The effect of the introduction of ChatGPT on the stock market return of competing companies

14 August 2023

Research: Master Thesis Business Economics
Student: B.T. (Bas) Volkert
Registration number: 1025338
Supervisor: Dr. T.P.F. (Tobias) Dalhaus
Course Code: BEC-80436
Period: Jan 2023 - Aug 2023

Disclaimer

WAGENINGEN
UNIVERSITY & RESEARCH

Acknowledgements

I would like to express my appreciation to my supervisor, Tobias Dalhaus, for his constructive feedback and sparring partnership during my thesis. His knowledge and experience helped me grasp the information needed for this research. Thanks for all the help, and good luck in obtaining your permits for the farm! I'm also grateful to the BMO group for giving me the opportunity to write my Master Thesis.

I am also grateful for my parents, who supported me in the last phase of my thesis with some writing assistance, but most importantly, they supported my new cycling hobby. It somewhat extricated me from my day-to-day thesis life, which was greatly beneficial.

Lastly, I would like to thank my fellow students, especially Stefan, for helping me prepare for my thesis presentation, and particularly Ian for aiding me in understanding the 5-factor.Lastly, when conducting this research about the stock prices of competing companies to ChatGPT, as dyslexic I made sure to explore competitiveness of ChatGPT's help with spelling.

Abstract

This research examines the repercussions of the introduction of ChatGPT on the stock market returns of prominent competing firms, namely Alphabet, Baidu, Verizon and Microsoft. Employing an event study methodology, the study focuses on the introduction of ChatGPT as the pivotal event, occurring on the 30th of November 2022. Six distinct event windows are analysed to assess the variations in stock market returns centered around the event. To provide a comprehensive assessment, the study incorporates the Fama & French 5-factor model supplemented by three additional models. The results underscore significant high abnormal returns, with a particularly high abnormal returns within the extended event windows. Remarkably, all four models utilized in the analysis demonstrate a high level of statistical significance, reinforcing the robustness of the conclusions. Strikingly, the research reveals a substantial disparity between anticipated returns and the realized returns observed during the event study. Importantly, this research highlights the compelling influence of ChatGPT's introduction on the stock market performance of competing companies, showing significant stock market responses that surpass conventional expectations set out by the Efficient Market Hypothesis. Namely, despite the efficient market hypothesis, investors were still able to achieve outperformance compared to the benchmark. This implies that in the future, investors may recognize these patterns and actively seek out similar opportunities to potentially benefit from them.

Table of Content

Disclaimer 2
Acknowledgements 3
Abstract 4
Inhoud 5
1 Introduction 6
1.1 Hypotheses 7
1.2 Structure of research paper 8
2 Background 9
2.1 ChatGPT Theoretical Framework 9
2.2 Theoretical Framework 9
2.3 Search Engine Companies 11
2.4 Microsoft's Involvement in OpenAI 11
3 Methodology & Data 13
3.1 The Event 13
3.2 Event Window 13
3.3 Benchmark 15
3.4 Data 15
3.5 Methodology 15
4 Findings 18
4.1 Cumulative Abnormal Returns 18
4.2 P-Values 19
4.3 Actual Returns 20
4.4 Factor Data 22
5 Discussion 25
5. 1 Theory 25
5.2 Benchmark 26
5.3 Factors 27
5.4 Microsoft 27
5.5 Baidu 28
5.6 Estimation Window 29
5.7 Event Window 30
5.8 Implications for Investors 31
6 Conclusion 33
7 References 35
8 Appendix – R-Code 39

WAGENINGEN
UNIVERSITY & RESEARCH

1 Introduction

In times of rapid change innovations can be found everywhere. Companies are trying to improve their products and services or invent entirely new products. These new innovations can influence different aspects of organizations, such as the performance of organizations (Gunday, Ulusoy, Kilic, and Alpkan, 2011), (Lee, Lee, and Garrett, 2019) and the stock price (Srinivasan, Pauwels, Silva-Risso & Hanssens, 2009). It is important to get more insight on the extent to how certain innovations do affect competing companies, especially AI related innovations. By researching this, one can gain knowledge on how to deal with these innovations regarding the stock market returns. If these innovations do in fact have an influence on the stock price return of competing organizations, it is very important for shareholders of competing companies to be able to know what effects certain innovations could have on companies of which they are shareholders.

This research focuses on the effect of an innovation on competing companies. Specifically, to what extend the release of OpenAI’s ChatGPT has influenced the stock market return of competing companies, Alphabet, Baidu, Verizon, and Microsoft, within a set time period called the event window (Armitage, 1995). This research will fill the current knowledge gap that exists surrounding the stock market return of competing organizations on implementing new innovations.

This research is set up in order to fill the current knowledge gap that exists regarding the effect of an innovation on a competitive company, in particular the stock market return of competing organizations. In the past, research has been conducted on the effects of events, so called event studies, on daily stock market returns (Brown & Warner, 1985). This existing research shows that one can measure abnormal returns within stock market returns triggered by events. Moreover, research on events such as acquisitions was done by Elad & Zivney (2017). The research showed that the event measured, acquisition news, affected stock returns. Elad & Zivney focused on the stock market returns of the organizations executing the acquisition. Research has also been conducted on how the release of new products affects the maker's stock price (Lee & Chen, 2009). Mahajan, Sharma, and Buzzell (1993) suggested that the effect of announcing a new product on the competitors mainly depend on two factors. Namely, the way the market has an expansion effect and possible substitutions. In the research of Chen, Ho, and Ik (2005), they found that the announcement of new products can have a negative so called ‘wealth effect’ on competing organizations. They even state that the effect can be more unfavourable for rivalling technology organizations. In addition to Chen et al (2005), Ashok (1994) showed that rivalling companies often suffer from negative wealth effect after a rivalling company announced a new product. On the other hand, Lee (1995) found a positive wealth effect after a rival announced a new product within communication and computer industries. However, what is still missing is research on the influence of new innovations on the stock prices of competing tech/ai companies.

To achieve the goal of this research, an event was chosen. This event is the introduction of OpenAI's software called ChatGPT. The event took place on 30 November 2022. This study will look at the impact on stock market returns of competing companies during the event window. The event study will examine the overall change in stock prices of the with OpenAI competing firms beyond what is expected. These unexpected returns are called abnormal return (Strong, 1992). The event study can be used in order to find the price reaction of a stock during an event (Brown & Warner, 1985), (Campbell et al., 1997). The Event study will be used in order to find excess returns. To be able to do so first an estimation window needs to be defined (MacKinlay, 1997). During the estimation window all excess returns will be identified. These excess returns are needed to make an estimating of the abnormal returns during and after the event.

In order to identify excess returns four different models will be used. Each model builds upon the previous model. The simplest one is the CAMP model (Lintner, 1965; Sharpe, 1964). Fama & French improved this model leading to the Fama-French 3-factor model (Fama & French, 1993), Carhart (Carhart, 1997), and Fama & French 5-factor model (Fama & French, 2015). The 5-factor model is designed to be able to encapsulate the relation of the returns of a security against a benchmark. Within the 5-factor model corrections for size, value, value, and investment factors are included.

1.1 Hypotheses

The Null hypothesis states that there will not be a significant difference in stock return of the competing companies in comparison to the benchmark. To test the significance of the event the following hypotheses will be tested (Corrado & Zivney, 1992):

- H0: The introduction of ChatGPT does not have any relationship with the abnormal returns of competing firms.
- H1: The introduction of ChatGPT does appear to be related significantly to the abnormal returns of competing firms.

After conducting the hypothesis testing, the research will entail the analysis and discussion of the potential occurrence of abnormal returns on competing organizations as a result of the event. This study will utilize stock market data obtained from Yahoo Finance (Yahoo Finance, 2023) and factor data from French factor data base (French, 2023).

1.2 Structure of research paper

This research paper will comprise five sections, each contributing to the examination of the hypotheses. These sections include Context, Theoretical Framework, Methodology & Data, Findings, and Conclusion & Discussion.

The Context section will begin by providing an overview of the current landscape of innovations impacting security prices. Along with background information and a literature review, the paper will present insights into previous studies conducted on the subject.

The Theoretical Framework section will delve into the theories that currently influence stock price returns during new innovations introduced by competing organizations. It will provide a comprehensive explanation of these theories.

The Methodology & Data section will discuss the models employed to conduct the research. Additionally, the section will outline the process of extracting daily stock price data from exchanges.

Within the Findings segment, the hypotheses will be tested by analysing data before and after the event to determine the existence of abnormal returns within the event window.

The discussion section will evaluate the identified theories based on their alignment with the research findings and their contrasting perspectives. Alongside with potential learnings of investors. Finally, the section will provide a conclusion that encompasses the research setup, findings, and limitations. The objective is to address the current knowledge gap regarding the potential for innovations from competing organizations to generate abnormal returns.

2 Background

2.1 ChatGPT Theoretical Framework

ChatGPT is a large language model that's owned and trained by OpenAI (Introducing ChatGPT, 2022). When given a prompt of text ChatGPT will generate a response like a human (Ramponi, 2023). Because ChatGPT is trained with a large amount of data it can for fill a wide range of tasks like, having a conversation, looking up facts, translations, and language checks. For example, when ChatGPT is given the prompt: "can you explain how you work?" ChatGPT replies with:" *Certainly! I am a language model that uses a deep neural network to process text and generate responses to user inputs. I was trained on a large amount of textual data to analyse and understand the patterns and structures of language. When a user interacts with me, I use natural language processing techniques to understand their input and generate a relevant response*." This reply perfectly demonstrates the capabilities of ChatGPT.

2.2 Theoretical Framework

The theoretical framework of this study encompasses two fundamental concepts in finance: Modern Portfolio Theory and Efficient Market Hypothesis. The integration of these frameworks enables us to examine the potential influence of ChatGPT, an advanced AI-driven language model, on the stock market prices of companies engaged in direct competition with it. Modern Portfolio Theory, originally proposed by Harry Markowitz (1952), revolutionized investment theory by introducing the notion of portfolio diversification. Modern portfolio theory emphasizes the importance of constructing portfolios with a diverse mix of assets to achieve an optimal balance between risk and return. Investors adopting modern portfolio theory seek to maximize the efficiency of their portfolios by incorporating assets with varying risk profiles, aiming to mitigate overall risk exposure. Given the disruptive nature of AI technologies, such as ChatGPT, and the uncertainties they introduce, the application of modern portfolio theory becomes particularly relevant. Companies competing in the domain of AI may face unique risks and complexities, necessitating prudent risk management and asset allocation strategies advocated by modern portfolio theory. AI could potentially lead to higher amounts of risk and uncertainty (Ashta & Herrmann, 2021), the increase in risk can force investors to re-evaluate their portfolios to either reduce some exposure to these competing companies or to invest more in them.

Efficient Market Hypothesis, developed by Eugene Fama (1970), posits that financial markets rapidly and accurately reflect all available information in the prices of traded assets. According to efficient market hypothesis, the market efficiency hypothesis, stock prices fully incorporate all publicly available information, making it difficult for investors to consistently outperform the market based solely on publicly accessible data (Timmermann & Granger, 2004). In the context of companies engaged in competition with ChatGPT, which operates within the dynamic landscape of AI technologies, understanding the implications of efficient market hypothesis is crucial. The rapid assimilation of information related to AI-driven innovations into stock prices can lead to swift and efficient market reactions, potentially limiting opportunities for investors to achieve sustained market-beating returns. Grasping the implications of efficient market hypothesis on the valuation of AI-based companies is essential for investors and market participants navigating the AI-driven landscape. For example, AI is one of the buzz words of 2023 (Seeking Alpha, 2023). Everywhere you look companies are talking about AI. The number of references to AI in earning calls is up 77% from last year (fortune, 2023; Bloomberg 2023).

By integrating the principles of modern portfolio theory and efficient market hypothesis, this study aims to illuminate how these theoretical frameworks may influence the stock market dynamics of companies directly competing with ChatGPT. This analytical lens explores how investors allocate their resources among competing companies, taking into account the risk-return trade-offs associated with AI-driven businesses. Additionally, discerning how the market incorporates information about AI technologies into stock prices may potentially shape the competitive landscape for these companies. The adoption of modern portfolio theory and acknowledgment of the principles of efficient market hypothesis provide a solid theoretical foundation for investigating the stock market behaviour of companies competing with ChatGPT. However, it is important to note that the modern portfolio theory and the efficient market hypothesis are based on certain assumptions like investors rationality & normal distributed returns, which will not always be true (Swisher & Kasten, 2005). Besides the Efficient market hypothesis and the Morden portfolio theory the stock market can be influenced by multiple other factors. Frist of all, macro-economic conditions (Diebold & Yilmaz, 2008; Damiran, Dorjdagva, Sukhee, & Myagmarsuren, 2022). For example, during the 2008 financial crisis prices of assets did not reflect their actual value. Economic conditions made investors seeking for cash and therefore sell their stock at detrimental moments. Which led to huge price drops that could not be declared by the efficient market hypothesis. Secondly, geopolitical events (Zhang, He, He, & Li, 2023). Sunardi, Noviolla, Supramono, and Hermanto (2023) showed that geopolitical discussion led to negative abnormal returns. Lastly, investors sentiment, which can lead to stock prices that do not reflect the intrinsic value of the underlying asset (Balcilar et al., 2018; Poon & Taylor, 1991).

2.3 Search Engine Companies

The primary competitive market in which OpenAI's ChatGPT operates is the search engine market. Consequently, this study focuses on competing companies within this market segment. A prominent contender in this arena is Google, currently dominating with approximately 85% of the market share (Global Search Engine Desktop Market Share 2023, 2023). Google operates under the umbrella of its parent company, Alphabet. Beyond Google, other significant players in this market include Bing (8.21%), Yahoo! (2.44%), Yandex (1.42%), DuckDuckGo (0.74%), and Baidu (0.41%). Bing is a subsidiary of Microsoft Corporation (Microsoft, 2023). The situation involving Yahoo! is somewhat intricate. The Verizon Acquisition of 2017 (Verizon, 2017) marked an agreement through which Yahoo! sold its core internet business, comprising email services, websites, and digital advertising platforms, to Verizon Communications. This transaction was successfully completed in June 2017. Post the divestiture of its core business to Verizon, Yahoo! underwent a transformation, rebranding itself as Altaba Inc. and transitioning into a holding company structure. Since Verizon is publicly traded, it warrants inclusion among the list of competing companies. Baidu, a Chinese technology enterprise, has developed a search engine using sophisticated algorithms for web information retrieval (Company Overview | Baidu Inc, 2023). Similarly, Yandex, a Russian technology firm, operates a search engine as part of its business. However, recent developments linked to Russia's Ukraine incursion have substantially impacted Yandex's stock market listing on the NASDAQ Global Select Market, introducing a host of influential factors affecting its performance. Consequently, Yandex will be omitted from the set of competing search engine companies subject to investigation. Notably, DuckDuckGo, being privately held, lacks a public stock market listing, thus rendering it beyond the scope of this research.

2.4 Microsoft’s Involvement in OpenAI

Microsoft's investment in OpenAI, which occurred in 2019 (Feiner, 2019), involved a significant amount of 1 billion USD. While the specific details pertaining to the exact size or percentage of the stake that Microsoft acquired in OpenAI have not been publicly disclosed, it established a financial relationship between the two entities. In 2023 Microsoft made an extra step by committing 10 billion USD (Capoot, 2023). This investment signifies a strategic partnership between Microsoft and OpenAI, aligning their interests in advancing artificial general intelligence and AI technologies. It should be noted that OpenAI maintains its autonomy as an independent organization, retaining control over its research agenda and decision-making processes. Microsoft's investment grants it a financial interest in the success of OpenAI, while the partnership primarily focuses on collaboration, resource sharing, and leveraging Microsoft's cloud infrastructure and services to support OpenAI's research and development efforts. Specific details regarding the ownership stake resulting from Microsoft's investment have not been made available to the public. The collaboration between Microsoft and OpenAI is centred on mutual goals rather than direct control or majority ownership.

3 Methodology & Data

3.1 The Event

OpenAI released the first version of GPT in 2018. The program was significant smaller and not open for public. On 30th November 2022 OpenAI released GPT-3 (Introducing ChatGPT, 2022). This time ChatGPT was trained with more data then every before. Besides that, an easy-to-use interface was implemented. This allowed the public to better use and understand ChatGPT. Included with the interface upgrades, OpenAI also shipped the language model BERT (Devlin et al., 2019). Meaning that it could now better understand and work with multiple languages. Day by day the number of users of ChatGPT began to grow. Leading to mayor news outlets starting to post about the new AI technology that OpenAI introduced with ChatGPT (Vallance, 2022; Vanian, 2022). As seen in Figure 1 before the 30th of November almost no one was searching for ChatGPT on Google. After the announcement on the 30th of November the number of searches including: "ChatGPT" increased dramatically. So, the event date used for this paper will be set to the 30th of November 2022.

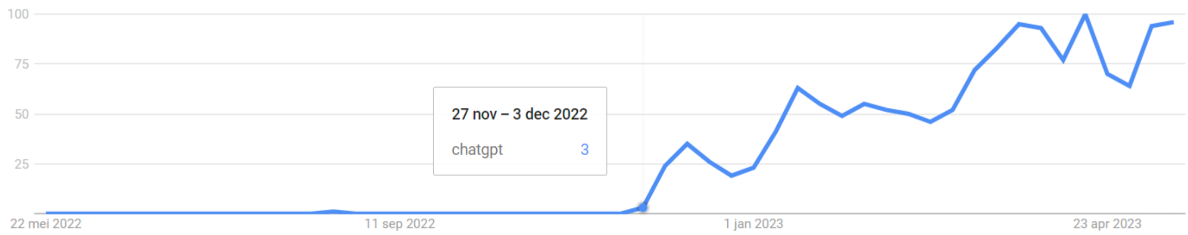

Figure 1 Google Trends: ChatGPT, A value of 100 is the peak popularity for that term. A value of 50 means the term is half as popular. A score of 0 means there is insufficient data available for that term.

3.2 Event Window

To measure possible excess returns an event window and estimation window need to be set. The estimation window will consist of 120 trading days before the event window. MacKinlay (1997) argues that an estimation window consisting of 120 trading days will cover enough trading days in order to get an accurate movement of the stock price within the estimation window. For the event window multiple approaches are used within existing literature. Bash and Alsaifi (2019) uses multiple event windows including [-1,+1], [-3,+3], [-5,+5], [-10,+10], [-10,+15] and [-10,+20]. This to cross validate if stock prices moved within different event windows. By using multiple event windows Bash and Alsaifi can use the mean of the different outcomes. Oler et al., (2008) discusses the use of even longer event windows. They studied literature that used event windows <5 days, 6-60 days and >60. They conclude that the use of event windows >60 will have negative results on the outcome of the research.

Hence for this research the event window will not be longer than +20 days. This to make use that no other events will contaminate the data set. So in for this research the ‘six event windows’ used by Bash and Alsaifi (2019) will be used. A study done by Sun et al. (2021) made use of a gap between the evaluation window and the event window. They did so to make sure the evaluation window data will not be contaminated with data that could be influenced by the event. This will be particularly interesting if a short event window will be used. Hence that in this paper an event gap will be used of ten trading days between the estimation window and event window. Figure 2 illustrates the sequence of the different event stages.

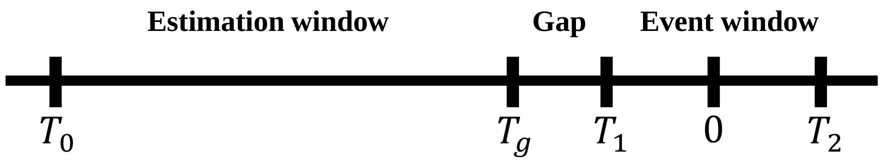

Figure 2 Time Event Study

For all the different event windows the amount of trading days between T0 and Tg will be 120. Tg, T1 will consist of ten trading days. T1 to 0 will vary from 1, 3, 5, 10, 10, 10. 0 Will always be the moment the event occurred, 30 November 2022. At last, 0 to T2 will range from 1, 3, 5, 10, 15, 20. Since it hard to point out when exactly this event has ended, if it has even ended is not sure, no post-event window will be used. The dates of the different event windows are displayed in table 1.

Table 1 Event Window Date's

	Estimation window	**Gap**		**Event window**	
Event windows	T_0	T_g	T_1	Event	T_2
1	24 May 2022	14 Nov 2022	29 Nov 2022	30 Nov 2022	1 Dec 2022
2	20 May 2022	10 Nov 2022	25 Nov 2022	30 Nov 2022	5 Dec 2022
3	18 May 2022	8 Nov 2022	22 Nov 2022	30 Nov 2022	7 Dec 2022
4	11 May 2022	1 Nov 2022	15 Nov 2022	30 Nov 2022	14 Dec 2022
5	11 May 2022	1 Nov 2022	15 Nov 2022	30 Nov 2022	21 Dec 2022
6	11 May 2022	1 Nov 2022	15 Nov 2022	30 Nov 2022	29 Dec 2022

3.3 Benchmark

In order to compare the stock market returns of the stocks that will be analysed in this paper a benchmark is needed. With the guidelines from Battaglia & Musar (2000) there where three benchmarks that could be of interest. First, the S&P500 and secondly the Nasdaq-100, lastly the All-World index. The S&P500 (*S&P 500*, 2023) is index with broad view of the total market. It consists of the 500 large cap publicly traded companies that are listed in the United States. On the other side the NASDAQ 100 (*Nasdaq-100®Index*,2023) is in index that includes the 100 largest non-financial companies that are listed on the NASDAQ. Which mainly consists of technology and growth-oriented companies. At last, the All-World index (*Vanguard FTSE*, 2023) includes markets from all over the world. Thus, being a perfect index to use for benchmarking against the global market. One could argue that it is fair to compare the researched companies, which are of course technology companies against a benchmark that is predominantly tech focussed. One argument: for this could be that other tech companies stock prices could also have been influenced by prompting innovation of OpenAI's ChatGPT. The main question here is do you want to compare the stock market returns to the broader market or to a more tech focused market? Because the factor data will be imported from the French (2023) data base, a limited number of benchmarks can be used. The suited benchmark for this research is a combination of the Nasdaq and S&P 500. This unfortunately means that the All-world index will be excluded.

3.4 Data

Stock market data is collected from the following companies: Microsoft (NASDAQ: MSFT), Alphabet (NASDAQ: GOOGL), Verizon (NYSE: VZ), Baidu (NASDAQ: BIDU). All stock market data are be extracted from Yahoo Finance (Finance Yahoo). For each trading day the Closing price is used. The reason for only using the closing price is that fluctuations within a trading day will not have any influence on the data. The factor data are retrieved from Kenneth R. French data base (French, 2023). This data base has been created by one of the researchers that came up with the 3 and 5-factor models. Its main purpose is the provide the factor data needed in order to calculated excess returns using the Fama-French models.

3.5 Methodology

To be able to test the hypothesis, excess returns need to be found. In order to do so the model is trained during a 120-day estimation window. The excess returns that are found in the 120-day estimation window are used in order to predict the abnormal returns within the different event windows. These event windows are discussed in 3.2. Next, the abnormal returns are added together to the cumulative abnormal returns. These then are tested against the null hypnotises. To be able to find the excess returns, four different economic market models are used that help identify excess returns. These models are:

(1) $R_{jkt} = \alpha_{jk} + \beta_{1jk}R_{mt} + \varepsilon_{jkt}$

(2) $R_{jkt} = \alpha_{jk} + \beta_{1jk}R_{mt} + \beta_{2jk}(SMBt) + \beta_{3jk}(HMLt) + \varepsilon_{jkt}$

(3) $R_{jkt} = \alpha_{jk} + \beta_{1jk}R_{mt} + \beta_{2jk}(SMBt) + \beta_{3jk}(HMLt) + \beta_{4jk}(RMWt) + \varepsilon_{jkt}$

(4) $R_{jkt} = \alpha_{jk} + \beta_{1jk}R_{mt} + \beta_{2jk}(SMBt) + \beta_{3jk}(HMLt) + \beta_{4jk}(RMWt) + \beta_{5jk}(CMAt) + \varepsilon_{jkt}$

The first model is Capital Asset Prices Model (Sharpe, 1964), secondly the Fama-French 3-factor model (Fama & French, 1993), thirdly Carhart's 4 factor model (Carhart, 1997). Fourthly, Fama- French 5-factor model (Fama & French, 2015).

Multiple variables will be used within these models. Within these models R_{jkt} stands for the excess return of the kth stock at time *t*. R_{mt} is implies the excess return of the market ETF at time *t*. $\beta_{1jk,\text{ up to }5jk}$ are the coefficients of the different factors. Rjkt is the return of the asset at time t. α_{jk} stands for the relative performance of variable R_{jkt} against R_{mt}.

The following factors are used: SMBt (Small Minus Big) represents the size factor. HMLt (High Minus Low) represents the value factor. RMWt (Robust Minus Weak) is the profitability factor. CMA (Conservative Minus Aggressive) stands for the investment factor. All at time t. ε_{jkt} is used for the error term. Next the differences between the observed values and the estimated values are used to estimate the abnormal returns within their respective models will need to be calculated (MacKinlay, 1997). To do so the following formula's will be used:

(5) $AR_{jkt} = R_{jkt} - \alpha_{jk} - \beta_{1jk}R_{mt}$

(6) $AR_{jkt} = R_{jkt} - \alpha_{jk} - \beta_{1jk}R_{mt} - \beta_{2jk}(SMBt) - \beta_{3jk}(HMLt)$

(7) $AR_{jkt} = R_{jkt} - \alpha_{jk} - \beta_{1jk}R_{mt} - \beta_{2jk}(S\ MBt) - \beta_{3jk}(HMLt) - \beta_{4jk}(RMWt)$

(8) $AR_{jkt} = R_{jkt} - \alpha_{jk} - \beta_{1jk}R_{mt} - \beta_{2jk}(SMBt) - \beta_{3jk}(HMLt) - \beta_{4jk}(RMWt) - \beta_{5jk}(CMAt)$

At last, the abnormal returns need to be summed up over the total event window. Meaning t = T_1 + 1, … , T_2 , shown in (9).

(9) $CAR_{jk}(t_1, t_2) = \sum_{t=t_1}^{t_2} \mathrm{AR}_{jkt}$

All the CAR's (Cumulative abnormal returns) will reflect the interaction between the tested stock and the benchmark index within the tested period. Even though the CAR_{jk} gives somewhat of an indication on the outcome it's not good enough in order to used it for testing the null hypothesis. That's why Kothari & Warner (2007) came up with a statistical testing method. Within statistical testing method, AR_{jkt} is expected to be normally distributed. Besides being normally distributed the mean and constant variance will be zero. Kothari and Warner came up with the following formula.

(10) $\frac{CAR_{jk}(t_1,t_2)}{[\sigma^2(t_1,t_2)]_{jkt}^{1/2}}$

For calculating $[\sigma^2(t_1, t_2)]_{jtk}$ formula 11 are used.

(11) $\sigma^2(t_1, t_2)_{jkt} = L\sigma^2(AR_{jkt})$

Where L stands for the length of the event window, this includes 3, 7, 11, 21, 26 & 31 days. The code of the programmed model can be found in the8 Appendix – R-Code.

4 Findings

4.1 Cumulative Abnormal Returns

The cumulative abnormal returns of the four different models can be found in Table 2, Table 3,Table 4 & Table 5. Each table represents one of the four models used.

Table 2 cumulative abnormal returns CAPM Model

CAPM	MSFT	GOOGL	VZ	BIDU
[-1, 1]	0.037841	0.038099	0.019149	0.134886
[-3, 3]	0.068421*	0.082023**	0.019295	0.269208*
[-5, 5]	0.089544**	0.095234**	0.061776	0.303116**
[-10, 10]	0.177145***	0.159522*	0.153641**	0.451783**
[-10, 15]	0.227891***	0.178005**	0.216212***	0.479646**
[-10, 20]	0.272694***	0.217186**	0.300414***	0.550978**

** Indicates significance at 95%, ** indicates significance at 99%, *** indicates significance at 99,9%*

Table 3 cumulative abnormal returns Fama-French 3-factor model

3 Factor	MSFT	GOOGL	VZ	BIDU
[-1, 1]	0.038254	0.032628	0.018723	0.134573
[-3, 3]	0.0689*	0.06611**	0.020079	0.259388*
[-5, 5]	0.088397**	0.074218**	0.059341	0.304841**
[-10, 10]	0.188495***	0.119173*	0.147854**	0.489656**
[-10, 15]	0.2265***	0.148413**	0.209455***	0.524424**
[-10, 20]	0.273675***	0.193955**	0.292722***	0.602280**

** Indicates significance at 95%, ** indicates significance at 99%, *** indicates significance at 99,9%*

Table 4 cumulative abnormal returns Carhart model

Carhart	MSFT	GOOGL	VZ	BIDU
[-1, 1]	0.013881	0.033611	0.018816	0.137112
[-3, 3]	0.063462*	0.065901*	0.020063	0.260008*
[-5, 5]	0.079579**	0.073264**	0.059541	0.308332**
[-10, 10]	0.163556***	0.118372*	0.147156**	0.505489**
[-10, 15]	0.192779***	0.14759**	0.208737***	0.540719**
[-10, 20]	0.231345***	0.193286**	0.292139***	0.615507**

** Indicates significance at 95%, ** indicates significance at 99%, *** indicates significance at 99,9%*

Table 5 cumulative abnormal returns Fama-French 5-factor model

5 Factor	MSFT	GOOGL	VZ	BIDU
[-1, 1]	0.038254	0.033657	0.018816	0.131831
[-3, 3]	0.065887*	0.070106**	0.013298	0.260622**
[-5, 5]	0.082417**	0.07884**	0.056133	0.307836***
[-10, 10]	0.169668***	0.135124**	0.138072**	0.495314***
[-10, 15]	0.199776***	0.166766**	0.198337***	0.529072***
[-10, 20]	0.239871***	0.216654***	0.279467***	0.601313***

** Indicates significance at 95%, ** indicates significance at 99%, *** indicates significance at 99,9%*

The results show an outperformance of the competing companies compared to the benchmark. First, all the cumulative abnormal returns are positive. Meaning that the competing companies outperformed the benchmark in every event window within the four models. The cumulative abnormal returns range from 1.4% to 61%. The lowest returns are found within the shortest event window. This is as expected because the shortest event window only consists of 3 trading days. While ChatGPT got released on the 30th of November it did take some time for the public to notice, as can be seen at Figure 1 Google Trends: ChatGPT. The first event window only consisted of two trading days on which ChatGPT was available. The efficient market hypothesis states that asset prices will reflect all available information. Within the first trading days most of the investors were not up to date with all the information about ChatGPT's abilities leading to the information not being priced in. Meaning the competing companies stock price with too little amount of time to outperform the benchmark. On the other hand, the highest cumulative abnormal returns are found within the longest event windows. There is a clear correlation between the length of the event windows and the amount of cumulative abnormal returns. Contributing to this is the fact that within the long event windows the stock prices of the competing companies have more time to accumulate more returns. Over time following the Efficient market hypothesis investors had the time to correctly price in the new information about ChatGPT within the individual stock prices. Interesting to note is that the knowledge about ChatGPT lead to an increase in stock price for most of the competing companies and an outperformance of the benchmark for all of them.

4.2 P-Values

The p-values of the cumulative abnormal returns indicate no significant cumulative abnormal returns within the 3-trading day event window. However, when the event window length increases the cumulative abnormal returns will be more significant. To the point that starting from the [10,-10] event window up and until the largest event window, all cumulative abnormal returns have at least an 95% significance. Even at a significance level of 99% and 99.9% a decent amount of cumulative abnormal returns appears to be significant. For Baidu, Alphabet and Microsoft only one out of the six event windows did not return significant cumulative abnormal returns. In contrast to the rest, Verizon had the least amount of significant cumulative abnormal returns. Only the event windows with 21 and more trading days delivered significant cumulative abnormal returns.

4.3 Actual Returns

As can be seen in Figure 3, Figure 4, Figure 5 & Figure 6, the high cumulative abnormal returns are caused by the discrepancy between the predicted and the actual returns of all four of the models. All four models predicted increasing negative returns over time. The longer the event window the higher the predicted negative returns. In contrast to the models' predictions, Alphabet was the only company with a negative return over the long event window, Microsoft broke even, Baidu and Verizon had a positive return. Meanwhile the models predicted that they would have a negative return. With the models predicting negative returns and the competing companies besides Alphabet delivering negative returns, led to a large difference between the actual and predicted returns, thus high cumulative abnormal returns.

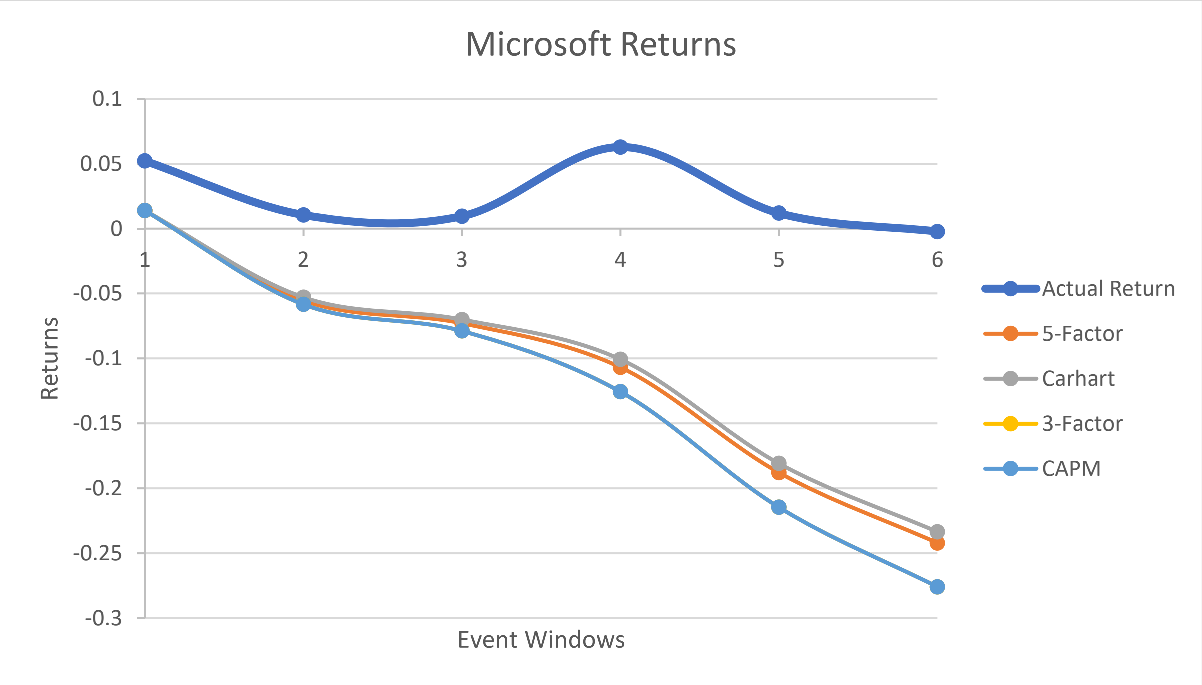

Figure 3 Microsoft Returns; Showing the returns of the four models and the actual returns for Microsoft's stock for the six event windows.

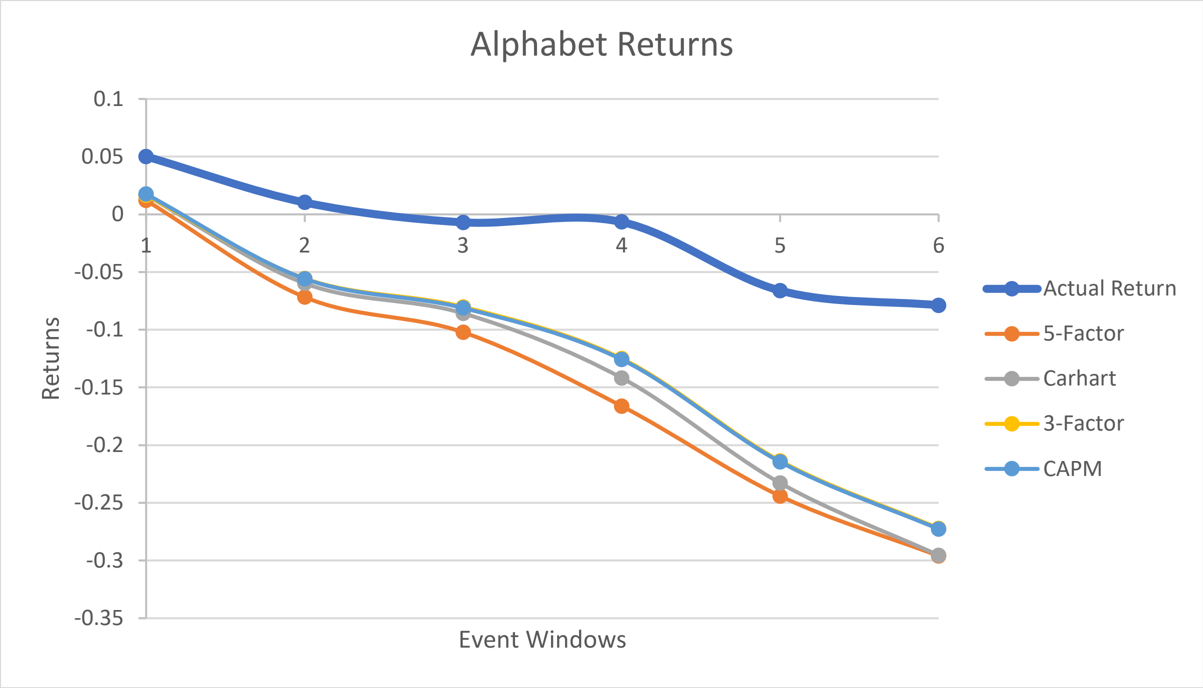

Figure 4 Alphabet Returns; Showing the returns of the four models and the actual returns for Alphabet's stock for the six event windows.

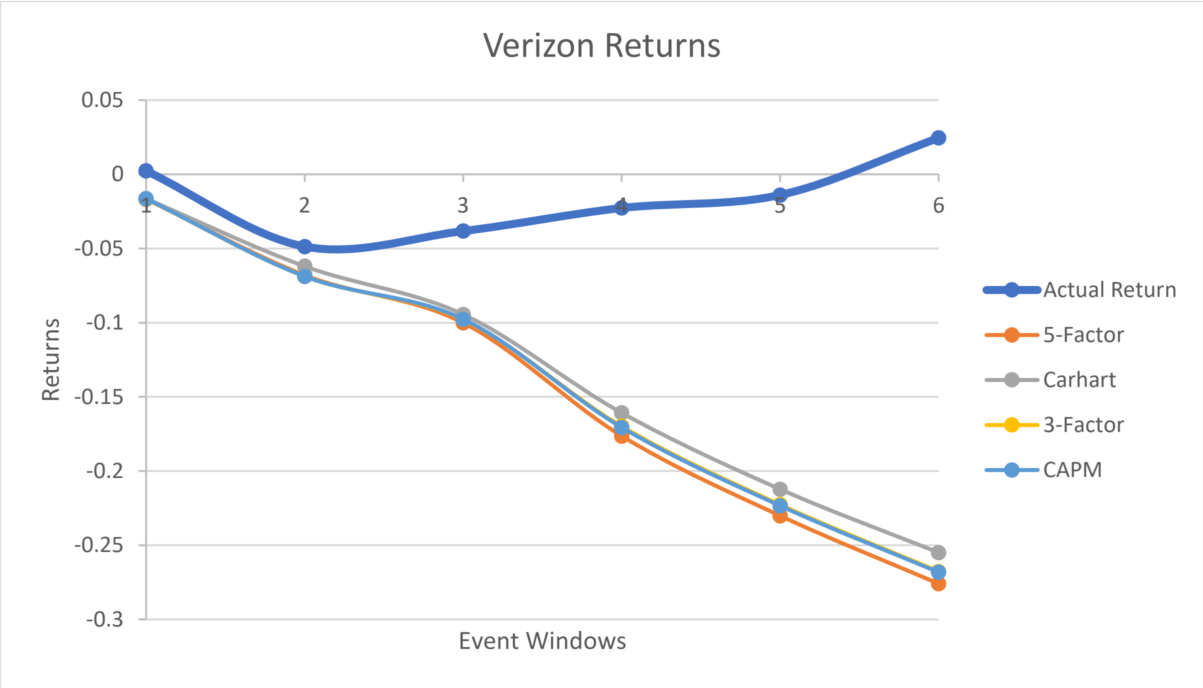

Figure 5 Verizon Returns; Showing the returns of the four models and the actual returns for Verizon's stock for the six event windows.

WAGENINGEN
UNIVERSITY & RESEARCH

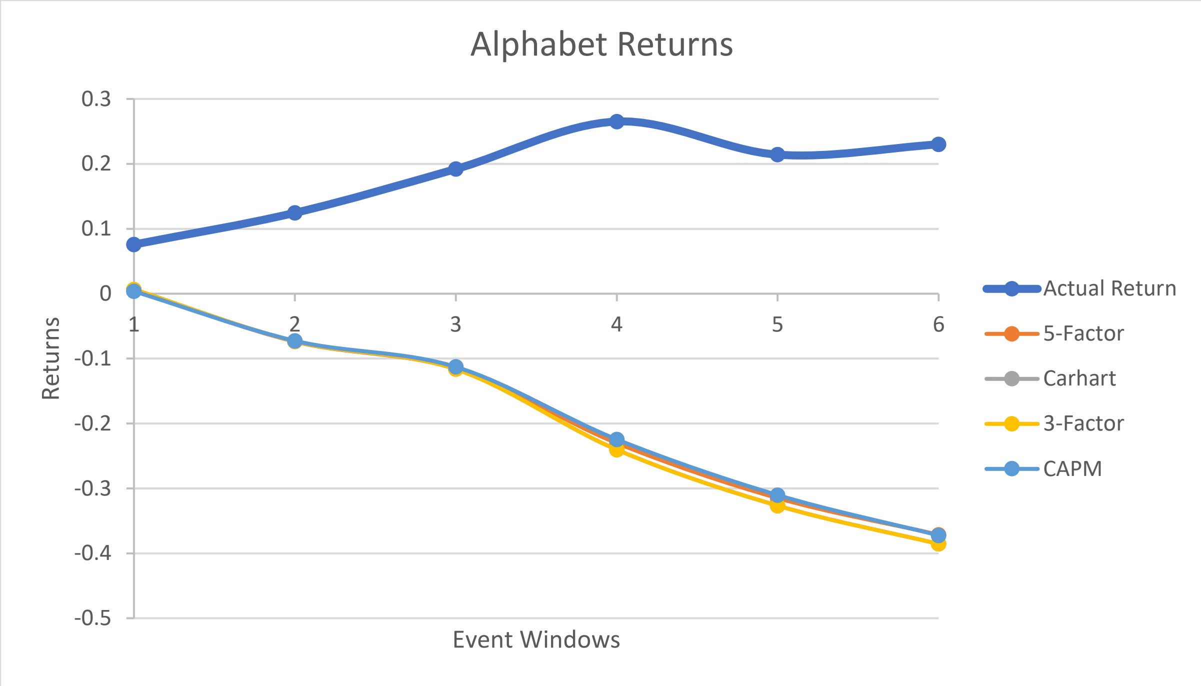

Figure 6 Baidu Returns; Showing the returns of the four models and the actual Baidu's stock for the six event windows.

4.4 Factor Data

To understand the significance of the results of this research it is important to validate the accuracy of the four models that have been used during this research. As Figure 3, Figure 4, Figure 5, & Figure 6, demonstrate, all of the models behaved closely to each other. There are no major differences between the predictions of the four models. The differences are especially close between the 3-Factor and Carhart model. The reason for the similar behaviour between the models can be explained by the factors that mainly contribute to the outcome of the models. The main contributing factors are Market factor ($\beta_{1jk}R_{mt}$) and the Size factor ($\beta_{2jk}SMBt$).This can be seen in Table 6 Microsoft β factor values,

Table 7 Alphabet β factor values,

Table 8 Verizon β factor values, &

Table 9 Baidu β factor values. Important to note is that when calculating the average β of the factors six data points haven been used, one for each of the six event windows. The estimation period of event window: [-10, 10], [-10, 10], and [-10, 10] consist of the same trading days. They all start 140 trading days before the actual event based on 10 days prior to the event, 10 days for the event window gap, and 120 days estimation window. They all end 20 before the event, 10 days prior to the event, and 10 days for the gap. Meaning that for these three event windows the estimation window ranges from 11th of May 2022 until 15th of November 2022. As a consequence of this the model is trained with the same data. Resulting in the fact that the β of the factors within these estimation windows is the same. This heavily influences the average of the factor β's since three out of the six data points are the same.

Table 6 Microsoft β factor values

Microsoft	5-Factor	Carhart	3-Factor	CAPM	Average
Mkt.RF	0.011415	0.011483	0.011351	0.011787	0.011509
SMB	-0.009	-0.0088	-0.01049	-	-0.00943
HML	-0.00361	-0.00453	-0.00402	-	-0.00405
RMW	0.002547	0.002535	-	-	0.002541
CMA	-0.00362	-	-	-	-0.00162

Table 7 Alphabet β factor values

Alphabet	5-Factor	Carhart	3-Factor	CAPM	Average
Mkt.RF	0.011545	0.011691	0.011624	0.01288	0.011935
SMB	-0.00641	-0.00597	-0.0069	-	-0.00642
HML	-0.00382	-0.00589	-0.00561	-	-0.00511
RMW	0.001419	0.001398	-	-	0.001409
CMA	-0.00365	-	-	-	-0.00365

Table 8 Verizon β factor values

Verizon	5-Factor	Carhart	3-Factor	CAPM	Average
Mkt.RF	0.004445	0.004323	0.004303	0.003957	0.004257
SMB	-0.00215	-0.00248	-0.00266	-	-0.00243
HML	-0.00125	0.000304	0.000356	-	-0.0002
RMW	0.000261	0.000279	-	-	0.00027
CMA	0.002721	-	-	-	0.002721

Table 9 Baidu β factor values

BAIDU	**5-Factor**	**Carhart**	**3-Factor**	**CAPM**	**Average**
Mkt.RF	0.008506	0.008459	0.00903	0.011221	0.009304
SMB	0.006005	0.005833	0.012004	-	0.007947
HML	-0.00261	-0.00182	-0.00362	-	-0.00268
RMW	-0.00092	-0.00091	-	-	-0.00092

CMA	0.0014	-	-	-	0.0014

The Market factor is included in all four of the models and the Size factor is included in three out of the four models. The size factor is only absent for the CAPM model. Since these are most of the time the largest contributors to the outcomes, the cumulative abnormal returns do not have a big variance between the four models. Moreover, the β of the RMW (Robust Minus Weak) factor is often very small. Leading to it having a low influence on the outcome of the model. Besides the low β of RMW the β's of Carhart and 3-factor model are often very close. The combination of the low β of RMW and the small difference between the β's of the Carhart and 3-Factor model result in the fact that the predictions of both models are very close to each other. All in all, this means that the predictions of the models are mostly influenced by the market, and Size factors. The fact that the predictions of the four models are close to each other goes to show that the results, namely the cumulative abnormal returns are robust to the different model specifications. Validating that the actual returns of the companies outperformed every model.

5 Discussion

5. 1 Theory

Up until today, little research has been done on the impact of new AI technology on the stock price of competing companies. However, research has been done on the consequence of acquisitions on stock market prices of competing companies (Elad & Zivney, 2017). As well as how new products affect the maker's and competing companies stock price (Lee & Chen, 2009). Mahajan, Sharma, and Buzzell (1993) already concluded that the effect on competitors of announcing a new product is mainly dependent on possible substitutions and most important the way the market has an expansion effect. Chen, Ho, and Ik (2005), stated that the announcement of new products could lead to a negative effect on the market price of competing organization, an effect that will get priced in the stock price of the competing organization. All in all, a lot of different events that could influence the stock market price of competing companies.

The analysis of the effect of ChatGPT's introduction on the stock market returns of competing companies has yielded intriguing insights that align with both Modern Portfolio Theory and Efficient Market Hypothesis. The event window revealed a notable trend, with the stock prices of three out of the four competing companies experiencing an upward surge. This positive response to the introduction of ChatGPT suggests that investors perceived the AI-driven innovation as a valuable asset, potentially indicating a higher demand for companies embracing such technologies. Enholm, Papagiannidis, Mikalef, and Krogstie (2021) found that that there are multiple options for companies to implement of AI technology in companies in order to increase their revenue. Moreover, Mikalef and Gupta (2021) discovered that companies which have established a methodical strategy for integrating and utilizing AI, along with building organizational proficiency in these innovative technologies, have experienced improvements in their financial performance.

From the perspective of Modern Portfolio Theory, these results can be interpreted as an endorsement of diversification strategies that encompass companies adopting AI-driven innovations (Markowitz, 1952). The observed outperformance of these companies against the benchmark indicates that investors may have benefited from incorporating AI-focused firms into their portfolios. This aligns with Modern Portfolio Theory's core principle of optimizing risk-return trade-offs through diversification, as investors bought more AI related stocks

(Singh, 2023), they may have achieved enhanced returns by strategically including stocks of companies operating in the AI landscape.

In light of Efficient Market Hypothesis, the findings pose intriguing questions regarding market efficiency. The fact that investors could identify and respond to the value potential of competing AI-based companies, leading to positive stock price movements, challenges the notion of complete market efficiency. Behavioural finance (Kapoor & Prosad, 2017) has significantly altered the neoclassical perspective on financial markets (Shefrin, 2002), causing fundamental disruptions. It contends that investors are not always capable of accurately assessing the value of alternative choices, struggle with estimating and updating event probabilities, and often fail to diversify adequately as prescribed by Markowitz's portfolio theory. The collective irrational behaviour of a large segment of market participants, acting in a coordinated manner, can exert influence over asset prices (Szyszka, 2013). The once-assumed risk-free and cost-effective nature of arbitrage has been challenged, always undermining the notion that the self-regulating market mechanism functions flawlessly. Consequently, the rectification of mispricing does not consistently occur rapidly or sufficiently. This leads to the rejection of the validity of the Efficient Market Hypothesis (Szyszka 2007). In essence, the observed positive stock price movements suggest that investors were able to gain an advantage by responding to the potential value and growth prospects associated with AI-driven companies. This challenges the strict interpretation of the Efficient Market Hypothesis, as it indicates that market participants were able to identify and act upon certain informational advantages, potentially leading to outcomes that deviate from the hypothesis of instantaneous and complete assimilation of all available information into stock prices. The efficient assimilation of AI-related information by investors may have influenced the observed outperformance, suggesting that certain informational advantages may persist despite the assumption of Efficient Market Hypothesis that stock prices fully reflect all available information. The findings raise intriguing questions regarding market efficiency and the role of information dissemination in the context of AI technologies; How quickly do investors incorporate information about AI technologies into their investment decisions?

5.2 Benchmark

For the benchmark, a combination of the S&P 500 and Nasdaq indices has been utilized, focusing solely on stocks listed in the US. Despite being a Chinese company, Baidu's listing on the Nasdaq includes it in the benchmark. Ideally, a more comprehensive benchmark consisting of global stocks would have been preferred. However, the unavailability of factor data on French's database site (2023) prevented the use of a worldwide index. In an ideal scenario, a custom-created benchmark encompassing the world index would have provided a more robust reference for comparison. However, due to the limitations imposed by data availability, the chosen benchmark serves as a relevant approximation for evaluating Baidu's performance against US-based stocks. It is essential to acknowledge these limitations when interpreting the findings.

5.3 Factors

As stated in the results chapter, an examination of the five factors employed in this research reveals that not all factors contributed equally to the model's outcomes. In certain instances, the RMW (Robust Minus Weak) factor exhibited such negligible influence that the differences between the Carhart and 3-Factor models became almost negligible. This finding prompts a critical question about the representativeness of the RMW factor for the companies analysed in this study. A plausible explanation for the limited impact of the RMW factor could be the composition of the analysed companies. All the companies included in this research demonstrate robust financials, boasting a market capitalization exceeding 50 billion USD and consistent profitability. Consequently, none of these companies are represented in weak operating profitability portfolios, rendering the RMW factor, which considers both robust and weak organizations, less relevant to the robust companies exclusively considered in this research. While the RMW factor may lack meaningful significance within the models, it is evident that both the SMB (Small Minus Big) and Mk. RF (Market Risk-Free) factors exert substantial influence. This influence can be attributed to the widespread presence and prominence of companies like Microsoft, Alphabet, and Verizon, which are highly recognized and commonly included in diverse portfolios. As a result, these companies are present in both small and large portfolios, amplifying their impact on the models. In summary, the results indicate that the RMW factor does not significantly contribute to the models used in this research due to the exclusive focus on robust companies with strong financials. On the other hand, the SMB and Mk. RF factors play pivotal roles in shaping the outcomes, driven by the presence of widely known companies in various portfolios.

5.4 Microsoft

It is important to talk about to what extent Microsoft is actually a competitive company. As discussed in the background, it is clear to see that Microsoft has mingled with OpenAI (Capoot, 2023). A 10 billion USD investment cannot be seen as neutral position. Therefore, this is an important aspect to include in the conclusion. To what extent can Microsoft be seen as a competitive company. This remains difficult to say until it is made clear what the exact relationship between Microsoft and OpenAI is. As such, there is nothing on Microsoft's recent quarterly earnings report about Microsoft's stake in OpenAI (Investor Relations - Microsoft, 2023). It seems that Microsoft is doing everything possible to make the relationship with OpenAI as unclear as possible. Reason may be that they fear monopoly claims. Since the revenue turned by OpenAI currently does not go on Microsoft's balance sheet, Microsoft will be included in the companies. Should more news come out in the future about Microsoft's interest in OpenAI, it will be of great importance to see if Microsoft can still be seen as a competitive company. Should this not be the case, Microsoft should be removed from the list of competitive companies and if not included within the results of this study.

5.5 Baidu

When examining the cumulative abnormal returns, Baidu has demonstrated exceptional financial performance, achieving a remarkable 60% abnormal returns. These high cumulative abnormal returns can be attributed primarily to Baidu's outstanding performance during the event window compared to its performance during the training period. Baidu's unique case can be further understood when considering its distinct geographical focus. As a technology company mainly based in Asia (Baidu Inc, 2023), Baidu operates within a specific market landscape, which influences its growth trajectory and business dynamics. Its extensive presence and deep-rooted connections within the Asian market have granted Baidu an advantageous position, leading to greater opportunities for expansion and revenue generation. One of the intriguing factors impacting Baidu's potential and cumulative abnormal returns is the unavailability of ChatGPT software in China due to restrictive signup processes. This limitation creates a market void in the country for AI-powered language models, opening the door for domestic companies like Baidu to explore and capitalize on this unmet demand. With Baidu's vast expertise in AI research and development, the company could seize this opportunity to design and introduce its proprietary software that addresses the unique linguistic and cultural nuances of the Chinese market. As investors keenly recognize the potential for Baidu to fill the ChatGPT gap in China, they could view it as an enticing avenue for additional revenue generation. This optimism could be evident in how investors incorporate this foreseeable revenue stream into their valuation of Baidu's stock, potentially explaining the remarkable outperformance of the company in comparison to the benchmark.

However, Baidu's trajectory is not without its potential uncertainties. The political landscape in China can be fluid, leading to shifting regulatory policies and market conditions, which could impact Baidu's stock price and overall performance. Research conducted by Liu (2012) underscores the dynamic nature of investing in China, with risks and opportunities arising unexpectedly. Furthermore, Tao et al. (2020) found that fluctuations in global economic policy uncertainty can significantly sway the performance of the Chinese stock market. Given that the benchmark predominantly focuses on the US market, it may not fully reflect the unique volatility and intricacies present in Chinese stocks. In light of these complex factors, it is essential to consider both the AI opportunity for Baidu and the specific volatility of the Chinese stock market when interpreting the observed cumulative abnormal returns.

5.6 Estimation Window

The dataset utilized for training consists of 120 data points, representing individual trading days. In line with MacKinlay's recommendation (1997), meaning that the estimation window encompass 120 trading days. Notably, the majority of the estimation windows exhibited a prevailing negative performance for both benchmarks and stocks, as evidenced by the downward trend depicted in Figure 7. While occasional positive returns were observed within some estimation windows, the model's training data contained limited instances of such positive trading days. In contrast to the estimation's windows, the event windows, most stocks displayed positive returns, the model encountered challenges in accurately predicting these outcomes. To mitigate this limitation, a potential solution could involve adopting a longer estimation window, one that includes a more diverse mix of trading days with both positive and negative returns. By doing so, the model can gain a more comprehensive understanding of the complex relationships between various factors and stock performance. Future research endeavours might explore the effects of employing such an extended estimation window, incorporating both upward and downward trends, to enhance the model's predictive capabilities.

Figure 7 Google Finance: MSFT, GOOGL, VZ, & BIDU Graph

5.7 Event Window

For the purpose of this research, an event window duration of up until 20 trading days was chosen, based on a study by Oler et al. (2008). Cautioned against using event windows longer than 30 trading days, citing potential inaccuracies0 in their estimation. Despite this cautionary note, it can be argued that exploring the results with longer event windows could offer valuable insights. One compelling rationale for considering extended event windows is the dynamic nature of news dissemination, particularly concerning a technology as groundbreaking as ChatGPT. As illustrated in Figure 1, there is a sustained increase in views and traffic towards ChatGPT, suggesting that its impact might not be fully absorbed within the initial 20 trading days. This phenomenon indicates that a significant portion of potential users may not yet have fully grasped the possibilities and opportunities presented by ChatGPT. Moreover, Figure 8 reveals a notable surge in AI-related stock prices, such as the Xtrackers Artificial Intelligence and Big Data UCITS ETF, after the termination of the longest event window on 30 December 2022. Unfortunately, this upward trend lies beyond the scope of the current research due to the limited event window length. However, the intriguing combination of Figure 1 and Figure 8 sparks curiosity about the potential benefits of utilizing longer event windows to capture more extended trends and effects. Despite this consideration, it is pertinent to acknowledge that the current results already demonstrate significant cumulative abnormal returns. Nonetheless, it is worth contemplating that these outcomes might be partly influenced by the model's inadequate exposure to positive trading days during its training phase. Thus, future research endeavours could prove fruitful by extending both the estimation window and the event window, affording a more comprehensive and nuanced analysis of the intricate dynamics related to ChatGPT and AI-related stocks.

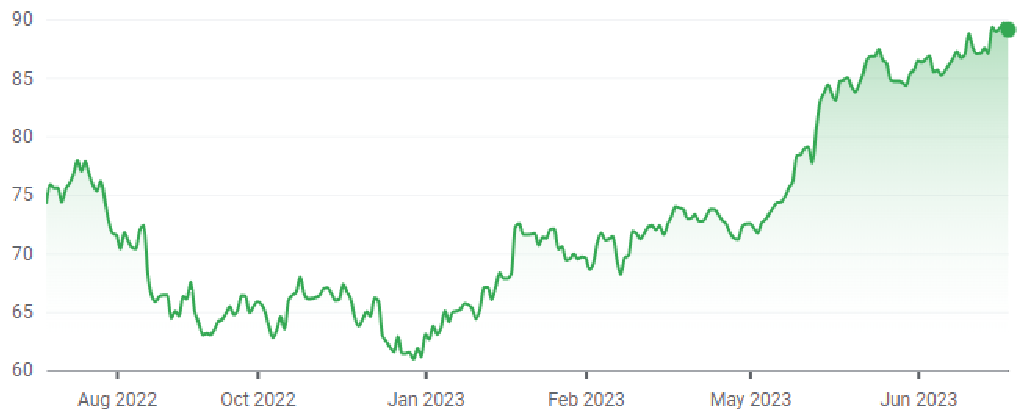

Figure 8 Google Finance: Xtrackers Artificial Intelligence UCITS ETF performance

An important constraint in this research lies in the limited number of competing companies considered, currently standing at just four and potentially reducing further depending on the extent of Microsoft's stake in OpenAI. This limitation arises from the stringent selection criteria applied to identify relevant competitors, specifically focusing on search engine companies that are publicly listed on the stock market. As a result, the set of competitors analysed in this study is inherently constrained. However, it is essential to recognize that the competition faced by OpenAI's ChatGPT extends far beyond the search engine market, encompassing various other domains like writing tools, spelling checkers, programming code, and more. The existence of these alternative aspects indicates that ChatGPT likely competes with a much larger and diverse array of companies than those explicitly addressed in this research. To gain a more comprehensive understanding of the competitive landscape surrounding ChatGPT, it would be valuable to conduct further research that expands the database to include a broader scope of stock market-listed competitors. Companies like Oracle, that recently launched its generative AI program("Faster Insights With Oracle AI and Machine Learning," 2023) and Adobe's AI Photoshop ("AI Photo Editing With Photoshop - Adobe," 2023). By doing so, a more nuanced and comprehensive analysis of the competitive dynamics faced by ChatGPT can be attained, shedding light on the full extent of its competition in the market.

5.8 Implications for Investors

Since the introduction of ChatGPT led to cumulative abnormal returns, it is important to comprehend how this information could be strategically employed to capitalize on future opportunities. As a consequence of the Efficient Market Hypothesis not holding true for this event, investors could have leveraged the event's information by strategically investing in the competing companies. Delving into the factors that caused this deviation from market expectations and exploring the specific aspects that rendered the event inefficient in terms of market response could shed light on the unique dynamics at play.

Although not established conclusively within this study, there appears to be an indication that a greater number of AI-related stocks outperformed the benchmark subsequent to the event (Investors Business Daily, 2023; Mitchell, 2023). While this research primarily focuses on the introduction of ChatGPT, there's an opportunity to discuss specific investment strategies that could have been employed by investors during and immediately after the event. Examining historical case studies or hypothetical scenarios may offer valuable insights into different approaches investors could adopt.

In the future, investors may endeavour to detect these rare events in their early stages in order to exploit their potential. However, identifying such opportunities at an early juncture is likely to be a challenging endeavour. Nevertheless, this research shows the potential for investors to profit from the subsequent escalation in asset prices.

6 Conclusion

The primary objective of this research was to investigate the effect of the introduction of ChatGPT on the stock market return of competing companies, namely Microsoft, Alphabet, Verizon, and Baidu. To achieve this, four estimation models, including CAPM, Fama-French 3 factor model, Carhart, and Fama-French 5 factor model, were employed to calculate estimated returns and abnormal returns. The estimation window, spanning 120 trading days, was utilized to estimate excess returns, while the event window ranged from 3 to 31 trading days to identify abnormal returns. Subsequently, returns were aggregated across the event window to derive cumulative abnormal returns, which were then subjected to statistical significance testing.

The results from the study appeared to be significant at a 90% significance level. Only the smaller event windows did not yield significant cumulative abnormal returns. Starting from +10 trading days event windows, all returns appeared to be significant. Within the larger event windows, most of the results were significant at a 99% significance level, and some even at a 99.9% significance level. Three out of the four competing companies had a positive return within the event window, and all of the companies surpassed the models' predictions. Overall, the results conclude that the introduction of ChatGPT led to significant abnormal returns. Meaning that H0 can be rejected and H1 can be assumed stating: The introduction of ChatGPT does appear to be related significantly to the abnormal returns of competing firms.

However, the research does have some limiting factors. Firstly, the number of competing companies, with only four being used, increases the risk of the results being influenced by any single company's behaviour. Fortunately, all four competing companies exhibited similar patterns, showing significant cumulative abnormal returns. Secondly, the length of the estimation and event window prevented the model from capturing the upward trend that started after the 30th of December 2022. Lastly, mostly negative trading days were included within the estimation window, leading to the model not having enough positive trading days to train effectively.

For future research, two suggestions can be made. Firstly, research could explore adding competing companies outside of the search engine market, thereby increasing the pool of competing companies for a more comprehensive analysis. Secondly, conducting research with longer estimation windows and event windows could be beneficial in gaining deeper insights into the long-term effects of ChatGPT's introduction on stock prices and market behaviour.

The information within this research can serve as a motivating factor for investors to recognize that during significant innovation events, an outperformance of the benchmark is attainable. It falls upon the individual investor to devise strategies to capitalize on the insights stemming from these innovations. Acknowledging that, despite the Efficient market hypothesis it is possible to benefit from these events.

7 References

AI photo editing with Photoshop - Adobe. (2023). Retrieved from https://www.adobe.com/products/photoshop/ai.html

AI Stocks: Tech Giants, Cloud Titans, Chipmakers Battle For An Edge. (2023). Retrieved from https://www.investors.com/news/technology/artificial-intelligence-stocks/

Armitage, S. (1995). EVENT STUDY METHODS AND EVIDENCE ON THEIR PERFORMANCE. *Journal of Economic Surveys*, *9*(1), 25–52. https://doi.org/10.1111/j.1467-6419.1995.tb00109.x

Ashok, T. A. (1994). A study of product and financial market interactions. Retrieved February 20, 2023, from https://www.proquest.com/openview/1299b3e0a9c7ed8e98b7ea8a0fbc9b69/1?pq-origsite=gscholar&cbl=18750&diss=y

Ashta, A., & Herrmann, H. (2021). Artificial intelligence and fintech: An overview of opportunities and risks for banking, investments, and microfinance. *Strategic Change*, *30*(3), 211–222. https://doi.org/10.1002/jsc.2404

Baidu announces Fourth quarter and Fiscal Year 2022 results | Baidu Inc. (2023, February 22). Retrieved from https://ir.baidu.com/news-releases/news-release-details/baidu-announces-fourth-quarter-and-fiscal-year-2022-results

Balcilar, M., Bonato, M., Demirer, R., & Gupta, R. (2018). Geopolitical risks and stock market dynamics of the BRICS. *Economic Systems*, *42*(2), 295–306. https://doi.org/10.1016/j.ecosys.2017.05.008

Bash, A., & Alsaifi, K. (2019). Fear from uncertainty: An event study of Khashoggi and stock market returns. *Journal of Behavioral and Experimental Finance*, *23*, 54–58. https://doi.org/10.1016/j.jbef.2019.05.004

Battaglia, J. S., & Musar, R. C. (2000). Picking the right benchmark - ProQuest. *Journal of Accountancy*, *Vol. 190, Iss. 2, 63-70*. Retrieved from https://www.proquest.com/docview/206760263?parentSessionId=Wcp8D539QuR2NA%2BiiUfF%2BTsiz1KO%2F2Ih1o2ffR%2BuVyE%3D&accountid=27871

Brown, S. J., & Warner, J. B. (1985). Using daily stock returns. *Journal of Financial Economics*, *14*(1), 3–31. https://doi.org/10.1016/0304-405x(85)90042-x

Capoot, A. (2023, February 7). Microsoft announces new multibillion-dollar investment in ChatGPT-maker OpenAI. *CNBC*. Retrieved from https://www.cnbc.com

Carhart, M. M. (1997). On Persistence in Mutual Fund Performance. *Journal of Finance*, *52*(1), 57–82. https://doi.org/10.1111/j.1540-6261.1997.tb03808.x

Chen, S., Ho, K. J., & Ik, K. H. (2005). The Wealth Effect of New Product Introductions on Industry Rivals. *The Journal of Business*, *78*(3), 969–996. https://doi.org/10.1086/429650

Company Overview | Baidu Inc. (n.d.). Retrieved from https://ir.baidu.com/company-overview

Constantz, J. (2023, February 3). California's New Gold Rush: Big Tech Moves to Gain the Edge in AI. Retrieved from https://www.bloomberg.com/news/articles/2023-02-03/big-tech-earnings-call-mentions-of-ai-spike-after-chatgpt-went-viral#xj4y7vzkg

Corrado, C. J., & Zivney, T. L. (1992). The Specification and Power of the Sign Test in Event Study Hypothesis Tests Using Daily Stock Returns. *The Journal of Financial and Quantitative Analysis*, *27*(3), 465. https://doi.org/10.2307/2331331

Damiran, S., Dorjdagva, O., Sukhee, B., & Myagmarsuren, T. (2022). Macroeconomic determinants of stock market volatility: Evidence from post socialist countries. *Journal of Eastern European and Central Asian Research*, *9*(4), 569–580. https://doi.org/10.15549/jeecar.v9i4.966

Devlin, J., Change, M., Lee, K., & Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Diebold, F. X., & Yilmaz, K. (2008, August). *Macroeconomic volatility and stock market volatility, worldwide*. https://doi.org/10.3386/w14269

WAGENINGEN
UNIVERSITY & RESEARCH

Elad, F. L., & Bongbee, N. S. (2016). Event Study on the Reaction of Stock Returns to Acquisition News. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2863243

Enholm, I. M., Papagiannidis, E., Mikalef, P., & Krogstie, J. (2021). Artificial Intelligence and Business Value: a Literature Review. *Information Systems Frontiers*, *24*(5), 1709–1734. https://doi.org/10.1007/s10796-021-10186-w

Event Studies in Economics and Finance on JSTOR. (n.d.). Retrieved from https://www.jstor.org/stable/2729691

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, *25*(2), 383. https://doi.org/10.2307/2325486

FAMA, E. F., & FRENCH, K. R. (1992). The Cross-Section of Expected Stock Returns. *The Journal of Finance*, *47*(2), 427–465. https://doi.org/10.1111/j.1540-6261.1992.tb04398.x

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*(1), 3–56. https://doi.org/10.1016/0304-405x(93)90023-5

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, *116*(1), 1–22. https://doi.org/10.1016/j.jfineco.2014.10.010

Faster Insights with Oracle AI and Machine Learning. (2023). Retrieved August 12, 2023, from https://www.oracle.com/artificial-intelligence/

Feiner, L. (2019, July 22). Microsoft invests " billion in artificial intelligence project co-founded by Elon Musk. *CNBC*. Retrieved from https://www.cnbc.com

French, K. R. (2023). Kenneth R. French - Data Library. Retrieved from http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

French, K. R. & Tuck School of Business at Dartmouth College. (2022.). Description of Fama/French 5 Factors (2x3). Retrieved March 2, 2023, from https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_5_factors_2x3.html

Global search engine desktop market share 2023 | Statista. (2023, April 26). Retrieved from https://www.statista.com/statistics/216573/worldwide-market-share-of-search-engines/

Gunday, G., Ulusoy, G., Kilic, K., & Alpkan, L. (2011). Effects of innovation types on firm performance. *International Journal of Production Economics*, *133*(2), 662–676. https://doi.org/10.1016/j.ijpe.2011.05.014

Introducing ChatGPT. (n.d.). Retrieved from https://openai.com/blog/chatgpt

Investor Relations - Microsoft. (2023). Retrieved from https://www.microsoft.com/en-us/investor/earnings/FY-2023-Q4/press-release-webcast

Kapoor, S., & Prosad, J. M. (2017). Behavioural Finance: a review. *Procedia Computer Science*, *122*, 50–54. https://doi.org/10.1016/j.procs.2017.11.340

Kenneth R. French - Data Library. (2023.). Retrieved from http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

Kothari, S., & Warner, J. B. (2007). Econometrics of Event Studies**We thank Espen Eckbo, Jon Lewellen, Adam Kolasinski, and Jay Ritter for insightful comments, and Irfan Safdar and Alan Wancier for research assistance. In *Elsevier eBooks* (pp. 3–36). https://doi.org/10.1016/b978-0-444-53265-7.50015-9

Lee, H. (1995). Shareholder wealth effects of new product rivalry: First movers, second movers, and laggards. *Ph.D. Thesis, University Of Maryland.*

Lee, R., Lee, J. H., & Garrett, T. C. (2019). Synergy effects of innovation on firm performance. *Journal of Business Research*, *99*, 507–515. https://doi.org/10.1016/j.jbusres.2017.08.032

Lee, R. P., & Chen, Q. (2009). The Immediate Impact of New Product Introductions on Stock Price: The Role of Firm Resources and Size*. *Journal of Product Innovation Management*, *26*(1), 97–107. https://doi.org/10.1111/j.1540-5885.2009.00337.x

Lintner, J. (1965). The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets. *The Review of Economics and Statistics*, *47*(1), 13. https://doi.org/10.2307/1924119

Liu, Y. (2012). Investing in China: opportunities and risks in the future. *Journal of Financial Risk Management*, *01*(01), 1–6. https://doi.org/10.4236/jfrm.2012.11001

MacKinlay, A. C. (1997). Event Studies in Economics and Finance. *Journal of Economic Literature*, *35*(1), 13–39. Retrieved from https://www.jstor.org/stable/2729691

Mahajan, V., Sharma, S., & Buzzell, R. D. (1993). Assessing the Impact of Competitive Entry on Market Expansion and Incumbent Sales. *Journal of Marketing*, *57*(3), 39–52. https://doi.org/10.1177/002224299305700303

Microsoft. (2023, February 22). Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web - The Official Microsoft Blog. Retrieved from https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/

Mitchell, C. (2023, August 1). 10 Best AI Stocks Of August 2023. Retrieved from https://www.forbes.com/advisor/investing/best-ai-stocks/

Nasdaq-100®Index. (n.d.). Retrieved from https://www.nasdaq.com/solutions/nasdaq-100

Oler, D., Harrison, J. S., & Allen, M. R. (2008). The danger of misinterpreting short-window event study findings in strategic management research: an empirical illustration using horizontal acquisitions. *Strategic Organization*, *6*(2), 151–184. https://doi.org/10.1177/1476127008090008

Poon, S. S., & Taylor, S. (1991). MACROECONOMIC FACTORS AND THE UK STOCK MARKET. *Journal of Business Finance & Accounting*, *18*(5), 619–636. https://doi.org/10.1111/j.1468-5957.1991.tb00229.x

Ramponi, M. (2023). How ChatGPT actually works. *News, Tutorials, AI Research*. Retrieved from https://www.assemblyai.com/blog/how-chatgpt-actually-works/

Sharpe, W. F. (1964). CAPITAL ASSET PRICES: A THEORY OF MARKET EQUILIBRIUM UNDER CONDITIONS OF RISK*. *The Journal of Finance*, *19*(3), 425–442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Shefrin, H. (2002). Beyond greed and fear. *Oxford University Press eBooks*. https://doi.org/10.1093/0195161211.001.0001

Singh, M. (2023, July 6). Retail traders pile into US stocks; focus shifts to EVs from AI. *Reuters*. Retrieved from https://www.reuters.com

S&P 500®. (n.d.). Retrieved from https://www.spglobal.com/spdji/en/indices/equity/sp-500/#overview

Srinivasan, S., Pauwels, K., Silva-Risso, J., & Hanssens, D. M. (2009). Product Innovations, Advertising, and Stock Returns. *Journal of Marketing*, *73*(1), 24–43. https://doi.org/10.1509/jmkg.73.1.024

Stock Market Insights | Seeking Alpha. (2023). Retrieved from https://seekingalpha.com/article/4587399-artificial-intelligence-may-be-the-hottest-topic-of-early-2023

Strong, N. (1992). MODELLING ABNORMAL RETURNS: A REVIEW ARTICLE. *Journal of Business Finance &Amp; Accounting*, *19*(4), 533–553. https://doi.org/10.1111/j.1468-5957.1992.tb00643.x

Sun, Y., Wu, M., Zeng, X., & Peng, Z. (2021). The impact of COVID-19 on the Chinese stock market: Sentimental or substantial? *Finance Research Letters*, *38*, 101838. https://doi.org/10.1016/j.frl.2020.101838

Sunardi, S., Noviolla, C., Supramono, S., & Hermanto, Y. B. (2023). Stock market reaction to government policy on determining coal selling price. *Heliyon*, *9*(2), e13454. https://doi.org/10.1016/j.heliyon.2023.e13454

Swisher, P., & Kasten, G. W. (2005). Post-Modern Portfolio Theory. *Trade Journal*, *18*(9). Retrieved from https://www.proquest.com/docview/217540934?pq-origsite=gscholar&fromopenview=true

Szyszka, A. (2007). From the Efficient Market Hypothesis to Behavioral Finance: How investors' psychology changes the vision of financial markets. *Social Science Research Network*. https://doi.org/10.2139/ssrn.1266862

Szyszka, A. (2013). Behavioral finance and capital markets. *Palgrave Macmillan US eBooks* (Vols. 8–9). https://doi.org/10.1057/9781137366290

Tao, L., Ma, F., Zhang, X., & Zhang, Y. (2020). Economic policy uncertainty and the Chinese stock market volatility: Novel evidence. *Economic Modelling*, *87*, 24–33. https://doi.org/10.1016/j.econmod.2019.07.002

Timmermann, A., & Granger, C. W. (2004). Efficient market hypothesis and forecasting. *International Journal of Forecasting*, *20*(1), 15–27. https://doi.org/10.1016/s0169-2070(03)00012-8

Tobin, M., Brown, R., Patnaik, S., & Bloomberg. (2023, March 1). A.I. is the star of earnings calls as mentions skyrocket 77% with companies saying they'll use for everything from medicine to cybersecurity. Retrieved from https://fortune.com/2023/03/01/a-i-earnings-calls-mentions-skyrocket-companies-say-search-cybersecurity-medicine-customer-service/

Vallance, B. C. (2022, December 7). ChatGPT: New AI chatbot has everyone talking to it. *BBC News*. Retrieved from https://www.bbc.com

Vanguard Asset Management | Personal Investing in the UK. (n.d.). Retrieved from https://www.vanguardinvestor.co.uk/investments/vanguard-ftse-all-world-ucits-etf-usd-distributing/overview

Vanian, J. (2022, December 13). Why tech insiders are so excited about ChatGPT, a chatbot that answers questions and writes essays. *CNBC*. Retrieved from https://www.cnbc.com

Verizon completes Yahoo acquisition, creating a diverse house of 50+ brands under new Oath subsidiary. (2017, July 18). Retrieved from https://www.verizon.com/about/news/verizon-completes-yahoo-acquisition-creating-diverse-house-50-brands-under-new-oath-subsidiary

Yahoo Finance. (2023). Retrieved May 1, 2023, from https://finance.yahoo.com/

Zhang, Y., He, J., He, M., & Li, S. (2023). Geopolitical risk and stock market volatility: A global perspective. *Finance Research Letters*, *53*, 103620. https://doi.org/10.1016/j.frl.2022.103620

8 Appendix – R-Code

```
library(quantmod)
start_date <- "2022-05-11"
end_date <- "2022-12-30"
ranges <- data.frame (x = c(1,3,5,10,10,10),y = c(1,3,5,10,15,20))

#1. Getting and sorting the Data ----
# Creating a function that creates a data set with all the factors and all the daily returns of the companies
getting_data <- function(start_date,end_date){

  # Calling the ff_data.csv and assigning the tickers
  ff_data <- read.table("ff_data.csv",header=TRUE, sep=",")
  tickers <- c("MSFT", "GOOGL", "VZ", "BIDU")

  # Creating a returns table
  returns_table <- data.frame(Date = character(), stringsAsFactors = FALSE)
  for (ticker in tickers) {
    stock_data <- getSymbols(ticker, from = start_date, to = end_date, auto.assign = FALSE)
    daily_returns <- dailyReturn(stock_data, type = "log")
    colnames(daily_returns) <- ticker
    if (nrow(returns_table) == 0) {
      returns_table <- daily_returns
    } else {
      returns_table <- merge(returns_table, daily_returns, all = TRUE)
    }
  }
  # Combining Factor data and stock data in to one data frame
  combined_df <-cbind(ff_data, returns_table)
  return(combined_df)
}

# Creating a data frame that includes all the factor and daily return data for the given start and end data.
data <- getting_data(start_date,end_date)

# Making a function in order to split the data, train data
splitting_data_train <- function(ff_data,e, g, x, y){
  N <- 141
  start <- N - x - g -e
  end <- N - x - g
  train_data <- ff_data[start:end, ]
  return(train_data)
}

# Making a function in order to split the data, test data
splitting_data_test <- function(ff_data, e, g, x , y){
  N <- 141
  start <- N - x
  end <- N + y
```


WAGENINGEN
UNIVERSITY & RESEARCH

```
  test_data <- ff_data[start:end, ]
  return(test_data)
}

# Making a function in order to create data sets for the different event windows
Loop_data <- function(ff_data, ranges) {
  train_tables <- list()
  test_tables <- list()

  for (i in seq_len(nrow(ranges))) {
    x_val <- ranges$x[i]
    y_val <- ranges$y[i]

    train_data <- splitting_data_train(ff_data, 120, 10, x_val, y_val)
    test_data <- splitting_data_test(ff_data, 120, 10, x_val, y_val)

    train_tables[[i]] <- train_data
    test_tables[[i]] <- test_data
  }

  names(train_tables) <- paste("TrainTable", seq_len(nrow(ranges)), sep = "_")
  names(test_tables) <- paste("TestTable", seq_len(nrow(ranges)), sep = "_")

  return(list(train_tables = train_tables, test_tables = test_tables))
}

# Call the function
result <- Loop_data(data, ranges)

# Access the train and test tables
train_tables <- result$train_tables
test_tables <- result$test_tables

# Extracting the data from the train_tables
for (i in seq_along(train_tables)) {
  table <- train_tables[[i]]  #
  table_name <- paste0("train_table", i)
  assign(table_name, table)
}

# Extracting the data from the test_tables
for (i in seq_along(test_tables)) {
  table <- test_tables[[i]]  #
  table_name <- paste0("test_table", i)
  assign(table_name, table)
}

#2.1 MSFT 5 Factor lines ----
# Function in order to assign the right factors for the Train data set; MSFT, 5 Factor
fitting_train_factors_MSFT_5F <- function(train_table, test_table){
  train_regression <- lm(MSFT - RF ~ Mkt.RF + SMB + HML + RMW + CMA, train_table)
```

```
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[8]
  print(summary(train_regression_5F))
  p <- predict(train_regression, test_table)
  a <- test_table$MSFT
  sum_p <- sum(p)
  sum_a <- sum(test_table$MSFT)
  ap <-cbind(a, p)

  return(ap)
}
# Function in order to assign the right factors for the Train data set; MSFT, 4 Factor
fitting_train_factors_MSFT_4F <- function(train_table, test_table){
  train_regression <- lm(MSFT - RF ~ Mkt.RF + SMB + HML + RMW, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[8]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  a <- test_table$MSFT
  sum_p <- sum(p)
  sum_a <- sum(test_table$MSFT)
  ap <-cbind(a, p)

  return(ap)
}
# Function in order to assign the right factors for the Train data set; MSFT, 3 Factor
fitting_train_factors_MSFT_3F <- function(train_table, test_table){
  train_regression <- lm(MSFT - RF ~ Mkt.RF + SMB + HML, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[8]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  a <- test_table$MSFT
  sum_p <- sum(p)
  sum_a <- sum(test_table$MSFT)
  ap <-cbind(a, p)

  return(ap)
}
# Function in order to assign the right factors for the Train data set; MSFT, CAPM
fitting_train_factors_MSFT_F <- function(train_table, test_table){
  train_regression <- lm(MSFT - RF ~ Mkt.RF, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[8]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  a <- test_table$MSFT
  sum_p <- sum(p)
  sum_a <- sum(test_table$MSFT)
  ap <-cbind(a, p)
```

```
  return(ap)
}

# Function to calculate AR, CAR, t-value and P-value for 5 Factor
MSFT_data_5F <- function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_MSFT_5F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_MSFT_5F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_MSFT_5F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_MSFT_5F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_MSFT_5F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_MSFT_5F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 5 Factor model")
```

```
  print(paste("MSFT:p-value 1", pt1))
  print(paste("MSFT:p-value 2", pt2))
  print(paste("MSFT:p-value 3", pt3))
  print(paste("MSFT:p-value 4", pt4))
  print(paste("MSFT:p-value 5", pt5))
  print(paste("MSFT:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for 4 Factor
MSFT_data_4F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_MSFT_4F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_MSFT_4F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_MSFT_4F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_MSFT_4F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_MSFT_4F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_MSFT_4F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
```

```
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 4 Factor model")
  print(paste("MSFT:p-value 1", pt1))
  print(paste("MSFT:p-value 2", pt2))
  print(paste("MSFT:p-value 3", pt3))
  print(paste("MSFT:p-value 4", pt4))
  print(paste("MSFT:p-value 5", pt5))
  print(paste("MSFT:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for 3 Factor
MSFT_data_3F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_MSFT_3F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_MSFT_3F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_MSFT_3F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_MSFT_3F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_MSFT_3F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)
```

```
  R6 <- data.frame(fitting_train_factors_MSFT_3F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 3 Factor model")
  print(paste("MSFT:p-value 1", pt1))
  print(paste("MSFT:p-value 2", pt2))
  print(paste("MSFT:p-value 3", pt3))
  print(paste("MSFT:p-value 4", pt4))
  print(paste("MSFT:p-value 5", pt5))
  print(paste("MSFT:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for CAPM
MSFT_data_F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_MSFT_F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_MSFT_F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_MSFT_F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_MSFT_F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_MSFT_F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
```

```
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_MSFT_F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for Factor model")
  print(paste("MSFT:t-value 1", pt1))
  print(paste("MSFT:t-value 2", pt2))
  print(paste("MSFT:t-value 3", pt3))
  print(paste("MSFT:t-value 4", pt4))
  print(paste("MSFT:t-value 5", pt5))
  print(paste("MSFT:t-value 6", pt6))
}

# Calling all four functions to received the P-values
MSFT_div_5f <-
MSFT_data_5F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
MSFT_div_4f <-
MSFT_data_4F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
MSFT_div_3f <-
MSFT_data_3F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
MSFT_div_F <-
MSFT_data_F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)

#2.2 GOOGL 5 Factor lines ----

# Function in order to assign the right factors for the Train data set; GOOGL, 5 factor
fitting_train_factors_GOOGL_5F <- function(train_table, test_table){
  train_regression <- lm(GOOGL - RF ~ Mkt.RF + SMB + HML + RMW + CMA, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[9]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$GOOGL
  ap <-cbind(a, p)

  return(ap)
}
# Function in order to assign the right factors for the Train data set; GOOGL, 4 factor
```

```
fitting_train_factors_GOOGL_4F <- function(train_table, test_table){
  train_regression <- lm(GOOGL - RF ~ Mkt.RF + SMB + HML + RMW, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[9]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$GOOGL
  ap <-cbind(a, p)

  return(ap)
}
# Function in order to assign the right factors for the Train data set; GOOGL, 3 factor
fitting_train_factors_GOOGL_3F <- function(train_table, test_table){
  train_regression <- lm(GOOGL - RF ~ Mkt.RF + SMB + HML, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[9]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$GOOGL
  ap <-cbind(a, p)

  return(ap)
}
# Function in order to assign the right factors for the Train data set; GOOGL, CAPM
fitting_train_factors_GOOGL_F <- function(train_table, test_table){
  train_regression <- lm(GOOGL - RF ~ Mkt.RF, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[9]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$GOOGL
  ap <-cbind(a, p)

  return(ap)
}

# Function to calculate AR, CAR, t-value and P-value for 5 Factor
GOOGL_data_5F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_GOOGL_5F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)
```


WAGENINGEN UNIVERSITY & RESEARCH

```
  R2 <- data.frame(fitting_train_factors_GOOGL_5F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_GOOGL_5F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_GOOGL_5F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_GOOGL_5F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_GOOGL_5F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 5 Factor model")

  print(paste("GOOGL:p-value 1", pt1))
  print(paste("GOOGL:p-value 2", pt2))
  print(paste("GOOGL:p-value 3", pt3))
  print(paste("GOOGL:p-value 4", pt4))
  print(paste("GOOGL:p-value 5", pt5))
  print(paste("GOOGL:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for 4 Factor
GOOGL_data_4F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_GOOGL_4F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
```

```
 var_ar <- var(R1$a_p)
 CAR <- sum(R1$a_p)
 t1 <- CAR/sqrt(var_ar*3)
 pt1 <- pt(t1,2, lower.tail = FALSE)

 R2 <- data.frame(fitting_train_factors_GOOGL_4F(train_table2, test_table2))
 R2$a_p <- (R2$a- R2$p)
 var_ar <- var(R2$a_p)
 CAR <- sum(R2$a_p)
 t2 <- CAR/sqrt(var_ar*7)
 pt2 <- pt(t2,6, lower.tail = FALSE)

 R3 <- data.frame(fitting_train_factors_GOOGL_4F(train_table3, test_table3))
 R3$a_p <- (R3$a- R3$p)
 var_ar <- var(R3$a_p)
 CAR <- sum(R3$a_p)
 t3 <- CAR/sqrt(var_ar*11)
 pt3 <- pt(t3,10, lower.tail = FALSE)

 R4 <- data.frame(fitting_train_factors_GOOGL_4F(train_table4, test_table4))
 R4$a_p <- (R4$a- R4$p)
 var_ar <- var(R4$a_p)
 CAR <- sum(R4$a_p)
 t4 <- CAR/sqrt(var_ar*21)
 pt4 <- pt(t4,20, lower.tail = FALSE)

 R5 <- data.frame(fitting_train_factors_GOOGL_4F(train_table5, test_table5))
 R5$a_p <- (R5$a- R5$p)
 var_ar <- var(R5$a_p)
 CAR <- sum(R5$a_p)
 t5 <- CAR/sqrt(var_ar*26)
 pt5 <- pt(t5,25, lower.tail = FALSE)

 R6 <- data.frame(fitting_train_factors_GOOGL_4F(train_table6, test_table6))
 R6$a_p <- (R6$a- R6$p)
 var_ar <- var(R6$a_p)
 CAR <- sum(R6$a_p)
 t6 <- CAR/sqrt(var_ar*31)
 pt6 <- pt(t6,30, lower.tail = FALSE)

 print("These are the values for 4 Factor model")
 print(paste("GOOGL:p-value 1", pt1))
 print(paste("GOOGL:p-value 2", pt2))
 print(paste("GOOGL:p-value 3", pt3))
 print(paste("GOOGL:p-value 4", pt4))
 print(paste("GOOGL:p-value 5", pt5))
 print(paste("GOOGL:p-value 6", pt6))

}
# Function to calculate AR, CAR, t-value and P-value for 3 Factor
```

```
GOOGL_data_3F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_GOOGL_3F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_GOOGL_3F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_GOOGL_3F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_GOOGL_3F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_GOOGL_3F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_GOOGL_3F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 3 Factor model")
  print(paste("GOOGL:p-value 1", pt1))
  print(paste("GOOGL:p-value 2", pt2))
  print(paste("GOOGL:p-value 3", pt3))
  print(paste("GOOGL:p-value 4", pt4))
```

```
  print(paste("GOOGL:p-value 5", pt5))
  print(paste("GOOGL:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for CAPM
GOOGL_data_F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_GOOGL_F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_GOOGL_F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_GOOGL_F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_GOOGL_F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_GOOGL_F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_GOOGL_F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for Factor model")
```

```
  print(paste("GOOGL:p-value 1", pt1))
  print(paste("GOOGL:p-value 2", pt2))
  print(paste("GOOGL:p-value 3", pt3))
  print(paste("GOOGL:p-value 4", pt4))
  print(paste("GOOGL:p-value 5", pt5))
  print(paste("GOOGL:p-value 6", pt6))
}

# Calling all four functions to received the P-values
GOOGL_div_5F <-
GOOGL_data_5F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
GOOGL_div_4f <-
GOOGL_data_4F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
GOOGL_div_3F <-
GOOGL_data_3F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
GOOGL_div_F <-
GOOGL_data_F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)

#2.3 VZ 5 Factor lines ----
# Function in order to assign the right factors for the Train data set; VZ, 5 factor
fitting_train_factors_VZ_5F <- function(train_table, test_table){
  train_regression <- lm(VZ - RF ~ Mkt.RF + SMB + HML + RMW + CMA, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[10]
  print(summary(train_regression))
  p <- predict(train_regression, test_table)
  sum_p <- sum(p)
  print(sum_p)
  a <- test_table$VZ
  ap <-cbind(a, p)

  return(ap)
}
# Function in order to assign the right factors for the Train data set; VZ, 4 factor
fitting_train_factors_VZ_4F <- function(train_table, test_table){
  train_regression <- lm(VZ - RF ~ Mkt.RF + SMB + HML + RMW, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[10]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$VZ
  ap <-cbind(a, p)
  print(sum_p)
  return(ap)
}
# Function in order to assign the right factors for the Train data set; VZ, 3 factor
```

```
fitting_train_factors_VZ_3F <- function(train_table, test_table){
  train_regression <- lm(VZ - RF ~ Mkt.RF + SMB + HML, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[10]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$VZ
  ap <-cbind(a, p)
  print(sum_p)
  return(ap)
}
# Function in order to assign the right factors for the Train data set; VZ, CAPM
fitting_train_factors_VZ_F <- function(train_table, test_table){
  train_regression <- lm(VZ - RF ~ Mkt.RF, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[10]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$VZ
  ap <-cbind(a, p)
  print(sum_p)
  return(ap)
}

# Function to calculate AR, CAR, t-value and P-value for 5 Factor
VZ_data_5F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_VZ_5F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_VZ_5F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_VZ_5F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)
```

```
  R4 <- data.frame(fitting_train_factors_VZ_5F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_VZ_5F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_VZ_5F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 5 Factor model")

  print(paste("VZ:p-value 1", pt1))
  print(paste("VZ:p-value 2", pt2))
  print(paste("VZ:p-value 3", pt3))
  print(paste("VZ:p-value 4", pt4))
  print(paste("VZ:p-value 5", pt5))
  print(paste("VZ:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for 4 Factor
VZ_data_4F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_VZ_4F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_VZ_4F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_VZ_4F(train_table3, test_table3))
```

```
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_VZ_4F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_VZ_4F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_VZ_4F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 4 Factor model")
  print(paste("VZ:p-value 1", pt1))
  print(paste("VZ:p-value 2", pt2))
  print(paste("VZ:p-value 3", pt3))
  print(paste("VZ:p-value 4", pt4))
  print(paste("VZ:p-value 5", pt5))
  print(paste("VZ:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for 3 Factor
VZ_data_3F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_VZ_3F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_VZ_3F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
```

```
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_VZ_3F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_VZ_3F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_VZ_3F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_VZ_3F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 3 Factor model")
  print(paste("VZ:p-value 1", pt1))
  print(paste("VZ:p-value 2", pt2))
  print(paste("VZ:p-value 3", pt3))
  print(paste("VZ:p-value 4", pt4))
  print(paste("VZ:p-value 5", pt5))
  print(paste("VZ:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for CAPM
VZ_data_F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_VZ_F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)
```

```
 R2 <- data.frame(fitting_train_factors_VZ_F(train_table2, test_table2))
 R2$a_p <- (R2$a- R2$p)
 var_ar <- var(R2$a_p)
 CAR <- sum(R2$a_p)
 t2 <- CAR/sqrt(var_ar*7)
 pt2 <- pt(t2,6, lower.tail = FALSE)

 R3 <- data.frame(fitting_train_factors_VZ_F(train_table3, test_table3))
 R3$a_p <- (R3$a- R3$p)
 var_ar <- var(R3$a_p)
 CAR <- sum(R3$a_p)
 t3 <- CAR/sqrt(var_ar*11)
 pt3 <- pt(t3,10, lower.tail = FALSE)

 R4 <- data.frame(fitting_train_factors_VZ_F(train_table4, test_table4))
 R4$a_p <- (R4$a- R4$p)
 var_ar <- var(R4$a_p)
 CAR <- sum(R4$a_p)
 t4 <- CAR/sqrt(var_ar*21)
 pt4 <- pt(t4,20, lower.tail = FALSE)

 R5 <- data.frame(fitting_train_factors_VZ_F(train_table5, test_table5))
 R5$a_p <- (R5$a- R5$p)
 var_ar <- var(R5$a_p)
 CAR <- sum(R5$a_p)
 t5 <- CAR/sqrt(var_ar*26)
 pt5 <- pt(t5,25, lower.tail = FALSE)

 R6 <- data.frame(fitting_train_factors_VZ_F(train_table6, test_table6))
 R6$a_p <- (R6$a- R6$p)
 var_ar <- var(R6$a_p)
 CAR <- sum(R6$a_p)
 t6 <- CAR/sqrt(var_ar*31)
 pt6 <- pt(t6,30, lower.tail = FALSE)

 print("These are the values for Factor model")
 print(paste("VZ:t-value 1", pt1))
 print(paste("VZ:t-value 2", pt2))
 print(paste("VZ:t-value 3", pt3))
 print(paste("VZ:t-value 4", pt4))
 print(paste("VZ:t-value 5", pt5))
 print(paste("VZ:t-value 6", pt6))
}

# Calling all four functions to received the P-values
VZ_div_5f <-
VZ_data_5F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,te
st_table2,test_table3,test_table4,test_table5,test_table6)
VZ_div_4f <-
VZ_data_4F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,te
st_table2,test_table3,test_table4,test_table5,test_table6)
```

```
VZ_div_3f <-
VZ_data_3F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
VZ_div_F <-
VZ_data_F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)


#2.4 BIDU 5 Factor lines ----
# Function in order to assign the right factors for the Train data set; BIDU, 5 factor
fitting_train_factors_BIDU_5F <- function(train_table, test_table){
  train_regression <- lm(BIDU - RF ~ Mkt.RF + SMB + HML + RMW + CMA, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[11]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  print(sum_p)
  a <- test_table$BIDU
  ap <-cbind(a, p)

  return(ap)
}
# Function in order to assign the right factors for the Train data set; BIDU, 4 factor
fitting_train_factors_BIDU_4F <- function(train_table, test_table){
  train_regression <- lm(BIDU - RF ~ Mkt.RF + SMB + HML + RMW, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[11]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$BIDU
  ap <-cbind(a, p)
  print(sum_p)


  return(ap)
}
# Function in order to assign the right factors for the Train data set; BIDU, 3 factor
fitting_train_factors_BIDU_3F <- function(train_table, test_table){
  train_regression <- lm(BIDU - RF ~ Mkt.RF + SMB + HML, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[11]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$BIDU
  ap <-cbind(a, p)
  print(sum_p)

  return(ap)
```

```
}
# Function in order to assign the right factors for the Train data set; BIDU, CAPM
fitting_train_factors_BIDU_F <- function(train_table, test_table){
  train_regression <- lm(BIDU - RF ~ Mkt.RF, train_table)
  table_name <- deparse(substitute(train_table))
  column_name <-names(train_table)[11]
  p <- predict(train_regression, test_table)
  print(summary(train_regression))
  sum_p <- sum(p)
  a <- test_table$BIDU
  ap <-cbind(a, p)
  print(sum_p)

  return(ap)
}

# Function to calculate AR, CAR, t-value and P-value for 5 Factor
BIDU_data_5F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_BIDU_5F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_BIDU_5F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_BIDU_5F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_BIDU_5F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_BIDU_5F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
```

```
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_BIDU_5F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 5 Factor model")

  print(paste("BIDU:p-value 1", pt1))
  print(paste("BIDU:p-value 2", pt2))
  print(paste("BIDU:p-value 3", pt3))
  print(paste("BIDU:p-value 4", pt4))
  print(paste("BIDU:p-value 5", pt5))
  print(paste("BIDU:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for 4 Factor
BIDU_data_4F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_BIDU_4F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_BIDU_4F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_BIDU_4F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_BIDU_4F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
```

```
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_BIDU_4F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_BIDU_4F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for 4 Factor model")
  print(paste("BIDU:p-value 1", pt1))
  print(paste("BIDU:p-value 2", pt2))
  print(paste("BIDU:p-value 3", pt3))
  print(paste("BIDU:p-value 4", pt4))
  print(paste("BIDU:p-value 5", pt5))
  print(paste("BIDU:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for 3 Factor
BIDU_data_3F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

  R1 <- data.frame(fitting_train_factors_BIDU_3F(train_table1, test_table1))
  R1$a_p <- (R1$a- R1$p)
  var_ar <- var(R1$a_p)
  CAR <- sum(R1$a_p)
  t1 <- CAR/sqrt(var_ar*3)
  pt1 <- pt(t1,2, lower.tail = FALSE)

  R2 <- data.frame(fitting_train_factors_BIDU_3F(train_table2, test_table2))
  R2$a_p <- (R2$a- R2$p)
  var_ar <- var(R2$a_p)
  CAR <- sum(R2$a_p)
  t2 <- CAR/sqrt(var_ar*7)
  pt2 <- pt(t2,6, lower.tail = FALSE)

  R3 <- data.frame(fitting_train_factors_BIDU_3F(train_table3, test_table3))
  R3$a_p <- (R3$a- R3$p)
  var_ar <- var(R3$a_p)
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)
```

```
 R4 <- data.frame(fitting_train_factors_BIDU_3F(train_table4, test_table4))
 R4$a_p <- (R4$a- R4$p)
 var_ar <- var(R4$a_p)
 CAR <- sum(R4$a_p)
 t4 <- CAR/sqrt(var_ar*21)
 pt4 <- pt(t4,20, lower.tail = FALSE)

 R5 <- data.frame(fitting_train_factors_BIDU_3F(train_table5, test_table5))
 R5$a_p <- (R5$a- R5$p)
 var_ar <- var(R5$a_p)
 CAR <- sum(R5$a_p)
 t5 <- CAR/sqrt(var_ar*26)
 pt5 <- pt(t5,25, lower.tail = FALSE)

 R6 <- data.frame(fitting_train_factors_BIDU_3F(train_table6, test_table6))
 R6$a_p <- (R6$a- R6$p)
 var_ar <- var(R6$a_p)
 CAR <- sum(R6$a_p)
 t6 <- CAR/sqrt(var_ar*31)
 pt6 <- pt(t6,30, lower.tail = FALSE)

 print("These are the values for 3 Factor model")
 print(paste("BIDU:p-value 1", pt1))
 print(paste("BIDU:p-value 2", pt2))
 print(paste("BIDU:p-value 3", pt3))
 print(paste("BIDU:p-value 4", pt4))
 print(paste("BIDU:p-value 5", pt5))
 print(paste("BIDU:p-value 6", pt6))
}
# Function to calculate AR, CAR, t-value and P-value for CAPM
BIDU_data_F <-
function(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6){

 R1 <- data.frame(fitting_train_factors_BIDU_F(train_table1, test_table1))
 R1$a_p <- (R1$a- R1$p)
 var_ar <- var(R1$a_p)
 CAR <- sum(R1$a_p)
 t1 <- CAR/sqrt(var_ar*3)
 pt1 <- pt(t1,2, lower.tail = FALSE)

 R2 <- data.frame(fitting_train_factors_BIDU_F(train_table2, test_table2))
 R2$a_p <- (R2$a- R2$p)
 var_ar <- var(R2$a_p)
 CAR <- sum(R2$a_p)
 t2 <- CAR/sqrt(var_ar*7)
 pt2 <- pt(t2,6, lower.tail = FALSE)

 R3 <- data.frame(fitting_train_factors_BIDU_F(train_table3, test_table3))
 R3$a_p <- (R3$a- R3$p)
 var_ar <- var(R3$a_p)
```

```
  CAR <- sum(R3$a_p)
  t3 <- CAR/sqrt(var_ar*11)
  pt3 <- pt(t3,10, lower.tail = FALSE)

  R4 <- data.frame(fitting_train_factors_BIDU_F(train_table4, test_table4))
  R4$a_p <- (R4$a- R4$p)
  var_ar <- var(R4$a_p)
  CAR <- sum(R4$a_p)
  t4 <- CAR/sqrt(var_ar*21)
  pt4 <- pt(t4,20, lower.tail = FALSE)

  R5 <- data.frame(fitting_train_factors_BIDU_F(train_table5, test_table5))
  R5$a_p <- (R5$a- R5$p)
  var_ar <- var(R5$a_p)
  CAR <- sum(R5$a_p)
  t5 <- CAR/sqrt(var_ar*26)
  pt5 <- pt(t5,25, lower.tail = FALSE)

  R6 <- data.frame(fitting_train_factors_BIDU_F(train_table6, test_table6))
  R6$a_p <- (R6$a- R6$p)
  var_ar <- var(R6$a_p)
  CAR <- sum(R6$a_p)
  t6 <- CAR/sqrt(var_ar*31)
  pt6 <- pt(t6,30, lower.tail = FALSE)

  print("These are the values for Factor model")
  print(paste("BIDU:p-value 1", pt1))
  print(paste("BIDU:p-value 2", pt2))
  print(paste("BIDU:p-value 3", pt3))
  print(paste("BIDU:p-value 4", pt4))
  print(paste("BIDU:p-value 5", pt5))
  print(paste("BIDU:p-value 6", pt6))
}

# Calling all four functions to received the P-values
BIDU_div_5f <-
BIDU_data_5F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
BIDU_div_4f <-
BIDU_data_4F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
BIDU_div_3f <-
BIDU_data_3F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
BIDU_div_F <-
BIDU_data_F(train_table1,train_table2,train_table3,train_table4,train_table5,train_table6,test_table1,test_table2,test_table3,test_table4,test_table5,test_table6)
```