ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics
Bachelor Thesis Economics and Business Economics

On the relationship between (EGARCH) Idiosyncratic Risk and Expected Returns and the effects of the COVID-19 pandemic

Evidence from European Stock Markets

Name student: Lodewijk Sweerts de Landas
Student ID number: 470043

Supervisor: Giovanni Cocco
Second assessor: Esad Smajlbegovic

Date final version: 09-07-2022

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Abstract

This thesis examines the relationship between idiosyncratic risk and expected stock returns from stocks listed on the STOXX Europe 600 index in the period of January 2014 until December 2021. Previous literature has shown that both negative and positive relationships have been found between idiosyncratic risk and expected stock returns in the cross section. Initially a negative relationship was established. However, when applying a before and during the COVID-19 pandemic split in the sample, to explore the effects of this pandemic on the relationship, the coefficient for idiosyncratic risk switches from significant negative to insignificant positive during the pandemic. Furthermore, this thesis corroborates the findings of Guo et al. (2014) by showing that EGARCH models, with Fama French three factors as the mean process, to estimate the out-of-sample expected idiosyncratic risk, do not have any explanatory power for expected stock returns in the cross section.

Keywords: idiosyncratic risk, EGARCH, cross-sectional returns, STOXX 600 Europe, Fama-French three factors

Contents

List of Tables

List of Figures

1 Introduction

According to the Capital Asset Pricing Model, CAPM (Sharpe, 1964), investors keep the efficient mean–variance market portfolio in equilibrium. This model predicts a linear relationship between expected returns and systematic risk, but idiosyncratic risk plays no role. CAPM provides no prediction on the role of idiosyncratic risk in under-diversified investor scenarios.

Diverse hypotheses, assumed by investors under diversification of their portfolios, led to the conclusion that the idiosyncratic risk of stocks is positively correlated with the expected returns of stocks in the cross section. Merton (1987) was among them. He expanded the CAPM and proposed a model that implied under-diversification among investors. He suggested that if investors are not able or unwilling to retain the market portfolio, they would consider overall risk in their asset allocation decisions and demand greater returns for stocks with more idiosyncratic risk to compensate for poor diversification. Ultimately, Merton (1987) discovered that idiosyncratic risk was positively correlated with expected stock returns.

Campbell et al. (2001) and Kearney & Potì (2008) demonstrated that the idiosyncratic risk of stocks has grown in recent years, meaning that modern portfolios require more stocks to diversify away from idiosyncratic risk. Nonetheless, according to Goetzmann & Kumar (2008), the majority of U.S. individual investors, in their sample from 1991 to 1996, held under-diversified portfolios. Goetzmann & Kumar (2008) indicated that 25% of investor portfolios comprised only one stock, while less than 10% of investor portfolios had ten or more stocks. It remains an empirical problem to accurately predict idiosyncratic risk, hence making it more difficult for modern investors to be completely diversified.

Various approaches to quantify expected idiosyncratic risk and to examine its empirical relationship with expected returns yielded contradictory results. Ranging from negative to positive to having no correlation. Fama & French (1993) employed cross-sectional regressions on a monthly basis and rejected the importance of idiosyncratic risk in the cross-section of returns. Malkiel & Xu (2002) extended the data of Fama & MacBeth (1973) and found that idiosyncratic risk is priced positively.

Ang et al. (2006) found that the relationship between monthly stock returns and the one-month lagged idiosyncratic volatilities is negative. These idiosyncratic volatilities were used in their research as proxies for idiosyncratic risk of the specific stocks. Fu (2009) on the other hand, did not agree that the one-month lagged idiosyncratic volatilities are good proxies for the expected value as they are time-varying. He pointed out that the relationship between idiosyncratic risk and expected returns should be contemporary. Fu (2009) showed that idiosyncratic risk was time-varying. He stated that models using lagged idiosyncratic risk as a proxy for its expected value are wrong. Instead, Fu (2009) used exponential GARCH (EGARCH) models for estimating the expected idiosyncratic volatilities and found a positive relation with the expected stock returns.

Due to the look-ahead bias it causes, Guo et al. (2014) criticized EGARCH models using in-sample estimations (particularly Fu, 2009). The maximum likelihood approach is a typical strategy for estimating an EGARCH model. The resultant log likelihood function will incorporate contemporaneous returns even if the model technical specification only uses lagged idiosyncratic volatility and historical stock returns. By establishing a "truly" out-of-sample idiosyncratic volatility, they were able to demonstrate that there is no significant positive association between anticipated idiosyncratic risk and stock returns. But it is crucial to stress that neither the findings of Ang et al. (2006, 2009) nor the common asset pricing models are supported by the results. It is possible that the relationship is in fact positive, however measuring idiosyncratic risk with EGARCH models, as undertaken by Fu (2009), produce a bias that renders the conclusions untrustworthy.

Further in time, the research of Poudeh et al. (2022) examined the influence of the COVID-19 pandemic on the relationship between idiosyncratic risk and expected stock returns at the portfolio and company levels. Portfolio analysis and cross-sectional regression findings revealed that throughout the pandemic era, the relationship between idiosyncratic risk and subsequent stock returns shifted in a negative to a positive direction.

With various conclusions regarding idiosyncratic volatility and its relationship to expected stock returns, opinions on the use of EGARCH models and their explanatory power for expected stock returns, and findings pertaining to the COVID-19 pandemic, this thesis tries to answer the following research questions:

1. How does idiosyncratic risk relate to expected stock returns in the European stock market?
2. How does idiosyncratic risk relate to expected stock returns in the European stock market before and during the COVID-19 pandemic?
3. Does EGARCH with out-of-sample estimated idiosyncratic risk has explanatory power for expected stock returns in the European stock market?
4. Does a zero-investment portfolio alternating between short and long positions with low and high estimated EGARCH idiosyncratic risk provide a positive or negative return?

When taking into account that many individual investors still have under-diversified portfolios which therefore contain idiosyncratic risk, the primary focus of this thesis is to ascertain whether investors are rewarded when they bear idiosyncratic risk in their portfolios. The results discovered in this thesis thereby can be relevant for multiple reasons. First, the results support earlier studies on idiosyncratic risk that have been conducted using a variety of samples from developed and emerging markets throughout the world. Also it supports literature that criticises the use of EGARCH in-sample estimations and proposes out-of-sample estimations. Second, empirical research on the empirical relationship between conditional idiosyncratic volatility and predicted stock returns using a European market sample is scarce to nonexistent, making this theory a research frontier for this specific topic in Europe. Being the research frontier, these findings thereby can also be used to stimulate further research on the potential relationship with a larger European stock sample. Finally, the results give evidence that investors on European stock markets are not rewarded for bearing idiosyncratic risk, in contrast to investors on other stock markets.

This thesis proceeds as follows: In Section 2, a literature review addresses research of idiosyncratic volatility, the effects of COVID-19 pandemic on the relationship and the use of EGARCH models followed by the development of my hypothesis. Section 3 describes data and methodology. All empirical results are presented in Section 4. Section 5 addresses robustness checks. Section 6 provides a conclusion.

2 Literature Review

2.1 Idiosyncratic risk and expected returns

Modern portfolio theory suggests that investors should maintain a portfolio of stocks in order to diversify their exposure to idiosyncratic risk. Based on this portfolio theory, the capital asset pricing model, CAPM (Sharpe, 1964) predicted that all investors will maintain a balanced market portfolio. Based on CAPM's premise of perfect indivisibility of an investment and the lack of transaction costs in the stock markets, Levy (1978) derived a theoretical conclusion stating that each investor possesses all market-available securities in his portfolio. He discovered that this notion was inconsistent with reality.

Another now well-known theory was discovered by Merton (1987). His fundamental assumption was that acquiring information about stocks was costly, making it neither desirable nor realistic for investors to monitor every security on the market. Consequently, investors maintain under-diversified portfolios and seek compensation for the peculiar idiosyncratic risk of their securities. In the cross section, Merton (1987) predicted a positive relationship between idiosyncratic risk and expected stock returns.

Malkiel & Xu (2002) discovered that idiosyncratic risk is more influential than beta or other size metrics in explaining the cross section of returns. In addition, they discovered a positive relationship between idiosyncratic risk and expected returns in the cross section.

Levy (1978), Merton (1987) and Malkiel & Xu (2002) showed that under-diversified investors demanded compensation for the idiosyncratic risk of the securities in their portfolios. At that time, all of their studies indicated that the idiosyncratic risk was positively correlated with the predicted returns of securities in the cross section. However Ang et al. (2006) made an end to this assumption. Their results proved the complete opposite of earlier studies described above. In their sample of US stocks from 1963 to 2006, they observed that stocks with high idiosyncratic volatility predicted the following month's relatively low returns on average. Although there was a high association between idiosyncratic risk and predicted returns in the cross section, it nevertheless showed a negative relationship. Exposure to aggregate volatility risk was insufficient to explain this pattern. Also, the influences of size, book-to-market, momentum, and liquidity were insufficient to explain this event. Ang et al. (2006) called this phenomenon considering the results on idiosyncratic risk "a substantive puzzle".

Ang et al. (2009) investigated whether or not this substantive problem from their 2006 study could be applied to an international sample. They analyzed information from 23 developed nations for the period 1980 and 2003. In contrast to their techniques in Ang et al. (2006), they conducted Fama-MacBeth regressions as opposed to building portfolios and discovered a negative relationship.

Not long after the studies of Ang et al. (2006, 2009), Fu (2009) was the one who showed the implications that led to the opposite results of their research. By conducting Dikecy-Fuller tests, Fu (2009) showed that idiosyncratic volatility, that was used in the research of Ang et al. (2006), was not following a random walk process. The results of the Dickey-Fuller tests indicated that the negative relationship between one-month lagged idiosyncratic volatility and average returns from Ang et al. (2006) did not imply a negative relationship between idiosyncratic risk and expected stock returns. Fu (2009) assumed that one-month lagged idiosyncratic volatility was not the most accurate representation of expected idiosyncratic risk. Consequently, he utilised exponential generalised autoregressive conditional heteroskedasticity (EGARCH) models to estimate the predicted idiosyncratic volatility, thereby capturing the time-varying aspect of idiosyncratic risk that Ang et al. (2006) neglected. Fu (2009) performed Fama-MacBeth regressions on monthly stock returns based on EGARCH estimations and other company characteristics which, proven by Fama & French (1992), are known to have descriptive power for cross-sectional returns. With this method, Fu (2009) found a significant positive relationship between the EGARCH estimated conditional idiosyncratic volatilities and expected returns for the same time period as Ang et al. (2006). Due to its potentially large contribution to the study of idiosyncratic risk, the EGARCH model technique has been frequently embraced in empirical studies[1] linked to this topic.

[1] For example: Spiegel & Wang (2005), Eiling (2006), Brockman et al. (2007),Okpara & Nwezeaku (2009), Huang et al. (2010), Bley & Saad (2012a), De Mendonça et al. (2012) and Perera & Ediriwickrama (2020)

However, according to Guo et al. (2014), when using the EGARCH idiosyncratic volatility approach, a look-ahead bias is introduced by incorporating the month t in the estimation of EGARCH model parameters. These parameters are used to construct the expected month t idiosyncratic volatility. In relatively small samples, Guo et al. (2014) indicated that this in-sample EGARCH idiosyncratic volatility can have a high dependency on the contemporaneous stock return. When the month t return is included in the estimation of EGARCH model parameters, the EGARCH idiosyncratic volatility of month t has an upward bias when the magnitude of month t is large. Given that the cross section of stock returns is positively skewed (Duffee, 1995), the positive correlation between the bias in in-sample EGARCH idiosyncratic volatility and predicted returns predominates in historical stock returns, resulting in a possible spurious predictability of cross-sectional stock returns generated by the look-ahead bias. Guo et al. (2014) replicated the results of Fu (2009) and showed that indeed the expected idiosyncratic risk is positively related to returns. However, when they substituted the in-sample EGARCH idiosyncratic risk with their own out-of-sample forecast of EGARCH idiosyncratic risk, the relationship became negative and lost its predictive potential for the cross section of stock returns.

2.2 International evidence on idiosyncratic risk and expected returns

While the majority of research on the cross-sectional relationship between idiosyncratic risk and expected stock returns has been conducted in the United States, the relationship has also been examined in other developed as well as emerging markets. For example, Okpara & Nwezeaku (2009) examined whether or not idiosyncratic risk may be priced on the Nigerian stock exchange market. They followed the research methods of Fu (2009), except that they used CAPM model with market beta to predict the variance of residuals. Their results showed that the stock's expected return is a positive and significant function of beta risk, implying that only the systematic risk is priced in the listed companies in Nigeria. Furthermore, Okpara & Nwezeaku (2009) found a negative and significant coefficient of idiosyncratic risk, estimated by EGARCH models.De Mendonça et al. (2012) also used the same research methodology as Fu (2009) and found a significant positive relationship between idiosyncratic risk, estimated by EGARCH models, and expected stock returns in the Brazilian market. Bley & Saad (2012a) investigated the pricing of idiosyncratic risk of seven frontier markets in six Gulf Cooperation Council (GCC) countries. Bley & Saad (2012a) provided evidence that stocks in the GCC with high idiosyncratic risk experienced a contemporaneous positive relationship with expected returns. They found a significant negative relationship between expected returns and one-month lagged idiosyncratic risk for only a few of these frontier markets. But when Bley & Saad (2012a) estimated idiosyncratic risk with EGARCH and AR models, the relationship turned positive.

Al Rahahleh et al. (2016) discovered a significant correlation between idiosyncratic risk and expected stock returns for Singaporean morally deficient vetted stocks. They observed that over their sample period the portfolio of stocks with the highest idiosyncratic risk generated higher average returns than the portfolio of stocks with the lowest idiosyncratic risk.

Kinnunen & Martikainen (2017) evaluated the price of idiosyncratic risk using MIDAS regression and a cross section of Russian industrial stock portfolios. In their initial sample period (before to the 2008-2009 financial crisis), they discovered that idiosyncratic risk was inversely correlated with anticipated returns. It attracted a negative annual risk premium. However, in their second sample period (after the financial crisis), they discovered a strong positive correlation.

2.3 European Stock Markets

Although there has not been a significant amount of research on the relationship between idiosyncratic risk and expected returns for the largest European stock markets, there are a few papers which explored a number of intriguing subjects relevant to this thesis. Similar to Campbell et al. (2001), Kearney & Potì (2008) observed an increase in the idiosyncratic volatility of European stocks. The findings of Kearney & Potì (2008) show that over time, European stock investors will require a greater number of stocks to diversify away idiosyncratic risk.

Furthermore, Angelidis & Tessaromatis (2008) aimed to determine whether or not idiosyncratic risk can predict stock returns for ten European stock markets. They discovered no evidence that idiosyncratic risk, whether equally or value weighted, may predict future market performance. Angelidis & Tessaromatis (2008) evaluated the relationship between idiosyncratic risk and subsequent stock returns in the 10 European markets by regressing value-weighted excess monthly market stock returns on a series of lagged volatility metrics. With equally-weighted idiosyncratic risk, they discovered a negligible relationship between market returns in practically every European market. Similar outcomes were observed with value-weighted idiosyncratic risk.

Annaert et al. (2022) contributed to the growing international evidence on the negative relationship between idiosyncratic risk and expected returns. They estimated idiosyncratic risk with the four factor model (Carhart, 1997) and found a statistically significant negative coefficient for the idiosyncratic risk premium among stocks in the European region.

2.4 The COVID-19 Pandemic

Recently a worldwide market decline occured during the start of 2020 as a result of the COVID-19 outbreak that developed into a pandemic. Poudeh et al. (2022) studied the effect of the COVID-19 pandemic on the relationship between idiosyncratic risk and expected stock returns in the US market. Using portfolio analysis and cross-sectional regressions, they discovered that the relationship between idiosyncratic risk and contemporaneous stock returns became positive (from negative) after the pandemic (during the pandemic). In addition, they discovered that the link is resilient to both the skewness of both periods. While there has not been any research on this phenomena for the European region, the results of Poudeh et al. (2022) might indicate that in Europe this could be the case as well.

2.5 Asset Pricing Models

Explaining cross-sectional stock returns is characterized by difficulty throughout the years and there is still a lot of debate on which factors should be included and which should not. The CAPM proposed that stock returns are positively correlated with the market return minus the risk-free rate. Fama & French (1993) expanded the CAPM model by adding size risk and value risk factors. These modifications enabled the model to account for the vast majority of the cross-sectional variance in average stock returns. Throughout the years, the Fama-French three factor asset pricing model has been utilised extensively in scientific research and for evaluating management performance.

Due to its simplicity and proven explanatory power, the three-factor model serves as the baseline model for estimating expected stock returns in this thesis. By utilising the three factor model, this thesis was consistent with the methods used by Ang et al. (2006, 2009) and Fu (2009) for estimating stock returns and generating a proxy for the monthly idiosyncratic risk.

2.6 Hypothesis Development

In this thesis, the main objective is to find out if investors in European stock markets are compensated for bearing idiosyncratic risk in their portfolios in the period from 2014 to the end 2021 and if the use of EGARCH models for estimating next month's has explanatory power for expected returns. I used the same methodology as Fu (2009) except that I used a different estimation approach with EGARCH models. To conduct this thesis I formulated my hypotheses, developed from my research questions and the literature reviewed. The hypotheses are as follows:

- **Hypothesis 1**: *Idiosyncratic risk and expected stock returns in the cross-section will be negatively related over total time period.*

- **Hypothesis 2**: *Idiosyncratic risk and expected stock returns in the cross-section will be negatively related before the COVID-19 pandemic.*

- **Hypothesis 3**: *Idiosyncratic risk and expected stock returns in the cross-section will be positively related during the COVID-19 pandemic.*

- **Hypothesis 4**: *Expected idiosyncratic risk, derived from out-of-sample EGARCH model estimations, and expected stock returns will have no significant relation.*

- **Hypothesis 5**: *A strategy of going short on a portfolio of low next month's expected idiosyncratic volatility and going long on a portfolio of high next month's expected idiosyncratic volatility, will yield no significant positive monthly return.*

My first hypothesis is developed based on the findings of Annaert et al. (2022). The second and third are developed on the findings of Poudeh et al. (2022) as one has to take into account that my sample period contains the period where the worldwide COVID-19 pandemic took place.

Notwithstanding the literature referenced above, the question remains what the real relationship between idiosyncratic volatility and expected returns should be. With EGARCH models, many studies attempted to forecast the idiosyncratic risk but made a common estimation mistake which accidentally introduced a look-ahead bias. My fourth hypothesis is therefore based on the evidence from Guo et al. (2014) who discovered that the findings of Fu (2009) were not reliable and should have been based on out-of-sample estimations. They criticized studies based on EGARCH in-sample estimations as these result in a look ahead bias. When accounting for out-of-sample estimation, Guo et al. (2014) found a negative coefficient for expected idiosyncratic risk with no predictive power for the same data sample as Fu (2009).

My fifth hypothesis is dependent on the fourth hypothesis. This means that if my fourth hypothesis is true, then so will be my fifth. However, I have formulated these hypotheses separately because I think it is important to analyse, not only from the company level, but also from the portfolio level.

3 Data and Methodology

3.1 Data

The data used to conduct this research is obtained via Refinitiv Eikon which contains databases from Datastream. The data comprises daily and monthly adjusted closing prices, monthly market capitalization and monthly price-to-book ratios of the companies listed on STOXX Europe 600 index during the period of January 2009 to December 2021. Adjusted closing prices are the closing prices after adjusting for corporate actions such as dividend payouts. The STOXX Europe 600 Index is derived from the STOXX Europe Total Market Index and is a subset of the STOXX Global 1800 Index (TMI). The Index includes major, medium, and small capitalisation companies from seventeen countries[2] located in the European region. Although, the Index has a fixed number of 600 companies with a yearly rebalancing, I include all stocks that were listed on STOXX Europe 600 Index during the period of January 2014 to December 2021, resulting in 829 stocks in total. From these 829, I selected stocks that are based on multiple criteria. First of all, I selected stocks of companies which traded actively for the entire period under review and not stike out. By doing this I was left with stocks that give daily adjusted closing prices for the whole period and do not contain missing values for certain months. Second, I removed stocks where market capitalisation occurred to be negative during the sample period. Third, I excluded all companies in the financial sector, taken into account the arguments from Fama & French (1992). They stated that financial companies typically experience high leverage which is normal for these kind of companies but is not normal for other non-financial companies. High leverage in non-financial companies usually indicates distress and therefore results may be distorted if financial companies are included in the sample. The application of the foregoing criteria resulted in 366 stocks that are used for this research. Furthermore, I used the Fama-French three factors that Kenneth R. French made available on his Web site[3].

3.2 Methodology

My research seeks to determine if under-diversified investors are rewarded for taking on idiosyncratic risk. Idiosyncratic risk indicates the volatile character of firm-specific information. From a theoretical standpoint, the risk and return trade-off should be contemporary. Investors gain returns during periods in which they take risk. As a natural proxy for idiosyncratic risk, if idiosyncratic volatility is priced in, it is anticipated that a positive empirical relationship between expected return and predicted idiosyncratic volatility would be found. However, neither of these can be observed. In cross-sectional regressions, it is common to utilise the realised return as the dependent variable in accordance with Fu (2009). On the right-hand side of the regressions, (expected) idiosyncratic volatility and control variables are placed.The equations for the regressions are given below.

[2] Austria, Belgium, Denmark, Finland, France, Germany, Ireland, Italy, Luxembourg, the Netherlands, Norway, Poland, Portugal, Spain, Sweden, Switzerland and the United Kingdom

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

$$R_{i,t} = \beta_0 + \beta_1 \mathrm{E}_{t-1}[IVOL_{i,t}] + \beta_2 \ln(MV)_{i,t} + \beta_3 \ln(B/M)_{i,t} + \epsilon_{i,t} \tag{1}$$

$$R_{i,t} = \gamma_0 + \gamma_1 IVOL_{i,t} + \gamma_2 \ln(MV)_{i,t} + \gamma_3 \ln(B/M)_{i,t} + \epsilon_{i,t} \tag{2}$$

$$R_{i,t} = \ln\left(\frac{P_{i,t}}{P_{i,t-1}}\right) \tag{3}$$

The dependent variable in equations (1) and (2), is the realized return for a stock i in period t. Returns are calculated with the natural logarithm difference as in equation (3) because natural logarithm returns have a smaller skewness than simple returns. P_t and P_{t-1} are the monthly adjusted closing prices of a stock. The difference between the two equations is that in the first equation the expected idiosyncratic volatility is used while in the second equation the actual idiosyncratic volatility is used. In equation (1), $\mathrm{E}_{t-1}[.]$ stands for the function of expectation conditional on all information until t-1. $\mathrm{E}_{t-1}[IVOL_{i,t}]$ thus stands for the expected idiosyncratic volatility for stock i at time t conditional on all information until t-1. In equation (2), $IVOL_{i,t}$ thus stands for the actual idiosyncratic volatility for stock i at time t. For the control variables, I have chosen company size (market capitalization), $\ln(MV)$, and book-to-market ratio, $\ln(BM)$, based on the paper of Fama and French (1992).

3.2.1 Estimation of idiosyncratic volatility

By definition, idiosyncratic risk is unrelated to market fluctuations. Following the methodology of Ang et al. (2006, 2009), I estimated the specific stock's idiosyncratic risk as follows. Daily excess returns of the individual stock were regressed on the daily Fama-French three factors in every month of every year:

1. (R_m - r_f), the excess return on a broad market portfolio.
2. *SMB* (small minus big), the difference between returns of portfolios existing of small and portfolios existing of big stocks
3. *HML* (high minus low), the difference between returns of portfolios existing of high book-to-market ratio stocks and portfolios existing of low book-to-market ratio stocks.

$$R_{i,\tau} - r_{f,\tau} = \alpha_0 + b_{i,t}(R_{m,\tau} - r_{f,\tau}) + s_{i,t} SMB_\tau + h_{i,t} HML_\tau + \epsilon_{i,\tau} \tag{4}$$

The subscript for the day is τ, while the subscript for the month is t, where $\tau \in t$. The factor sensitivities per stock are b_i, s_i and h_i. The idiosyncratic volatility of an individual stock is estimated as the standard deviation of the regression residuals. This standard deviation of daily return residuals is transformed into a monthly standard deviation by multiplying the daily standard deviation by the square root of the number of trading days in month t, formulated as n_t.

$$IVOL_{i,t} = \sqrt{Var\left(\sum \epsilon_{i,\tau}^2\right)} \cdot \sqrt{n_t} \tag{5}$$

In Table 1, descriptive statistics are shown of idiosyncratic volatility ($IVOL$). In the pooled sample of 35,121 stock-month observations (96 for each stock), the mean of the monthly idiosyncratic volatility is 5,57% and the standard deviation is 2.92%.

Table 1: Time-series properties of idiosyncratic volatility

Variable	N	Mean	Std dev.	Autocorrelation at lags				
				1	2	3	4	5
$IVOL$	35,121	5.57	2.92	0.344	0.29	0.25	0.17	0.14

Note: This table summarises the time-series property of individual stock idiosyncratic volatilities. The autocorrelations are the mean statistics across all 366 stocks.

3.2.2 Idiosyncratic volatility process

In their study, Ang et al. (2006) assumed that the idiosyncratic volatility process of a stock may be represented by a random walk process. This means that the first order autocorrelation should be one and then followed by zeros for all other lags. Based on the average autocorrelation at lags one to five, shown in Table 1, evidence is provided that this random walk hypothesis for the average stock's idiosyncratic volatility process is not suitable.

Furthermore, I conducted Dickey-Fuller tests on the estimated monthly idiosyncratic volatilities. According to the average results of these tests, shown in Table 2, I reject the null hypothesis of a random walk in 89% of the stocks. These results also show that the idiosyncratic volatility process of a common stock cannot be represented as a random walk. Consequently, the one-month lagged idiosyncratic volatility of a stock might result in significant measurement errors when used to estimate the volatility for the following month.

Table 2: Idiosyncratic volatilities Augmented Dickey-Fuller tests

Variable	N	t-stat	p-value	Crititcal Values			Rejected
				1%	5%	10%	
IVOL	35,121	-5.68	0.02	-3.47	-2.88	-2.58	89.07%

Note: This table presents average statistics of the cross-sectional t-statistics, p-values and critical values of the Augmented Dickey-Fuller tests conducted on all 366 companies individually. The last column presents the percentage of companies for which the null hypothesis is rejected.

3.2.3 Estimation of expected idiosyncratic volatility

The results in Table 1 and 2, suggest that in order to represent the time-varying feature of idiosyncratic volatility, a better model is required. Following Fu (2009), I made use of EGARCH models and tried to achieve this objective, except that I used out-of-sample estimations instead of in-sample, taken into account the arguments from Guo et al. (2014).

Engle (1982), presented the autoregressive conditional heteroskedasticity (ARCH) model to depict a series of which the volatility is changing. He demonstrated that this approach was useful for modelling the time-series behaviour of financial variables, namely financial market data. The ARCH model was so appealing due to its simultaneous estimation of the mean and variance processes. If this model is applied to stock returns, it assumes that investors adjust their estimations of the mean and variance of returns every period to account for return shocks from the previous period.

The ARCH model was extended to the generalised autoregressive conditional heteroskedasticity (GARCH) model by Bollerslev (1986). The GARCH model added an autoregressive component to the ARCH procedure by permitting the conditional variance to rely on both earlier innovations and their delays. Nonetheless, the GARCH model requires symmetry in its estimations, which is not the case for stock returns due to the well-known leverage effect. This effect, which was initially described by Black (1976), implies that he distribution of stock returns is often negatively biased. It refers to the observed correlation between a stock's volatility and its returns being negative. As a result, researchers devised asymmetric GARCH models as a solution to this issue, which made it feasible to capture the asymmetric feature of stock returns. The exponential GARCH (EGARCH) model was proposed by Nelson (1991), the GJR-GARCH model by Glosten et al. (1993), and the TGARCH model by Zakoian (1994). The EGARCH model utilises a logarithmic equation and adds a new component that enables the model to respond more or less to negative shocks than to positive shocks. Both the GJR-GARCH and TGARCH models employ an indicator function that enables them to respond to negative shocks to varying degrees. These three models have been frequently utilised for predicting volatility, and several studies have demonstrated that they yield accurate projections of stock returns.

For this thesis, I have chosen for the use of EGARCH models to model idiosyncratic volatilities. This model has been favored the most for the estimation of expected idiosyncratic volatility in many studies[4]. I used 9 different EGARCH(p, q) models, in which $1 \leq p \leq 3$ and $1 \leq q \leq 3$. The forms of the constructed the EGARCH(p, q) models are shown below:

$$\begin{gathered} R_{i,t} - r_{f,t} = \alpha_0 + \beta_{i,t}(R_{m,t} - r_{f,t}) + s_i SMB_t + h_i HML_t + \epsilon_{i,t}, \\ \epsilon_{i,t} \sim \mathrm{N}(0, \sigma^2_{i,t}), \end{gathered} \tag{6}$$

$$\ln(\sigma^2_{i,t}) = \alpha_{i,t} + \sum_{l=1}^{p} b_{i,l,t} \ln(\sigma^2_{i,t-l}) + \sum_{k=1}^{q} c_{i,k,t} \left\{ \theta \left(\frac{\epsilon_{i,t-k}}{\sigma_{i,t-k}} \right) + \gamma \left[\left| \frac{\epsilon_{i,t-k}}{\sigma_{i,t-k}} \right| - \left(\frac{2}{\pi} \right)^{\frac{1}{2}} \right] \right\} \tag{7}$$

The Fama-French three-factor model characterises the monthly return process, shown in equation (6). It is assumed that the idiosyncratic return ϵ_{it} at time t has a serially independent normal distribution with a mean of zero and the variance of σ^2_{it}. The conditional variance σ^2_{it} follows an EGARCH process, shown in equation (7). It is dependent on previous periods of residual variance (p) and return shocks (q). Permutation of orders p and q yields nine different EGARCH models: EGARCH(1, 1), EGARCH(1, 2), EGARCH(1, 3), EGARCH(2, 1), EGARCH(2, 2), EGARCH(2, 3), EGARCH(3, 1), EGARCH(3, 2) and EGARCH(3, 3). Each model is separately applied to every stock in the sample. In other words, nine estimates for conditional idiosyncratic volatility in month t+1 will be produced if the idiosyncratic volatility process of a stock for month t converges under all nine models. Consequently, the estimate with the lowest Akaike Information Criterion (AIC) will be selected. I also employed the Bayesian Information Criterion (BIC) in addition to the AIC, although the final findings were insensitive to the alternative selection criteria.

The goal was to estimate the conditional variance $\sigma^2_{i,t+1}$ with an out-of-sample estimation using parameters constructed from all data up to month t, shown in equation (8). The expected idiosyncratic volatility for month t+1 has been derived as in equation (9).

$$\ln(\sigma^2_{i,t+1}) = \alpha_{i,t} + \sum_{l=1}^{p} b_{i,l,t} \ln(\sigma^2_{i,t+1-l}) + \sum_{k=1}^{q} c_{i,k,t} \left\{ \theta \left(\frac{\epsilon_{i,t+1-k}}{\sigma_{i,t+1-k}} \right) + \gamma \left[\left| \frac{\epsilon_{i,t+1-k}}{\sigma_{i,t+1-k}} \right| - \left(\frac{2}{\pi} \right)^{\frac{1}{2}} \right] \right\} \tag{8}$$

[4] For example: Spiegel & Wang (2005), Brockman et al. (2007) and Fu (2009)

$$\mathrm{E}_t[IVOL_{i,t+1}] = \sqrt{e^{\ln\left(\sigma^2_{i,t+1}\right)}} \tag{9}$$

Fu (2009) used the full period data to estimate parameters of the EGARCH models (expanding window) with the requirement of at least 30 monthly return observations to estimate EGARCH models. However, I followed Spiegel & Wang (2005) with respect to the requirement of at least 60 monthly return observations. Their motivation takes into account Lundblad (2007), who stressed the need of a large number of data for obtaining more exact parameter estimates in GARCH-type nonlinear models. This implies that the EGARCH parameters, used for the forecasting of conditional idiosyncratic volatility at month *t*, are estimated from all data up through month *t-1*.

The mean of the expected idiosyncratic volatility (E($IVOL$)) is 5,63% and has a standard deviation of 3,67%. I excluded 15 observations that had an expected idiosyncratic volatility greater than 100%. The number of excluded observations only account for 0.04% of the whole sample. The correlation between the realized idiosyncratic volatility ($IVOL$) based on my method and the expected idiosyncratic volatility is 0.24 and is statistically significant at the 1% level. This correlation is substantially lower than the correlation of 0,46 that Fu (2009) found in his paper. Further empirical evidence of my model selection questions the conclusions drawn by Fu (2009) regarding the necessity to include more lags for the estimations of E($IVOL$).

For all stocks in every month starting from January 2014 (instead of 2009 given that 60 months are needed for the first estimation of company betas) to December 2021, monthly chosen model statistics were not in line with the findings of Fu (2009) for the largest part. Of all estimates, 44,95% were yielded by the EGARCH(1, q) models while Fu (2009) only had 26.67%. Only 24.04% were yielded by the EGARCH(3, q) models while Fu had 40.00%. Where Fu had the EGARCH(3, 1) model as the best fitting model for the most number of observations (16.58%), I consider the EGARCH(3, 2) model to be my best fitting model for the fewest number of observations (6.93%). And where Fu had the EGARCH(1, 1) model as the best fitting model for the fewest number of observations (7.41%), I consider this model as the best fitting model for the most number of observations (22.17%).

3.3 Descriptive statistics

Table 3 shows descriptive statistics of all variables. The mean monthly stock returns (RET) in my sample period is 0.74% with a standard deviation 7.87%. The large standard deviation is due to returns varying across time and companies. The measurement of systematic risk, $BETA$, is conducted with the same methodology as in Fama and French

Table 3: Descriptive statistics of the pooled sample

Variables	Mean	Std dev.	Median	$Q1$	$Q3$	Skew	N
RET	0.74	7.98	0.96	-3.46	5.29	-0.73	35,121
IVOL	5.57	2.92	4.90	3.70	6.61	2.71	35,121
E(IVOL)	5.63	3.67	4.91	3.56	6.73	4.72	35,121
BETA	1.13	0.34	1.15	0.91	1.37	-0.01	35,121
Ln(MV)	9.07	1.16	8.98	8.30	9.82	0.17	35,121
Ln(B/M)	-0.91	0.83	-0.83	-1.42	-0.31	-0.68	35,121

Note: This table presents the pooled descriptive statistics of the stocks listed on STOXX 600 in the period of January 2014 to December 2021. All variables are continuous variables. *RET*, *IVOL*, E(*IVOL*) are measured in percentages. *BETA* has no measurement. *Ln(MV)* and *Ln(B/M)* are the natural logarithms of market capitalization MV and book-to-market ratio *B/M.*

(1992).

For the estimation of company betas (β_i), with the same methodology as Fu (2009) using the market model, I utilised the 60 prior months results each month. I allocated equities to 10 x 10 portfolios according to market capitalization (size) and β. Month by month, this practice was repeated. I then calculated the equal weighted returns for each of the 100 portfolios. I did a full-period time series regression of each portfolio's portfolio return on the current and prior month's value-weighted market returns. The *BETA* of the portfolios was determined by adding these two market return coefficients. This was done to account for non-synchronous trade consequences. I assigned the *BETA* of a portfolio to each stock that belonged to that portfolio in month t, resulting in the *BETA*s. The mean *BETA* is 1.13 and the median is 1.15. Furthermore, because they were highly skewed, market capitalization and book-to-market ratios were converted to their natural logarithm. The book-to-market ratio is obtained by reversing the denominator and numerator of price-to-book ratio from Refinitiv Eikon. Average monthly natural logarithm of market capitalization (*Ln(MV)*) is 9.07 with a standard deviation of 1.16. While the Euro STOXX 600 represents large, mid and small capitalization companies in Europe, the results imply that the across companies, that are not in the financial sector of Euro STOXX 600, there is not much variation of market capitalization. Average monthly natural logarithm of book-to-market ratio (*Ln(B/M)*) is -0.91 with a standard deviation of 0.83. The negative value implies that the companies in my sample were generally overvaluated and are referred as growth stocks.

3.4 Cross-sectional correlations

Table 4 shows all cross-sectional correlations. The correlations can be regarded as univariate tests. The coefficients that are statistically significant at the 1% level are followed by *. The association between monthly stock returns and current idiosyncratic volatility is -0.133, which is statistically significant at the 1% level but contrasting to the results of Fu (2009). The one-month lagged idiosyncratic volatility however has a statistically significant correlation with monthly stock returns of 0.071 (not reported in table). These results show that the one-month lagged $IVOL$ may remain to be a suitable proxy for the expected value in the next month. The statistically significant correlation between monthly stock returns and conditional idiosyncratic volatility (E($IVOL$)) is 0.029. The low correlation coefficient does seem to confirm the finding of Guo et al. (2014) that there is no significant relationship between conditional idiosyncratic and expected stock returns.

Table 4: Cross-sectional correlations

Variables	*IVOL*	*E(IVOL)*	*BETA*	*Ln(MV)*	*Ln(B/M)*
RET	-0.133*	0.029*	-0.003	-0.001	-0.091*
IVOL		0.236*	0.171*	-0.248*	0.020*
E(*IVOL*)			0.159*	-0.173*	-0.015*
BETA				-0.290*	0.264*
Ln(MV)					-0.024*

Note: This table presents the cross-sectional correlations. Variables are defined in Table 3. *p< 0.01.

Consistent with past research, the monthly stock returns are negatively correlated with the natural logarithm of market capitalization (*Ln(MV)*). However this correlation is not statistically significant and therefore I am not able to confirm this relationship based on the univariate test itself. Not consistent is the negative correlation between the natural logarithm book-to-market ratio (*Ln(B/M)*) and returns as previous literature found a positive correlation. The correlation between stock returns (RET) and $BETA$ is flat which confirms the findings of Fama & French (1992). Conditional idiosyncratic volatility (*Ln(B/M)*) is negatively correlated with natural logarithms of size and book-to-market ratio, and positively correlated with $BETA$. All these results are in line with the findings of Fu (2009). Small companies typically exhibit greater idiosyncratic volatility on average than large companies (based on size) and idiosyncratic volatilities of growth companies on average are higher than those of value companies.

4 Results

In this section, I discuss the results of the cross-sectional regressions for idiosyncratic volatility and expected idiosyncratic volatility as well as the results from the portfolios formed on E($IVOL$). The statistical regression methods that I used for testing the effects are Fama-MacBeth, Fixed Effects and Random Effects. With the Fama-MacBeth regression method, I refer to the two stage Fama & MacBeth (1973) methodology. For the model specification of the first stage, I used the cross-sectional regressions formulated in equations (1) and (2). The first stage produces coefficient estimates for every month. In the second stage, I collected the first-stage monthly coefficient estimates to test the significance of the resulting time series using Newey & West (1987) corrected standard errors. This was done to control for any serial correlations. I computed the number of lags for the Newey and West standard errors using a technique utilised by many academics. The number of delays was followed by the integer resulting from the expression $T^{1/4}$. As the time-series has a duration of 96 months, this resulted in three lags in total for the Newey-West standard errors used in the Fama-MacBeth regressions. With the Fixed Effects regression method, I refer to the statistical regression model in which the constant of the regression model is allowed to vary freely across individual companies. I applied this model in order to control for any company-specific attributes that do not vary across time. With the Random Effects regression method, I refer to the statistical regression model to estimate the effect of company-specific characteristics that are unmeasurable. The hypotheses could only be confirmed by the Fama-MacBeth regressions as I am dealing with a time-series where time effects are much larger in contrast to company effects. To make a comparison between the various regressions, Fixed Effects and Random Effects needed to be adjusted also for the time effect. This is covered in more detail in Section 5 below.

4.1 Regressions full time period

Table 5 presents the regression results for the total time period. The first model from the Fama-MacBeth regressions reproduces the Fama and French results. Indeed, the relationship between market beta and average stock return is flat, and the average slope of $BETA$ is not statistically distinct from zero. This is in line with the results of Fama & French (1992). The cross-sectional relationship between size and average stock returns is negative. Small companies generate higher returns on average than their larger counterparts. However the relationship between book-to-market ratio and average stock returns is negative instead of positive in the cross section. This means that in my sample, growth companies on average have had higher returns than value companies.

Table 5: Idiosyncratic Volatility Regressions for full period

	Fama-MacBeth				Fixed Effects			Random Effects		
Variables	1	2	3	4	5	6	7	8	9	10
BETA	0.352 (0.391)									
Ln(MV)	-0.067 (0.062)	-0.081 (0.062)	-0.182*** (0.058)	-0.075 (0.061)	1.072*** (0.243)	1.065*** (0.199)	1.026*** (0.261)	0.010 (0.040)	-0.261*** (0.041)	0.107* (0.043)
Ln(B/M)	-0.739*** (0.154)	-0.696*** (0.179)	-0.710*** (0.176)	-0.720*** (0.175)	-2.317*** (0.355)	-1.759*** (0.320)	-2.608*** (0.370)	-0.870*** (0.088)	-0.857*** (0.095)	-0.896*** (0.088)
E(*IVOL*)		0.031 (0.023)			0.083*** (0.017)			0.060*** (0.013)		
IVOL			-0.100** (0.051)			-0.451*** (0.027)			-0.383*** (0.026)	
$IVOL_{t-1}$				0.033 (0.036)			0.269*** (0.022)			0.206*** (0.018)
Constant	0.277 (0.660)	0.663 (0.800)	2.334*** (0.670)	0.571 (0.749)	-11.571*** (2.048)	-8.026*** (1.668)	-12.453*** (2.224)	-0.479 (0.404)	4.453*** (0.427)	-2.184*** (0.452)
Observations	35,121	35,121	35,121	34,757	35,121	35,121	34,757	35,121	35,121	34,757
Adj. R-Squared	0.063	0.049	0.069	0.053	0.018	0.038	0.025	0.009	0.027	0.014
Number of periods	96	96	96	96						
Number of companies					366	366	366	366	366	366

Note: This table presents the time-series averages of the slopes in cross-sectional regression for the full time period using the standard Fama and MacBeth (1973) methodology as well as panel regression with Fixed Effects and Random Effects. Standard Errors used in the Fama-MacBeth regressions are Newey-West with three lags. Standard Errors for Fixed Effects and Random Effects are company clustered. Standard Errors are in parentheses. *p< 0.10, **p< 0.05, ***p< 0.01.

Models 2, 5 and 8 are regressions formulated as in equation (1) and show positive coefficients for expected idiosyncratic volatility. In model 2, the Fama-MacBeth regression slope for E($IVOL$) is not statistically significant and therefore I am not able to confirm that the relationship is different from zero. Besides the insignificant coefficient, the Adjusted R-Squared decreases after we include E($IVOL$) in the regression from model 1 without company *BETA*s. Models 5 and 8, adjusting for the company effect, show a highly statistical significant positive relationship between expected idiosyncratic volatility and average stock returns. But the slopes in all three models for E($IVOL$) are close to zero. This raises questions about the ability of typical investors to generate positive abnormal returns. From the insignificance of the coefficient of model 2, it is not possible to confirm the fourth hypothesis, which states that there is no significant relation.

Models 3, 6 and 9 are regressions formulated as in equation (2) and show a negative relationship between contemporaneous idiosyncratic volatility and average stock returns. The slope of $IVOL$ is negative in all three models and highly significant in all three models. These results thus confirm my first hypothesis. The negative slopes from the models indicates that in my sample of stocks listed on Euro STOXX 600, an individual stock with high idiosyncratic volatility in month t experienced a negative return on average more often than a positive return in the same month t. Furthermore, we see that the Adjusted R-Squared increases from model 1 to model 3. From a theoretical standpoint, however, it is not possible to draw any conclusions about expected stock returns based on these results. This is based on the possible correlation between the unexpected return shock (R_t - E(R_t)) and idiosyncratic volatility shock ($IVOL_t$ - E($IVOL_t$)), which Fu (2009) explains more deeply in his research.

Models 4, 7 and 10 examine the association between the one-month lagged $IVOL$ and stock returns. The regression results of all three models confirm the complete opposite of the findings of Ang et al. (2006). The monthly stock returns are positively related with the one-month lagged $IVOL$, based on the results of model 7 and 10. Although the reason for this is unclear, a possible cause may be linked to the statistics of my EGARCH model selections. As mentioned earlier, I have used the EGARCH(1, 3) model as my best fitting model for the most number of observations in the data sample. This model contains only one lag of residual variance for the forecast of expected idiosyncratic volatility.

4.2 Before the COVID-19 pandemic

Table 6: Idiosyncratic Volatility Regressions for period before COVID-19

	Fama-MacBeth			Fixed Effects			Random Effects		
Variables	1	2	3	4	5	6	7	8	9
$Ln(MV)$	-0.158*	-0.293***	-0.151*	1.324***	1.221***	1.284***	-0.120***	-0.300***	-0.94*
	(0.065)	(0.051)	(0.061)	(0.321)	(0.290)	(0.327)	(0.036)	(0.036)	(0.037)
$Ln(B/M)$	-0.630***	-0.653***	-0.558***	-2.120***	-1.867***	-2.268***	-2.268***	-0.744***	-0.754***
	(0.158)	(0.154)	(0.152)	(0.484)	(0.455)	(0.485)	(0.485)	(0.099)	(0.095)
E($IVOL$)	0.033			0.042*			0.026		
	(0.020)			(0.018)			(0.015)		
$IVOL$		-0.160***			-0.332***			-0.275***	
		(0.053)			(0.035)			(0.029)	
$IVOL_{t-1}$			0.036			0.107***			0.059**
			(0.031)			(0.028)			(0.021)
$Constant$	1.424	2.334***	1.325*	-13.330***	-10.203***	-13.432***	1.029**	4.219***	0.625
	(0.724)	(0.670)	(0.644)	(2.672)	(2.380)	(2.756)	(0.371)	(0.389)	(0.398)
Observations	26,338	26,338	25,974	26,338	26,338	25,974	26,338	26,338	25,974
Adj. R-Squared	0.041	0.064	0.045	0.017	0.027	0.017	0.008	0.017	0.008
Number of periods	72	72	72						
Number of companies				366	366	366	366	366	366

Note: This table presents the time-series averages of the slopes in cross-sectional regression for the time period before the COVID-19 pandemic using the standard Fama and MacBeth (1973) methodology as well as panel regression with Fixed Effects and Random Effects. Standard Errors used in the Fama-MacBeth regressions are Newey-West with three lags. Standard Errors for Fixed Effects and Random Effects are company clustered. Standard Errors are in parentheses. *p< 0.10, **p< 0.05, ***p< 0.01.

In Table 6, the regression models 2-10 from Table 5 have been re-used but only for the period from January 2014 until December 2019. Again the slope for E($IVOL$) is positive but this time only for the Fixed Effects regression method, shown in model 4. Models 2, 5 and 8 all together confirm my third hypothesis. From all three models, negative slopes for $IVOL$ are highly statistically significant. Also in this shorter sample are the slopes for $IVOL_{t-1}$ positive but only statistically significant for Fixed Effects and Random Effects regression methods. The results for $IVOL_{t-1}$ in model 6 and 9 are once more not in line with the findings of Ang et al. (2006).

4.3 During the COVID-19 pandemic

In Table 7, the regression models 2-10 from Table 5 have been re-used but now only for the period from January 2020 until December 2021 (generally accepted to be the duration of the COVID-19 pandemic). The slopes of E($IVOL$) are all positive but only for the Fixed Effects model significant at the 10% significance level, shown in model 4. $IVOL$ is only statistically significant with the Fixed Effects and Random Effects models (5 and 8) and both have a negative coefficient. Furthermore, the slope of $IVOL$ in the Fama-MacBeth regression (model 2) is positive but insignificant which makes it impossible to confirm my third hypothesis. However it is interesting to observe a change in coefficient from negative in model 2 from Table 6 to positive in model 2 from Table 7. While Poudeh et al. (2022) found the same results, their coefficients are both statistically significant where my second

Table 7: Idiosyncratic Volatility Regressions for period during COVID-19

	Fama-MacBeth			Fixed Effects			Random Effects		
Variables	1	2	3	4	5	6	7	8	9
Ln(MV)	-0.152*	0.153***	-0.148*	8.120***	4.667***	11.528***	-0.566***	-0.135***	-0.806*
	(0.138)	(0.105)	(0.143)	(1.092)	(1.227)	(1.031)	(0.100)	(0.089)	(0.108)
Ln(B/M)	-0.895***	-0.882***	-0.872***	-4.799***	-5.166***	-3.992***	-1.128***	-1.002***	-1.200***
	(0.547)	(0.544)	(0.541)	(1.347)	(1.403)	(1.231)	(0.104)	(0.111)	(0.114)
E($IVOL$)	0.023			0.454*			0.205		
	(0.071)			(0.065)			(0.036)		
$IVOL$		0.080			-0.513***			-0.576***	
		(0.094)			(0.057)			(0.047)	
$IVOL_{t-1}$			0.026			0.814***			0.472**
			(0.111)			(0.048)			(0.036)
Constant	-1.621	-1.747***	-1.658*	-81.853***	-44.022***	-115.134***	-6.877**	4.778***	-10.949
	(2.403)	(1.543)	(2.294)	(9.457)	(10.606)	(9.229)	(1.030)	(0.976)	(1.121)
Observations	8,783	8,783	8,783	8,783	8,783	8,783	8,783	8,783	8,783
Adj. R-Squared	0.074	0.081	0.773	0.079	0.088	0.127	0.017	0.051	0.039
Number of periods	24	24	24						
Number of companies				366	366	366	366	366	366

Note: This table presents the time-series averages of the slopes in cross-sectional regression for the time period during the COVID-19 pandemic using the standard Fama and MacBeth (1973) methodology as well as panel regression with Fixed Effects and Random Effects. Standard Errors used in the Fama-MacBeth regressions are Newey-West with three lags. Standard Errors for Fixed Effects and Random Effects are company clustered. Standard Errors are in parentheses. *p< 0.10, **p< 0.05, ***p< 0.01.

coefficient in model 2 from Table 7 is not. $IVOL_{t-1}$ has similar results as in Table 6 and show that for both periods (before and during the COVID-19 pandemic) the findings of Ang et al. (2006) are not reflected in my sample.

4.4 Portfolio Analysis formed on E($IVOL$)

On the basis of the cross-sectional regression coefficients, suggesting that E($IVOL$) has a potential positive relationship with average stock returns on the company level, I wanted to further investigate this potential relationship by also looking at the portfolio level. I analysed the returns of portfolios constructed by sorting E($IVOL$). The positive significant slopes from the Fixed Effect and Random Effect suggests that stocks with high E($IVOL$) on average experienced better returns than stocks with low E($IVOL$). Therefore, a zero-investment portfolio which is long in high E($IVOL$) and short in low E($IVOL$), may be yielding a positive return. In Table 8, the results of this portfolio analysis are shown.

I approached the portfolios with the same methodology of Fu (2009). The methodology is as described below. Each month, I sorted E($IVOL$) to create ten portfolios. But because there are only 366 companies in my sample I could not divide them perfectly into 10 portfolios. Therefore, all of my portfolios had 37 stocks each month, with the exception of portfolios 5 and 6, which contained 35 stocks each month. The first portfolio had the 37 stocks with the lowest predicted monthly idiosyncratic volatility. The final portfolio contained the 37 stocks with the highest projected idiosyncratic volatility over the following month. The descriptive data for all ten portfolios are presented in Table 8.

The average E($IVOL$) grows monotonically from 2.01% for the first portfolio to 13.48% for the final portfolio. The last portfolio containing the stocks with the highest E($IVOL$) has higher returns than the portfolio containing the stocks with the lowest E($IVOL$). The value-weighted portfolio returns continue to fluctuate around the same return height from the first portfolio to the ninth and with the highest portfolio it increases significantly. However the first portfolio has a value-weighted return of 0.53% and the last 1.03% and thus increases from low to high expected idiosyncratic volatility. The average monthly return for the portfolio with no investments is merely 0.50%. The equal-weighted portfolio returns show more promising results. While the returns continue to fluctuate around the same return height for the first seven portfolios, portfolios eight to ten have an increasing rise in returns. The return spread for the equal-weighted portfolio returns is even larger with 1.15%. These results show that there is a potential positive relationship between E($IVOL$) and individual stock returns.

Subsequently, I conducted the time-series regression of the value-weighted excess returns on the three Fama-French factors for all 10 portfolios individually. The regression intercepts are presented in the last row of Table 8. As we can see in Table 8, the al-

Table 8: Portfolios formed on out-of-sample EGARCH Idiosyncratic Volatility

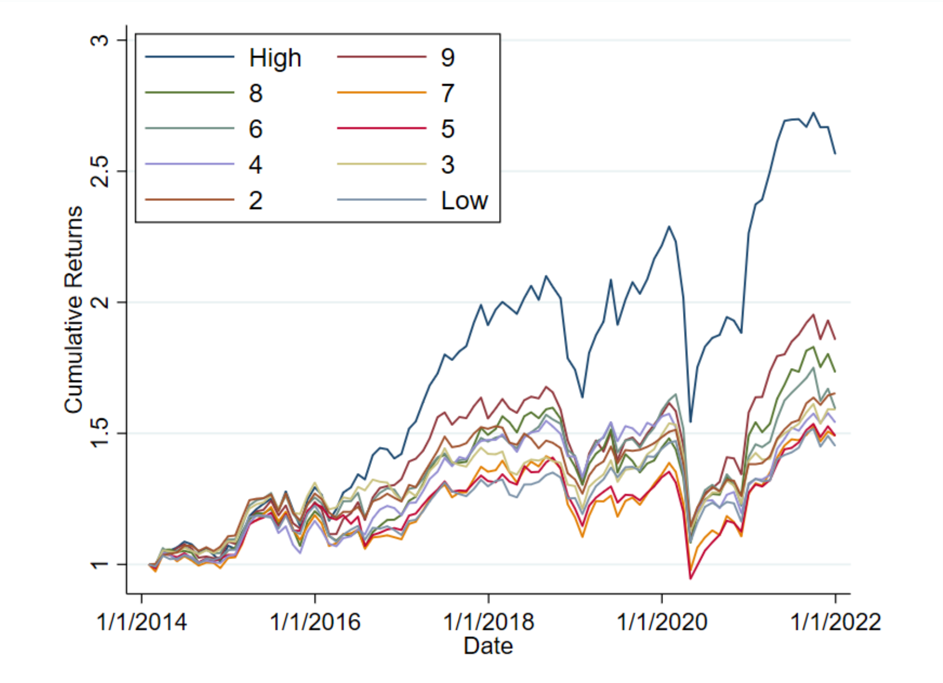
	Portfolios formed on E(*IVOL*)										
Variables	Low	2	3	4	5	6	7	8	9	High	High-Low
Port. RETVW	0.53	0.64	0.66	0.40	0.63	0.82	0.66	0.55	0.66	1.03	0.50
Port. RETEW	0.51	0.64	0.62	0.57	0.56	0.65	0.56	0.72	0.81	1.66	1.15
E(*IVOL*)	2.01	2.98	3.57	4.10	4.63	5.17	5.82	6.71	8.10	13.48	
IVOL	4.71	4.70	4.92	5.04	5.26	5.53	5.79	6.06	6.58	7.11	
BETA	1.02	1.04	1.07	1.08	1.12	1.14	1.17	1.18	1.20	1.24	
MV (€mld)	27,06	24,73	22,26	20,70	18,11	16,36	14,73	12,75	11,78	10,80	
B/M	0.55	0.52	0.50	0.50	0.53	0.55	0.56	0.58	0.59	0.56	
FF Alphas	0.26	0.34	0.29	0.09	0.25	0.50	0.32	0.17	0.22	0.64	0.38

Note: This table presents summary statistics for portfolios formed on out-of-sample EGARCH idiosyncratic volatilities. The first two rows display the time-series means of the value- and equal-weighted monthly portfolio returns, respectively. Other rows display the pooled means of variables for the specific portfolio. The last row displays the alphas derived from time-series regressions of the value-weighted portfolio excess returns on the three Fama-French factors.

pha for the first portfolio is 0.26% and for the last portfolio 0.64%. A hedging portfolio longing the portfolio that holds the 10% highest E(*IVOL*) and shorting the portfolio that holds the 10% lowest E(*IVOL*) will then yield a monthly return of 0.38%. However, this monthly return is not statistically significant as the GRS test statistic from Gibbons, Ross, & Shanken (1989) has a value of only 0.83. Because of this I can not reject the null hypothesis of the GRS test, stating that all alphas are jointly equal to zero. Based on this test statistic it is possible to confirm my fifth hypothesis. This however, does not automatically indicates that the alternative of my fifth hypothesis, stating that such hedging portfolio yields a positive or negative return, would not apply. The results are promising but the statistical significance is needed to draw any more definite conclusions concerning the relationship between expected idiosyncratic volatility and expected stock returns.

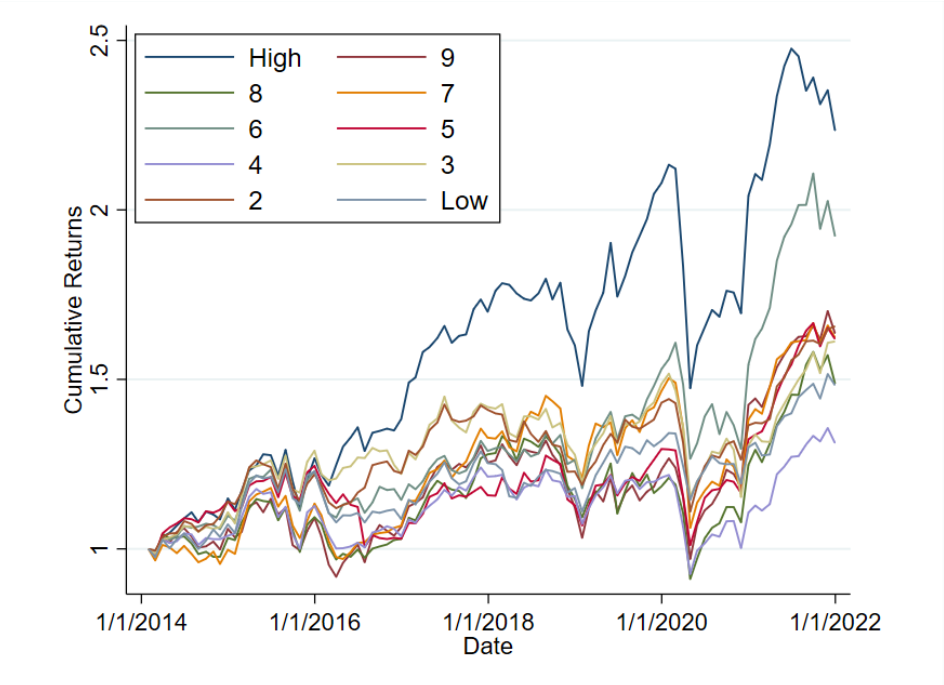
To elaborate further on this potential relationship, Figure 1 and 2 visualize the cumulative returns of all portfolios in order to get a better view of the progression of the portfolios over time. Figure 1 shows a potential positive relationship between E(*IVOL*) and individual stock returns based on equal-weighted portfolios. This is supported by Figure 2, which shows the cumulative returns of the value-weighted portfolios.

Figure 1: Cumulative returns of equal-weighted portfolios



Note: This figure presents the cumulative returns of equal-weighted portfolios formed on E($IVOL$)in the period of January 2014 to December 2021. Portfolios were monthly rebalanced.

Figure 2: Cumulative returns of value-weighted portfolios



Note: This figure presents the cumulative returns of value-weighted portfolios formed on E($IVOL$)in the period of January 2014 to December 2021. Portfolios were monthly rebalanced.

5 Robustness Check

In this section, I explore in more detail if there is any relationship between expected idiosyncratic volatility and expected returns as this is the variable of main interest. I applied an additional regression method and a different mean process for the EGARCH models. In addition, I analyzed at the differences when analyzing only the 10% smallest companies in size from my sample using the same regression methods described in methodology section and applied in the results section.

5.1 Comparing Fixed Effects and Random Effects models

While Fixed Effects and Random Effects models eventually adjust for the same company effect and having the same conclusion. Nonetheless, it is useful to know which model is more robust and with which model I can work further, letting it adjust for time effect. Using the Hausman-Wu test with a χ^2 distribution indicates the most efficient model is. Under the null hypothesis the Random Effects model is preferred and under the alternative hypothesis the fixed effects model. Utilising the Fixed Effects and Random Effects model, following equation (1), the χ^2-statistic (94.70) is very high and a p-value (0.000) near zero. Thus, I reject the null hypothesis and conclude that the Fixed Effects model is more robust.

Table 9: Out-of-sample EGARCH Idiosyncratic Volatility Regressions including Fixed Effects adjusted for Time Effect

Variables	Fama-MacBeth	Fixed Effects	Fixed Effects
Ln(MV)	-0.081 (0.062)	-2.317*** (0.355)	-1.542*** (0.299)
Ln(B/M)	-0.696*** (0.179)	1.072*** (0.243)	1.127*** (0.220)
E(*IVOL*)	0.031 (0.023)	0.083*** (0.017)	0.057*** (0.015)
Constant	0.662 (0.800)	-11.571*** (2.048)	-12.190*** (1.867)
Observations	35,121	35,121	31,151
Adj. R-Squared	0.049	0.018	0.279
Number of periods	96		
Number of companies		366	366
Company Effect		Yes	Yes
Time Effect		No	Yes

Note: This table presents the time-series averages of the slopes in cross-sectional regression for the time period during COVID-19 using the standard Fama and MacBeth (1973) methodology as well as panel regression with Fixed Effects and Fixed Effects adjusted for time effect. Standard Errors used in the Fama-MacBeth regressions are Newey-West with three lags. Standard Errors for both Fixed Effects models are company clustered. Standard Errors are in parentheses. *p< 0.10, **p< 0.05, ***p< 0.01.

Although there are no explicit tests to compare Fixed Effects models with Fama-MacBeth models, I adjusted the Fixed Effects model for time effect to get close to a legitimate comparison. As mentioned before, the time effect has a larger variation in stock returns time-series. So, to make a comparison I rewrote the equation for the fixed effects model as follows:

$$R_{i,t} = \delta_0 + \delta_1 \mathrm{E}_{t-1}[IVOL_{i,t}] + \delta_2 \ln(MV)_{i,t} + \delta_3 \ln(B/M)_{i,t} + \sum_{j=2}^{96} \mu_j MONTH_j + \epsilon_{i,t} \quad (10)$$

In equation (10) dummy variables for each month (the first month omitted to avoid omitted variable bias) are included in the model similar to equation (1).

Table 9 presents the results for the Fixed Effects regression method adjusted for time effect. Compared with the standard Fixed Effects model, the slope of E($IVOL$) is lower but has the same significance level. The decrease of the slope in the adjusted model, shows that the expected idiosyncratic volatility does have time-varying characteristics. Nevertheless, the Adjusted R-Squared increases dramatically with the new Fixed Effects model. However, the coefficient stays small which may indicate that EGARCH models are possibly not the best estimation models for the forecasting of next month's idiosyncratic volatility.

5.2 EGARCH with CAPM mean process

Okpara & Nwezeaku (2009), as previously discussed, examined the Nigerian stock market using EGARCH models with a different mean process than Fu (2009) applied. Instead of the Fama-French three factors, they used the CAPM factor for describing the mean process. Therefore, I wanted to see if there are differences when this different mean process is applied to my sample. The explicit functional forms of the EGARCH models with CAPM as mean process are as follows:

$$\begin{gathered} R_{i,t} - r_{f,t} = \alpha_0 + \beta_{i,t}(R_{m,t} - r_{f,t}) + \epsilon_{i,t}, \\ \epsilon_{i,t} \sim \mathrm{N}(0, \sigma^2_{i,t}), \end{gathered} \quad (11)$$

$$\ln(\sigma^2_{i,t}) = \alpha_{i,t} + \sum_{l=1}^{p} b_{i,l,t} \ln(\sigma^2_{i,t-l}) + \sum_{k=1}^{q} c_{i,k,t} \left\{ \theta \left(\frac{\epsilon_{i,t-k}}{\sigma_{i,t-k}} \right) + \gamma \left[\left| \frac{\epsilon_{i,t-k}}{\sigma_{i,t-k}} \right| - \left(\frac{2}{\pi} \right)^{\frac{1}{2}} \right] \right\} \quad (12)$$

Equation (11) describes the CAPM mean process. Equation (12) is similar to equation (7) as the volatility process stays the same. The expected idiosyncratic volatilities derived with the different mean process were estimated using the same method as described in the

Methodology section. Then the estimates are regressed following equation (1) with Fama-MacBeth and Fixed Effects (adjusted for time effect) regression methods. The results are presented in Table 10.

Table 10: Out-of-sample EGARCH (CAPM mean process) Idiosyncratic Volatility Regressions

	Fama-MacBeth		Fixed Effects	
Variables	FF-3	CAPM	FF-3	CAPM
Ln(MV)	-0.081 (0.062)	-0.095 (0.067)	-1.542*** (0.299)	1.083*** (0.214)
Ln(B/M)	-0.696*** (0.179)	-0.693*** (0.180)	1.127*** (0.220)	-1.540*** (0.298)
E($IVOL$)	0.031 (0.023)	0.017 (0.016)	0.057*** (0.015)	0.009 (0.005)
Constant	0.662 (0.800)	0.945 (0.896)	-12.190*** (1.867)	-11.502*** (1.791)
Observations	35,121	35,121	35,121	35,121
Adj. R-squared	0.049	0.044	0.279	0.278
Number of periods	96	96		
Number of companies			366	366
Company Effect			Yes	Yes
Time Effect			Yes	Yes

Note: This table presents the time-series averages of the slopes in cross-sectional regression for the full time period using the standard Fama and MacBeth (1973) methodology as well as panel regression with Fixed Effects adjusted for time effect. Standard Errors used in the Fama-MacBeth regressions are Newey-West with three lags. Standard Errors for Fixed Effects are company clustered. Standard Errors are in parentheses. *p< 0.10, **p< 0.05, ***p< 0.01.

Looking at Table 10, the slope of E($IVOL$) stays positive for both methods with the Fama-MacBeth regressions but decreases for the CAPM method. The CAPM also provides no statistical significant coefficient and because of this, it is not possible to make any statements as to which mean process works best for estimating expected idiosyncratic volatility. Therefore the CAPM mean process method also confirms my fourth hypothesis. From the Fixed Effects models the only difference is that the significance decreases to none at CAPM method. As a result, this might indicate that stock returns in the European region are not following the CAPM model.

Furthermore, I replicated the same portfolio methodology on the CAPM method estimates as in the Results section. Again I sorted E($IVOL$) in 10 portfolios. The results of the portfolios are presented in Table 11. Looking at the value-weighted and equal-weighted returns there are no monotonic increases but the spread between highest and lowest portfolio is positive for both returns. From the alphas there is also no clear monotonic increase from lowest to highest portfolio, however a zero investment portfolio could yield high abnormal monthly returns of 0.20%. For this mean process with CAPM, the GRS test statistic is very low (1.83), meaning that the null hypothesis, stating that all

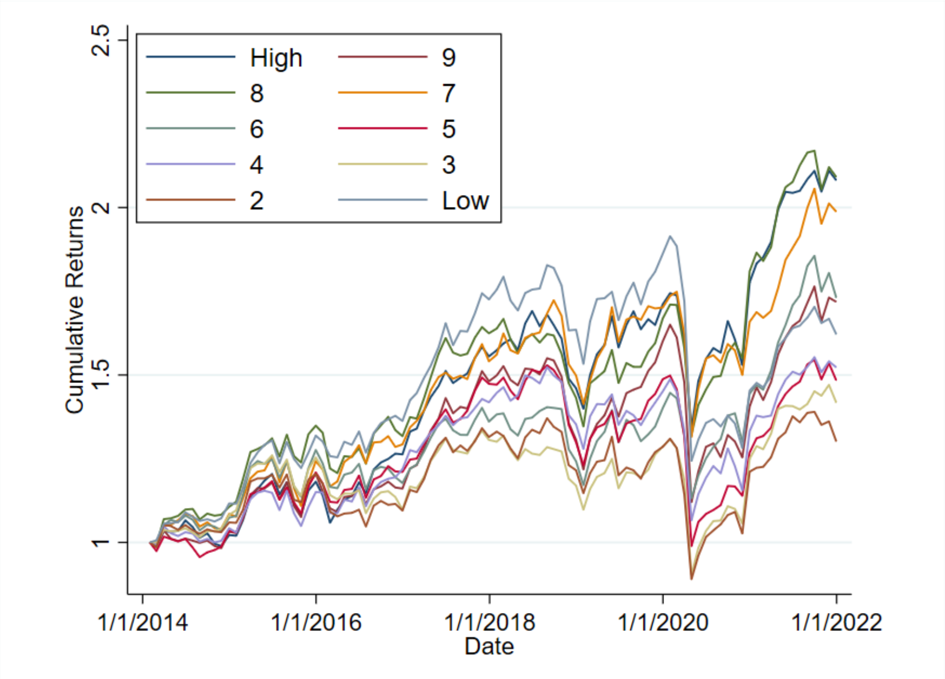
Table 11: Portfolios formed on out-of-sample EGARCH (CAPM mean process) Idiosyncratic Volatility

	Portfolios formed on E$(IVOL)_{CAPM}$										
Variables	Low	2	3	4	5	6	7	8	9	High	High-Low
Port. RETVW	0.46	0.56	0.33	0.60	0.37	0.62	1.09	0.82	0.71	0.66	0.20
Port. RETEW	0.65	0.42	0.53	0.58	0.56	0.72	0.85	0.92	0.72	0.89	0.24
E$(IVOL)_{CAPM}$	0.36	0.60	0.76	0.90	1.04	1.22	1.50	1.98	2.97	9.12	
IVOL	5.86	5.50	5.45	5.39	5.35	5.39	5.44	5.52	5.68	6.10	
BETA	1.13	1.14	1.14	1.14	1.12	1.12	1.11	1.12	1.12	1.14	
MV (€mld)	17.40	19.19	18.82	18.52	18.00	19.62	18.40	17.30	17.30	14.89	
B/M	0.61	0.59	0.59	0.56	0.53	0.51	0.50	0.51	0.51	0.52	
FF Alphas	0.21	0.20	0.01	0.31	0.06	0.22	0.68	0.50	0.34	0.41	0.20

Note: This table presents summary statistics for portfolios formed on out-of-sample CAPM EGARCH idiosyncratic volatilities. The first two rows display the time-series means of the value- and equal-weighted monthly portfolio returns, respectively. Other rows display the pooled means of variables for the specific portfolio. The last row displays the alphas derived from time-series regressions of the value-weighted portfolio excess returns on the three Fama-French factors.

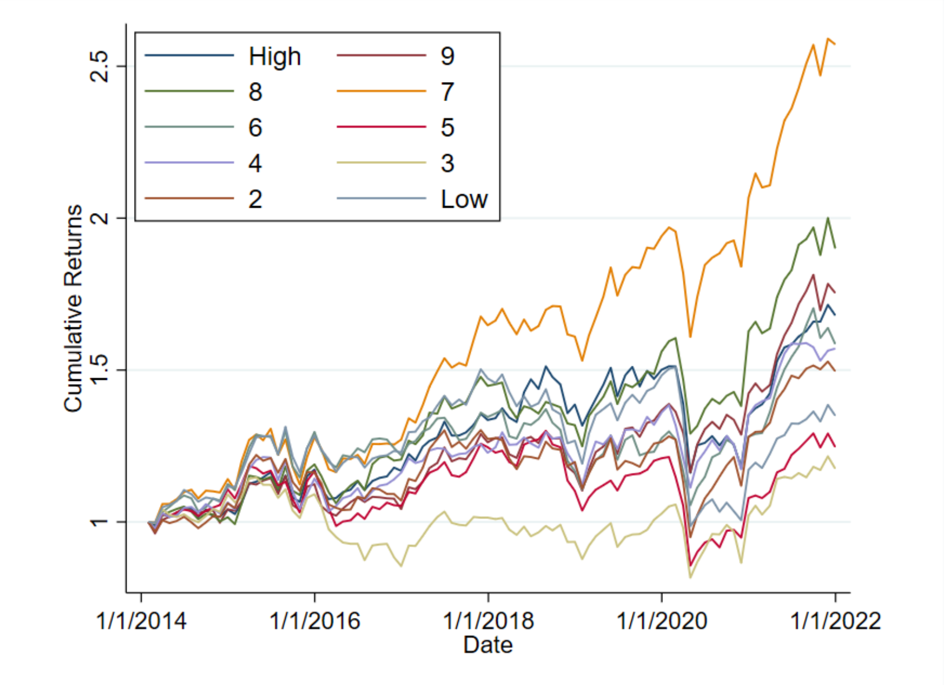
alphas jointly equal to zero, can not be rejected. This leads me to the same conclusion as that of E$(IVOL)$ with a Fama-French three factor mean process.

Figure 3: Cumulative returns of equal-weighted portfolios with CAPM mean process



Note: This figure presents the cumulative returns of equal-weighted portfolios formed on E$(IVOL)$, with CAPM mean process, in the period of January 2014 to December 2021. Portfolios were monthly rebalanced.

Figure 4: Cumulative returns of value-weighted portfolios with CAPM mean process



Note: This figure presents the cumulative returns of value-weighted portfolios formed on E($IVOL$), with CAPM mean process, in the period of January 2014 to December 2021. Portfolios were monthly rebalanced.

Figures 3 and Figure 4 visualize the cumulative returns for both equal-weighted (3) and value-weighted portfolios (4). In comparison with the E($IVOL$) derived from the EGARCH models with the Fama-French three factors for the mean process, it can be seen that the highest portfolio of a CAPM mean process did not achieve the highest returns. Therefore it is clear to say that a CAPM mean process is even less effective for the forecast of expected idiosyncratic volatility and for investing.

5.3 Analysis for 10% smallest (size) companies

In this last subsection, I want to zoom in on the possible relationship between expected idiosyncratic risk and small size companies. As shown in Table 3, the average natural log of market capitalization is significantly high (9.07) compared to the average of the sample (4.29) of Fu (2009). Therefore, I took a look at the coefficients that will be formed using Fama-MacBeth and Fixed Effects (adjusted for time effect) regressions for the 10% smallest companies. I chose the 10% lowest companies at the first month of my time period, January 2014. The mean natural log of market value of these companies is (7.49) which is still a lot higher than the average of the sample of Fu (2009) sample. However, I was still curious to see if there are any differences for this sub-sample of companies compared to my full sample.

Table 12: Out-of-sample EGARCH Idiosyncratic Volatility Regressions for 10% smallest size

	Fama-MacBeth	Fixed Effects
Variables	1	2
Ln(MV)	0.435	2.090***
	(0.303)	(0.526)
Ln(B/M)	-0.762**	0.208
	(0.275)	(0.9544)
E(IVOL)	0.017	-0.028
	(0.042)	(0.024)
Constant	-2.023	-11.504**
	(2.283)	(3.530)
Observations	3,552	3,552
Adj. R-squared	0.054	0.186
Number of periods	96	
Number of companies		37
Company Effect		Yes
Time Effect		Yes

Note: This table presents the time-series averages of the slopes in cross-sectional regression for the full time period using the standard Fama and MacBeth (1973) methodology as well as panel regression with Fixed Effects adjusted for time effect. Standard Errors used in the Fama-MacBeth regressions are Newey-West with three lags. Standard Errors for Fixed Effects are company clustered. Standard Errors are in parentheses. *p< 0.10, **p< 0.05, ***p< 0.01.

Table 12 shows the results of the regressions methods applied to the sub-sample. E($IVOL$) has a positive slope in the Fama-MacBeth regression (model 1) and a negative slope in the Fixed Effects regression that is adjusted for time effect (model 2). Both slopes are statistically insignificant. The main reason for the insignificance in both models is due to the low number of observations. Taking into account he magnitude of both coefficients as well, it does cast doubt on whether accounting for small size companies is needed as the slopes are close to zero.

6 Conclusion

In this thesis, from a sample of stocks from STOXX 600 Europe, I found empirical evidence for a negative and significant relationship between contemporaneous idiosyncratic risk and expected returns in the period from January 2014 to December 2021. From this evidence it can be concluded that in the analyzed sample period, investors in European stock markets, on average, were not rewarded for bearing idiosyncratic risk in their portfolios. In addition, I split the data into two sub-samples, one before the COVID-19 pandemic, the other during the pandemic. The coefficient of idiosyncratic volatility changed from significant negative in the sub-sample before the pandemic to a positive but insignificant coefficient in the sub-sample during the pandemic. These results are interesting and call for more extensive research in the coming years.

Ang et al. (2006) furthermore found a negative relationship between the one-month lagged idiosyncratic volatility and monthly stock returns. In my sample, I found contradictory results with the one-month lagged idiosyncratic risk. In line with the research of Fu (2009), I demonstrated that idiosyncratic volatilities vary with time and that the one-month lagged value is not a reliable explanatory factor for the expected value. Similar to Fu (2009) as criticized Guo et al. (2014), I employed EGARCH models in order to determine the expected idiosyncratic risk with an out-of-sample estimation instead of an in-sample (Fu, 2009). As the main purpose of this thesis was to find out if next month's expected idiosyncratic risk has explanatory power for expected stock returns, I found no relationship between expected idiosyncratic risk and expected stock returns. This thesis therefore demonstrates that there is no indication that investors anticipate a better or lower return from stocks with a higher forecast idiosyncratic risk. To confirm the non-descriptive power of the expected idiosyncratic risk, I also conducted a portfolio analysis on portfolios sorted on expected idiosyncratic risk. While the monthly positive return of a zero investment portfolio prior to transaction fees looks promising, it was not possible to confirm its statistically significance based on GRS test statistic.

In this thesis, an alternative mean process with CAPM for the EGARCH models was also examined. Although this is a much simpler mean process than the Fama-French three factors (baseline), the portfolio analysis and regression results did not improve but remained more or less the same. The CAPM mean process EGARCH models had a slightly larger GRS test statistic in comparison with the baseline model but not statistically significant as well, and thus indicated that CAPM mean process was no more effective in identifying the systematic risks that underlie stock performance.

Last, I looked at the smallest size companies in my sample to observe if there were any differences with EGARCH model estimations for this smaller sample. Because only few observations were left once I adjusted the sample of stocks for small size companies only, there was no visible relationship and coefficients were practically equal to zero.

6.1 Limitations and future research recommendations

The main objective of this thesis was to observe if EGARCH forecast idiosyncratic risk can predict expected stock returns in the European region. The reason for choosing the European region is because of the limited literature that has been written about it so far. However, this research has had to take into account, and has been limited by, several limitations. The first limitation regards the sample of stocks that was used in this thesis.

Despite having a significant share of the free-float market capitalization of the European stock market, STOXX 600 Europe comprises mostly of large- and mid-cap companies. The number of small-cap companies is limited. Second, my sample period included the COVID-19 pandemic which may also have a major impact on the results and reliability. Third, the number of EGARCH models to forecast expected idiosyncratic volatility and the combinations with different mean processes is low. It would be ideal to apply more model combinations, but owing to its complexity, this will demand more computational power, which was not available to me. Fourth, as a proxy for expected returns, I applied realized returns. Despite being a fair estimate of expected returns, the quality of the estimate is subject to improvement. Hence, I would make the following recommendations for future research to enhance the conclusions of this thesis:

1. **More small-cap companies**. To get an overall picture, more consideration should also be given to small-cap companies in the European stock market. This in order to avoid any biases in favor of size characteristics.

2. **Longer time interval**. This thesis only used a sample period from January 2014 to December 2021 (7 years in total), which is on the low side compared to previous literature. In further research, longer time intervals should be used. A longer time interval can also contribute to the optimisation of the EGARCH models. As discussed in Section 2, the higher the number of observations the more precise parameter estimates will be.

3. **Daily observations instead of monthly**. Daily observations may provide further optimisation of model parameters. But it must be taken into account that this requires more computational power since observations then increase by a factor of 22 (average number of trading days in a month).

4. **Testing additional non-linear ARCH model combinations**. As mentioned in Section 2, there are numerous other models available from the ARCH family. GARCH, GJR-GARCH and T-GARCH can also be used in future research to investigate any differences in the models and significance of the estimated expected idiosyncratic volatilities with expected stock returns. It is beyond the scope of this thesis to address additional ARCH model combinations.

5. **More factors in asset pricing models**. Having used the Fama-French three factor model as my baseline, the model is dated and newer models have mostly been applied in recent research.

6. **Better indicators of expected stock return**. As discussed by Fu (2009), the use of implied cost of capital (ICC) as an alternative indicator of expected stock returns should be considered. This should certainly not be omitted in future research.

References

Al Rahahleh, N., Adeinat, I., & Bhatti, I. (2016). On ethnicity of idiosyncratic risk and stock returns puzzle. *Humanomics*, *32*(1), 48–68.

Ang, A., Hodrick, R. J., Xing, Y., & Zhang, X. (2006). The cross-section of volatility and expected returns. *Journal of Finance*, *61*(1), 259–299.

Ang, A., Hodrick, R. J., Xing, Y., & Zhang, X. (2009). High idiosyncratic volatility and low returns: International and further us evidence. *Journal of Financial Economics*, *91*(1), 1–23.

Angelidis, T., & Tessaromatis, N. (2008). Does idiosyncratic risk matter? Evidence from European stock markets. *Applied Financial Economics*, *18*(2), 125–137.

Annaert, J., De Ceuster, M., & Van Doninck, F. (2022). Decomposing the idiosyncratic volatility anomaly among euro area stocks. *Finance Research Letters*.

Annaert, J., De Ceuster, M., & Verstegen, K. (2013). Are extreme returns priced in the stock market? European evidence. *Journal of Banking and Finance*, *37*(9), 3401–3411.

Black, F. (1976). Studies of stock price volatility changes. in: Proceedings of the 1976 meeting of the business and economic statistics section. *American Statistical Association*, 177-181.

Bley, J., & Saad, M. (2012a). Idiosyncratic risk and expected returns in frontier markets: Evidence from GCC. *Journal of International Financial Markets, Institutions and Money*, *22*(3), 538–554.

Bley, J., & Saad, M. (2012b). Idiosyncratic risk and expected returns in frontier markets: Evidence from gcc. *Journal of International Financial Markets*, *22*(3), 538–554.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, *31*(3), 307–327.

Brockman, P., Schutte, M., & Yu, W. (2007). Is idiosyncratic volatility priced? the international evidence. *Unpublished working paper. University of Missouri U Columbia.*

Campbell, J. Y., Lettau, M., Malkiel, B. G., & Xu, Y. (2001). Have Individual Stocks Become More Volatile? An Empirical Exploration of Idiosyncratic Risk. *The Journal of Finance*, *56*(1), 1–43.

Carhart, M. M. (1997). On persistence in mutual fund performance. *The Journal of finance*, *52*(1), 57–82.

De Mendonça, F. P., Klotzle, M. C., Pinto, A. C. F., & da Silva Montezano, R. M. (2012). A relação entre risco idiossincrático e retorno no mercado acionário brasileiro. *Revista Contabilidade & Finanças-USP*, *23*(60), 246–257.

Duffee, G. R. (1995). Stock returns and volatility a firm-level analysis. *Journal of financial Economics*, *37*(3), 399–420.

Eiling, E. (2006). Can nontradable assets explain the apparent premium for idiosyncratic risk? the case of industry-specific human capital. *Unpublished working paper. Tilburg University, Netherlands.*

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, *50*(4), 987–1007.

Fama, E. F., & French, K. R. (1992). The Cross-Section of Expected Stock Returns. *The Journal of Finance*, *47*(2), 427–465.

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.

Fama, E. F., & MacBeth, J. D. (1973). Risk, return, and equilibrium: Empirical tests. *Journal of political economy*, *81*(3), 607–636.

Fu, F. (2009). Idiosyncratic risk and the cross-section of expected stock returns. *Journal of Financial Economics*, *91*(1), 24–37.

Gibbons, M. R., Ross, S. A., & Shanken, J. (1989). A test of the efficiency of a given portfolio. *Econometrica: Journal of the Econometric Society*, 1121–1152.

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The journal of finance*, *48*(5), 1779–1801.

Goetzmann, W. N., & Kumar, A. (2008). Equity portfolio diversification. *Review of Finance*, *12*(3), 433–463.

Guo, H., Kassa, H., & Ferguson, M. F. (2014). On the relation between egarch idiosyncratic volatility and expected stock returns. *Journal of Financial and Quantitative Analysis*, *49*(1), 271–296.

Huang, W., Liu, Q., Rhee, S. G., & Zhang, L. (2010). Return reversals, idiosyncratic risk, and expected returns. *The Review of Financial Studies*, *23*(1), 147–168.

Kearney, C., & Potì, V. (2008). Have European stocks become more volatile? An empirical investigation of idiosyncratic and market risk in the Euro area. *European Financial Management*, *14*(3), 419–444.

Kinnunen, J., & Martikainen, M. (2017). Expected Returns and Idiosyncratic Risk: Industry-Level Evidence from Russia. *Emerging Markets Finance and Trade*, *53*(11), 2528–2544.

Levy, H. (1978). Equilibrium in an imperfect market: A constraint on the number of securities in the portfolio. *The American Economic Review*, *68*(4), 643–658.

Lundblad, C. (2007). The risk return tradeoff in the long run: 1836–2003. *Journal of Financial Economics*, *85*(1), 123–150.

Malkiel, B. G., & Xu, Y. (2002). Idiosyncratic risk and security returns. *Working Paper, University of Texas, Dallas*.

Merton, R. C. (1987). A simple model of capital market equilibrium with incomplete information. *Journal of Banking and Finance*, *42*(3), 483–510.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the Econometric Society*, *59*(2), 347–370.

Newey, W. K., & West, K. D. (1987). Hypothesis testing with efficient method of moments estimation. *International Economic Review*, 777–787.

Okpara, G. C., & Nwezeaku, N. C. (2009). Idiosyncratic risk and the cross-section of expected stock returns: Evidence from nigeria. *European Journal of Economics, Finance and Administrative Science*, *17*, 1–10.

Perera, H., & Ediriwickrama, T. (2020). Impact of idiosyncratic volatility on average stock returns: Evidence from sri lanka.

Poudeh, T., Reza, S., Sungchul, C., & Fu, C. (2022). The Effect of COVID-19 on the Relationship between Idiosyncratic Volatility and Expected Stock Returns. *Risks*, *10*(3).

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, *19*(3), 425–442.

Spiegel, M. I., & Wang, X. (2005). Cross-sectional variation in stock returns: Liquidity and idiosyncratic risk.

Tan, O. (2020). Stock return and expected idiosyncratic risk. *EUR repository*.

Tavakoli Baghdadabad, M., & Mallik, G. (2018). Global idiosyncratic risk moments. *Empirical Economics*, *55*(2), 731–764.

Zakoian, J.-M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and control*, *18*(5), 931–955.

Appendix

In this section, I present all python codes and the stata do file which are used for this thesis. The first python script is used to estimate $IVOL$ in and E($IVOL$). The second python script is used for calculating $BETA$s. The third python script is used for Portfolio construction for both E($IVOL$) estimates (FF3 and CAPM). Furthermore, I present my full STATA script from the do file in the fourth subsection of the Appendix.

estimate *IVOL* and E(*IVOL*)

```
import math
import numpy as np
from arch.univariate import LS, EGARCH
from arch.__future__ import reindexing
from statsmodels.tsa.stattools import adfuller
import statsmodels.api as sm
from itertools import repeat
from arch import arch_model
import time

#------------------------------------Daily Panel Data------------------------------------

dfDailyPrices = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\
    Data\Daily\Daily Adjusted Closing Prices without financials.xlsx', sheet_name=0)
dfFF3factorsDaily = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\
    Thesis\Data\Daily\ff 3 daily.xlsx')

# 1 ____Prep____

prices = dfDailyPrices.values.tolist()

# Array for all unique companies
companies = dfDailyPrices['ISIN'].unique()

MKT = dfFF3factorsDaily['Mkt-RF'].values.tolist()[1:]
MKT = MKT * len(companies)
MKT = np.array(MKT)
# MKT = MKT/100
SMB = dfFF3factorsDaily['SMB'].values.tolist()[1:]
SMB = SMB * len(companies)
SMB = np.array(SMB)
# SMB = SMB/100
HML = dfFF3factorsDaily['HML'].values.tolist()[1:]
HML = HML * len(companies)
HML = np.array(HML)
# HML = HML/100
RF = dfFF3factorsDaily['RF'].values.tolist()[1:]
RF = RF * len(companies)
RF = np.array(RF)
# RF = RF/100

# List for all prices for all companies
for i in range(len(companies)):
    prices[i].remove(companies[i])
```

```
# List for all dates
dates = dfDailyPrices.columns.values.tolist()
dates = dates[2:]


# 2 ____Making suitable columns for dataframe____

# Deriving returns
Returns = []

for row in prices:
    for i in range(len(row)):
        try:
            Returns.extend([math.log(row[i+1]/row[i])])
        except ZeroDivisionError:
            Returns.extend([0])
        except IndexError:
            break

Returns = np.array(Returns) * 100
Excess_Returns = Returns - RF

# Dates
Alldates = dates * len(companies)

# Companies
Allcompanies = []
for company in companies:
    l = []
    l.append(company)
    Allcompanies.extend(l * len(dates))
Allcompanies = np.array(Allcompanies)


# 3 ____Making dataframe____

# Constructing dataframe
dfDaily = pd.DataFrame({
    'Date': Alldates,
    'ISIN': Allcompanies,
    'Excess_Return': Excess_Returns,
    'MKT':MKT,
    'SMB':SMB,
    'HML':HML
    })


# 4 ____Deriving monthly standard deviations____

companies = dfDaily['ISIN'].unique()
std_devs = {}
count = 0
for company in companies:
    std_devs[company] = []
    dfC = dfDaily[dfDaily['ISIN'] == company]
    dfC['year'] = pd.DatetimeIndex(dfC['Date']).year
    dfC['month'] = pd.DatetimeIndex(dfC['Date']).month
    df_years = [dfC[dfC['year'] == y] for y in dfC['year'].unique()]
    count += 1
    print(count)
    for df_year in df_years:
        df_months = [df_year[df_year['month'] == m] for m in df_year['month'].unique()]
        for df_month in df_months:
            X = df_month[['MKT' ,'SMB' ,'HML']]
            y = df_month['Excess_Return']

            ls = LS(y, X)
            res = ls.fit()
            resids = np.std(res.resid)
            std_dev = resids * math.sqrt(len(res.resid))
            std_devs[company].append(std_dev)

# IVOL for monthly
IVOL = []
for company in companies:
    IVOL.extend(std_devs[company])

IVOL = np.array(IVOL)


#-----------------------------Monthly Panel Data-----------------------------

dfMonthlyPrices = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\Monthly\Monthly Data without financials.xlsx', sheet_name=0)
dfFF3factorsMonthly = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\Monthly\ff 3 monthly.xlsx')

# 1 ----Prep----

prices = dfMonthlyPrices.values.tolist()
for i in range(len(companies)):
    prices[i].remove(companies[i])

MKT = dfFF3factorsMonthly['Mkt-RF'].values.tolist()[1:]
MKT = MKT * len(companies)
MKT = np.array(MKT)
SMB = dfFF3factorsMonthly['SMB'].values.tolist()[1:]
SMB = SMB * len(companies)
SMB = np.array(SMB)
HML = dfFF3factorsMonthly['HML'].values.tolist()[1:]
HML = HML * len(companies)
HML = np.array(HML)
RF = dfFF3factorsMonthly['RF'].values.tolist()[1:]
RF = RF * len(companies)
RF = np.array(RF)

# List for all dates
dates = dfMonthlyPrices.columns.values.tolist()
dates = dates[2:]

# Companies
Allcompanies = []
for company in companies:
    l = []
    l.append(company)
    Allcompanies.extend(l * len(dates))
Allcompanies = np.array(Allcompanies)
```

```
# 2 ____Making suitable columns for dataframe____

# Deriving returns
Returns = []

for row in prices:
    for i in range(len(row)):
        try:
            Returns.extend([math.log(row[i+1]/row[i])])
        except ZeroDivisionError:
            Returns.extend([0])
        except IndexError:
            break

Returns = np.array(Returns) * 100
Excess_Returns = Returns - RF

# Dates column for in dataframe
Alldates = dates * len(companies)

# Dickey Fuller test for rejecting random walk ivol and Auto Coorrelation
Auto_Correlation = {}
dickey_fuller_info = {}
for key in std_devs:
    dickey_fuller_info[key] = []
    Auto_Correlation[key] = []

    data = std_devs[key]

    dickey_fuller_info[key] = adfuller(data)
    Auto_Correlation[key] = sm.tsa.acf(data, nlags = 5, fft=False)[1:]

Total_DF_info = {}
for key in dickey_fuller_info:
    Total_DF_info[key] = {}

    Total_DF_info[key]['t-stat'] = []
    Total_DF_info[key]['t-stat'] = dickey_fuller_info[key][0]

    Total_DF_info[key]['p-value'] = []
    Total_DF_info[key]['p-value'] = dickey_fuller_info[key][1]

    Total_DF_info[key]['1%'] = []
    Total_DF_info[key]['1%'] = dickey_fuller_info[key][4].get('1%')

    Total_DF_info[key]['5%'] = []
    Total_DF_info[key]['5%'] = dickey_fuller_info[key][4].get('5%')

    Total_DF_info[key]['10%'] = []
    Total_DF_info[key]['10%'] = dickey_fuller_info[key][4].get('10%')

Average_DF_info = []
num = 5
Average_DF_info = [[] for x in repeat(None, num)]

for key in Total_DF_info:
    Average_DF_info[0].append(Total_DF_info[key].get('t-stat'))
    Average_DF_info[1].append(Total_DF_info[key].get('p-value'))
    Average_DF_info[2].append(Total_DF_info[key].get('1%'))
    Average_DF_info[3].append(Total_DF_info[key].get('5%'))
    Average_DF_info[4].append(Total_DF_info[key].get('10%'))

# Output for average Dickey-Fuller test results
DF_output = {}
DF_output['t-stat'] = sum(Average_DF_info[0])/len(Average_DF_info[0])
DF_output['p-value'] = sum(Average_DF_info[1])/len(Average_DF_info[1])
DF_output['1%'] = sum(Average_DF_info[2])/len(Average_DF_info[2])
DF_output['5%'] = sum(Average_DF_info[3])/len(Average_DF_info[3])
DF_output['10%'] = sum(Average_DF_info[4])/len(Average_DF_info[4])

count = 0
for i in Average_DF_info[1]:
    if i <= 0.05:
        count += 1

DF_output['Percentage Rejected'] = round(count/len(Average_DF_info[1]), 4)


# Output for average of first five autocorrelation lags
Average_AC_info = [[] for x in repeat(None, num)]
for key in Auto_Correlation:
    Average_AC_info[0].append(Auto_Correlation[key][0])
    Average_AC_info[1].append(Auto_Correlation[key][1])
    Average_AC_info[2].append(Auto_Correlation[key][2])
    Average_AC_info[3].append(Auto_Correlation[key][3])
    Average_AC_info[4].append(Auto_Correlation[key][4])

AC_output = {}
AC_output['1st lag'] = sum(Average_AC_info[0])/len(Average_AC_info[0])
AC_output['2nd lag'] = sum(Average_AC_info[1])/len(Average_AC_info[1])
AC_output['3rd lag'] = sum(Average_AC_info[2])/len(Average_AC_info[2])
AC_output['4th lag'] = sum(Average_AC_info[3])/len(Average_AC_info[3])
AC_output['5th lag'] = sum(Average_AC_info[4])/len(Average_AC_info[4])


# 3 ____EGARCH with different lags____

dfIVOL = pd.DataFrame({
    'Date': Alldates,
    'ISIN': Allcompanies,
    'Excess_Return': Excess_Returns,
    'MKT':MKT,
    'SMB':SMB,
    'HML':HML
    })

# Function for forecasting IVOL with multiple models
def idiosyncratic_forecast(y,x):
    aic_scores = []
    model_params = []
    model_resid = []
    model_cond_var = []
    convergence = []
    for i in range(1, 4):
        for j in range(1, 4):
            model = arch_model(y, x, mean='LS', vol='EGARCH', p=i, o=i, q=j)
            res = model.fit(disp='off', options={'iter': 1000})
            convergence.append(res.convergence_flag)
            aic_scores.append(res.aic)
            model_params.append(res._params.T)
            model_resid.append(res.resid[-3:])
            model_cond_var.append(np.power(res.conditional_volatility[-3:],2))

    aic_scores_correct = []
    for i in range(0, len(convergence)):
        if convergence[i] == 0:
            aic_scores_correct.append(aic_scores[i])

    best_model = 0
    count = 0
    for i in range(0, len(aic_scores)):
        if aic_scores[i] == min(aic_scores_correct):
            best_model = count
            break
        else:
            count += 1

    best_model_aic = min(aic_scores_correct)

    if aic_scores[0] == best_model_aic: # 1,1,1 - p=1, q=1
        cond_var = (model_params[0][4]
        + model_params[0][5] * (abs(model_resid[0][2]/math.sqrt(model_cond_var[0][2])) -
            math.sqrt(2/math.pi))
        + model_params[0][6] * (model_resid[0][2]/math.sqrt(model_cond_var[0][2]))
        + model_params[0][7] * math.log(model_cond_var[0][2]))

    elif aic_scores[1] == best_model_aic: #1,1,2 - p=2, q=1
        cond_var = (model_params[1][4]
        + model_params[1][5] * (abs(model_resid[1][2]/math.sqrt(model_cond_var[1][2])) -
            math.sqrt(2/math.pi))
        + model_params[1][6] * (model_resid[1][2]/math.sqrt(model_cond_var[1][2]))
        + model_params[1][7] * math.log(model_cond_var[1][2])
        + model_params[1][8] * math.log(model_cond_var[1][1]))

    elif aic_scores[2] == best_model_aic: #1,1,3 - p=3, q=1
        cond_var = (model_params[2][4]
        + model_params[2][5] * (abs(model_resid[2][2]/math.sqrt(model_cond_var[2][2])) -
            math.sqrt(2/math.pi))
        + model_params[2][6] * (model_resid[2][2]/math.sqrt(model_cond_var[2][2]))
        + model_params[2][7] * math.log(model_cond_var[2][2])
        + model_params[2][8] * math.log(model_cond_var[2][1])
        + model_params[2][9] * math.log(model_cond_var[2][0]))

    elif aic_scores[3] == best_model_aic: #2,2,1 - p=1, q=2
        cond_var = (model_params[3][4]
        + model_params[3][5] * (abs(model_resid[3][2]/math.sqrt(model_cond_var[3][2])) -
            math.sqrt(2/math.pi))
        + model_params[3][6] * (abs(model_resid[3][1]/math.sqrt(model_cond_var[3][1])) -
            math.sqrt(2/math.pi))
        + model_params[3][7] * (model_resid[3][2]/math.sqrt(model_cond_var[3][2]))
        + model_params[3][8] * (model_resid[3][2]/math.sqrt(model_cond_var[3][1]))
        + model_params[3][9] * math.log(model_cond_var[3][2]))

    elif aic_scores[4] == best_model_aic: # 2,2,2 - p=2, q=2
        cond_var = (model_params[4][4]
        + model_params[4][5] * (abs(model_resid[4][2]/math.sqrt(model_cond_var[4][2])) -
            math.sqrt(2/math.pi))
        + model_params[4][6] * (abs(model_resid[4][1]/math.sqrt(model_cond_var[4][1])) -
            math.sqrt(2/math.pi))
        + model_params[4][7] * (model_resid[4][2]/math.sqrt(model_cond_var[4][2]))
        + model_params[4][8] * (model_resid[4][1]/math.sqrt(model_cond_var[4][1]))
        + model_params[4][9] * math.log(model_cond_var[4][2])
        + model_params[4][10] * math.log(model_cond_var[4][1]))

    elif aic_scores[5] == best_model_aic: # 2,2,3 - p=3, q=2
        cond_var = (model_params[5][4]
        + model_params[5][5] * (abs(model_resid[5][2]/math.sqrt(model_cond_var[5][2])) -
            math.sqrt(2/math.pi))
        + model_params[5][6] * (abs(model_resid[5][1]/math.sqrt(model_cond_var[5][1])) -
            math.sqrt(2/math.pi))
        + model_params[5][7] * (model_resid[5][2]/math.sqrt(model_cond_var[5][2]))
        + model_params[5][8] * (model_resid[5][1]/math.sqrt(model_cond_var[5][1]))
        + model_params[5][9] * math.log(model_cond_var[5][2])
        + model_params[5][10] * math.log(model_cond_var[5][1])
        + model_params[5][11] * math.log(model_cond_var[5][0]))

    elif aic_scores[6] == best_model_aic: #3,3,1 - p=1, q=3
        cond_var = (model_params[6][4]
        + model_params[6][5] * (abs(model_resid[6][2]/math.sqrt(model_cond_var[6][2])) -
            math.sqrt(2/math.pi))
        + model_params[6][6] * (abs(model_resid[6][1]/math.sqrt(model_cond_var[6][1])) -
            math.sqrt(2/math.pi))
        + model_params[6][7] * (abs(model_resid[6][0]/math.sqrt(model_cond_var[6][0])) -
            math.sqrt(2/math.pi))
        + model_params[6][8] * (model_resid[6][2]/math.sqrt(model_cond_var[6][2]))
        + model_params[6][9] * (model_resid[6][1]/math.sqrt(model_cond_var[6][2]))
        + model_params[6][10] * (model_resid[6][0]/math.sqrt(model_cond_var[6][0]))
        + model_params[6][11] * math.log(model_cond_var[6][2]))

    elif aic_scores[7] == best_model_aic: #3,3,2 - p=2, q=3
        cond_var = (model_params[7][4]
        + model_params[7][5] * (abs(model_resid[7][2]/math.sqrt(model_cond_var[7][2])) -
            math.sqrt(2/math.pi))
        + model_params[7][6] * (abs(model_resid[7][1]/math.sqrt(model_cond_var[7][1])) -
            math.sqrt(2/math.pi))
        + model_params[7][7] * (abs(model_resid[7][0]/math.sqrt(model_cond_var[7][0])) -
            math.sqrt(2/math.pi))
        + model_params[7][8] * (model_resid[7][2]/math.sqrt(model_cond_var[7][2]))
        + model_params[7][9] * (model_resid[7][1]/math.sqrt(model_cond_var[7][1]))
        + model_params[7][10] * (model_resid[7][0]/math.sqrt(model_cond_var[7][0]))
        + model_params[7][11] * math.log(model_cond_var[7][2])
        + model_params[7][12] * math.log(model_cond_var[7][1]))

    else: #3,3,3 - p=3, q=3
        cond_var = (model_params[8][4]
        + model_params[8][5] * (abs(model_resid[8][2]/math.sqrt(model_cond_var[8][2])) -
            math.sqrt(2/math.pi))
        + model_params[8][6] * (abs(model_resid[8][1]/math.sqrt(model_cond_var[8][1])) -
            math.sqrt(2/math.pi))
        + model_params[8][7] * (abs(model_resid[8][0]/math.sqrt(model_cond_var[8][0])) -
            math.sqrt(2/math.pi))
        + model_params[8][8] * (model_resid[8][2]/math.sqrt(model_cond_var[8][2]))
        + model_params[8][9] * (model_resid[8][1]/math.sqrt(model_cond_var[8][1]))
        + model_params[8][10] * (model_resid[8][0]/math.sqrt(model_cond_var[8][0]))
        + model_params[8][11] * math.log(model_cond_var[8][2])
        + model_params[8][12] * math.log(model_cond_var[8][1])
        + model_params[8][13] * math.log(model_cond_var[8][0]))


    return math.exp(cond_var), best_model

# Loop for forecasting IVOL with 9 different models
test_size = 60 # minimimum of 60 months required
exp_ivol = {}
model_used = {}
for company in companies:
    exp_ivol[company] = []
    model_used[company] = []
    dfC = dfIVOL[dfIVOL['ISIN'] == company]
    for i in range(0, len(dfC['Excess_Return']) - test_size):
        X = dfC[['MKT', 'SMB' ,'HML']].values[:(test_size + i)]
        # X = dfC['MKT'].values[:(test_size + i)] for CAPM mean process
        y = dfC['Excess_Return'].values[:(test_size + i)]
        ivol, model = idiosyncratic_forecast(y,X)
        exp_ivol[company].append(ivol)
        model_used[company].append(model)


# Count how many times each model has been used
model_used_total = []
for company in companies:
    model_used_total.extend(model_used[company])

model_0 = model_used_total.count(0)
model_1 = model_used_total.count(1)
model_2 = model_used_total.count(2)
model_3 = model_used_total.count(3)
model_4 = model_used_total.count(4)
model_5 = model_used_total.count(5)
model_6 = model_used_total.count(6)
model_7 = model_used_total.count(7)
model_8 = model_used_total.count(8)

model_total = len(model_used_total)

iv = []
eiv = []
for company in companies:
    iv.extend(std_devs[company][60:])
    eiv.extend(np.sqrt(exp_ivol[company]))
iv = np.array(iv)
eiv = np.array(eiv)


# 4 ----Making the dataframe----

# Dates
dates = dates[60:]
Alldates = dates * len(companies)

# Companies
Allcompanies = []
for company in companies:
    l = []
    l.append(company)
    Allcompanies.extend(l * len(dates))
Allcompanies = np.array(Allcompanies)

# Import Market Capitalization and Market-to-Book ratio's
dfMonthlyMV = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\
    Data\Monthly\Monthly Data without financials.xlsx', sheet_name=2)
dfMonthlyMtoB = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\
    Data\Monthly\Monthly Data without financials.xlsx', sheet_name=3)

prices = dfMonthlyPrices.values.tolist()
MV = dfMonthlyMV.values.tolist()
MtoB = dfMonthlyMtoB.values.tolist()

for i in range(len(companies)):
    prices[i].remove(companies[i])
    MV[i].remove(companies[i])
    MtoB[i].remove(companies[i])

MV_correct = []
MtoB_correct = []
for i in range(len(companies)):
    MV_correct.extend(MV[i][60:])
    MtoB_correct.extend(MtoB[i][60:])


# Turning MV to ln(MV)
ln_MV_total = []
for val in MV_correct:
    ln_MV_total.extend([math.log(val)])

# Turning M/B to ln(B/M)
ln_BtoM_total = []
for val in MtoB_correct:
    ln_BtoM_total.extend([math.log(1/val)])

# Deriving returns
Returns = []

for row in prices:
    for i in range(len(row[60:])):
        try:
            Returns.extend([math.log(row[60+i+1]/row[60+i])])
        except ZeroDivisionError:
            Returns.extend([0])
        except IndexError:
            break

Returns = np.array(Returns) * 100

# Constructing dataframe
dfMonthly = pd.DataFrame({
    'Date': Alldates,
    'ISIN': Allcompanies,
    'Return': Returns,
    'IVOL': iv,
    'Exp_IVOL': eiv,
    'Ln_MV': ln_MV_total,
    'Ln_BM': ln_BtoM_total,
    })
```

```
# 5 ____Export to .dta file____
dfMonthly.to_stata(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\
    Monthly\Monthly Data Test Final without fincials.dta')
```

Calculating ***BETA***s

```
import pandas as pd
import numpy as np
from itertools import repeat
from arch.univariate import LS
import math
from scipy.ndimage.interpolation import shift

dfFF3factorsMonthly = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\
    Thesis\Data\Monthly\ff 3 monthly.xlsx')
dfMonthlyPrices = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\
    Thesis\Data\Monthly\Monthly Data without financials.xlsx', sheet_name=0)
dfMonthlyMV = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\
    Data\Monthly\Monthly Data without financials.xlsx', sheet_name=2)

companies = dfMonthlyPrices['ISIN CODE'].unique()

MKT = dfFF3factorsMonthly['Mkt-RF'].values.tolist()[1:]
MKT = MKT * len(companies)
MKT = np.array(MKT)
MKT = MKT/100

prices = dfMonthlyPrices.values.tolist()
MV = dfMonthlyMV.values.tolist()

for i in range(len(companies)):
    prices[i].remove(companies[i])
    MV[i].remove(companies[i])

MV_correct = []
for i in range(len(companies)):
    MV_correct.extend(MV[i][60:])

# Deriving returns
Returns = []

for row in prices:
    for i in range(len(row)):
        try:
            Returns.extend([math.log(row[i+1]/row[i])])
        except ZeroDivisionError:
            Returns.extend([0])
        except IndexError:
            break

Returns = np.array(Returns)

dates = dfMonthlyPrices.columns.values.tolist()
dates = dates[2:]
Alldates = dates * len(companies)
```

```
Allcompanies = []
for company in companies:
    l = []
    l.append(company)
    Allcompanies.extend(l * len(dates))
Allcompanies = np.array(Allcompanies)

# Deriving returns
Returns = []

for row in prices:
    for i in range(len(row)):
        try:
            Returns.extend([math.log(row[i+1]/row[i])])
        except ZeroDivisionError:
            Returns.extend([0])
        except IndexError:
            break

Returns = np.array(Returns)

dfBETA = pd.DataFrame({
    'Date': Alldates,
    'ISIN': Allcompanies,
    'Return': Returns,
    'MKT':MKT,
    })

betas = {}
prior = 60
for company in companies:
    betas[company] = []
    dfC = dfBETA[dfBETA['ISIN'] == company]
    for i in range(0, len(dfC['Return']) - prior):
        x = np.array(dfC['MKT'].values[i:(prior + i)])*100
        y = np.array(dfC['Return'].values[i:(prior + i)])*100
        ls = LS(y, x)
        res = ls.fit()
        betas[company].append(res._params[1])

## ---------------------------------------------
## Portfolio constructing, monthly rebalancing
## ---------------------------------------------

MKT = dfFF3factorsMonthly['Mkt-RF'].values.tolist()[61:]
MKT = MKT * len(companies)
MKT = np.array(MKT)
MKT = MKT/100

prices = dfMonthlyPrices.values.tolist()

for i in range(len(companies)):
    prices[i].remove(companies[i])

dates = dfMonthlyPrices.columns.values.tolist()
dates = dates[62:]
Alldates = dates * len(companies)

Allcompanies = []
```

```
for company in companies:
    l = []
    l.append(company)
    Allcompanies.extend(l * len(dates))
Allcompanies = np.array(Allcompanies)

# Deriving returns
Returns = []

for row in prices:
    for i in range(len(row[60:])):
        try:
            Returns.extend([math.log(row[60+i+1]/row[60+i])])
        except ZeroDivisionError:
            Returns.extend([0])
        except IndexError:
            break

Returns = np.array(Returns)

# begint met de laatste in companies
betasFF = []
for company in companies:
    betasFF.extend(betas[company])

all_data = pd.DataFrame({
    'ISIN': Allcompanies,
    'Return': Returns,
    'BETA_FF':betasFF,
    'MV':MV_correct
    }, index = Alldates)

def vw_weighted_ret(portfolio, factor):
    weight = portfolio[factor]/np.sum(portfolio[factor])
    vw_ret = weight * portfolio['Return']
    return np.sum(vw_ret)

monthly_portfolios_comp = {}
monthly_portfolios_ret = {}
Market_Ret = []
for date in dates:
    first = all_data.loc[date]
    sort_size = first.sort_values('MV')
    monthly_portfolios_comp[date] = [[] for x in repeat(None, 100)]
    monthly_portfolios_ret[date] = [[] for x in repeat(None, 100)]

    # 10 MV
    sort_low = sort_size.iloc[:37]
    sort_2 = sort_size.iloc[37:74]
    sort_3 = sort_size.iloc[74:111]
    sort_4 = sort_size.iloc[111:148]
    sort_5 = sort_size.iloc[148:183]
    sort_6 = sort_size.iloc[183:218]
    sort_7 = sort_size.iloc[218:255]
    sort_8 = sort_size.iloc[255:292]
    sort_9 = sort_size.iloc[292:329]
    sort_high = sort_size.iloc[329:]

    # MV low, betas low —> high
```

```
sort_low_beta = sort_low.sort_values('BETA_FF') # 61

sort_low_low = sort_low_beta.iloc[:math.ceil(len(sort_low_beta)/10)]
monthly_portfolios_comp[date][0].append(sort_low_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][0].append(sort_low_low['Return'].values.tolist())

sort_low_2 = sort_low_beta.iloc[math.ceil(len(sort_low_beta)/10):round(len(
    sort_low_beta)/10)*2 + 1]
monthly_portfolios_comp[date][1].append(sort_low_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][1].append(sort_low_2['Return'].values.tolist())

sort_low_3 = sort_low_beta.iloc[round(len(sort_low_beta)/10)*2 + 1:round(len(
    sort_low_beta)/10)*3 + 1]
monthly_portfolios_comp[date][2].append(sort_low_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][2].append(sort_low_3['Return'].values.tolist())

sort_low_4 = sort_low_beta.iloc[round(len(sort_low_beta)/10)*3 + 1:round(len(
    sort_low_beta)/10)*4 + 1]
monthly_portfolios_comp[date][3].append(sort_low_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][3].append(sort_low_4['Return'].values.tolist())

sort_low_5 = sort_low_beta.iloc[round(len(sort_low_beta)/10)*4 + 1:round(len(
    sort_low_beta)/10)*5 + 1]
monthly_portfolios_comp[date][4].append(sort_low_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][4].append(sort_low_5['Return'].values.tolist())

sort_low_6 = sort_low_beta.iloc[round(len(sort_low_beta)/10)*5 + 1:round(len(
    sort_low_beta)/10)*6 + 1]
monthly_portfolios_comp[date][5].append(sort_low_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][5].append(sort_low_6['Return'].values.tolist())

sort_low_7 = sort_low_beta.iloc[round(len(sort_low_beta)/10)*6 + 1:round(len(
    sort_low_beta)/10)*7 + 1]
monthly_portfolios_comp[date][6].append(sort_low_7['ISIN'].values.tolist())
monthly_portfolios_ret[date][6].append(sort_low_7['Return'].values.tolist())

sort_low_8 = sort_low_beta.iloc[round(len(sort_low_beta)/10)*7 + 1:round(len(
    sort_low_beta)/10)*8 + 1]
monthly_portfolios_comp[date][7].append(sort_low_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][7].append(sort_low_8['Return'].values.tolist())

sort_low_9 = sort_low_beta.iloc[round(len(sort_low_beta)/10)*8 + 1:round(len(
    sort_low_beta)/10)*9 + 1]
monthly_portfolios_comp[date][8].append(sort_low_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][8].append(sort_low_9['Return'].values.tolist())

sort_low_high = sort_low_beta.iloc[-round(len(sort_low_beta)/10):]
monthly_portfolios_comp[date][9].append(sort_low_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][9].append(sort_low_high['Return'].values.tolist())

# MV 2, betas low --> high
sort_2_beta = sort_2.sort_values('BETA_FF') # 61

sort_2_low = sort_2_beta.iloc[:math.ceil(len(sort_2_beta)/10)]
monthly_portfolios_comp[date][10].append(sort_2_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][10].append(sort_2_low['Return'].values.tolist())

sort_2_2 = sort_2_beta.iloc[math.ceil(len(sort_2_beta)/10):round(len(sort_2_beta)/10)
    *2 + 1]
```

```
monthly_portfolios_comp[date][11].append(sort_2_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][11].append(sort_2_2['Return'].values.tolist())

sort_2_3 = sort_2_beta.iloc[round(len(sort_2_beta)/10)*2 + 1:round(len(sort_2_beta)
    /10)*3 + 1]
monthly_portfolios_comp[date][12].append(sort_2_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][12].append(sort_2_3['Return'].values.tolist())

sort_2_4 = sort_2_beta.iloc[round(len(sort_2_beta)/10)*3 + 1:round(len(sort_2_beta)
    /10)*4 + 1]
monthly_portfolios_comp[date][13].append(sort_2_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][13].append(sort_2_4['Return'].values.tolist())

sort_2_5 = sort_2_beta.iloc[round(len(sort_2_beta)/10)*4 + 1:round(len(sort_2_beta)
    /10)*5 + 1]
monthly_portfolios_comp[date][14].append(sort_2_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][14].append(sort_2_5['Return'].values.tolist())

sort_2_6 = sort_2_beta.iloc[round(len(sort_2_beta)/10)*5 + 1:round(len(sort_2_beta)
    /10)*6 + 1]
monthly_portfolios_comp[date][15].append(sort_2_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][15].append(sort_2_6['Return'].values.tolist())

sort_2_7 = sort_2_beta.iloc[round(len(sort_2_beta)/10)*6 + 1:round(len(sort_2_beta)
    /10)*7 + 1]
monthly_portfolios_comp[date][16].append(sort_2_7['ISIN'].values.tolist())
monthly_portfolios_ret[date][16].append(sort_2_7['Return'].values.tolist())

sort_2_8 = sort_2_beta.iloc[round(len(sort_2_beta)/10)*7 + 1:round(len(sort_2_beta)
    /10)*8 + 1]
monthly_portfolios_comp[date][17].append(sort_2_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][17].append(sort_2_8['Return'].values.tolist())

sort_2_9 = sort_2_beta.iloc[round(len(sort_2_beta)/10)*8 + 1:round(len(sort_2_beta)
    /10)*9 + 1]
monthly_portfolios_comp[date][18].append(sort_2_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][18].append(sort_2_9['Return'].values.tolist())

sort_2_high = sort_2_beta.iloc[-round(len(sort_2_beta)/10):]
monthly_portfolios_comp[date][19].append(sort_2_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][19].append(sort_2_high['Return'].values.tolist())

# MV 3, betas low --> high
sort_3_beta = sort_3.sort_values('BETA_FF')

sort_3_low = sort_3_beta.iloc[:round(len(sort_3_beta)/10)]
monthly_portfolios_comp[date][20].append(sort_3_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][20].append(sort_3_low['Return'].values.tolist())

sort_3_2 = sort_3_beta.iloc[round(len(sort_3_beta)/10):round(len(sort_3_beta)/10)*2]
monthly_portfolios_comp[date][21].append(sort_3_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][21].append(sort_3_2['Return'].values.tolist())

sort_3_3 = sort_3_beta.iloc[round(len(sort_3_beta)/10)*2:round(len(sort_3_beta)/10)
    *3]
monthly_portfolios_comp[date][22].append(sort_3_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][22].append(sort_3_3['Return'].values.tolist())
```

```
sort_3_4 = sort_3_beta.iloc[round(len(sort_3_beta)/10)*3:round(len(sort_3_beta)/10)
    *4]
monthly_portfolios_comp[date][23].append(sort_3_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][23].append(sort_3_4['Return'].values.tolist())

sort_3_5 = sort_3_beta.iloc[round(len(sort_3_beta)/10)*4:round(len(sort_3_beta)/10)
    *5]
monthly_portfolios_comp[date][24].append(sort_3_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][24].append(sort_3_5['Return'].values.tolist())

sort_3_6 = sort_3_beta.iloc[round(len(sort_3_beta)/10)*5:round(len(sort_3_beta)/10)
    *6]
monthly_portfolios_comp[date][25].append(sort_3_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][25].append(sort_3_6['Return'].values.tolist())

sort_3_7 = sort_3_beta.iloc[round(len(sort_3_beta)/10)*6:round(len(sort_3_beta)/10)
    *7]
monthly_portfolios_comp[date][26].append(sort_3_7['ISIN'].values.tolist())
monthly_portfolios_ret[date][26].append(sort_3_7['Return'].values.tolist())

sort_3_8 = sort_3_beta.iloc[round(len(sort_3_beta)/10)*7:round(len(sort_3_beta)/10)
    *8]
monthly_portfolios_comp[date][27].append(sort_3_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][27].append(sort_3_8['Return'].values.tolist())

sort_3_9 = sort_3_beta.iloc[round(len(sort_3_beta)/10)*8:round(len(sort_3_beta)/10)
    *9]
monthly_portfolios_comp[date][28].append(sort_3_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][28].append(sort_3_9['Return'].values.tolist())

sort_3_high = sort_3_beta.iloc[-round(len(sort_3_beta)/10):]
monthly_portfolios_comp[date][29].append(sort_3_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][29].append(sort_3_high['Return'].values.tolist())

# MV 4, betas low —> high
sort_4_beta = sort_4.sort_values('BETA_FF')

sort_4_low = sort_4_beta.iloc[:round(len(sort_4_beta)/10)]
monthly_portfolios_comp[date][30].append(sort_4_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][30].append(sort_4_low['Return'].values.tolist())

sort_4_2 = sort_4_beta.iloc[round(len(sort_4_beta)/10):round(len(sort_4_beta)/10)*2]
monthly_portfolios_comp[date][31].append(sort_4_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][31].append(sort_4_2['Return'].values.tolist())

sort_4_3 = sort_4_beta.iloc[round(len(sort_4_beta)/10)*2:round(len(sort_4_beta)/10)
    *3]
monthly_portfolios_comp[date][32].append(sort_4_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][32].append(sort_4_3['Return'].values.tolist())

sort_4_4 = sort_4_beta.iloc[round(len(sort_4_beta)/10)*3:round(len(sort_4_beta)/10)
    *4]
monthly_portfolios_comp[date][33].append(sort_4_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][33].append(sort_4_4['Return'].values.tolist())

sort_4_5 = sort_4_beta.iloc[round(len(sort_4_beta)/10)*4:round(len(sort_4_beta)/10)
    *5]
monthly_portfolios_comp[date][34].append(sort_4_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][34].append(sort_4_5['Return'].values.tolist())
```

```
sort_4_6 = sort_4_beta.iloc[round(len(sort_4_beta)/10)*5:round(len(sort_4_beta)/10)
    *6]
monthly_portfolios_comp[date][35].append(sort_4_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][35].append(sort_4_6['Return'].values.tolist())

sort_4_7 = sort_4_beta.iloc[round(len(sort_4_beta)/10)*6:round(len(sort_4_beta)/10)
    *7]
monthly_portfolios_comp[date][36].append(sort_4_7['ISIN'].values.tolist())
monthly_portfolios_ret[date][36].append(sort_4_7['Return'].values.tolist())

sort_4_8 = sort_4_beta.iloc[round(len(sort_4_beta)/10)*7:round(len(sort_4_beta)/10)
    *8]
monthly_portfolios_comp[date][37].append(sort_4_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][37].append(sort_4_8['Return'].values.tolist())

sort_4_9 = sort_4_beta.iloc[round(len(sort_4_beta)/10)*8:round(len(sort_4_beta)/10)
    *9]
monthly_portfolios_comp[date][38].append(sort_4_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][38].append(sort_4_9['Return'].values.tolist())

sort_4_high = sort_4_beta.iloc[-round(len(sort_4_beta)/10):]
monthly_portfolios_comp[date][39].append(sort_4_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][39].append(sort_4_high['Return'].values.tolist())

# MV 5, betas low --> high
sort_5_beta = sort_5.sort_values('BETA_FF')

sort_5_low = sort_5_beta.iloc[:round(len(sort_5_beta)/10)]
monthly_portfolios_comp[date][40].append(sort_5_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][40].append(sort_5_low['Return'].values.tolist())

sort_5_2 = sort_5_beta.iloc[round(len(sort_5_beta)/10):round(len(sort_5_beta)/10)*2]
monthly_portfolios_comp[date][41].append(sort_5_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][41].append(sort_5_2['Return'].values.tolist())

sort_5_3 = sort_5_beta.iloc[round(len(sort_5_beta)/10)*2:round(len(sort_5_beta)/10)
    *3]
monthly_portfolios_comp[date][42].append(sort_5_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][42].append(sort_5_3['Return'].values.tolist())

sort_5_4 = sort_5_beta.iloc[round(len(sort_5_beta)/10)*3:round(len(sort_5_beta)/10)
    *4]
monthly_portfolios_comp[date][43].append(sort_5_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][43].append(sort_5_4['Return'].values.tolist())

sort_5_5 = sort_5_beta.iloc[round(len(sort_5_beta)/10)*4:round(len(sort_5_beta)/10)
    *5]
monthly_portfolios_comp[date][44].append(sort_5_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][44].append(sort_5_5['Return'].values.tolist())

sort_5_6 = sort_5_beta.iloc[round(len(sort_5_beta)/10)*5:round(len(sort_5_beta)/10)
    *6]
monthly_portfolios_comp[date][45].append(sort_5_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][45].append(sort_5_6['Return'].values.tolist())

sort_5_7 = sort_5_beta.iloc[round(len(sort_5_beta)/10)*6:round(len(sort_5_beta)/10)
    *7]
monthly_portfolios_comp[date][46].append(sort_5_7['ISIN'].values.tolist())
```

```
monthly_portfolios_ret[date][46].append(sort_5_7['Return'].values.tolist())

sort_5_8 = sort_5_beta.iloc[round(len(sort_5_beta)/10)*7:round(len(sort_5_beta)/10)
    *8]
monthly_portfolios_comp[date][47].append(sort_5_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][47].append(sort_5_8['Return'].values.tolist())

sort_5_9 = sort_5_beta.iloc[round(len(sort_5_beta)/10)*8:round(len(sort_5_beta)/10)
    *9]
monthly_portfolios_comp[date][48].append(sort_5_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][48].append(sort_5_9['Return'].values.tolist())

sort_5_high = sort_5_beta.iloc[-round(len(sort_5_beta)/10):]
monthly_portfolios_comp[date][49].append(sort_5_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][49].append(sort_5_high['Return'].values.tolist())

# MV 6, betas low --> high
sort_6_beta = sort_6.sort_values('BETA_FF')

sort_6_low = sort_6_beta.iloc[:round(len(sort_6_beta)/10)]
monthly_portfolios_comp[date][50].append(sort_6_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][50].append(sort_6_low['Return'].values.tolist())

sort_6_2 = sort_6_beta.iloc[round(len(sort_6_beta)/10):round(len(sort_6_beta)/10)*2]
monthly_portfolios_comp[date][51].append(sort_6_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][51].append(sort_6_2['Return'].values.tolist())

sort_6_3 = sort_6_beta.iloc[round(len(sort_6_beta)/10)*2:round(len(sort_6_beta)/10)
    *3]
monthly_portfolios_comp[date][52].append(sort_6_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][52].append(sort_6_3['Return'].values.tolist())

sort_6_4 = sort_6_beta.iloc[round(len(sort_6_beta)/10)*3:round(len(sort_6_beta)/10)
    *4]
monthly_portfolios_comp[date][53].append(sort_6_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][53].append(sort_6_4['Return'].values.tolist())

sort_6_5 = sort_6_beta.iloc[round(len(sort_6_beta)/10)*4:round(len(sort_6_beta)/10)
    *5]
monthly_portfolios_comp[date][54].append(sort_6_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][54].append(sort_6_5['Return'].values.tolist())

sort_6_6 = sort_6_beta.iloc[round(len(sort_6_beta)/10)*5:round(len(sort_6_beta)/10)
    *6]
monthly_portfolios_comp[date][55].append(sort_6_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][55].append(sort_6_6['Return'].values.tolist())

sort_6_7 = sort_6_beta.iloc[round(len(sort_6_beta)/10)*6:round(len(sort_6_beta)/10)
    *7]
monthly_portfolios_comp[date][56].append(sort_6_7['ISIN'].values.tolist())
monthly_portfolios_ret[date][56].append(sort_6_7['Return'].values.tolist())

sort_6_8 = sort_6_beta.iloc[round(len(sort_6_beta)/10)*7:round(len(sort_6_beta)/10)
    *8]
monthly_portfolios_comp[date][57].append(sort_6_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][57].append(sort_6_8['Return'].values.tolist())

sort_6_9 = sort_6_beta.iloc[round(len(sort_6_beta)/10)*8:round(len(sort_6_beta)/10)
    *9]
```

```
monthly_portfolios_comp[date][58].append(sort_6_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][58].append(sort_6_9['Return'].values.tolist())

sort_6_high = sort_6_beta.iloc[-round(len(sort_6_beta)/10):]
monthly_portfolios_comp[date][59].append(sort_6_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][59].append(sort_6_high['Return'].values.tolist())

# MV 7, betas low --> high
sort_7_beta = sort_7.sort_values('BETA_FF')

sort_7_low = sort_7_beta.iloc[:round(len(sort_7_beta)/10)]
monthly_portfolios_comp[date][60].append(sort_7_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][60].append(sort_7_low['Return'].values.tolist())

sort_7_2 = sort_7_beta.iloc[round(len(sort_7_beta)/10):round(len(sort_7_beta)/10)*2]
monthly_portfolios_comp[date][61].append(sort_7_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][61].append(sort_7_2['Return'].values.tolist())

sort_7_3 = sort_7_beta.iloc[round(len(sort_7_beta)/10)*2:round(len(sort_7_beta)/10)
    *3]
monthly_portfolios_comp[date][62].append(sort_7_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][62].append(sort_7_3['Return'].values.tolist())

sort_7_4 = sort_7_beta.iloc[round(len(sort_7_beta)/10)*3:round(len(sort_7_beta)/10)
    *4]
monthly_portfolios_comp[date][63].append(sort_7_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][63].append(sort_7_4['Return'].values.tolist())

sort_7_5 = sort_7_beta.iloc[round(len(sort_7_beta)/10)*4:round(len(sort_7_beta)/10)
    *5]
monthly_portfolios_comp[date][64].append(sort_7_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][64].append(sort_7_5['Return'].values.tolist())

sort_7_6 = sort_7_beta.iloc[round(len(sort_7_beta)/10)*5:round(len(sort_7_beta)/10)
    *6]
monthly_portfolios_comp[date][65].append(sort_7_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][65].append(sort_7_6['Return'].values.tolist())

sort_7_7 = sort_7_beta.iloc[round(len(sort_7_beta)/10)*6:round(len(sort_7_beta)/10)
    *7]
monthly_portfolios_comp[date][66].append(sort_7_7['ISIN'].values.tolist())
monthly_portfolios_ret[date][66].append(sort_7_low['Return'].values.tolist())

sort_7_8 = sort_7_beta.iloc[round(len(sort_7_beta)/10)*7:round(len(sort_7_beta)/10)
    *8]
monthly_portfolios_comp[date][67].append(sort_7_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][67].append(sort_7_8['Return'].values.tolist())

sort_7_9 = sort_7_beta.iloc[round(len(sort_7_beta)/10)*8:round(len(sort_7_beta)/10)
    *9]
monthly_portfolios_comp[date][68].append(sort_7_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][68].append(sort_7_9['Return'].values.tolist())

sort_7_high = sort_7_beta.iloc[-round(len(sort_7_beta)/10):]
monthly_portfolios_comp[date][69].append(sort_7_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][69].append(sort_7_high['Return'].values.tolist())

# MV 8, betas low --> high
sort_8_beta = sort_8.sort_values('BETA_FF')
```

```
sort_8_low = sort_8_beta.iloc[:round(len(sort_8_beta)/10)]
monthly_portfolios_comp[date][70].append(sort_8_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][70].append(sort_8_low['Return'].values.tolist())

sort_8_2 = sort_8_beta.iloc[round(len(sort_8_beta)/10):round(len(sort_8_beta)/10)*2]
monthly_portfolios_comp[date][71].append(sort_8_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][71].append(sort_8_2['Return'].values.tolist())

sort_8_3 = sort_8_beta.iloc[round(len(sort_8_beta)/10)*2:round(len(sort_8_beta)/10)
    *3]
monthly_portfolios_comp[date][72].append(sort_8_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][72].append(sort_8_3['Return'].values.tolist())

sort_8_4 = sort_8_beta.iloc[round(len(sort_8_beta)/10)*3:round(len(sort_8_beta)/10)
    *4]
monthly_portfolios_comp[date][73].append(sort_8_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][73].append(sort_8_4['Return'].values.tolist())

sort_8_5 = sort_8_beta.iloc[round(len(sort_8_beta)/10)*4:round(len(sort_8_beta)/10)
    *5]
monthly_portfolios_comp[date][74].append(sort_8_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][74].append(sort_8_5['Return'].values.tolist())

sort_8_6 = sort_8_beta.iloc[round(len(sort_8_beta)/10)*5:round(len(sort_8_beta)/10)
    *6]
monthly_portfolios_comp[date][75].append(sort_8_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][75].append(sort_8_6['Return'].values.tolist())

sort_8_7 = sort_8_beta.iloc[round(len(sort_8_beta)/10)*6:round(len(sort_8_beta)/10)
    *7]
monthly_portfolios_comp[date][76].append(sort_8_7['ISIN'].values.tolist())
monthly_portfolios_ret[date][76].append(sort_8_7['Return'].values.tolist())

sort_8_8 = sort_8_beta.iloc[round(len(sort_8_beta)/10)*7:round(len(sort_8_beta)/10)
    *8]
monthly_portfolios_comp[date][77].append(sort_8_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][77].append(sort_8_8['Return'].values.tolist())

sort_8_9 = sort_8_beta.iloc[round(len(sort_8_beta)/10)*8:round(len(sort_8_beta)/10)
    *9]
monthly_portfolios_comp[date][78].append(sort_8_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][78].append(sort_8_9['Return'].values.tolist())

sort_8_high = sort_8_beta.iloc[-round(len(sort_8_beta)/10):]
monthly_portfolios_comp[date][79].append(sort_8_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][79].append(sort_8_high['Return'].values.tolist())

# MV 9, betas low --> high
sort_9_beta = sort_9.sort_values('BETA_FF') # 61

sort_9_low = sort_9_beta.iloc[:math.ceil(len(sort_9_beta)/10)]
monthly_portfolios_comp[date][80].append(sort_9_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][80].append(sort_9_low['Return'].values.tolist())

sort_9_2 = sort_9_beta.iloc[math.ceil(len(sort_9_beta)/10):round(len(sort_9_beta)/10)
    *2 + 1]
monthly_portfolios_comp[date][81].append(sort_9_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][81].append(sort_9_2['Return'].values.tolist())
```

```
sort_9_3 = sort_9_beta.iloc[round(len(sort_9_beta)/10)*2 + 1:round(len(sort_9_beta)
    /10)*3 + 1]
monthly_portfolios_comp[date][82].append(sort_9_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][82].append(sort_9_3['Return'].values.tolist())

sort_9_4 = sort_9_beta.iloc[round(len(sort_9_beta)/10)*3 + 1:round(len(sort_9_beta)
    /10)*4 + 1]
monthly_portfolios_comp[date][83].append(sort_9_4['ISIN'].values.tolist())
monthly_portfolios_ret[date][83].append(sort_9_4['Return'].values.tolist())

sort_9_5 = sort_9_beta.iloc[round(len(sort_9_beta)/10)*4 + 1:round(len(sort_9_beta)
    /10)*5 + 1]
monthly_portfolios_comp[date][84].append(sort_9_5['ISIN'].values.tolist())
monthly_portfolios_ret[date][84].append(sort_9_5['Return'].values.tolist())

sort_9_6 = sort_9_beta.iloc[round(len(sort_9_beta)/10)*5 + 1:round(len(sort_9_beta)
    /10)*6 + 1]
monthly_portfolios_comp[date][85].append(sort_9_6['ISIN'].values.tolist())
monthly_portfolios_ret[date][85].append(sort_9_6['Return'].values.tolist())

sort_9_7 = sort_9_beta.iloc[round(len(sort_9_beta)/10)*6 + 1:round(len(sort_9_beta)
    /10)*7 + 1]
monthly_portfolios_comp[date][86].append(sort_9_7['ISIN'].values.tolist())
monthly_portfolios_ret[date][86].append(sort_9_7['Return'].values.tolist())

sort_9_8 = sort_9_beta.iloc[round(len(sort_9_beta)/10)*7 + 1:round(len(sort_9_beta)
    /10)*8 + 1]
monthly_portfolios_comp[date][87].append(sort_9_8['ISIN'].values.tolist())
monthly_portfolios_ret[date][87].append(sort_9_8['Return'].values.tolist())

sort_9_9 = sort_9_beta.iloc[round(len(sort_9_beta)/10)*8 + 1:round(len(sort_9_beta)
    /10)*9 + 1]
monthly_portfolios_comp[date][88].append(sort_9_9['ISIN'].values.tolist())
monthly_portfolios_ret[date][88].append(sort_9_9['Return'].values.tolist())

sort_9_high = sort_9_beta.iloc[-round(len(sort_9_beta)/10):]
monthly_portfolios_comp[date][89].append(sort_9_high['ISIN'].values.tolist())
monthly_portfolios_ret[date][89].append(sort_9_high['Return'].values.tolist())

# MV high, betas low --> high
sort_high_beta = sort_high.sort_values('BETA_FF')  # 61

sort_high_low = sort_high_beta.iloc[:math.ceil(len(sort_high_beta)/10)]
monthly_portfolios_comp[date][90].append(sort_high_low['ISIN'].values.tolist())
monthly_portfolios_ret[date][90].append(sort_high_low['Return'].values.tolist())

sort_high_2 = sort_high_beta.iloc[math.ceil(len(sort_high_beta)/10):round(len(
    sort_high_beta)/10)*2 + 1]
monthly_portfolios_comp[date][91].append(sort_high_2['ISIN'].values.tolist())
monthly_portfolios_ret[date][91].append(sort_high_2['Return'].values.tolist())

sort_high_3 = sort_high_beta.iloc[round(len(sort_high_beta)/10)*2 + 1:round(len(
    sort_high_beta)/10)*3 + 1]
monthly_portfolios_comp[date][92].append(sort_high_3['ISIN'].values.tolist())
monthly_portfolios_ret[date][92].append(sort_high_3['Return'].values.tolist())

sort_high_4 = sort_high_beta.iloc[round(len(sort_high_beta)/10)*3 + 1:round(len(
    sort_high_beta)/10)*4 + 1]
    monthly_portfolios_comp[date][93].append(sort_high_4['ISIN'].values.tolist())
    monthly_portfolios_ret[date][93].append(sort_high_4['Return'].values.tolist())

    sort_high_5 = sort_high_beta.iloc[round(len(sort_high_beta)/10)*4 + 1:round(len(
        sort_high_beta)/10)*5 + 1]
    monthly_portfolios_comp[date][94].append(sort_high_5['ISIN'].values.tolist())
    monthly_portfolios_ret[date][94].append(sort_high_5['Return'].values.tolist())

    sort_high_6 = sort_high_beta.iloc[round(len(sort_high_beta)/10)*5 + 1:round(len(
        sort_high_beta)/10)*6 + 1]
    monthly_portfolios_comp[date][95].append(sort_high_6['ISIN'].values.tolist())
    monthly_portfolios_ret[date][95].append(sort_high_6['Return'].values.tolist())

    sort_high_7 = sort_high_beta.iloc[round(len(sort_high_beta)/10)*6 + 1:round(len(
        sort_high_beta)/10)*7 + 1]
    monthly_portfolios_comp[date][96].append(sort_high_7['ISIN'].values.tolist())
    monthly_portfolios_ret[date][96].append(sort_high_7['Return'].values.tolist())

    sort_high_8 = sort_high_beta.iloc[round(len(sort_high_beta)/10)*7 + 1:round(len(
        sort_high_beta)/10)*8 + 1]
    monthly_portfolios_comp[date][97].append(sort_high_8['ISIN'].values.tolist())
    monthly_portfolios_ret[date][97].append(sort_high_8['Return'].values.tolist())

    sort_high_9 = sort_high_beta.iloc[round(len(sort_high_beta)/10)*8 + 1:round(len(
        sort_high_beta)/10)*9 + 1]
    monthly_portfolios_comp[date][98].append(sort_high_9['ISIN'].values.tolist())
    monthly_portfolios_ret[date][98].append(sort_high_9['Return'].values.tolist())

    sort_high_high = sort_high_beta.iloc[-round(len(sort_high_beta)/10):]
    monthly_portfolios_comp[date][99].append(sort_high_high['ISIN'].values.tolist())
    monthly_portfolios_ret[date][99].append(sort_high_high['Return'].values.tolist())

    market_ret = vw_weighted_ret(first, 'MV')
    Market_Ret.append(market_ret)


portfolio_returns = [[] for x in repeat(None, 100)]
for i in range(0, 100):
    for date in dates:
        portfolio_returns[i].append(np.average(monthly_portfolios_ret[date][i]))

Market_Ret = np.array(Market_Ret)
Market_Ret_lag = shift(Market_Ret, 1, cval=0)

portfolio_betas = []
for i in range(0, len(portfolio_returns)):
    y = np.array(portfolio_returns[i]) * 100
    x = np.stack((Market_Ret, Market_Ret_lag), axis=-1) * 100
    ls = LS(y,x)
    res = ls.fit()
    beta = res._params[1] + res._params[2]
    portfolio_betas.append(beta)

## assigning betas to firms on monthly basis
portfolio_index = {}
for company in companies:
    portfolio_index[company] = []
    for date in dates:
        count = 0
        for l in range(0, len(monthly_portfolios_comp[date])):
            try:
                monthly_portfolios_comp[date][l][0].index(company)
                portfolio_index[company].append(count)
                break
            except ValueError:
                count += 1

company_betas = {}
for company in companies:
    company_betas[company] = []
    for i in range(0, len(dates)):
        company_betas[company].append(portfolio_betas[portfolio_index[company][i]])


## Making dataframe to stata
betasFF = []
for company in companies:
    betasFF.extend(company_betas[company])

Allcompanies = []
for company in companies:
    l = []
    l.append(company)
    Allcompanies.extend(l * len(dates))
Allcompanies = np.array(Allcompanies)

Alldates = dates * len(companies)

dfBETA = pd.DataFrame({
    'Date': Alldates,
    'ISIN': Allcompanies,
    'BETA_FF':betasFF,
    })

dfBETA.to_stata(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\Monthly\
    Monthly BETA FF without financials.dta')
```

Portfolio construction for both E($IVOL$) estimates (FF3 and CAPM)

```
import pandas as pd
import numpy as np
from itertools import repeat
from arch.univariate import LS
import matplotlib.pyplot as plt

# ----------------------------------------------
#                FF-3 Factors
# ----------------------------------------------

dfFF3factorsMonthly = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\
    Thesis\Data\Monthly\ff 3 monthly.xlsx')
dfFrame = pd.read_stata(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\
    Monthly\Final 1.dta')
dfFrame = dfFrame.drop(['index',
                        'Ln_MV',
                        'Ln_BM',
                         'company_id',
                         '_merge',
                         'IVOLlag1'], axis=1)

companies = dfFrame['ISIN'].unique()
dates = dfFrame['Date'].unique()
RF = dfFF3factorsMonthly['RF'].values.tolist()[61:] * 366
RF = np.array(RF)
MKT = dfFF3factorsMonthly['Mkt-RF'].values.tolist()[61:]
MKT = np.array(MKT)
SMB = dfFF3factorsMonthly['SMB'].values.tolist()[61:]
SMB = np.array(SMB)
HML = dfFF3factorsMonthly['HML'].values.tolist()[61:]
HML = np.array(HML)

dfFrame['Excess_Return'] = dfFrame['Return'] - RF

def cum_returns(months, excess_returnx):

    cum_ret = [[] for x in repeat(None, months)]
    cum_ret[0] = 1
    for i in range(0, (len(excess_returnx)-1)):
        cum_ret[i+1] = cum_ret[i]*(1 + excess_returnx[i]/100)

    return cum_ret

def vw_weighted_ret(portfolio, factor):
    weight = portfolio[factor]/np.sum(portfolio[factor])
    vw_ret = weight * portfolio['Return']
    return np.sum(vw_ret)

def portfolio(x1, x2):

    MV = []
    BM = []
    BETA = []
    IVOL = []
    Exp_IVOL = []
    excess_return_ew = []
    excess_return_vw = []

    for date in dates:
        dfMonth = dfFrame[dfFrame['Date'] == date]
        dfPortfolio = dfMonth.sort_values('Exp_IVOL', ascending=False)
        dfPortfolio = dfPortfolio.iloc[x1:x2]
        MV_total = np.sum(dfPortfolio['MV'].to_numpy())
        returns = dfPortfolio['Excess_Return'].to_numpy()

        # equal-weighted
        excess_return_ew.append(np.average(dfPortfolio['Excess_Return'].to_numpy()))

        #statistics
        MV.append(np.average(dfPortfolio['MV'].to_numpy()))
        BM.append(np.average(dfPortfolio['BM'].to_numpy()))
        BETA.append(np.average(dfPortfolio['BETA_FF'].to_numpy()))
        IVOL.append(np.average(dfPortfolio['IVOL'].to_numpy()))
        Exp_IVOL.append(np.average(dfPortfolio['Exp_IVOL'].to_numpy()))

        vw_return = []
        for i in range(0, len(dfPortfolio)):
            weight = dfPortfolio['MV'].iloc[i]/MV_total
            vw_return.append(weight * returns[i])
        # value-weighted
        excess_return_vw.append(np.sum(vw_return))

    y = np.array(excess_return_vw)
    x = np.array([MKT, SMB, HML]).T
    ls = LS(y, x)
    res = ls.fit(disp='off')
    coeff = round(res._params[0],3)
    pvalue = round(res.pvalues.Const,3)

    return coeff, pvalue, excess_return_ew, excess_return_vw, cum_returns(len(dates),
        excess_return_ew), cum_returns(len(dates), excess_return_vw), np.average(MV), np.
        average(BM), np.average(BETA), np.average(IVOL), np.average(Exp_IVOL)

coeff1, pvalue1, excess_return1, vw_excess_return1, cum_ret1, vw_cum_ret1, av_MV_1,
    av_BM_1, av_beta_1, av_ivol_1, av_eivol_1 = portfolio(0, 37)
coeff2, pvalue2, excess_return2, vw_excess_return2, cum_ret2, vw_cum_ret2, av_MV_2,
    av_BM_2, av_beta_2, av_ivol_2, av_eivol_2 = portfolio(37, 74)
coeff3, pvalue3, excess_return3, vw_excess_return3, cum_ret3, vw_cum_ret3, av_MV_3,
    av_BM_3, av_beta_3, av_ivol_3, av_eivol_3 = portfolio(74, 111)
coeff4, pvalue4, excess_return4, vw_excess_return4, cum_ret4, vw_cum_ret4, av_MV_4,
    av_BM_4, av_beta_4, av_ivol_4, av_eivol_4 = portfolio(111, 148)
coeff5, pvalue5, excess_return5, vw_excess_return5, cum_ret5, vw_cum_ret5, av_MV_5,
    av_BM_5, av_beta_5, av_ivol_5, av_eivol_5 = portfolio(148, 183)
coeff6, pvalue6, excess_return6, vw_excess_return6, cum_ret6, vw_cum_ret6, av_MV_6,
    av_BM_6, av_beta_6, av_ivol_6, av_eivol_6 = portfolio(183, 218)
coeff7, pvalue7, excess_return7, vw_excess_return7, cum_ret7, vw_cum_ret7, av_MV_7,
    av_BM_7, av_beta_7, av_ivol_7, av_eivol_7 = portfolio(218, 255)
coeff8, pvalue8, excess_return8, vw_excess_return8, cum_ret8, vw_cum_ret8, av_MV_8,
    av_BM_8, av_beta_8, av_ivol_8, av_eivol_8 = portfolio(255, 292)
coeff9, pvalue9, excess_return9, vw_excess_return9, cum_ret9, vw_cum_ret9, av_MV_9,
    av_BM_9, av_beta_9, av_ivol_9, av_eivol_9 = portfolio(292, 329)
coeff10, pvalue10, excess_return10, vw_excess_return10, cum_ret10, vw_cum_ret10, av_MV_10
    , av_BM_10, av_beta_10, av_ivol_10, av_eivol_10 = portfolio(329, 366)

av_excess_return1 = np.average(excess_return1)
av_excess_return2 = np.average(excess_return2)
av_excess_return3 = np.average(excess_return3)
av_excess_return4 = np.average(excess_return4)
av_excess_return5 = np.average(excess_return5)
av_excess_return6 = np.average(excess_return6)
av_excess_return7 = np.average(excess_return7)
av_excess_return8 = np.average(excess_return8)
av_excess_return9 = np.average(excess_return9)
av_excess_return10 = np.average(excess_return10)

av_excess_return1_vw = np.average(vw_excess_return1)
av_excess_return2_vw = np.average(vw_excess_return2)
av_excess_return3_vw = np.average(vw_excess_return3)
av_excess_return4_vw = np.average(vw_excess_return4)
av_excess_return5_vw = np.average(vw_excess_return5)
av_excess_return6_vw = np.average(vw_excess_return6)
av_excess_return7_vw = np.average(vw_excess_return7)
av_excess_return8_vw = np.average(vw_excess_return8)
av_excess_return9_vw = np.average(vw_excess_return9)
av_excess_return10_vw = np.average(vw_excess_return10)
```

```
# Plot cumulative returns
plt.plot(dates, cum_ret1, label = "High")
plt.plot(dates, cum_ret2, label = "9")
plt.plot(dates, cum_ret3, label = "8")
plt.plot(dates, cum_ret4, label = "7")
plt.plot(dates, cum_ret5, label = "6")
plt.plot(dates, cum_ret6, label = "5")
plt.plot(dates, cum_ret7, label = "4")
plt.plot(dates, cum_ret8, label = "3")
plt.plot(dates, cum_ret9, label = "2")
plt.plot(dates, cum_ret10, label = "Low")
plt.legend()
plt.show()

# Plot value weighted cumulative returns
plt.plot(dates, vw_cum_ret1, label = "High")
plt.plot(dates, vw_cum_ret2, label = "9")
plt.plot(dates, vw_cum_ret3, label = "8")
plt.plot(dates, vw_cum_ret4, label = "7")
plt.plot(dates, vw_cum_ret5, label = "6")
plt.plot(dates, vw_cum_ret6, label = "5")
plt.plot(dates, vw_cum_ret7, label = "4")
plt.plot(dates, vw_cum_ret8, label = "3")
plt.plot(dates, vw_cum_ret9, label = "2")
plt.plot(dates, vw_cum_ret10, label = "Low")
plt.legend()
plt.show()

# UMD portfolio based on long High and short Low
excess_returnumd = np.array(excess_return1) - np.array(excess_return10)
cum_retumd = cum_returns(len(dates), excess_returnumd)
plt.plot(dates, cum_retumd, label = "UMDew")
plt.legend()
plt.show()

# value weighted UMD portfolio based on long High and short Low
vw_excess_returnumd = np.array(vw_excess_return1) - np.array(vw_excess_return10)
vw_cum_retumd = cum_returns(len(dates), vw_excess_returnumd)
plt.plot(dates, vw_cum_retumd, label = "UMDvw")
plt.legend()
plt.show()

# Export spread returns to stata for GRS test
dfGRS = pd.DataFrame({
    'UMD': vw_excess_returnumd,
    'high': vw_excess_return1,
    '9': vw_excess_return2,
    '8': vw_excess_return3,
    '7': vw_excess_return4,
    '6': vw_excess_return5,
    '5': vw_excess_return6,
    '4': vw_excess_return7,
    '3': vw_excess_return8,
    '2': vw_excess_return9,
    '1ow': vw_excess_return10,
    'MKT': MKT,
    'HML': HML,
    'SMB': SMB,
    })

dfFF3 = pd.DataFrame({
    'dates': dates,
    'high_ew': cum_ret1,
    '9_ew': cum_ret2,
    '8_ew': cum_ret3,
    '7_ew': cum_ret4,
    '6_ew': cum_ret5,
    '5_ew': cum_ret6,
    '4_ew': cum_ret7,
    '3_ew': cum_ret8,
    '2_ew': cum_ret9,
    'low_ew': cum_ret10,
    'high_vw': vw_cum_ret1,
    '9_vw': vw_cum_ret2,
    '8_vw': vw_cum_ret3,
    '7_vw': vw_cum_ret4,
    '6_vw': vw_cum_ret5,
    '5_vw': vw_cum_ret6,
    '4_vw': vw_cum_ret7,
    '3_vw': vw_cum_ret8,
    '2_vw': vw_cum_ret9,
    'low_vw': vw_cum_ret10,
    })

dfFF3.to_stata(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\Figuren\
    FF3.dta')


dfGRS.to_stata(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\Monthly\
    GRS.dta')

##----------------------------------------------
##                      CAPM
##----------------------------------------------

dfFF3factorsMonthly = pd.read_excel(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\
    Thesis\Data\Monthly\ff 3 monthly.xlsx')
dfFrame = pd.read_stata(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\
    Monthly\Final 1 CAPM.dta')
dfFrame = dfFrame.drop(['index',
                        'Ln_MV',
                        'Ln_BM',
                        'company_id',
                        '_merge'], axis=1)
dfFrame['Excess_Return'] = dfFrame['Return'] - RF

coeff1, pvalue1, excess_return1, vw_excess_return1, cum_ret1, vw_cum_ret1, av_MV_1,
    av_BM_1, av_beta_1, av_ivol_1, av_eivol_1 = portfolio(0, 37)
coeff2, pvalue2, excess_return2, vw_excess_return2, cum_ret2, vw_cum_ret2, av_MV_2,
    av_BM_2, av_beta_2, av_ivol_2, av_eivol_2 = portfolio(37, 74)
coeff3, pvalue3, excess_return3, vw_excess_return3, cum_ret3, vw_cum_ret3, av_MV_3,
    av_BM_3, av_beta_3, av_ivol_3, av_eivol_3 = portfolio(74, 111)
coeff4, pvalue4, excess_return4, vw_excess_return4, cum_ret4, vw_cum_ret4, av_MV_4,
    av_BM_4, av_beta_4, av_ivol_4, av_eivol_4 = portfolio(111, 148)
coeff5, pvalue5, excess_return5, vw_excess_return5, cum_ret5, vw_cum_ret5, av_MV_5,
    av_BM_5, av_beta_5, av_ivol_5, av_eivol_5 = portfolio(148, 183)
```

```
coeff6, pvalue6, excess_return6, vw_excess_return6, cum_ret6, vw_cum_ret6, av_MV_6,
    av_BM_6, av_beta_6, av_ivol_6, av_eivol_6 = portfolio(183, 218)
coeff7, pvalue7, excess_return7, vw_excess_return7, cum_ret7, vw_cum_ret7, av_MV_7,
    av_BM_7, av_beta_7, av_ivol_7, av_eivol_7 = portfolio(218, 255)
coeff8, pvalue8, excess_return8, vw_excess_return8, cum_ret8, vw_cum_ret8, av_MV_8,
    av_BM_8, av_beta_8, av_ivol_8, av_eivol_8 = portfolio(255, 292)
coeff9, pvalue9, excess_return9, vw_excess_return9, cum_ret9, vw_cum_ret9, av_MV_9,
    av_BM_9, av_beta_9, av_ivol_9, av_eivol_9 = portfolio(292, 329)
coeff10, pvalue10, excess_return10, vw_excess_return10, cum_ret10, vw_cum_ret10, av_MV_10
    , av_BM_10, av_beta_10, av_ivol_10, av_eivol_10 = portfolio(329, 366)

av_excess_return1 = np.average(excess_return1)
av_excess_return2 = np.average(excess_return2)
av_excess_return3 = np.average(excess_return3)
av_excess_return4 = np.average(excess_return4)
av_excess_return5 = np.average(excess_return5)
av_excess_return6 = np.average(excess_return6)
av_excess_return7 = np.average(excess_return7)
av_excess_return8 = np.average(excess_return8)
av_excess_return9 = np.average(excess_return9)
av_excess_return10 = np.average(excess_return10)

av_excess_return1_vw = np.average(vw_excess_return1)
av_excess_return2_vw = np.average(vw_excess_return2)
av_excess_return3_vw = np.average(vw_excess_return3)
av_excess_return4_vw = np.average(vw_excess_return4)
av_excess_return5_vw = np.average(vw_excess_return5)
av_excess_return6_vw = np.average(vw_excess_return6)
av_excess_return7_vw = np.average(vw_excess_return7)
av_excess_return8_vw = np.average(vw_excess_return8)
av_excess_return9_vw = np.average(vw_excess_return9)
av_excess_return10_vw = np.average(vw_excess_return10)

# Plot cumulative returns
plt.plot(dates, cum_ret1, label = "High")
plt.plot(dates, cum_ret2, label = "9")
plt.plot(dates, cum_ret3, label = "8")
plt.plot(dates, cum_ret4, label = "7")
plt.plot(dates, cum_ret5, label = "6")
plt.plot(dates, cum_ret6, label = "5")
plt.plot(dates, cum_ret7, label = "4")
plt.plot(dates, cum_ret8, label = "3")
plt.plot(dates, cum_ret9, label = "2")
plt.plot(dates, cum_ret10, label = "Low")
plt.legend()
plt.show()

# Plot value weighted cumulative returns
plt.plot(dates, vw_cum_ret1, label = "High")
plt.plot(dates, vw_cum_ret2, label = "9")
plt.plot(dates, vw_cum_ret3, label = "8")
plt.plot(dates, vw_cum_ret4, label = "7")
plt.plot(dates, vw_cum_ret5, label = "6")
plt.plot(dates, vw_cum_ret6, label = "5")
plt.plot(dates, vw_cum_ret7, label = "4")
plt.plot(dates, vw_cum_ret8, label = "3")
plt.plot(dates, vw_cum_ret9, label = "2")
plt.plot(dates, vw_cum_ret10, label = "Low")
plt.legend()
plt.show()

# UMD portfolio based on long High and short Low
excess_returnumd = np.array(excess_return1) − np.array(excess_return10)
cum_retumd = cum_returns(len(dates), excess_returnumd)
plt.plot(dates, cum_retumd, label = ”UMDew”)
plt.legend()
plt.show()

# value weighted UMD portfolio based on long High and short Low
vw_excess_returnumd = np.array(vw_excess_return1) − np.array(vw_excess_return10)
vw_cum_retumd = cum_returns(len(dates), vw_excess_returnumd)
plt.plot(dates, vw_cum_retumd, label = ”UMDvw”)
plt.legend()
plt.show()

# Export spread returns to stata for GRS test
dfGRSCAPM = pd.DataFrame({
    'UMD': vw_excess_returnumd,
    'high': vw_excess_return1,
    '9': vw_excess_return2,
    '8': vw_excess_return3,
    '7': vw_excess_return4,
    '6': vw_excess_return5,
    '5': vw_excess_return6,
    '4': vw_excess_return7,
    '3': vw_excess_return8,
    '2': vw_excess_return9,
    '1ow': vw_excess_return10,
    'MKT': MKT,
    'HML': HML,
    'SMB': SMB,
    })

dfCAPM = pd.DataFrame({
    'dates': dates,
    'high_ew': cum_ret1,
    '9_ew': cum_ret2,
    '8_ew': cum_ret3,
    '7_ew': cum_ret4,
    '6_ew': cum_ret5,
    '5_ew': cum_ret6,
    '4_ew': cum_ret7,
    '3_ew': cum_ret8,
    '2_ew': cum_ret9,
    '1ow_ew': cum_ret10,
    'high_vw': vw_cum_ret1,
    '9_vw': vw_cum_ret2,
    '8_vw': vw_cum_ret3,
    '7_vw': vw_cum_ret4,
    '6_vw': vw_cum_ret5,
    '5_vw': vw_cum_ret6,
    '4_vw': vw_cum_ret7,
    '3_vw': vw_cum_ret8,
    '2_vw': vw_cum_ret9,
    '1ow_vw': vw_cum_ret10,
    })
```

```
dfCAPM.to_stata(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\Figuren\
    CAPM.dta')

dfGRSCAPM.to_stata(r'C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\
    Monthly\GRS CAPM.dta')
```

STATA do file

```
* -----------------------------------------------------------------------------
*                    Descriptive Statistics and Results
* -----------------------------------------------------------------------------


cd "C:\Users\Lodew\Documents\Bachelor Economie Vakken\Thesis\Data\Monthly"

* -Import Dataset-
use "Monthly Data Test Final without financials.dta", clear


gen date_only = dofc(Date)
drop Date
recast int date_only
format %tdnn/dd/CCYY date_only
rename date_only Date

egen company_id = group(ISIN)

merge 1:1 company_id Date using "Monthly BETA FF without financials.dta" // Merge for
    Betas

drop if BETA_FF == .

xtset company_id Date

sort company_id
by company_id: gen IVOLlag1 = IVOL[_n-1] // Create lagged IVOL

drop if Exp_IVOL > 100 // Remove extreme expected IVOL values, in total 15 will be
    removed
gen Year = year(Date) // Create year variable for splitting sample regressions

* -- Descriptive statistics --

* 4.3
tabstat Return IVOL Exp_IVOL BETA_FF Ln_MV Ln_BM , c(stat) stat(mean sd p50 p25 p75 skew
    n)

* 4.4
pwcorr Return IVOL BETA_FF Exp_IVOL Ln_MV Ln_BM IVOLlag1, sig star(.01)

* -- Regression for total time period --

* Fama and French
// Fama-MacBeth
asreg Return BETA_FF Ln_MV Ln_BM, fmb newey(3)
estimates store FF_factors_1
```

```
* IVOL
// Fama–MacBeth
asreg Return IVOL Ln_MV Ln_BM, fmb newey(3)
estimates store FMB_IVOL_1

// Fixed Effects
xtreg Return IVOL Ln_MV Ln_BM, fe vce(cluster company_id)
estimates store FE_IVOL_1

// Random Effects
xtreg Return IVOL Ln_MV Ln_BM, re vce(cluster company_id)
estimates store RE_IVOL_1

* E[IVOL]
// Fama–MacBeth
asreg Return Exp_IVOL Ln_MV Ln_BM, fmb newey(3)
estimates store FMB_EIVOL_1

// Fixed Effects
xtreg Return Exp_IVOL Ln_MV Ln_BM, fe vce(cluster company_id)
estimates store FE_EIVOL_1

// Random Effects
xtreg Return Exp_IVOL Ln_MV Ln_BM, re vce(cluster company_id)
estimates store RE_EIVOL_1

* Lagged IVOL
// Fama–MacBeth
asreg Return IVOLlag1 Ln_MV Ln_BM, fmb newey(3)
estimates store FMB_IVOL1_1

// Fixed Effects
xtreg Return IVOLlag1 Ln_MV Ln_BM, fe vce(cluster company_id)
estimates store FE_IVOL1_1

// Random Effects
xtreg Return IVOLlag1 Ln_MV Ln_BM, re vce(cluster company_id)
estimates store RE_IVOL1_1

* — Regression for before COVID–19 time period —

* IVOL
// Fama–MacBeth
asreg Return IVOL Ln_MV Ln_BM if Year < 2020, fmb newey(3)
estimates store FMB_IVOL_2
// Fixed Effects
xtreg Return IVOL Ln_MV Ln_BM if Year < 2020, fe vce(cluster company_id)
estimates store FE_IVOL_2
// Random Effects
xtreg Return IVOL Ln_MV Ln_BM if Year < 2020, re vce(cluster company_id)
estimates store RE_IVOL_2

* E[IVOL]
// Fama–MacBeth
asreg Return Exp_IVOL Ln_MV Ln_BM if Year < 2020, fmb newey(3)
estimates store FMB_EIVOL_2
// Fixed Effects
xtreg Return Exp_IVOL Ln_MV Ln_BM if Year < 2020, fe vce(cluster company_id)
estimates store FE_EIVOL_2
```

```
// Random Effects
xtreg Return Exp_IVOL Ln_MV Ln_BM if Year < 2020, re vce(cluster company_id)
estimates store RE_EIVOL_2

* Lagged IVOL
// Fama–MacBeth
asreg Return IVOLlag1 Ln_MV Ln_BM if Year < 2020, fmb newey(3)
estimates store FMB_IVOL1_2
// Fixed Effects
xtreg Return IVOLlag1 Ln_MV Ln_BM if Year < 2020, fe vce(cluster company_id)
estimates store FE_IVOL1_2
// Random Effects
xtreg Return IVOLlag1 Ln_MV Ln_BM if Year < 2020, re vce(cluster company_id)
estimates store RE_IVOL1_2

* — Regression for during COVID–19 time period —

* IVOL
// Fama–MacBeth
asreg Return IVOL Ln_MV Ln_BM if Year >= 2020, fmb newey(3)
estimates store FMB_IVOL_3
// Fixed Effects
xtreg Return IVOL Ln_MV Ln_BM if Year >= 2020, fe vce(cluster company_id)
estimates store FE_IVOL_3
// Random Effects
xtreg Return IVOL Ln_MV Ln_BM if Year >= 2020, re vce(cluster company_id)
estimates store RE_IVOL_3

* E[IVOL]
// Fama–MacBeth
asreg Return Exp_IVOL Ln_MV Ln_BM if Year >= 2020, fmb newey(3)
estimates store FMB_EIVOL_3
// Fixed Effects
xtreg Return Exp_IVOL Ln_MV Ln_BM if Year >= 2020, fe vce(cluster company_id)
estimates store FE_EIVOL_3
// Random Effects
xtreg Return Exp_IVOL Ln_MV Ln_BM if Year >= 2020, re vce(cluster company_id)
estimates store RE_EIVOL_3

* Lagged IVOL
// Fama–MacBeth
asreg Return IVOLlag1 Ln_MV Ln_BM if Year >= 2020, fmb newey(3)
estimates store FMB_IVOL1_3
// Fixed Effects
xtreg Return IVOLlag1 Ln_MV Ln_BM if Year >= 2020, fe vce(cluster company_id)
estimates store FE_IVOL1_3
// Random Effects
xtreg Return IVOLlag1 Ln_MV Ln_BM if Year >=2020, re vce(cluster company_id)
estimates store RE_IVOL1_3


* — Display all regressions —
// Coeff, SE per time period
estimates table FF_factors_1 FMB_EIVOL_1 FMB_IVOL_1 FMB_IVOL1_1 FE_EIVOL_1 FE_IVOL_1
    FE_IVOL1_1 RE_EIVOL_1 RE_IVOL_1  RE_IVOL1_1, b(%5.3f) se(%5.3f)
estimates table FMB_EIVOL_2 FMB_IVOL_2 FMB_IVOL1_2 FE_EIVOL_2 FE_IVOL_2 FE_IVOL1_2
    RE_EIVOL_2 RE_IVOL_2 RE_IVOL1_2, b(%5.3f) se(%5.3f)
estimates table FMB_EIVOL_3 FMB_IVOL_3 FMB_IVOL1_3 FE_EIVOL_3 FE_IVOL_3 FE_IVOL1_3
    RE_EIVOL_3 RE_IVOL_3 RE_IVOL1_3, b(%5.3f) se(%5.3f)
```

```
// Significance per time period
estimates table FF_factors_1 FMB_EIVOL_1 FMB_IVOL_1 FMB_IVOL1_1 FE_EIVOL_1 FE_IVOL_1
    FE_IVOL1_1 RE_EIVOL_1 RE_IVOL_1  RE_IVOL1_1, star(.05 .01 .001)

estimates table FMB_EIVOL_2 FMB_IVOL_2 FMB_IVOL1_2 FE_EIVOL_2 FE_IVOL_2 FE_IVOL1_2
    RE_EIVOL_2 RE_IVOL_2 RE_IVOL1_2, star(.05 .01 .001)
estimates table FMB_EIVOL_3 FMB_IVOL_3 FMB_IVOL1_3 FE_EIVOL_3 FE_IVOL_3 FE_IVOL1_3
    RE_EIVOL_3 RE_IVOL_3 RE_IVOL1_3, star(.05 .01 .001)
// Adj. R-Squared per time period
estimates table FF_factors_1 FMB_EIVOL_1 FMB_IVOL_1 FMB_IVOL1_1 FE_EIVOL_1 FE_IVOL_1
    FE_IVOL1_1 RE_EIVOL_1 RE_IVOL_1  RE_IVOL1_1, stats(N r2_a)
estimates table FMB_EIVOL_2 FMB_IVOL_2 FMB_IVOL1_2 FE_EIVOL_2 FE_IVOL_2 FE_IVOL1_2
    RE_EIVOL_2 RE_IVOL_2 RE_IVOL1_2, stats(N r2_a)
estimates table FMB_EIVOL_3 FMB_IVOL_3 FMB_IVOL1_3 FE_EIVOL_3 FE_IVOL_3 FE_IVOL1_3
    RE_EIVOL_3 RE_IVOL_3 RE_IVOL1_3, stats(N r2_a)

* GRS for portfolio

* -Import Dataset-
use "GRS.dta", clear

grstest2 _low _2 _3 _4 _5 _6 _7 _8 _9 high , flist(MKT SMB HML) alphas

* ------------------------------------------------------------
*                    Robustness Check
* ------------------------------------------------------------

* 5.1

// Hausmann tests for random or fixed effect
xtreg Return Exp_IVOL Ln_MV Ln_BM, fe // without clustered standard errors
estimates store FE_EIVOL_1_nc
xtreg Return Exp_IVOL Ln_MV Ln_BM, re // without clustered standard errors
estimates store RE_EIVOL_1_nc

hausman FE_EIVOL_1_nc RE_EIVOL_1_nc // null hypothesis is rejected

// Continuing with fixed effects model to adjust for time-effect
egen month_id = group(Date)
tabulate month_id, generate(month_dum) //dummy variable for every month

xtreg Return Exp_IVOL Ln_MV Ln_BM month_dum2-month_dum96, fe vce(cluster company_id)
estimates store FE_EIVOL_TE

estimates table FMB_EIVOL_1 FE_EIVOL_1 FE_EIVOL_TE, b(%5.3f) se(%5.3f) keep(Exp_IVOL
    Ln_MV Ln_BM _cons)
estimates table FMB_EIVOL_1 FE_EIVOL_1 FE_EIVOL_TE, star keep(Exp_IVOL Ln_MV Ln_BM _cons)

estimates table FMB_EIVOL_1 FE_EIVOL_1 FE_EIVOL_TE, stats(N r2_a) keep(Exp_IVOL Ln_MV
    Ln_BM _cons)

* 5.2

// Import Dataset
use "Monthly Data Test Final without financials CAPM.dta", clear

gen date_only = dofc(Date)
drop Date
recast int date_only
```

```
format %tdnn/dd/CCYY date_only
rename date_only Date

egen company_id = group(ISIN)

merge 1:1 company_id Date using ”Monthly BETA FF without financials.dta” // Merge for
    Betas

drop if BETA_FF == .

xtset company_id Date

sort company_id

// drop same observations as in ff3 dataframe
drop if index == 3648
drop if index == 30197
drop if index == 26092
drop if index == 7988
drop if index == 28611
drop if index == 26027
drop if index == 29509
drop if index == 33954
drop if index == 2990
drop if index == 2794
drop if index == 7995
drop if index == 26016
drop if index == 21987
drop if index == 27846
drop if index == 17774

// Fama-MacBeth
asreg Return Exp_IVOL Ln_MV Ln_BM, fmb newey(3)
estimates store FMB_IVOL_1_CAPM

// Fixed Effects adjusted with time effect
egen month_id = group(Date)
tabulate month_id, generate(month_dum) //dummy variable for every month
xtreg Return Exp_IVOL Ln_MV Ln_BM month_dum2-month_dum96, fe vce(cluster company_id)
estimates store FE_EIVOL_TE_CAPM

estimates table FMB_EIVOL_1 FMB_IVOL_1_CAPM FE_EIVOL_TE FE_EIVOL_TE_CAPM, b(%5.3f) se
    (%5.3f) keep(Exp_IVOL Ln_MV Ln_BM _cons)
estimates table FMB_EIVOL_1 FMB_IVOL_1_CAPM FE_EIVOL_TE FE_EIVOL_TE_CAPM, star keep(
    Exp_IVOL Ln_MV Ln_BM _cons)
estimates table FMB_EIVOL_1 FMB_IVOL_1_CAPM FE_EIVOL_TE FE_EIVOL_TE_CAPM, stats(N r2_a)
    keep(Exp_IVOL Ln_MV Ln_BM _cons)

* -Import Dataset-
use ”GRS CAPM.dta”, clear

grstest2 _low _2 _3 _4 _5 _6 _7 _8 _9 high , flist(MKT SMB HML) alphas


* 5.3

* -Import Dataset-
use ”Monthly Data Test Final without financials.dta”, clear
```

```
gen date_only = dofc(Date)
drop Date
recast int date_only
format %tdnn/dd/CCYY date_only
rename date_only Date

egen company_id = group(ISIN)

merge 1:1 company_id Date using ”Monthly BETA FF without financials.dta” // Merge for
    Betas

drop if BETA_FF == .

xtset company_id Date

sort company_id

// sorted on Ln_MV and then picked the company ID's of the 37 lowest values for size
keep if (company_id == 358 | company_id == 207 | company_id == 276 | company_id == 364 |
    company_id == 116 | company_id == 313 | company_id == 23 | company_id == 326 |
    company_id == 107 | company_id == 237 | company_id == 37 | company_id == 14 |
    company_id == 292 | company_id == 64 | company_id == 361 | company_id == 246 |
    company_id == 222 | company_id == 106 | company_id == 19 | company_id == 211 |
    company_id == 359 | company_id == 109 | company_id == 203| company_id == 290 |
    company_id == 75 | company_id == 30 | company_id == 257 | company_id == 196 |
    company_id == 267 | company_id == 139 | company_id == 198 | company_id == 354 |
    company_id == 322 | company_id == 103 | company_id == 323 | company_id == 142|
    company_id == 144)

egen month_id = group(Date)
tabulate month_id, generate(month_dum) //dummy variable for every month


* E[IVOL]
// Fama–MacBeth
asreg Return Exp_IVOL Ln_MV Ln_BM, fmb newey(3)
estimates store FMB_EIVOL_1_10p

// Fixed Effects
xtreg Return Exp_IVOL Ln_MV Ln_BM month_dum2–month_dum96, fe vce(cluster company_id)
estimates store FE_EIVOL_1_10p

estimates table FMB_EIVOL_1_10p FE_EIVOL_1_10p , b(%5.3f) se(%5.3f) keep(Exp_IVOL Ln_MV
    Ln_BM _cons)
estimates table FMB_EIVOL_1_10p FE_EIVOL_1_10p, star keep(Exp_IVOL Ln_MV Ln_BM _cons)

estimates table FMB_EIVOL_1_10p FE_EIVOL_1_10p, stats(N r2_a) keep(Exp_IVOL Ln_MV Ln_BM
    _cons)
```