output is only 10-K/A
            file_dict = {}
            file_dict['file_type'] = filing_type
            file_dict['file_number'] = filing_num
            file_dict['file_date'] = filing_date
            file_dict['links'] = {}
            file_dict['links']['documents'] = filing_doc_link
            file_dict['links']['interactive_data'] = filing_int_link
            file_dict['links']['filing_number'] = filing_num_link

            year = parser.parse(filing_date).year
            word_df[filing_date] = ''

            # append dictionary to master list
            master_list.append(file_dict)

# Loop through to get the links from the dictionary

error_list = []
for report in master_list[:]:
    try:
        doc_filing = report['links']['documents']
        date = report['file_date']
        year = parser.parse(date).year
        doc_files = requests.get(doc_filing)

        #proceed to next page w/ 10-K for a specific year, extract tag with url
        soup2 = BeautifulSoup(doc_files.content, 'html.parser')
        doc_table2 = soup2.findAll('table', class_='tableFile')
```

```
            doc_row = doc_table2[0].find_all('tr')[1]
            colonel = doc_row.find_all('td')[2]
            fil_doc_href = colonel.find_all('a', {'href':True})

            #extract url in text form and create a full url of html-10K
            s1 = str(fil_doc_href[0])
            start = s1.find("") + 1
            end = s1.find("", start)
            fil_url = s1[start:end]
            url_k = base_url_sec + fil_url
            print('__'*20)
            print('DOC_URL', url_k)
            print('DATE:', date)

            #open and read 10K, extract text elements and apply Natural Language processor to tokenize
elements
            r = requests.get(url_k)
            html = r.text
            soup3 = BeautifulSoup(html, 'lxml')
            text = soup3.get_text()
            tokens = tokenizer.tokenize(text)

            a_counter = Counter(tokens)

            #use Counter function and count number of tokens corresponding to Loughran & McDonald
            master = []
            for word in words:
                word_count = a_counter[word]
                master.append(word_count)
            word_df[date] = master

        except:
            error_list.append(cik)
```

```
        print('error has occured', error_list)


    print(word_df)
    print("--- %s seconds ---" % (time.time() - start_time))
    word_df.to_excel(os.path.join(os.path.dirname(__file__),
                                  "/Users/karelsarapatka1/Desktop/Loughran/" + str(cik) + ".xlsx"))
```

Once all word occurrences are collected and saved separately for individual companies, we proceed with year-level ordering. The weights are calculated and assigned for each firm/year. Lastly, quintiles for each year are computed and companies are assigned either 0 or 1, according to the sum of their uncertainty-word values. Data frames were saved throughout the code to allow for sanity test of the output.
The whole process can be seen in the commented code below:

```
import os
import pandas as pd
import glob
import re
import time
import numpy as np


tone_list = [os.path.basename(x) for x in sorted(glob.glob("/Users/karelsarapatka1/Desktop/Loughran/*"))]
ciks = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/cik_num1.xlsx"))
years = (2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009)
word = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/tonal_unc.xlsx"))
words = word['words'].tolist()


start_time = time.time()


#Getting tf(i), order all companies and respective occurrence of words according to filing years, save in
xlsx format
for year in years:
    tones_df = pd.DataFrame({'Words':words})
    for i in os.listdir("/Users/karelsarapatka1/Desktop/Loughran/"):
```

```
        tones = []
        try:
            tgt = re.split('_|, |\.|\n', i)
            cik = int(tgt[0])
            df = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/Loughran/"+i)).drop(['Unnamed: 0', 'words'], axis = 1)
            row = (ciks.loc[ciks['cik'] == cik, 'company']).tolist()
            df.columns = pd.to_datetime(df.columns)
            df.columns = pd.to_datetime(df.columns).year

            tones_df[row[0]] = df[year]
            print(year)

        except:
            print( 'Error has occured for_____:',i)

    #tones_df.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/Lough_order/" + str(year) + ".xlsx"))

#Calculate respective weights for each word according to Formula in the thesis (get df,a, N)
tones_yr = os.listdir("/Users/karelsarapatka1/Desktop/Lough_order/")
df4 = pd.DataFrame({'words':words})
lengths = {}
averages = {}
num_of_k = []
sums = []
sum_word = {}
for i in tones_yr:
    try:
        y = re.split('_|, |\.|\n', i)
        year = y[0]
        df_y = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/Lough_order/"+i)).drop(['Unnamed: 0'], axis = 1)
```

```
        nums = len(df_y.columns) - 2
        num_of_k.append(nums)
        lengths[year] = [nums]

        df4[year] = df_y['Counter']

        df5 = df_y.drop(['Counter', 'Words'], axis = 1)
        sum_all = df5.sum().sum()
        sum_w = df5.sum(axis = 1)
        sum_word[year] = sum_w
        sums.append(sum_all)
        averages[year] = [sum_all / nums]

    except:
        print('Error for the following:',i)

sum_word = pd.DataFrame(sum_word)
word_sums = sum_word.sum(axis = 1)

aver = pd.DataFrame.from_dict(averages)
df4['totals']= np.sum(df4,axis=1)
N = sum(num_of_k)
dfi = df4['totals']

RHS = []
for x in dfi:
    rhs = np.log(N/x)
    RHS.append(rhs)
rhs_term = pd.DataFrame({'rhs':RHS})

#get descriptive summary statistics for all years
for i in tones_yr:
    try:
        df6 = {}
```

```
        y = re.split('_|, |\.|\n', i)
        year = y[0]
        year_avg = aver.loc[0, year]
        a = 1/(1 + np.log(year_avg))
        #print('a = ',a)
        rhs_term['rhs_r'] = rhs_term['rhs']*a
        df_y =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/Lough_order/" +
i)).drop(['Unnamed: 0', 'Counter', 'Words'], axis=1)

        for z in df_y.columns:
            col = (1 + np.log(df_y[z]))*(rhs_term['rhs_r'])
            df6[z] = col

        df_loughran = pd.DataFrame(df6)

#df_loughran.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/Lough_orde
r/weights/" + str(year) + ".xlsx"))

    except:
        print('Not included in the set or error:', i)

tones_y = os.listdir("/Users/karelsarapatka1/Desktop/Lough_order/weights/")
means = []
medians = []
stds = []
for i in tones_y:
    try:
        df = pd.read_excel(os.path.join(os.path.dirname(__file__),
                                        "/Users/karelsarapatka1/Desktop/Lough_order/weights/"+ str(i))).drop(['Unnamed:
0'], axis=1)
        y = re.split('_|, |\.|\n', i)
        year = y[0]
        mean = np.mean(df.stack())
```

```
        means.append(mean)
        median = np.median(df.stack())
        medians.append(median)
        stand_dev = np.std(df.stack())
        stds.append(stand_dev)

    except:
        print('ERROR:', i)

mean_ = pd.DataFrame({'Means':means})
median_ = pd.DataFrame({'Medians':medians})
stds_ = pd.DataFrame({'std':stds})

mea = np.mean(mean_)
med = np.median(median_)
std = np.mean(stds_)

#get upper quintile for each year and assign either 0 or 1 if company is below or above, respectively, save
as xlsx.
for i in tones_y:
    print(i)
    try:
        df = pd.read_excel(os.path.join(os.path.dirname(__file__),
                                "/Users/karelsarapatka1/Desktop/Lough_order/weights/"+ str(i))).drop(['Unnamed:
0'], axis=1)
        y = re.split('_|, |\.|\n', i)
        year = y[0]
        words_firm = df.sum(axis = 0)
        firms = df.columns.values.tolist()
        df2 = pd.DataFrame({'firms':firms, 'sums':words_firm}).reset_index(drop= True,inplace = False)
        quintiles = np.percentile(df2.sums, 80)
        print('QUINTILE:', quintiles)
        df2['quint'] = np.where(df2['sums']>= quintiles, 1, 0)
```

```
#df2.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/Lough_order/quintile
s/" + str(year) + ".xlsx"))


    except:
        print('Exception occured:', i)
print("--- %s seconds ---" % (time.time() - start_time))
```

With few adjustments, the two segments of code above were repurposed for extraction of independent auditor companies (from Exhibit 23.1 of 10-K filing) and for collection of names of CEO and Board of Directors over the observed period (these were taken from directly from 10-K filings).

The most notable difference from to the above code for applications in CEO duality extraction was the use of full sentence tokenizer (nltk library). Which separated the filing content into full, and approximately correct, sentences. These were parsed for occurrence of commonly observed corporate title descriptions we collected manually from over 300 10-K filings.

The exceptions with respect to the first code segment are as follows:

1. Introduction of nltk punctuation module for natural sentences recognition.

```
try:
    _create_unverified_https_context = ssl._create_unverified_context
except AttributeError:
    pass
else:
    ssl._create_default_https_context = _create_unverified_https_context


nltk.download('punkt')
```

2. Extraction of text from 10-K filing, application of sentence tokenizer and counting of occurrences of corporate title words. Controlling for accidental co-occurrence (e.g. “Our former CEO and Chairman…”)

```
r = requests.get(url_k)
html = r.content
soup3 = BeautifulSoup(html, 'html.parser')
text = soup3.find_all(text = True)


output = ''
```

```
blacklist = ['[document]', 'noscript', 'header', 'html', 'meta', 'head', 'input', 'script', 'style']

for t in text:
    if t.parent.name not in blacklist:
        output += '{}'.format(t)

sentence = nltk.sent_tokenize(output)

master_sum = []
alt_counter = []
for word in words:
    alt_c = []
    for sent in sentence:
        if word in sent and 'separate' not in sent and 'former' not in sent:
            alt_c.append(1)
    sum_alt = np.sum(alt_c)
    alt_counter.append(sum_alt)
word_df[date] = alt_counter
```

Data ordering, calculation of ratios and determinants

All quarterly and year-end filing values are ordered to match campaign announcement dates while considering differing filing and reporting period dates. This leaves us ultimately with 32 fiscal quarters of interest. The code below can be applied to any income, cash flow and balance sheet statement with minor adjustments. For brevity's sake, I shall include only the segment dealing with *TACC*, other determinants' respective code is written in a similar fashion (with the difference in directories with either balance sheet or income statement data).

```
import pandas as pd
import numpy as np
import os
from pandas.tseries.offsets import BDay
import time

start_time = time.time()

Total_a = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/determinants/financials/totalassets.xlsx")).drop(['Unnamed: 0'], axis= 1)

tgt_dates = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/quarts.xlsx"))

dates = tgt_dates['Date']
quarterly = pd.DataFrame({'date': dates})

# income statement and cash flow statements on TTM-basis

for w in os.listdir("/Users/karelsarapatka1/Desktop/income_accs"):
    try:
        quarterly = pd.DataFrame({'date': dates})
        df = pd.read_excel(os.path.join(os.path.dirname(__file__),
                                        "/Users/karelsarapatka1/Desktop/income_accs/" + w))
        if 'Unnamed: 0' in df.columns.values.tolist():
            df = df.drop(['Unnamed: 0'], axis=1)

        col_list = df.columns
        col_ = col_list[col_list[:].str.contains('date')]
        col_ex = col_list[~col_list[:].str.contains('date')]
```

```
        #exclusion of sparsely populated data
        for x, z in zip(col_, col_ex):
            vals = []
            datess = []
            for i in dates:
                try:
                    listt = list(df[x])
                    len_l = len(df[x].dropna().unique().tolist())

                    if  len_l >= 10:
                        dats = df[x][(df[x] + BDay(40)) <= i].head(1).tolist()

                        if i.year == dats[0].year or i.year == dats[0].year + 1:
                            #(i - dats[0]).days<= 70:
                            row = listt.index(dats[0])
                            val = df.loc[row, z]

                            past_3q = df.loc[row:row + 3, z]
                            ttm = np.sum(past_3q)
                            datess.append(i)
                            vals.append(ttm)
                except:
                    print('Nowhere to be found', x, z)

            values = pd.DataFrame({'date': datess, z: vals})
            quarterly = pd.merge(quarterly, values, on='date', how='outer')

            quarterly.to_excel(os.path.join(os.path.dirname(__file__),
                                            "/Users/karelsarapatka1/Desktop/qtr_ordered/net_inc.xlsx"))
    except:
        print('Corrupted file name', w)

#balance sheet accounts, no sums across prior periods. Here, the file is fed directly without going through
directiories
"""for x, z in zip(col_, col_ex):
    vals = []
    datess = []
    for i in dates:
        try:
```

```
            listt = list(df[x])
            len_l = len(df[x].dropna().unique().tolist())

            dats = df[x][(df[x] + BDay(40)) <= i].head(1).tolist()

            if i.year == dats[0].year or i.year == dats[0].year + 1:
                row = listt.index(dats[0])
                val = df.loc[row, z]
                datess.append(i)
                vals.append(val)
        except:
            print('Nowhere to be found',x,z )

    values = pd.DataFrame({'date':datess, z:vals})
    quarterly = pd.merge(quarterly,values, on = 'date', how = 'outer')
    print(values)"""

#calculation of Accruals and with some adjustments, any ratio/value needed
ni = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/qtr_ordered/net_inc.xlsx")).drop(['Unnamed: 0'], axis= 1)
ocf = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/qtr_ordered/ocf.xlsx")).drop(['Unnamed: 0'], axis= 1)
ta = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/qtr_ordered/total_assets.xlsx")).drop(['Unnamed: 0'], axis= 1)

dates = ocf['date']
col_ex = ni.columns[~ni.columns[:].str.contains('date')]
col_ex2 = ocf.columns[~ocf.columns[:].str.contains('date')]

dif = len(col_ex) - len(col_ex2)
print(dif)

dfs = pd.DataFrame({'date': dates})
vals = []

ocf = ocf.fillna(0)
ni = ni.fillna(0)
df = []
#for x,y in zip(col_ex,col_ex2):
```

```
for x in col_ex:
    for y in col_ex2:
        if x == y:
            len_dif = ni[x].astype(bool).sum() - ocf[y].astype(bool).sum()
            dif = ni[x] - ocf[y]
            df.append(dif)

df = pd.DataFrame(df).transpose()
df.insert(column = 'date', value = dfs['date'], loc = 0)

tac = []
for x in col_ex2:
    for i in ta.columns[~ta.columns[:].str.contains('date')]:
        try:
            if x == i:
                tacc = df[x] / ta[i]
                tac.append(tacc)
        except:
            print('Not in df:', i)

tac = pd.DataFrame(tac).transpose()
tac.insert(column = 'date', value = dfs['date'], loc = 0)
targs =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/cap_peers.xlsx"))

nulls = []
for x in tac.columns:
    if x != 'date':
        try:
            for z in range(len(tac)):
                if tac.loc[z, x] >= 10 or tac.loc[z, x] <= -30:
                    tac = tac.drop([x], axis = 1)
        except:
            nulls.append(x)

print(tac)
tac.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/determinants/tacc.xls
x"))

print("--- %s seconds ---" % (time.time() - start_time))
```

P/E-Ratio

Several exceptions to the above occurred, since the data provided by Intrinio did not always prove to be either fully reflective of financial realities of companies in question or did not encompass financial history in its entirety.
One of such discrepant ratios was the Price-to-earnings ratio calculated below.

```
import pandas as pd
import os
from pandas.tseries.offsets import BDay

dff = pd.read_excel(os.path.join(os.path.dirname(__file__), "/Users/karelsarapatka1/Desktop/quarts.xlsx"))
dates = dff['Date']

def datecut (df):
    col_x = df.columns[~df.columns[:].str.contains('date')]
    col_d = df.columns[df.columns[:].str.contains('date')]
    return col_x, col_d

#step 1. = check whether collected eps matches ttm or qtr values = __qtr values__

#step 2. = if eps matches qtr values = sum past 4 months
eps_raw =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/adjbasiceps.xlsx")).
drop(['Unnamed: 0'], axis = 1)

val = []
ttm = []
col_x = datecut(eps_raw)[0]
col_d = datecut(eps_raw)[1]
for x,y in zip(col_x, col_d):
    col = eps_raw.loc[::-1,x].rolling(min_periods=1, window=4).sum()[::-1]
    date = eps_raw[y]

    values = pd.DataFrame({'date': date, x: col})
    print(values)
    ttm.append(values)
ttm = pd.concat(ttm,axis= 1, join = 'outer')

# step 3. = divide collected ttm EPS by shareprice at respective date
```

```
prices =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/determinants/bulk/eod.xlsx"))

col_ex = datecut(ttm)[0]
col_ed = datecut(ttm)[1]

# .drop(['Unnamed: 0'], axis = 1)
comp_list = prices.columns.values.tolist()
pe_ratio = []
counter = 0
for x, y in zip(col_ex, col_ed):
    counter = counter + 1
    eod_price = []
    eod_date = []
    location = [i for i, z in enumerate(comp_list) if z == x]

    if location:
        comp_col = location[0]
        date_col = location[0] - 1

        for d in ttm[y]:
            price_dates = prices.iloc[:, date_col]
            loc_d = [i for i, z in enumerate(price_dates) if z == d]

            if loc_d:
                eod_p = prices.iloc[loc_d[0], comp_col]
                eod_price.append(eod_p); eod_date.append(d)

            if not loc_d:
                date_shift = d - BDay(2)
                loc_misdated = [i for i, z in enumerate(price_dates) if z == date_shift]
                if loc_misdated:
                    eod_p2 = prices.iloc[loc_misdated[0], comp_col]
                    eod_price.append(eod_p2); eod_date.append(d)

    eod_df = pd.DataFrame({'date': eod_date, x:eod_price})
    P_E = eod_df[x] / ttm[x]
    PE_df = pd.DataFrame({'date':ttm[y] ,x:P_E})
    pe_ratio.append(PE_df)
```

```
    print(counter)

pe_ratio = pd.concat(pe_ratio,axis= 1, join = 'outer')

# step 4. = match ttm eps on respective quarter dates of interest
col_x = datecut(pe_ratio)[0]
col_d = datecut(pe_ratio)[1]

collection = pd.DataFrame({'date':dates})
for x, z in zip(col_d, col_x):
    vals = []
    datess = []
    for i in dates:
        try:
            listt = list(pe_ratio[x])
            dats = pe_ratio[x][(pe_ratio[x] + BDay(40)) <= i].head(1).tolist()

            if i.year == dats[0].year or i.year == dats[0].year + 1:
                row = listt.index(dats[0])
                val = pe_ratio.loc[row, z]
                datess.append(i)
                vals.append(val)
        except:
            print('Nowhere to be found',x,z )

    values = pd.DataFrame({'date':datess, z:vals})
    print(values)
    collection = pd.merge(collection,values, on = 'date', how = 'outer')

print(collection)
print('__' * 20)
collection.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/pe_ordered.xls
x"))
```

Beneish M-Score

The last of the determinants to be displayed in detail is the Beneish M-Score.
Due to our emphasis on correctly assigning SEC filings data to relevant time-windows, the varying filing dates and fiscal quarter periods made any filtering and data processing comparably intricate. As in the previous code segments, columns had to be handled one by one on a row basis rather than a whole, substantially increasing runtime of the code.

```
import pandas as pd
import numpy as np
import os
from datetime import timedelta
from pandas.tseries.offsets import BDay
import time


def dateselect(df):
    col_x = df.columns[~df.columns[:].str.contains('date')]
    col_d = df.columns[df.columns[:].str.contains('date')]
    return col_x, col_d

def locator(x, df, new_name):
    df_loc = df.columns.get_loc(x)
    cutout = df.iloc[:, df_loc - 1:df_loc + 1].dropna()
    cutout.rename(columns={list(cutout)[0]: x, x: new_name}, inplace=True)
    return cutout

def shifter(i, x, date, df):
    dat = i + pd.Timedelta(-1, unit='y')
    dat_p = dat + pd.Timedelta(15, unit='d')
    dat_m = dat + pd.Timedelta(-15, unit='d')
    mask = (date > dat_m) & (date <= dat_p)
    shift = df.loc[mask]
    return shift

def intersect(df, df1):
    inter = [x for x in df if x in df1 and '.1' not in str(x)]
    return inter

dirnm = '/Users/karelsarapatka1/Desktop/'
```

```
targets = pd.read_excel(os.path.join(os.path.dirname(__file__), dirnm + "cap_peers.xlsx")).drop(['Unnamed:
0'], axis=1)
targs = targets['Company']

# 1.part = GMI, grossm(t-1)/grossm(t)
gross = pd.read_excel(os.path.join(os.path.dirname(__file__), dirnm +
"beneish/grossmargin_edit.xlsx")).drop(['Unnamed: 0'], axis = 1)
col_x = dateselect(gross)[0]

GMI = []
outarange = []
for x in col_x:
    try:
        gmi_ = locator(x,gross,'gmi')
        date = gmi_['gmi']
        gmi_v = []
        gmi_d = []
        for i in date:
            try:
                gmi_shift = shifter(i,x,date,gmi_)
                gmi = list(gmi_shift['gmi'])

                if gmi:
                    gmi_d.append(i)
                    gmi_v.append(gmi[0])
            except:
                outarange.append(x)
                print('missing val', x)

        gmis = pd.DataFrame({x:gmi_d, 'gmi_': gmi_v})
        whole_gmi = gmi_.merge(gmis, on=x, how='outer')
        whole_gmi['GMI'] = whole_gmi['gmi'] / whole_gmi['gmi_']

        gmi_df = pd.DataFrame({'date': whole_gmi[x], x: whole_gmi['GMI']})
        GMI.append(gmi_df)
    except:
        outarange.append(x)

GMI = pd.concat(GMI, axis = 1, join = 'outer')
print(GMI)
```

```
GMI.to_excel(os.path.join(os.path.dirname(__file__),dirnm + "/beneish/raw/gmi.xlsx"))

# 2.part DSR = (rcv_t/sales_t)/(rcv_t-1/sales_t-1)
dsr_1 = pd.read_excel(os.path.join(os.path.dirname(__file__),dirnm + "arturnover.xlsx")).drop(['Unnamed:
0'], axis = 1)
col_list = list(dsr_1.columns)
dsr_x = dateselect(dsr_1)[0]

sr_prior = []
DSR_total = []
for x in dsr_x:
    try:
        dsr_ = locator(x, dsr_1, 'dsri')
        date = dsr_['dsri']

        dsr_v = []
        dsr_d = []
        for i in date:
            try:
                dsr_shift = shifter(i,x,date,dsr_)
                dsr = list(dsr_shift['dsri'])

                if dsr:
                    dsr_v.append(dsr[0])
                    dsr_d.append(i)
            except:
                print('missing val', x)

        drs_df = pd.DataFrame({x:dsr_d, 'dsri_':dsr_v})
        whole_dsr = dsr_.merge(drs_df, on = x, how = 'outer')
        whole_dsr['DSR'] = whole_dsr['dsri']/whole_dsr['dsri_']

        dsr_indic = pd.DataFrame({'date':whole_dsr[x], x:whole_dsr['DSR']})
        DSR_total.append(dsr_indic)

    except:
        print('Missing val', x)
DSR_total = pd.concat(DSR_total,axis = 1, join = 'outer')
DSR_total.to_excel(os.path.join(os.path.dirname(__file__), dirnm + "beneish/raw/dsr.xlsx"))
```

```
# Part 3 LEVI = D/A
TA = pd.read_excel(os.path.join(os.path.dirname(__file__),dirnm +
"beneish/totalassets.xlsx")).drop(['Unnamed: 0'], axis = 1)
debt = pd.read_excel(os.path.join(os.path.dirname(__file__),dirnm + "beneish/debt.xlsx")).drop(['Unnamed:
0'], axis = 1)

col_x2 = list(dateselect(TA)[0]); col_x3 = list(dateselect(debt)[0])
coincide = [x for x in col_x2 if x in col_x3]

Leverage = []
debt_asset = []
for x in coincide:
    df1 = locator(x,TA,'T_A')
    df2 = locator(x,debt,'debt')

    combined_ = df2.merge(df1, on = x, how = 'inner')
    combined_['D/A'] = combined_['debt']/combined_['T_A']

    date = combined_[x]
    try:
        val_da = []
        date_da = []
        for i in date:
            try:
                lever_shift = shifter(i,x,date,combined_)
                D_A = list(lever_shift['D/A'])

                if D_A:
                    val_da.append(D_A[0])
                    date_da.append(i)
            except:
                print('Missing val', x)
        D_A_ = pd.DataFrame({x:date_da,'D/A_':val_da})
        whole_DA = combined_.merge(D_A_, on = x, how = 'outer')
        whole_DA['LEVI'] = whole_DA['D/A']/whole_DA['D/A_']

        lever = pd.DataFrame({'date':whole_DA[x],x:whole_DA['LEVI']})
        debtotasset = pd.DataFrame({'date':whole_DA[x], x:whole_DA['D/A'] })

        Leverage.append(lever)
```

```
        debt_asset.append(debtotasset)
    except:
        print('Corrupted filename:',x)

Leverage = pd.concat(Leverage,axis = 1, join = 'outer')
debt_asset = pd.concat(debt_asset,axis = 1, join = 'outer')
Leverage.to_excel(os.path.join(os.path.dirname(__file__), dirnm + "beneish/raw/LEVI.xlsx"))
debt_asset.to_excel(os.path.join(os.path.dirname(__file__), dirnm + "debt_toasset.xlsx"))

# part 4 Asset Quality index (AQI) = 1 - [(PPE + CA)/TA]/last year's
ppe = pd.read_excel(os.path.join(os.path.dirname(__file__),dirnm +
"beneish/netppe.xlsx")).drop(['Unnamed: 0'], axis = 1)
ca = pd.read_excel(os.path.join(os.path.dirname(__file__),dirnm +
"beneish/totalcurrentassets.xlsx")).drop(['Unnamed: 0'], axis = 1)
TA = pd.read_excel(os.path.join(os.path.dirname(__file__),dirnm +
"beneish/totalassets.xlsx")).drop(['Unnamed: 0'], axis = 1)

col_x4 = dateselect(ppe)[0]
col_x5 = dateselect(ca)[0]
col_x6 = dateselect(TA)[0]

intersect1 = intersect(col_x4, col_x5)
intersect2 = intersect(col_x6, intersect1)

missing_vals = []
AQI = []
for x in intersect2:
    ppe_df = locator(x, ppe, 'ppe')
    ca_df = locator(x, ca, 'ca')
    ta_df = locator(x, TA, 'ta')

    combine = ta_df.merge(ca_df, on = x, how = 'inner')
    combi = combine.merge(ppe_df, on = x, how = 'inner')
    combi['aqi1'] = 1-((combi['ca'] + combi['ppe'])/combi['ta'])

    date_c = combi[x]
    aqi_v = []
    aqi_d = []
    for i in date_c:
        try:
```

```
            shift = shifter(i,x,date_c,combi)
            aqi_ = list(shift['aqi1'])

            if aqi_:
                aqi_v.append(aqi_[0])
                aqi_d.append(i)
        except:
            missing_vals.append(x)

    aqis = pd.DataFrame({x:aqi_d,'aqi_':aqi_v})

    if not combi.empty:
        whole_aqi = combi.merge(aqis, on=x, how='outer')
        whole_aqi['AQI'] = whole_aqi['aqi1']/whole_aqi['aqi_']
        whole = pd.DataFrame({'date':whole_aqi[x], x:whole_aqi['AQI']})
        AQI.append(whole)

AQI = pd.concat(AQI,axis = 1, join = 'outer')
AQI.to_excel(os.path.join(os.path.dirname(__file__), dirnm + "beneish/raw/AQI.xlsx"))

# part 5 = SGI = Sales(t)/sales(t-1)
sales = pd.read_excel(os.path.join(os.path.dirname(__file__), dirnm +
"beneish/sales_.xlsx")).drop(['Unnamed: 0'], axis = 1)

sales_x = dateselect(sales)[0]
SGI = []
for x in sales_x:
    sales_cut = locator(x,sales,'sales')
    sales_date = sales_cut[x]

    sgi_d = []
    sgi_v = []
    for i in sales_date:
        try:
            sales_shift = shifter(i,x, sales_date,sales_cut)
            sgi = list(sales_shift['sales'])

            if sgi:
                sgi_d.append(i)
                sgi_v.append(sgi[0])
```

```
        except:
            print('missing val', x)
    sgis = pd.DataFrame({x:sgi_d, 'sales_':sgi_v})

    if not sales_cut.empty:
        whole_sgi = sales_cut.merge(sgis, on = x, how = 'outer')
        whole_sgi['SGI'] = whole_sgi['sales']/whole_sgi['sales_']
        print(whole_sgi)
        sgi_df = pd.DataFrame({'date':whole_sgi[x], x:whole_sgi['SGI']})
        SGI.append(sgi_df)

SGI = pd.concat(SGI, axis = 1, join = 'outer')
SGI.to_excel(os.path.join(os.path.dirname(__file__), dirnm + "beneish/raw/SGI.xlsx"))

misses = []
for x in targs:
    if x not in SGI.columns:
        misses.append(x)
print('__'*20)
print(len(misses))

# Part 6 = DEPI = (deprec_rate(t-1)/depr_rate(t), depr_rate = depreciation/depreciation + PPE;
depreciation already at ttm values
dep = pd.read_excel(os.path.join(os.path.dirname(__file__),dirnm +
"beneish/depreciationexpense.xlsx")).drop(['Unnamed: 0'], axis = 1)
ppe_x = dateselect(ppe)[0]
dep_x = dateselect(dep)[0]
intersec_depi = [x for x in ppe_x if x in dep_x]

DEPI = []
for x in intersec_depi:
    ppe_df = locator(x, ppe, 'ppe')
    dep_df = locator(x, dep, 'dep')

    combine_dep = ppe_df.merge(dep_df, on=x, how='inner')
    combine_dep['depi'] = combine_dep['dep']/(combine_dep['dep']+combine_dep['ppe'])
    date_dep = combine_dep[x]

    dep_d = []
    dep_v = []
```

```
    for i in date_dep:
        try:
            sales_shift = shifter(i, x, date_dep, combine_dep)
            depi = list(sales_shift['depi'])

            if depi:
                dep_d.append(i)
                dep_v.append(depi[0])
        except:
            print('missing val', x)

    depis = pd.DataFrame({x: dep_d, 'depi_': dep_v})

    if not combine_dep.empty:
        whole_depi = combine_dep.merge(depis, on=x, how='outer')
        whole_depi['DEPI'] = whole_depi['depi_']/whole_depi['depi']
        depi_df = pd.DataFrame({'date': whole_depi[x], x: whole_depi['DEPI']})
        DEPI.append(depi_df)

DEPI = pd.concat(DEPI, axis=1, join='outer')
DEPI.to_excel(os.path.join(os.path.dirname(__file__), dirnm +"beneish/raw/DEPI.xlsx"))

# step 7.1. : sort SGA to reflect ttm values, save locally
sga = pd.read_excel(os.path.join(os.path.dirname(__file__),dirnm
+"beneish/sgaexpense.xlsx")).drop(['Unnamed: 0'], axis = 1)
sga_x = dateselect(sga)[0]

SGA_ttm = []
for x in sga_x:
    sga_cut = locator(x,sga,'sga')
    date = sga_cut[x]
    vals = list(sga_cut['sga'])

    sga_ttm = []
    date_ttm = []
    for i in date:
        s_loc = list(sga_cut[x]).index(i)
        s_shift = shifter(i,x,date,sga_cut).index
        if not s_shift.empty:
            try:
```

```
            sum_sga = sum(vals[s_loc:s_shift[0]])
            sga_ttm.append(sum_sga)
            date_ttm.append(i)
        except:
            print('Exception:',x)

    sga_tt = pd.DataFrame({'date':date_ttm, x:sga_ttm})
    if len(sga_tt)>= 1:
        SGA_ttm.append(sga_tt)
        print(sga_tt)

sga_ttm = pd.concat(SGA_ttm,axis = 1, join = 'outer')

# Step 7.2 SGAI = [SGA(t)/Sales(t)]/[SGA(t-1)/Sales(t-1)]
sga_ttm_x = dateselect(sga_ttm)[0]
intersect_sgai = [x for x in sga_ttm_x if x in sales_x]

SGAI = []
for x in intersect_sgai:
    sga_df = locator(x,sga, 'sg&a')
    sales_df = locator(x, sales, 'sales')

    combine_sgai = sga_df.merge(sales_df, on=x, how='inner')
    combine_sgai['sgai'] = combine_sgai['sg&a'] / combine_sgai['sales']
    date_sgai = combine_sgai[x]

    sga_d = []; sga_v = []
    for i in date_sgai:
        try:
            sga_shift = shifter(i, x, date_sgai, combine_sgai)
            sgai = list(sga_shift['sgai'])

            if sgai:
                sga_d.append(i)
                sga_v.append(sgai[0])
        except:
            print('missing val', x)

    sgais = pd.DataFrame({x: sga_d, 'sgai_': sga_v})
```

```
    if not combine_sgai.empty:
        whole_sgai = combine_sgai.merge(sgais, on=x, how='outer')
        whole_sgai['SGAI'] = whole_sgai['sgai'] / whole_sgai['sgai_']
        sgai_df = pd.DataFrame({'date': whole_sgai[x], x: whole_sgai['SGAI']})
        SGAI.append(sgai_df)

SGAI = pd.concat(SGAI, axis=1, join='outer')
SGAI.to_excel(os.path.join(os.path.dirname(__file__),dirnm + "/beneish/raw/SGAI.xlsx"))

# Step 8 = sort all indicators according to quarters of interest

dir_ = '/Users/karelsarapatka1/Desktop/beneish/'

dff = pd.read_excel(os.path.join(os.path.dirname(__file__), dirnm +"quarts.xlsx"))
dates = dff['Date']

for w in os.listdir(dir_+ "raw/")[1:]:
    try:
        quarterly = pd.DataFrame({'date': dates})
        df = pd.read_excel(os.path.join(os.path.dirname(__file__),dir_+'raw/' + w))

        if 'Unnamed: 0' in df.columns.values.tolist():
            df = df.drop(['Unnamed: 0'], axis=1)
        x_col = dateselect(df)[0]
        d_col = dateselect(df)[1]

        for x, z in zip(d_col, x_col):
            vals = []
            datess = []
            for i in dates:
                try:
                    listt = list(df[x])
                    len_l = len(df[x].dropna().unique().tolist())

                    dats = df[x][(df[x] + BDay(40)) <= i].head(1).tolist()

                    if i.year == dats[0].year or i.year == dats[0].year + 1:
                        row = listt.index(dats[0])
                        val = df.loc[row, z]
                        datess.append(i)
```

```
                    vals.append(val)
                except:
                    print('Nowhere to be found', x, z)

            values = pd.DataFrame({'date': datess, z: vals})
            print(values)
            quarterly = pd.merge(quarterly, values, on='date', how='outer')

        print(quarterly)
        print('__' * 20)
        quarterly.to_excel(os.path.join(os.path.dirname(__file__),dirnm +"beneish_ordered/_"+w))

    except:
        print('Unsupported format:',w)

dir_1 = '/Users/karelsarapatka1/Desktop/beneish_ordered/'
sgai = pd.read_excel(
    os.path.join(os.path.dirname(__file__), dir_1 + '_SGAI.xlsx')).drop(
    ['Unnamed: 0'], axis=1)
sgi = pd.read_excel(
    os.path.join(os.path.dirname(__file__), dir_1 + '_SGI.xlsx')).drop(
    ['Unnamed: 0'], axis=1)
aqi = pd.read_excel(
    os.path.join(os.path.dirname(__file__), dir_1 + '_AQI.xlsx')).drop(
    ['Unnamed: 0'], axis=1)
dsri = pd.read_excel(
    os.path.join(os.path.dirname(__file__), dir_1 + '_dsri.xlsx')).drop(
    ['Unnamed: 0'], axis=1)
gmi = pd.read_excel(
    os.path.join(os.path.dirname(__file__), dir_1 + '_gmi.xlsx')).drop(
    ['Unnamed: 0'], axis=1)
levi = pd.read_excel(
    os.path.join(os.path.dirname(__file__), dir_1 + '_LEVI.xlsx')).drop(
    ['Unnamed: 0'], axis=1)
depi = pd.read_excel(
    os.path.join(os.path.dirname(__file__), dir_1 + '_depi.xlsx')).drop(
    ['Unnamed: 0'], axis=1)
tata = pd.read_excel(
    os.path.join(os.path.dirname(__file__), dir_1 + 'tata.xlsx')).drop(
    ['Unnamed: 0'], axis=1)
```

```
# beneish formula = -4.84 + .920*DSR + .528*GMI + .404*AQI + .892*SGI + .115*DEPI-.172*SGAI
+4.679*ACCRUALS - .327*LEVI

date_ = tata['date']

columns = []
for x in os.listdir(dir_1):
    try:
        df = pd.read_excel(os.path.join(os.path.dirname(__file__), dir_1 + x))
        df = df.fillna(0)
        columns.append(df.columns)
        # print(x, len(df.columns))
    except:
        print('Corrupted filename:', x)

interse = list(set(columns[0]).intersection(*columns))
match_targ = [x for x in targs if x not in interse]
interse.sort()

beneish_m = pd.DataFrame({'date': date_})
missval = []
beneish = []

for y in interse:
    start = time.time()
    if y != 'date':
        try:
            val = []
            for i in range(len(beneish_m)):

                bene_v = -4.84 + 0.920 * dsri.loc[i, y] + 0.528 * gmi.loc[
                  i, y] + 0.404 * aqi.loc[i, y] + 0.892 * sgi.loc[i, y] + 0.115 * depi.loc[i, y] - 0.172 * sgai.loc[i, y] + 4.679 *
tata.loc[i, y] - 0.327 * levi.loc[i, y]
                val.append(bene_v)
            bene = pd.DataFrame({y: val})
            beneish.append(bene)
        except:
            missval.append(y)
```

```
beneish = pd.concat(beneish, axis=1, join='outer')
beneish.insert(0, 'date', date_)
print(beneish)
print('Missed values:', len(missval), missval)
beneish.to_excel(os.path.join(os.path.dirname(__file__), dirnm +"m_score.xlsx"))
```

Processing of variables and creation of Summary statistics

Variables calculated in previous steps are now ordered by quarter. Non-targeted and targeted firm-quarter values are combines separately at first. Mean, standard deviation, median, variance, first and last quintile are computed for targets and non-targets. Subsequently, t-statistics, critical values and p-values for means of both groups are calculated. The variables analyzed in the below section are in their original form, i.e. prior to transformation of continuous variables to binary variables. The code applied to binary variables differs only in several lines and thus we refrain from presenting it in its full form anew.

```
import pandas as pd
import numpy as np
import os
from scipy import stats

# clear the dataset off microcaps (below 5th marketcap percentile)
mcap = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/determinants/financials/marketcap.xlsx")).drop(['Unnamed: 0'], axis = 1)
mean = mcap.mean(axis = 0)
menval = pd.DataFrame({'mean':mean})
quants = menval.quantile(0.2)
deviants = menval.index[menval['mean']<= quants[0]].tolist()

# descriptive statistics = before turning continuous vars to binary

tgts = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/tgt_dates.xlsx")).drop(['Unnamed: 0'], axis=1)
comps = list(tgts['Company'])
collect = pd.DataFrame({'company': comps})
dirname = "/Users/karelsarapatka1/Desktop/determinants/"

#quarter-wise ordering of all variables
for l in range(0, 32):
    combine_df = []
    df = pd.DataFrame()
    for i in os.listdir(dirname)[::-1]:
        try:
            if '.xlsx' in str(i):
                det = i.split('.')[0]
```

```
            deter = []
            dets =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/determinants/" +
i)).drop(['Unnamed: 0'], axis=1)

            targ_excl = [x for x in dets if
                    x not in comps and x != 'date' and '.1' not in x and x not in deviants]
            dets_ex = dets[targ_excl]
            row = dets_ex.iloc[l, :]

            #5th and 95th percentile exclusion
            rows = pd.DataFrame({det: row})
            quant2 = rows.quantile(0.05)
            quant3 = rows.quantile(0.95)

            vals = rows.index[rows[det] <= quant3[0]].tolist()
            vals2 = rows.index[rows[det] >= quant2[0]].tolist()

            intersect = [x for x in vals if x in vals2]
            rows_ex = rows.loc[intersect]

            combine_df.append(rows_ex)

        except:
            print('invalid pathname:', i)

    combine_df = pd.concat(combine_df, axis=1, join='outer')

    combine_df.to_excel(os.path.join(os.path.dirname(__file__),dirname + "regression_sets/" + str(l + 1) +
".xlsx"))

    mean = combine_df.mean(axis=0)
    stand = combine_df.std(axis=0)
    statistics = pd.DataFrame({'mean': mean, 'std': stand})

    statistics.to_excel(os.path.join(os.path.dirname(__file__),dirname + "regression_sets/stats/st" + str(l + 1)
+ ".xlsx"))

#_____
```

```
# summary of all values across all fiscal quarters
#_____________________________________________________________________________

descriptives = pd.DataFrame(index=['count', 'mean', 'std', 'var', 'last_quintile', 'first_quintile'])
dirname_ = "/Users/karelsarapatka1/Desktop/determinants/regression_sets/"
col_df = pd.read_excel(os.path.join(os.path.dirname(__file__), dirname_ + '1.xlsx')).drop(['Unnamed: 0'],
axis=1)

col = col_df.columns
for i in col:
    stat = []
    for x in os.listdir(dirname_):
        try:
            df = pd.read_excel(
                os.path.join(os.path.dirname(__file__), dirname_ + x)).drop(
                ['Unnamed: 0'], axis=1)
            stat.append(df[i])
        except:
            print('invalid name', x)

    stats = pd.concat(stat, axis=0, join='outer', ignore_index=True)
    stat_mean = stats.mean()
    stat_count = stats.count()
    stat_std = stats.std()
    stat_var = stats.var()
    stat_upper = stats.quantile(0.8)
    stat_lower = stats.quantile(0.2)

    descriptives[i] = [stat_count, stat_mean, stat_std, stat_var, stat_upper,
                       stat_lower]
    descriptives.to_excel(os.path.join(os.path.dirname(__file__),
                                       "/Users/karelsarapatka1/Desktop/summary_statistics.xlsx"))
    print(descriptives)
# ____________________________________________________________________________


descript_targs = pd.DataFrame(
    index=['count', 'mean', 'std', 'var', 'last_quintile', 'first_quintile'])
targets = pd.read_excel(os.path.join(os.path.dirname(__file__),
                                     "/Users/karelsarapatka1/Desktop/descriptive_targets.xlsx")).drop(
```

```
    ['Unnamed: 0'], axis=1)
for x in targets.columns:
    if x != 'company':
        col = targets[x]
        t_mean = col.mean()
        t_count = col.count()
        t_std = col.std()
        t_var = col.var()
        t_upper = col.quantile(0.8)
        t_lower = col.quantile(0.2)

        descript_targs[x] = [t_count, t_mean, t_std, t_var, t_upper, t_lower]
        print(descript_targs)
        descript_targs.to_excel(os.path.join(os.path.dirname(__file__),
                                    "/Users/karelsarapatka1/Desktop/summary_stat_targets.xlsx"))
# ____________________________________________________________________

dirname_ =  "/Users/karelsarapatka1/Desktop/determinants/regression_sets/"

t_tests = pd.DataFrame(index = ['df','t-stat', 'p-val'])
col_df = pd.read_excel(os.path.join(os.path.dirname(__file__), dirname_ + '1.xlsx')).drop(['Unnamed: 0'],
axis = 1)
targets =
pd.read_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/descriptive_targets.
xlsx")).drop(['Unnamed: 0'], axis = 1)
t_stats = pd.DataFrame(index = ['cv1','cv2', 'cv3' ,'t-stat', 'p-val'])

col = col_df.columns
for i in col:
    stat = []

    tar_col = targets[i]
    for x in os.listdir(dirname_):
        try:
            df = pd.read_excel(os.path.join(os.path.dirname(__file__), dirname_ + x)).drop(['Unnamed: 0'], axis
= 1)
            stat.append(df[i])
        except:
            print('invalid name', x)
```

```
sta = pd.concat(stat, axis=0, join='outer', ignore_index=True)
stat_mean = sta.mean()
stat_count = sta.count()
stat_std = sta.std()
stat_var = sta.var(ddof = 1)

tar_mean = tar_col.mean()
tar_count = tar_col.count()
tar_std = tar_col.std()
tar_var = tar_col.var(ddof = 1)

#__________std deviation_________________
s = np.sqrt((stat_var + tar_var)/2)

#t-statistic___(2 alternative ways to get t = ts & t2)
ts = (tar_mean - stat_mean) / (np.sqrt((stat_var / stat_count) + (tar_var / tar_count)))

se1 = tar_std / np.sqrt(tar_count)
se2 = stat_std/np.sqrt(stat_count)
sed = np.sqrt(se1**2 + se2**2)
t2 = (tar_mean - stat_mean)/sed
alpha1 = 0.1; alpha2 = 0.05; alpha3 = 0.01

df = tar_count + stat_count - 2
#__________p-val_____________________________
p = 1 - stats.t.cdf(np.abs(t2), df = df)

#_____________critical values (cv) for different signif.levels__________
cv1 = stats.t.ppf(1-alpha1, df)
cv2 = stats.t.ppf(1-alpha2, df)
cv3 = stats.t.ppf(1-alpha3, df)

t_stats[i] = [cv1, cv2, cv3]
t_tests[i] = [df, ts, p]
t_tests.to_excel(os.path.join(os.path.dirname(__file__), "/Users/karelsarapatka1/Desktop/t_testss.xlsx"))
print(t_tests)
```

Pearson Chi-square test

Binarized quarterly values are grouped according to whether the firm was targeted or not. Variables are then transformed into a contingency table on which the Pearson Chi square is applied. Lastly, critical values, Chi-sq. statistic and p-values are calculated for statistical significance of mean differences.

```
import os
import pandas as pd
import numpy as np
from scipy.stats import chi2_contingency
from scipy.stats import chi2

#df.astype(bool).sum(axis=0)
# regset binary-> column w/ group {0,1}

dirname = "/users/karelsarapatka1/Desktop/determinants/regset_binary/"
df = pd.read_excel(os.path.join(os.path.dirname(__file__),dirname + "1.xlsx" )).drop(['Unnamed: 0'], axis =
1)

controls = ["debtasset_sort", "lnsize", "STD_sorted"]

dfxs = []
for x in range(1,33):
    dfx = pd.read_excel(os.path.join(os.path.dirname(__file__),dirname +str(x) + ".xlsx" )).drop(['Unnamed:
0'], axis = 1)
    dfxs.append(dfx)

dfxs = pd.concat(dfxs, axis = 0, join = 'outer', ignore_index= True)

df2 = dfxs.groupby(['target']).sum()
df2.drop(controls, axis = 1, inplace = True)
del df2.index.name

zero = pd.DataFrame(dfxs.loc[dfxs['target'] == 0].count()).transpose()
zero_t = pd.DataFrame(dfxs.loc[dfxs['target'] == 1].count()).transpose()

df3 = pd.DataFrame(data = [zero.iloc[0], zero_t.iloc[0]], index = [0, 1])

statistics = pd.DataFrame(index = ['dof', 'chi', 'p-val', 'critical'])
for i in range(len(df2.columns)):
```

```
    col = df2.columns[i]
    totals = df3.iloc[:,i]
    ones = df2.iloc[:,i]
    nulls = totals - ones

    df1 = pd.DataFrame({'nulls':nulls, 'one':ones})
    array = df1.values

    stat, p, dof, expected = chi2_contingency(array)
    print('stat:', stat)
    prob = 0.95
    critical = chi2.ppf(prob, dof)

    statistics[col] = [dof,stat,p, critical]

    if abs(stat)>= critical:
        print('Dependent (reject H0)')
    else:
        print('Independent (fail to reject H0)')
    print(statistics)
    print('__'*20)

statistics.to_excel(os.path.join(os.path.dirname(__file__),"/Users/karelsarapatka1/Desktop/chisq_test.xlsx")
)
```

Pearson Correlations among variables

Datasets are cleared off missing values, such that only available data is paired with non-missing values. Pearson correlations themselves are calculated with the use of scipy module and saved locally together with corresponding p-values.

```
import pandas as pd
import os
import numpy as np
from scipy import stats
from scipy.stats import pearsonr
help(pearsonr)

df1 = pd.read_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/descriptive_targets.xlsx")).drop(['Unnamed: 0'], axis = 1)
```

```
df1.set_index(['company'], drop = True, inplace = True)
pearson = pd.DataFrame(columns = [df1.columns], index = [df1.columns])
#print(pearson)

dirname_ =  "/Users/karelsarapatka1/Desktop/determinants/regression_sets/"

sta = []
sta.append(df1)
for x in os.listdir(dirname_):
    try:
        df2 = pd.read_excel(os.path.join(os.path.dirname(__file__), dirname_ + x)).drop(['Unnamed: 0'], axis =
1)
        sta.append(df2)
    except:
        print('corrupted format', x)

df = pd.concat(sta, axis=0, join='outer', ignore_index=True)

correl = []
pval = []
d= 0
for i in range(len(df.columns)):
        col = []
        colP = []
        for x in range(len(df.columns)):

                array1 = df.iloc[:,i].values
                array2 = df.iloc[:,x].values

                na1 = np.logical_or(np.isnan(array1), np.isnan(array2))
                pears = stats.pearsonr(array1[~na1], array2[~na1])
                col.append(pears[1])
                colP.append((pears[0]))

        var = df.columns[i]
        pearson = pd.DataFrame({var:col})
        pvals = pd.DataFrame({var:colP})
        correl.append(pearson)
        pval.append(pvals)
```

```
correl = pd.concat(correl, axis = 1, join = 'outer')
pvalues = pd.concat(pval, axis = 1, join = 'outer')



correl.to_excel(os.path.join(os.path.dirname(__file__), "/Users/karelsarapatka1/Desktop/correl_pval.xlsx"))
pvalues.to_excel(os.path.join(os.path.dirname(__file__),
"/Users/karelsarapatka1/Desktop/correl_pearson.xlsx"))
```

Data cluster analysis

```
import pandas as pd
import os
import sklearn
from sklearn.model_selection import train_test_split
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
from yellowbrick.cluster import KElbowVisualizer
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score
import matplotlib.pyplot as plt
import matplotlib.cm as cm

dirname = '/Users/karelsarapatka1/Desktop/XXX/'
dataset = []

for i in sorted(os.listdir(dirname)):
    try:
        dfx = pd.read_excel(os.path.join(os.path.dirname(__file__), dirname + i)).drop(['Unnamed: 0'], axis=1)
        dataset.append(dfx)

    except:
        print('Invalid path:', i)
```

```
df = pd.concat(dataset, axis=0, join='outer').dropna()
df= df.drop(['comp'], axis = 1)
df= df.drop(['ind'], axis = 1)
df = sklearn.utils.shuffle(df)

train = df.iloc[:round(0.7*len(df))]
test = df.iloc[round(0.7*len(df)):]

"""X = train.drop(['target'], 1)

km = KMeans(n_clusters=10, max_iter=600, algorithm= 'auto').fit(df)
cluster_map = pd.DataFrame()
cluster_map['data_index'] = df.index.values
cluster_map['cluster'] = km.labels_
print(cluster_map[cluster_map.cluster == 1])

X = np.array(train.drop(['target'], 1).astype(float))
y = np.array(train['target'])

scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)

kmeans = KMeans(n_clusters = 2, max_iter=600, algorithm= 'auto')
kmeans.fit(X_scaled)

correct = 0

for i in range(len(X)):
    predict_me = np.array(X[i].astype(float))
    predict_me = predict_me.reshape(-1, len(predict_me))
    prediction = kmeans.predict(predict_me)
    if prediction[0] == y[i]:
        correct += 1

print(correct / len(X))"""

range_n_clusters = [2, 3, 4, 5,6,7, 8, 9, 10, 11, 15, 20, 30, 40, 50, 60]

test_size = 0.3
```

```
seed = 10
X = df.drop(['target'], axis = 1).values
x = df.drop(['target'],axis = 1)
y = df['target'].values

X,y = make_blobs()

for n_clusters in range_n_clusters:
    fig, (ax1, ax2) = plt.subplots(1, 2)
    fig.set_size_inches(18, 7)
    ax1.set_xlim([-0.1, 1])
    ax1.set_ylim([0, len(X) + (n_clusters + 1) * 10])

    clusterer = KMeans(n_clusters=n_clusters, random_state=100)
    cluster_labels = clusterer.fit_predict(X)

    km = KMeans(n_clusters=n_clusters, max_iter=600, algorithm='auto').fit(x)
    cluster_map = pd.DataFrame()
    cluster_map['data_index'] = x.index.values
    cluster_map['cluster'] = km.labels_
    print(cluster_map)

    silhouette_avg = silhouette_score(X, cluster_labels)
    print("For n_clusters =", n_clusters,
          "The average silhouette_score is :", silhouette_avg)

    sample_silhouette_values = silhouette_samples(X, cluster_labels)

    y_lower = 10
    for i in range(n_clusters):
        ith_cluster_silhouette_values = \
            sample_silhouette_values[cluster_labels == i]
        ith_cluster_silhouette_values.sort()

        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i

        color = cm.nipy_spectral(float(i) / n_clusters)
        ax1.fill_betweenx(np.arange(y_lower, y_upper),
                          0, ith_cluster_silhouette_values,
```

```
                    facecolor=color, edgecolor=color, alpha=0.7)

        ax1.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))
        y_lower = y_upper + 10

    ax1.set_title("The silhouette plot for data clusters.")
    ax1.set_xlabel("The silhouette coefficient values")
    ax1.set_ylabel("Cluster label")

    ax1.axvline(x=silhouette_avg, color="red", linestyle="--")

    ax1.set_yticks([])  # Clear the yaxis labels / ticks
    ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])

    colors = cm.nipy_spectral(cluster_labels.astype(float) / n_clusters)
    ax2.scatter(X[:, 0], X[:, 1], marker='.', s=30, lw=0, alpha=0.7,
                c=colors, edgecolor='k')

    centers = clusterer.cluster_centers_

    ax2.scatter(centers[:, 0], centers[:, 1], marker='o',
                c="white", alpha=1, s=200, edgecolor='k')

    for i, c in enumerate(centers):
        ax2.scatter(c[0], c[1], marker='$%d$' % i, alpha=1,
                    s=50, edgecolor='k')

    ax2.set_title("The visualization of the clustered data.")
    ax2.set_xlabel("Feature space for the 1st feature")
    ax2.set_ylabel("Feature space for the 2nd feature")

    plt.suptitle(("Silhouette analysis for KMeans clustering on sample data "
                  "with n_clusters = %d" % n_clusters),
                 fontsize=14, fontweight='bold')

plt.show()

#for n_clusters in range_n_clusters:
model = KMeans(random_state=0)
```

```
visualizer = KElbowVisualizer(model, k=(2,20), metric='silhouette', timings=False)
visualizer.fit(X)
visualizer.show()
```

Logit Model

In the below code, we combine assigned data clusters established in the previous section to firms in the sample. To satisfy the requirement of completeness of data for logit regression, several firm-quarter datasets had to be dropped. Two libraries were tested for the logit regression itself, Statsmodels and Scikit-Learn. Upon introduction of the intersect in Scikit model, the variable coefficient values corresponded. Ultimately, we opt for Statsmodels logit, since it provides p-values and t-statistics, which is a feature not yet introduced in Scikit. 2 dummies corresponding to K-means clusters are introduced, standard errors are clustered at firm level.

```
from __future__ import division
import pandas as pd
import os
import numpy as np
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from statsmodels.discrete.discrete_model import Logit
from sklearn.linear_model import LogisticRegression



dirname = '/Users/karelsarapatka1/Desktop/XXX/'
dataset = []

cluster_set = pd.read_excel(os.path.join(os.path.dirname(__file__),
'/Users/karelsarapatka1/Desktop/regressions/clusters/clustergroup_3.xlsx')).drop(['Unnamed: 0'], axis = 1)
cluster_set = cluster_set.sort_values(by = ['comp'])

for i in sorted(os.listdir(dirname)):
    try:

        dfx = pd.read_excel(os.path.join(os.path.dirname(__file__), dirname + i)).drop(['Unnamed: 0'], axis=1)
        all_cols = list(dfx.columns)
        industries = list((dfx.loc[dfx['target'] == 1, 'ind']))
        dfx1 = dfx.loc[dfx['ind'].isin(industries), all_cols]

        dataset.append(dfx)
    except:
        print('Invalid path:', i)

dfs = pd.concat(dataset, axis=0, join='outer').dropna()
```

```
dfs = dfs.sort_values(by = ['comp'])
dfs.reset_index(inplace = True, drop = True)
cluster_set.reset_index(inplace = True, drop = True)

dfs['cluster'] = np.where((dfs['comp'] == cluster_set['comp']), cluster_set['cluster'], np.nan)
pd.get_dummies(dfs['cluster'])
print(dfs)
dummy_clusters = pd.get_dummies(dfs['cluster'], prefix = 'cluster_')

dummy_clusters= dummy_clusters.iloc[:,1:]
print(dummy_clusters)

variables = ["pe_ordered", "Loughran_sorted", "Z_score", "ADV_sorted", "tacc__",
            "big4", "momentum_ordered", "debtC", "spread_ordered", "overInv",
            "duality_ordered", "m_score", "AnDisp_ordered", "PB_sorted",
            "cROA", "LnAn_ordered"]

#data = dfs[variables].join(dummy_clusters)

master = []
master2 = []
master3 = []
#cluster = dfs['cluster'].values
overval = ["pe_ordered", "tacc__", "overInv", "Z_score", "momentum_ordered", "PB_sorted",
"cROA","debtC"]
ambiguity = ["ADV_sorted", "spread_ordered","big4", "Loughran_sorted", "m_score", "duality_ordered",
"AnDisp_ordered", "LnAn_ordered"]

combo = [overval, ambiguity]
z = 'combined'

for z in variables:
    controls = ['lnsize', 'STD_sorted', 'debtasset_sort', z]
    #valuess = ['const', 'lnsize', 'STD_sorted', 'debtasset_sort', 'determinant']
    valuess = ['const', 'lnsize', 'STD_sorted', 'debtasset_sort', 'determinant', 'cl_2', 'cl_3']
    data = dfs[controls].join(dummy_clusters)

#introduction of constant to the model

    X = sm.add_constant(data).values
```

```
    y = dfs['target'].values

    logit_model = sm.Logit(y,X)

#Clustering of standard errors at company level

    result = logit_model.fit(cov_type= 'cluster', cov_kwds={'groups':dfs['comp']})

    me = result.get_margeff(at = 'mean').summary()
    print(result.summary())
    print(z)
    print(me)

    parameters = pd.DataFrame(data = result.params, columns = [z], index = valuess)
    pvalues = pd.DataFrame(data=result.pvalues, columns=[z], index=valuess)
    tvalues = pd.DataFrame(data=result.tvalues, columns=[z], index=valuess)

    master.append(parameters)
    master2.append(pvalues)
    master3.append(tvalues)
    print('Exponential logits:',np.exp(result.params))

master = pd.concat(master, axis= 1, join = 'outer')
master2 = pd.concat(master2, axis= 1, join = 'outer')
master3 = pd.concat(master3, axis= 1, join = 'outer')
master2 = master2.round(decimals = 4)
master3 = master3.round(decimals = 4)

#master.to_excel(os.path.join(os.path.dirname(__file__),
'/Users/karelsarapatka1/Desktop/clustered_comb.xlsx'))
#master2.to_excel(os.path.join(os.path.dirname(__file__),
'/Users/karelsarapatka1/Desktop/clustered_comb_p.xlsx'))
#master3.to_excel(os.path.join(os.path.dirname(__file__),
'/Users/karelsarapatka1/Desktop/clustered_comb_t.xlsx'))
```

OLS Regression of Abnormal Returns

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
import pandas as pd
import numpy as np
import os
import statsmodels.api as sm

dirname = '/Users/karelsarapatka1/Desktop/'

targs = pd.read_excel(os.path.join(os.path.dirname(__file__), dirname +
'model_data.xlsx')).drop(['Unnamed: 0'], axis = 1)

returns = pd.read_excel(os.path.join(os.path.dirname(__file__), dirname +
'returns_determinants.xlsx')).drop(['Unnamed: 0'], axis = 1)

targets = targs.drop(['LnAn_ordered', 'AnDisp_ordered'], axis = 1)

sets = ["cROA",    "PB_sorted","pe_ordered", "overInv", "Z_score", "ADV_sorted", "m_score",
"spread_ordered"]
comp_list = []
for i in sets:
    quant1 = targets[i].quantile(0.01)
    quant2 = targets[i].quantile(0.99)
    targets[i] = targets[i].apply(lambda x: np.NaN if x >= quant2 else (np.NaN if x <= quant1 else x))

targets = targets.drop(["pe_ordered" ], axis = 1)
targets = targets.dropna()

company = list(targets['company'])
company3 = list(returns['Company'])

intersect3 = [x for x in company if x in company3]
regressors = targets.loc[targets['company'].isin(intersect3), targets.columns].sort_values(by = ['company'])
regressand = returns.loc[returns['Company'].isin(intersect3), returns.columns].sort_values(by =
['Company'])

regressand.reset_index(inplace = True, drop = True)
regressors.reset_index(inplace = True, drop = True)
regressand = regressand.drop(['Company'], axis = 1)
```

```
X = regressors[regressors.columns[1:]]
row = X.columns[:].tolist();rows = ['const']+ row[:]

def calc_vif_(X, thresh=5.0):
    variables = list(range(X.shape[1]))
    dropped = True
    while dropped:
        dropped = False
        vif = [variance_inflation_factor(X.iloc[:, variables].values, ix)
               for ix in range(X.iloc[:, variables].shape[1])]

        maxloc = vif.index(max(vif))
        if max(vif) > thresh:
            print('dropping \'' + X.iloc[:, variables].columns[maxloc] +
                  '\' at index: ' + str(maxloc))
            del variables[maxloc]
            dropped = True

    print('Remaining vars:')
    print(X.columns[variables])
    return X.iloc[:, variables]

X = calc_vif_(X)

master = []
master1 = []
master2 = []
for y in regressand.columns:
    Y =regressand[y]
    X = sm.add_constant(X)
    model = sm.OLS(Y,X)
    results = model.fit(cov_type= 'cluster', cov_kwds={'groups':regressors ['company']})

    print(results.summary())
```