

Implied volatilities and Euro Stoxx 50 returns

How the implied volatility skew in the option market may function as a predictor for future stock returns

Laurens-Jan Bovens – s4666143

Master Thesis, final version

Master Financial Economics; Radboud Universiteit Nijmegen

Supervisor: dr. J. Qiu

20-07-2017, Leiden

Abstract

The volatility smile may be an indication of the presence of informed traders and therefore may have predictive power for future stock returns. This thesis reports results of an analysis of the predictability of the volatility smirk present in call and in put options on stocks that make up the Euro Stoxx 50. A Fama-Macbeth regression is applied and subsequently portfolios based on differences in implied volatility are constructed. In this sample, the information in the volatility smirk harnesses no significant predictive power for future stock returns.

Table of contents

1. Introduction	6
2. Literature Review	7
2.1 The Black-Scholes-Merton Model	8
2.2 Volatility	10
2.3 The Volatility Smile	11
2.4 Implications of the Volatility Smile	12
3. Data	15
3.1 Main Variables	15
3.2 Control Variables	17
4. Methodology	18
4.1 Fama-MacBeth Regression	18
4.2 Portfolio Construction	19
5. Results	20
5.1 Fama-MacBeth Results	20
5.2 Portfolio Strategy Results	22
6. Conclusion and discussion	24
References	26
Appendix	29
Alternative FM-Regressions	29
MATLAB script for FM regression	32
VBA code for option selection and portfolio construction	40

1. Introduction

Strategies which allow one to generate abnormal returns in the stock market have always fascinated investors. Such strategies have often very different foundations, but they have in common that they seem to defy the efficient market hypothesis. Past stock returns (Jegadeesh & Titman, 1993), delayed price response (Bernard & Thomas, 1989), news (Chan, 2003) and implied volatilities of options (An, Ang, Bali, & Cakici, 2014; Ang, Hodrick, Xing, & Zhang, 2006; Xing, Zhang, & Zhao, 2010); all have been used as the foundation for the construction of abnormal return strategies. Accompanied with an explanation of their strategy and analysis authors tend to present annualized abnormal returns that are not uncommon to exceed the ten percent mark. Needless to say, economists approach such work with appropriate scepticism. Previously presented theories are refuted or confirmed, by testing with different samples, methodology and variables.

This thesis reports an analysis of a trading strategy based on differences in implied volatility in the option market. I analyse whether the volatility smile harnesses the ability to generate abnormal returns. The term volatility smile refers to a non-flat structure of implied volatilities of options with the same time to maturity and underlying asset, but with different exercise prices. Both the steepness of the volatility curve (Xing, Zhang, & Zhao, 2010) as well as the change in the shape of the volatility curve over time (An, Ang, Bali, & Cakici, 2014) are reported to be significant predictors for future stock returns, for up to 6 months. The economic justification to the existence of this theory is the presence of informed traders. Informed traders who want to exploit their informational advantage are more likely to use high leveraged financial products to do so (Black, 1975). An out of the money option is a prime example of such a product. The result of this behaviour is that the option market contains more information than the stock market and leads the price discovery process. Based on the model presented by Gârleanu, Pedersen, & Poteshman (2009) I use the shape of the volatility smile as a proxy for the presence of informed traders. Gârleanu et al. (2009) construct a model where the expensiveness of an option is directly used to end-user demand. Thus, a disproportionate difference in the implied volatility of an at the money option and an out of the money option may be the result of the trading behaviour of informed traders. Based on this assumption I test whether or not the shape of the volatility smile functions as a significant predictor for future stock returns.

My analysis differs from existing research in the size and time period of my data sample. I use a sample of daily option data on the firms that make up the Euro Stoxx 50 over a five

year period, starting on the first of January 2010. That means that my sample size is smaller, but more recent. The methodology is in line with the one used by Xing et al. (2010); first a Fama-Macbeth regression is performed and subsequently weekly quintile portfolios are constructed, based on the implied volatility difference between out of the money options and at the money options. This means that my quintile portfolios contain a maximum of 10 stocks each, whereas the portfolios of Xing et al. (2010) contain 168 stocks each. My data sample puts a strain on the predictability that is embedded in the volatility smile – if present at all – since firm level noise is diversified to a lesser extent in each portfolio. If the effect is still present and its returns exceed the transaction costs, then this strategy may provide an actual real-world trading opportunity.

The performed Fama-MacBeth regression reports that both the shape of the volatility smile of call options as well as the shape of the volatility smile of put options don't function as a predictor for next week's stock returns. The portfolio trading strategy shows in a heuristic manner that the call option volatility smile yields a return that is very close to zero. The put option volatility smile seems to yield abnormal returns of 0.14% on a weekly basis, but when comparing this to the results of the Fama-MacBeth regression it becomes clear that company size is the only significant explanatory variable.

The second section of this thesis is reserved for an exploration of existing relevant literature. I begin with an explanation of the Black-Scholes-Merton model before I dive deeper into the existence of the volatility smile. In the third section I elaborate on the data sample used for this thesis, subsequently I explain my methods of analysis in the section after that. The fifth section of this paper is reserved for the results of my analysis. Finally, the conclusion and the accompanied discussion is to be found in the sixth section.

2. Literature Review

This thesis falls within the literature that researches the way in which new information is incorporated in asset prices. A broad subject that has occupied economists for a long time now. Unsurprisingly then, there is a rich amount of literature available on the topic. Although the way in which the process of information incorporation occurs has not received a clear and unequivocal answer, there seems to be evidence that this process does not occur with equal speed in different markets with the same underlying asset. The reasoning behind this is that different markets have distinct characteristics, which

makes certain markets more attractive to informed traders (Xing, Zhang, & Zhao, 2010). The result is that different markets may not reflect the same information at the same point in time. One market may lag in the information incorporation process, compared to another market. This relationship has also been tested for the stock market and the option market for the same underlying asset, which is under consideration in this thesis. Before I discuss the literature on the relationship between the option market and the stock market, I will subsequently explain the Black-Scholes-Merton model, volatility and the volatility smile in the next sub-sections. This does not only support the discussion of relevant literature on information incorporation, it is also beneficial for supporting the methodology and result section of this thesis.

2.1 The Black-Scholes-Merton Model

The reason for discussing the Black-Scholes-Merton model (BSM) is that it is the standard for pricing options. The model was introduced by Black and Scholes (1973) and Merton (1973) – by coincidence at almost the same time as the opening of the Chicago Board Options Exchange (CBOE) – and is still relevant to this day. The original model allows the user to calculate the price of a European call option on a non-dividend paying stock, using only a limited number of – mainly observable – variables. The formula for a European call option on a stock is as follows:

$$C = S_0 N(d_1) - K e^{-rT} N(d_2) \quad (2.1)$$

Where:

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{1}{2} \sigma^2\right)T}{\sigma\sqrt{T}} \quad (2.2)$$

$$d_2 = d_1 - \sigma\sqrt{T} \quad (2.3)$$

From these formulas, we can observe that five inputs are required: S_0 is the current stock price; K is the exercise price of the call option; r is the continuously compounded risk-free rate; T the time to expiration; and σ is the volatility of the stock. The $N(x)$ -function represents the cumulative normal distribution function. All but one of the variables required to calculate the option price are rather self-explanatory and clearly observable, meaning that no assumptions must be made. The only variable that is open to discussion

is σ , the volatility of the stock. The true volatility is not known, but using historical volatility as an approximation is common practice. Historical volatility, then, is used in many of the models as a starting point¹.

Although calculating the price of an option makes the model already very valuable, identifying that all risk associated with an option can be eliminated through continuous hedging is a characteristic of the model that is at least as important². However, the model is based on several assumptions that aren't necessarily a veracious reflection of actual markets. In the model the market consists of a risk-free bond, a European call option and a non-dividend paying stock. The possibility exists to borrow and lend at the risk-free rate. Both the stock and the option can be bought and sold in fractions, without transaction costs. The stock price follows a geometric Brownian motion with a deterministic volatility factor that is constant over time. The motion is represented by the following equation

$$dS(t) = \alpha S(t)dt + \sigma S(t)dZ(t) \quad (2.4)$$

In this equation $dS(t)$ represents the change in stock price at a terminal point in time. Variable α is the continuously compounded expected return of the stock (also known as drift) and σ is the measure of volatility. Both drift and volatility are proportional to the stock price.

A geometric Brownian motion implies that the stock price at a certain point in time, conditional upon its current price, is lognormally distributed. For a number of reasons, it is not just the distribution of the price at a terminal point in time that is useful, but also the path of that price over a period of time; geometric Brownian motion is able to describe such a path (McDonald, 2014). Figure 1 represents 500 paths of a price simulation of an asset, according to a geometric Brownian motion stochastic process.

¹ Since a discussion of these models is outside the scope of this thesis, I would like to refer to Robert L. McDonald (2014), who provides a clear overview of such models.

² In the official press release in response to receiving the Nobel Prize for their contribution, the advantages for risk management through using this model were explicitly named. The press release, dating from 1997, can be found here: http://www.nobelprize.org/nobel_prizes/economic-sciences/laureates/1997/press.html

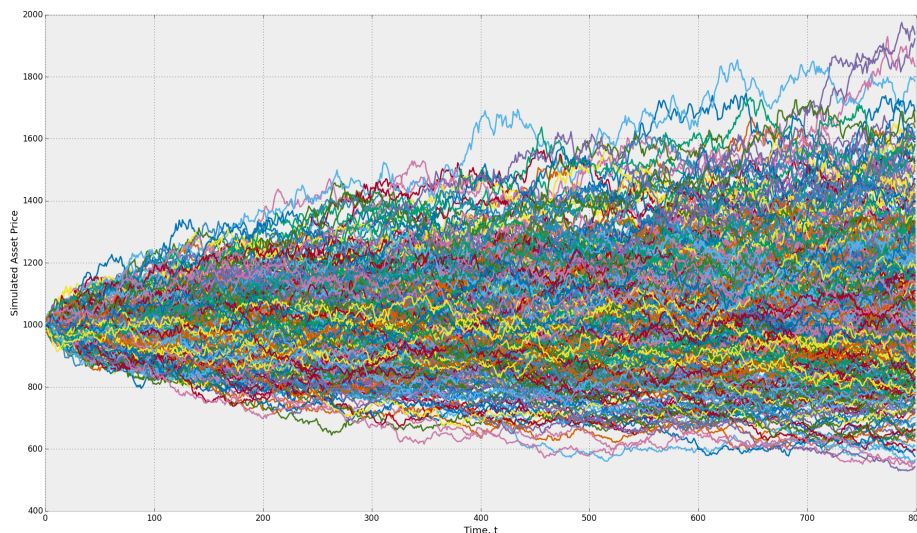


Figure 1 – “Asset Prices Simulated using the Geometric Brownian Motion Stochastic Process” (Reid, 2015)

Since the BSM model is based on the assumptions as explained above, it naturally means that it is not universally accepted. It has also received criticism in the 44 years since its publication. The observation of the following empirical phenomena is a frequent source of criticism: results show that the actual distribution of stock prices is closer to an asymmetric leptokurtic distribution – meaning that the distribution is skewed to the left, has a higher peak and two heavier tails – than the normal distribution assumed in the BSM and that there exists a phenomenon called the volatility smile (Kou, 2002)³. If the BSM were fully accurate, there would not be a phenomenon like the volatility smile. Prior to discussing the concept of the volatility smile, I explain the role of volatility in option pricing in the next sub-section, for the sake of clarity.

2.2 Volatility

The first subsection of this chapter mentions that the BSM only needs five input variables to calculate the price of a European option on a non-dividend paying stock. Four of those variables are clearly observable and give little room for discussion. The one variable that is a subject for discussion is volatility. Of all the input variables, volatility is the most difficult to understand. It is of a more abstract nature than the other variables, yet an accurate assessment of its value is crucial. Small changes may have severe

³ Kou (2002) discusses a large variety of papers, in which models take one of the phenomena into account. However, all fail to find an answer to both empirical phenomena.

consequences. Discussing volatility is not an unambiguous matter; volatility itself may be interpreted in various ways. Therefore, I first define the different types of volatility below.

The one volatility that we definitely can't observe is *future volatility*. The true volatility is the value that best describes the future distribution of the underlying stock. Market participants try to make an assessment of future volatility based on data that they can observe. *Historical volatility* is an example of observable data. It is the actual volatility of the underlying over a past period. Historical volatility may provide a good starting point – as mentioned in the previous sub-section – but actual volatility is not constant over time. *Forecast volatility*, then, is the result of an assessment of future volatility. It is an interpretation of all available information over a period that usually is identical to the remaining life of an option. Lastly, *implied volatility*, represents a different approach. It is the volatility of the underlying asset that is implied by market prices of options. Instead of trying to make an estimate of future volatility, the four observable required inputs and the option price are entered into the BSM (Natenberg, 1994). Using implied volatility may not be as straightforward as it seems, as evidenced by the following sub-section.

2.3 The Volatility Smile

On any large stock, options exist with different exercise prices. Different exercise prices consequently mean different option prices. However, the underlying of the option is exactly identical in all cases. A logical conclusion, then, is that the implied volatility of all options should be the same. The contrary is told by an empirical phenomenon called the volatility smile. This term refers to an imbalance of implied volatilities between at-the-money options and options that are not at-the-money. Bates (1991) was the first to notice that prior to the market crash of 1987 out of the money put options were trading at unusually high prices. This was a period in which the S&P500 favoured the bulls strongly, until the crash in October of 1987. The S&P500 lost 23% of its value on October the 19th and 20th of that year. With a perception of increased market risk, the price – and thus the implied volatility – of out-of-the-money put options rose. Where Rubinstein (1985) found that the BSM provided reasonably accurate values from 1976 to 1978, he later found that from 1986 onwards the functioning of the BSM rapidly deteriorated (Rubinstein, 1994). In 1986, implied volatilities of different options that were not at-the-money deviated on average only between 0.3 and 0.7 percent from options that were at-the-money. This deviation increased to between 4.9 and 15.3 percent on average in 1992. From that moment on the imbalance in implied volatility has been frequently reported, for a wide range of asset types. Sheikh (1991)

examined the pricing of S&P100 calls during a 14 month period and found that the smile is a bias in the BSM that is both statistically and economically significant.

Later research indicated that implied volatility was not only a function of moneyness – the degree to which an option is in the money – but also of the time to maturity (Bollen & Whaley, 2004). Based on these two variables, an implied volatility surface can be created. Figure 3 is a clear example of a stylized graphical depiction of the implied volatility surface. The volatility smile becomes more pronounced as time to maturity decreases.

Since the volatility smile was first observed in equity options, the phenomenon has also been observed in other asset types in different forms. Some papers report on volatility smiles, whereas other papers observe volatility smirks. Volatility smirks are defined by Xing et al. (2010) as the difference between implied volatilities of out-of-the-money put options and the implied volatilities of at-the-money call options. Xing et al. (2010) find that over 90% of the observations from 1996 to 2005 exhibit either a volatility smile or a volatility smirk. As mentioned above, Rubinstein (1994) reported that options on S&P 500 stocks tend to form volatility smiles, with increasing severity over time. Apart from stocks, the smile is also reported for equity indices. Bates (1991) reported the existence of the volatility smile for the S&P500 index and Sheikh (1991) did so for the S&P100. The smile also forms in indices of other markets: options on the FTSE-100 (Alexander, 2001), the DAX-30 (Wallmeier & Hafner, 2000) and the IBEX-35 (Pena, Rubio, & Serna, 1999) demonstrate this phenomenon. Volatility smiles have also been reported for options on currencies (Chalamandaris & Tsekrekos, 2010; Carr & Wu, 2007; Derman, 2003) as well as commodities (Derman, 2003; Trolle & Schwartz, 2009), albeit with distinctively skewed smiles.

2.4 Implications of the Volatility Smile

Option pricing models are often developed under the assumption of a perfect information flow – also known as a no-arbitrage relation – between the stock market and the option market. Heston (1993) does so in a model with stochastic volatility. Conrad, Dittmar, and Ghysels (2013) examine skewness and implied volatility by using a risk-neutral density function, also under the same perfect information flow assumption. But how realistic is this assumption? Empirical phenomena like post-earnings-announcement drift (Ball & Brown, 1968) and momentum and reversals (Jegadeesh & Titman, 1993) seem to contradict the concept of perfect information flow. Ofek, Richardson and Whitelaw (2004) examined the

no-arbitrage relation between the option and stock market and found that this was economically and significantly violated, even after accounting for transaction costs. Gârleanu, Pedersen and Poteshman (2009) find, by using a demand-pressured option pricing model in which intermediaries can't perfectly hedge their option positions, a positive relationship between option prices measured in implied volatility and end-user demand. This might indicate an information advantage to the end-user, leading to higher demand for certain options, which in turn means that option prices may predict future stock returns. A train of thought that is at the foundation of the paper of Xing et al. (2010). Their work, in turn again, forms the basis for this thesis. Xing et al. examine the predictability and the information that is embedded in the implied volatility smirks of options on equity. They compare the implied volatility of out-of-the-money (OTM) put options with the implied volatility of at-the-money (ATM) call options and find that this difference is significantly related to future earnings. Stocks exhibiting the steepest volatility smirks – the difference in implied volatility between OTM put and ATM call – underperform the stocks with flattest implied volatility curve by almost 11%. A result that lasts up to 6 months. Based on these results, the authors conclude that informed traders might choose to trade in the option market to benefit from their informational advantage. The volatility smirk is the result of the presence of the informed trader, implying that the option market leads the price discovery process. The reasons for informed traders to prefer options instead of the underlying asset is because of high financial leverage (an argument first suggested by Black (1975)), low transaction costs, and few short-selling restrictions. Cremers and Weinbaum (2010) recognize that, among other factors, the presence of informed traders may have implications for options prices. They use the difference in implied volatility between the call and put options on the same underlying asset, with the same exercise price and maturity as a convenient means to identify price pressure in the option market. Using a data sample of all exchange listed call and put options on U.S. equities from 1996 to 2005, Cremers and Weinbaum find that deviations in implied volatility contain economically and significant information about future stock returns. Constructing a portfolio in which stocks with relatively expensive calls are purchased and stocks with relatively expensive puts are sold on a weekly basis, earns an average factor-adjusted abnormal return of 50 basis points per week. Doran and Krieger (2010) confirm these findings by yielding similar results with a different data sample. The most recent paper, to my knowledge, is by Feng, Zhang and Friesen (2015) who use a sample of 2510 stocks from 1996 to 2008.

The papers in the above paragraph look at implied volatility for predictability, but other characteristics of the option market also indicate that this market is leading the stock market. Easley, O'Hara and Srinivas (1998)⁴ examine the informational link between the option and stock market based on option trade volume. Using a model that makes a distinction between positive and negative news, they find significant results that the option trade volumes affect the stock market. Their principal finding is that the option market is a venue for informed trading. Ofek et al. (2004) contribute to the literature on the link between the option market and the stock market by examining rebate rate spreads⁵ and the violations of put-call parity. Their results indicate that rebate rate spreads and violations of the put-call parity are significant predictors for future stock returns. Cao, Chen, and Griffin (2005) examine the price discovery process for both the option market and the stock market prior to takeover announcements. They find that in the preannouncement period, imbalances in call volumes are a strong predictor for next-day returns. Pan and Poteshman (2006) also find evidence that option trading volume contains information about future stock prices. In their paper, they find that stocks with a low put-call ratio in volume outperform stocks with a high put-call ratio by over 1% on a weekly basis.

Considering the relevant papers within the information transmission literature, it is the paper by Xing et al. (2010) that is related the closest to this thesis. It is the only other paper that examines the information embedded in the shape of the implied volatility smile of options and its implications for the link with the stock market. This thesis examines the implications of the information embedded in the volatility smile for the stock market, but for a different data sample. In addition to the paper by Xing et al. (2010), I also examine the implications for stock predictability of the difference between OTM call options and ATM call option in implied volatilities.

⁴ Easley, O'Hara and Srinivas also give an extensive overview of other papers on the link between the option market and the stock market in their paper. However, that falls not within the scope of this thesis.

⁵ The rebate rate is the interest rate or dividend owed by the owner to the short seller of a security. The rebate rate spread, then, is the difference between the rebate rate and the normal rate of trading securities.

3. Data

3.1 Main Variables

The data for this thesis runs from the 1st of January 2010 to the 31st December of 2014, spanning a period of five years. All data is from the Eikon data base. The data set consists of option and equity data on all companies that comprise the Euro Stoxx 50; Europe's leading blue-chip index for the Eurozone. The Euro Stoxx 50 covers stock data of companies from 11 countries of the Eurozone (Stoxx, 2017). Since the companies of this index are leading companies on a global scale, the average company had about 8,000 options listed on its equity on European exchanges over the sample period. With daily data on all of these options over the entire sample period for multiple variables the initial unfiltered sample contains 3.5 billion data points.

In calculating the implied volatility smirk I follow Xing et al. (2010). All implied volatilities used for this research are annualized values. The Implied volatility smirk measure is the difference between the implied volatility of an OTM option minus the implied volatility of an ATM option:

$$CALL\ SKEW_{i,t} = VOL_{i,t}^{OTMC} - VOL_{i,t}^{ATMC} \quad (3.1)$$

In this equation $CALL\ SKEW_{i,t}$ indicates the smirk in call options for firm i in week t . $VOL_{i,t}^{OTMC}$ and $VOL_{i,t}^{ATMC}$ are the implied volatilities of an OTM and an ATM call options. The smirk measure for put options is then:

$$PUT\ SKEW_{i,t} = VOL_{i,t}^{OTMP} - VOL_{i,t}^{ATMP} \quad (3.2)$$

In this equation $PUT\ SKEW_{i,t}$ indicates the smirk in put options for firm i in week t . The implied volatility of an OTM put options is denoted by $VOL_{i,t}^{OTMP}$, whereas the implied volatility of an ATM put option is denoted by $VOL_{i,t}^{ATMP}$.

An option is defined at the money when the ratio of exercise price to stock price falls between 0.95 and 1.05. A call option is defined out of the money when the ratio of exercise price to stock price is higher than 1.05, but does not exceed 1.20. Lastly, a put option is

defined out of the money when the ratio of exercise price to stock price falls between 0.8 and 0.95. This ensures that only more liquid options are included in the final sample.

Empirically implied volatility smiles become more pronounced with decreasing maturity. Decreasing maturity is also linked to increasing liquidity. For those two reasons I only include options with a time to expiration of 10 to 60 days. Xing et al. (2010) use a similar measure. Other measures are also common: for instance, An, Ang, Bali, & Cakici (2014) – who investigate the predictive powers of increases in implied volatilities – look on a daily basis at options with 30 days left to maturity. Combining such a strict measure with my moneyness restriction would leave too little observations for my final data sample.

Combining the maturity and the moneyness criteria may have as a result that on certain days within the sample period there are no suitable options for a given company. However, in most cases there will be more than one option that will meet the criteria. When this is the case I select ATM options that have a ratio that is closest to 1. For OTM options I select call

Summary Statistics						
Variable	Mean	5%	25%	50%	75%	95%
Call Skew	-0.007	-0.041	-0.017	-0.007	0.002	0.027
Put Skew	0.036	0.009	0.024	0.032	0.044	0.076
Stock Return	0.002	-0.063	-0.020	0.002	0.024	0.065
Lagged Return	0.057	-0.234	-0.051	0.049	0.149	0.342
Stock Volatility	0.273	0.137	0.189	0.241	0.314	0.521
Volatility Premium	0.007	-0.113	-0.024	0.010	0.048	0.113
Size	30.2	10.2	20.8	33.9	52.3	82.2

Table 1 – a representation of the summary statistics of all variables used

options with a moneyness ratio closest to 1.05 and put options with a moneyness ratio closest to 0.95. Based on a daily selection of options with non-negative open interest I construct an average with a weekly skew measure as a result. In the case that there is no suitable option for a certain company on a given day, that day will not be taken into account. This is a measure that is also used by Xing et al. (2010) and Ofek et al. (2004). An alternative measure is a volume-weighted volatility skew. However, Eikon retrieved

primarily missing values for daily traded volumes of the options in my sample, making a volume-weighted measure unreliable. Xing et al. (2010) include both measures in their research, but don't find significantly different results.

The weekly skew is calculated from Wednesday till Tuesday. The reason for choosing this timeframe over Monday till Friday is that there must be as little time as possible between the skew measure and the stock return. If I research the effect of a skew based on Monday till Friday on the stock return of next week from Monday till Friday, the jump with which the stock exchange opens on Monday may affect the results of the analysis.

Table 1 represents the summary statistics of all variables used for the analyses. The average of the call skew is negative, but very close to zero. The percentiles indicate a distribution where negative values are more pronounced than positive values. The put skew measure has a mean of 0.036, indicating an average 3.6 percent point difference between out of the money put options and in the money put options. The average weekly stock return is 0.2%. The returns closely resemble a normal distribution.

3.2 Control Variables

To isolate the effect of the put and call skew as much as possible I control for four characteristics that may have an influence on the stock return. All variables are also shown in table 1. The first control is the lagged return. This is the stock return over the past 6 months and is used to control for the momentum effect. Jegadeesh and Titman (1993) showed that strategies based on momentum can generate significant abnormal returns. The average lagged return is 5.7% over half a year, with 90% of all observations lie between a negative return of -23.4% and a positive return of 34.2%. The second variable used for controlling the effect of skewness is the stock volatility. Ang, Hodrick, Xing, & Zhang (2006) have linked higher values of historical stock volatility to significantly lower stock returns. The historical stock volatility in this thesis is the annualized volatility, based on one month of stock returns. Volatility premium is the third control variable. This variable is constructed by calculating the difference between the implied volatility of ATM call options – the most liquid option available - and the annualized historical stock volatility. The reason for including this variable is to control for volatility risk premium. Bakshi and Kapadia (2003) found that this premium is negative and differs significantly from zero. Investors are likely to pay extra for options with volatile underlying assets. The difference between historical volatility and implied volatilities tend to be a measure of investors beliefs and may be an indication of

future returns. The mean of the difference in ATM call options and historical stock volatility is in this sample 0.7%. 90% of all measurements fall in a range of -11.3% and 11.3%. This is a measure that very much depends on how you calculate historical volatility. As mentioned in chapter 2 of this thesis, there is a vast range of ways in which historical volatility can be calculated. The last control variable is company size measured in the natural log of the market capitalization. A measure that has often been linked to returns. Naturally this data sample contains only large companies. Something that is the direct consequence of requiring an underlying asset that has multiple liquid options on daily observations. The average company size in this sample is 30.2 billion euro. The lowest 5 percentile has a market cap of 10.2 billion euro, whereas the highest 95 percentile has a market cap of 82.2 billion euro.

The VBA code that processes the 3.5 billion data points to produce the final file is included in the appendix of this thesis.

4. Methodology

The method of analysis for this thesis can be divided in two parts. The first part is a Fama-MacBeth (FM) regression and is performed in MATLAB. The second part consists of constructing a portfolio to check for abnormal results.

4.1 Fama-MacBeth Regression

A FM regression is a regression that is a common method for estimating the risk premium for any sort of risk factors. It allows for time-varying coefficients, which becomes clearer in the following explanation. The regression consists of two stages. The first stage is a cross section. The equation for this stage is as follows:

$$STOCK\ RETURN_{i,t} = \beta_{0,t} + \beta_{1,t}SKEW_{i,t-1} + \beta_{2,t}CONTROLS_{i,t-1} + e_{i,t} \quad (4.1)$$

In this equation $STOCK\ RETURN_{i,t}$ is the stock return in week t in percent of company i . Since the goal is to analyse the predictive power of the skew in implied volatility, all variables are lagged by one week. This means that $SKEW_{i,t-1}$ is the skew measure of the implied volatility of company i in week $t-1$, the week previous to the stock return before the equal sign. Both Xing et al. (2010) and Feng et al. (2015) use a similar method, with the exception that Feng et al. look at monthly stock returns and monthly skew measures.

This analysis results in a time series of 260 different slope coefficients and 260 different error terms. The second and final stage is averaging all individual coefficients of all variables and error terms for each individual time period, under the assumption that coefficients are independent and identically distributed over time. I also account for autocorrelation and heteroscedasticity in my sample, by applying a Newey-West adjustment. The MATLAB code that performs all tests and the results of the Newey-West adjusted t-scores are included in the appendix of this thesis. The Newey-West support script for my analysis was written by J. P. LeSage⁶. However, the weakness of Fama-Macbeth that remains is that you can't exploit time series variation in the independent variables and how those relate to the dependent variable.

4.2 Portfolio Construction

A more heuristic approach is to display to predictive power of the skew measure is to construct portfolios on a weekly basis. The stocks will be sorted in five portfolios of ten stocks each. A process that will be done for both the call skew, as well as the put skew. This results in 10 portfolios for every week. The trading strategy based on the call skew will be completely independent of the trading strategy based on the put skew. In case of missing values, the top ranking and lowest ranking portfolios are still constructed. The middle portfolio will contain less than 10 stocks. Xing et al. (2010) and Feng et al. (2015) also apply this quintile portfolio construction strategy, but have a bigger sample size. Bigger samples allow for more diversification and thus reduce firm-level noise in each portfolio. Due to the scope of this thesis and constraints of data availability I am forced to a sample of 50 companies.

Since a high call skew may be the result of informed traders who buy OTM call options – expecting to profit from future stock returns – the strategy follows the pattern of buying the portfolio of stocks with the biggest skews in call options and shorting the portfolio of stocks with the smallest skews in call options. The resulting return is tested for significance. Naturally, the strategy based on the put skew does the exact opposite, because a high skew may result from informed traders buying OTM put options expecting to profit from a price drop in the underlying stock.

⁶ The original source file can be found on: <http://www.spatial-econometrics.com/regress/nwest.m>

5. Results

5.1 Fama-MacBeth Results

The results of the FM regression are reported in table 2 below. All dependent variables are regressed with a one period lag. Table 2 also reports the results of a FM regression where the t-stat scores are Newey-West adjusted, to account for autocorrelation and heteroscedasticity in the error term of the equation. The coefficients are the result of an analysis where no companies are dropped from the sample due to missing values. Analyses with three different filters are included in the appendix. These filters differ in the approach of missing values. The strictest filter drops all firms with at least one missing value, whereas two other filters drop all firms with at least five or ten missing values respectively. Although the number of firms changes, the filters do not seem to yield distinctively different results in the coefficients of interest.

Panel A of table 2 analyses the predictive power of the call skew, with and without controls, on the weekly stock return. The coefficient of the first regression shows a coefficient of 0.0304. This indicates that as the difference between implied volatility of OTM and ATM call options increases, the return of stocks in the next week also increase. The coefficient means that for every percentage point increase in the call skew, the returns increase by 3.04 basis points, equating to 1.6% on a yearly basis. Then the yearly interquartile difference – reported in the summary statistics in table 1 – in return is $((0.2 - (-1.7)) * (0.000304)) + 1)^{52} - 1 = 3.05\%$. However, the result is with a t-stat of 1.1423 not significant. After controlling for company size, returns in the past 6 months, stock volatility and the volatility premium, both the coefficient of the call skew and the t-stat of the call skew decrease. Although the coefficient of the call skew is of the expected sign, the only variable that is significant is company size. An increase in company size seems to lead to lower future returns, which is in line with economic theory.

Panel B of table 2 reports the predictive power of the put skew. The first regression solely analyzes the predictive power of the put skew, whereas the second regression in the panel includes the four control variables. The first regression shows a coefficient of -0.0316, indicating the preferred sign. Again, this is in line with economic theory; informed traders tend to exploit their information advantage through more leveraged financial products and higher buying pressure of OTM put options leads to an increase in the put skew. Hence, a relatively high implied volatility for OTM put options may result in lower future returns. The

coefficient of -0.0296 indicates that a percentage point increase in implied volatility difference results in a return that is 2.96 basis points lower in the subsequent week. Over a year the return would equate to -1.53% difference in return for just one percentage point

Fama-MacBeth Regression							
	Call Skew	Put Skew	Log Size	Lagged Return	Volatility Stock	Volatility Premium	R ²
<i>Panel A. FM-regression with call skew</i>							
Coefficient	0.0304						0.011
T-statistic	1.1423						
NW-adjusted	1.0703						
Coefficient	0.0133		-0.0014	-0.0020	-0.0010	0.0111	0.320
T-statistic	0.6411		-2.3210**	-0.6787	-0.1210	1.2323	
NW-adjusted	0.5887		-2.2372**	-0.7345	-0.1081	1.1337	
<i>Panel B. FM-regression with put skew</i>							
Coefficient		-0.0296					0.005
T-statistic		-1.5088					
NW-adjusted		-1.6973					
Coefficient		-0.0102	-0.0014	-0.0018	-0.0026	0.0112	0.315
T-statistic		-0.7738	-2.4379**	-0.6000	-0.3008	1.2405	
NW-adjusted		-0.7592	-2.3837**	-0.6607	-0.2785	1.1700	
<i>Panel C. FM-regression with call skew and put skew</i>							
Coefficient	0.0124	-0.0316					0.019
T-statistic	0.4471	-1.5504					
NW-adjusted	0.4165	-1.7062					
Coefficient	-0.0052	-0.0094	-0.0014	-0.0022	-0.0019	0.0110	0.324
T-statistic	-0.2424	-0.6988	-2.2893**	-0.7444	-0.2225	1.1629	
NW-adjusted	-0.2271	-0.6863	-2.2123**	-0.8071	-0.1996	1.0643	

*, **, and *** indicate significance at the 10%, 5% and 1% confidence level respectively.

Table 2 – a representation of the coefficients and t-stats analyzed through a FM-regression

difference in the put skew. The interquartile difference is 2 percentage points, which means that the yearly return would be $((4.4 - (-1.7)) * (-0.000296)) + 1)^{52} - 1 = -3.03\%$. But, as well as with coefficient of the call skew, the coefficient is not significant: the t-statistic has a value of -1.5088. When the control variables are included in the equation the effect is very similar to the regression results of panel A. The value of the coefficient is closer to zero and the t-stat also becomes smaller. Company size again seems to have a significant predictive power for next week's returns. So, in both cases the sign of the coefficient of the skew is in line with economic theory, but its predictive power is not significant.

Panel C combines both skew measures in one regression. Without control variables the coefficient of the call skew is 0.0124 and the coefficient of the put skew is -0.0316. With t-statistics of respectively 0.4471 and -1.5504 they both do not function as significant predictors for the stock returns in the subsequent week. When the control variables are introduced, the coefficients of the call and put skew have values closer to zero, with values of -0.0052 and -0.0094 respectively. Although the coefficient of the call skew is now negative, its t-statistic has a value of -0.2424, indicating that it is far from significant. The t-statistic of the put skew also shows a decrease to -0.6988. Again, size is significant at the 5% confidence level and provides evidence toward the economic theory that returns of bigger firms are lower than those of smaller firms.

Adjusting for autocorrelation and heteroscedasticity seems to have very little effect. In all panels the Newey-West adjusted t-statistics can be found below the normal t-statistics. Significance of both the skew measures and control variables are very similar: the skew measures remain insignificant and drop even further in significance when the control variables are introduced.

5.2 Portfolio Strategy Results

The portfolio construction strategy depends on the skew measures. Every week ten portfolios are constructed. Five portfolios are constructed from the entire data sample based on the call skew measure and five portfolios are constructed from that same data sample based on the put skew measure. Table 3 reports the results of the portfolio construction. Panel A reports the results of the portfolios based on the call skew measures. All numbers are averages of the 260 portfolios that are constructed over every week. Portfolio 1, on the first line of panel A, contains the stocks of the firms of which the options have the highest difference in implied volatility between ATM and OTM. Portfolio 5, then, contains those

Portfolio Strategy							
	Stock Return	Call Skew	Put Skew	Log Size	Lagged Return	Volatility Stock	Volatility Premium
<i>Panel A. Portfolio constructions based on call skew</i>							
Portfolio 1	0.02%	0.18%	0.35%	10.161	0.45%	2.45%	-0.02%
Portfolio 2	0.03%	-0.02%	0.33%	10.267	0.59%	2.59%	-0.02%
Portfolio 3	0.02%	-0.08%	0.33%	10.364	0.65%	2.66%	0.04%
Portfolio 4	0.02%	-0.13%	0.33%	10.379	0.73%	2.74%	0.10%
Portfolio 5	0.02%	-0.30%	0.41%	10.374	0.47%	3.18%	0.24%
Low – High	0.00%	0.48%					
<i>Panel B. Portfolio construction based on put skew</i>							
Portfolio 1	0.10%	-0.87%	6.71%	10.397	3.07%	26.61%	1.42%
Portfolio 2	0.16%	-0.82%	3.90%	10.365	5.11%	26.50%	0.19%
Portfolio 3	0.20%	-0.67%	3.30%	10.395	6.67%	26.10%	0.43%
Portfolio 4	0.35%	-0.66%	2.77%	10.250	6.98%	27.34%	0.08%
Portfolio 5	0.24%	-0.36%	0.79%	10.149	7.12%	29.48%	1.28%
Low – High	0.14%		5.92%				

Table 3 – a representation of the portfolios constructed based on the weekly skew measures stocks with the smallest difference. The average difference in skew between portfolio 1 and portfolio 5 over the 5 year period is 0.48%. The left column – the average stock returns – seems to confirm the results from the FM-regression. Sorting based on the call skew does not seem to cause a pattern in returns in my sample. The difference in stock return is less than 0.01% on a weekly basis. The averages of the log of the company size seem to be slightly higher in the portfolios with lower call skews. Lagged stock return shows no distinct pattern over the portfolios. It is slightly higher in portfolio 3 and portfolio 4, but there seems to be no reason to link this to call skewness. The historical volatility of the stock and the volatility premium are higher in the portfolios with lower call skew measures. Hence, the options of firms with low call skew measures are relatively expensive compared to historical stock volatility. Although historical stock volatility, call skew and volatility premium are related, the FM-regressions in previous section indicated that none of these variables harnessed any form of predictive power in this sample.

Panel B reports the result of the portfolios constructed on the basis of the put skew measure. Portfolio contains the 10 stocks of which the options have the biggest difference between OTM put options and ATM put options. Following this strategy produces clearer results when it comes to average weekly stock returns. The first two portfolios have lower weekly returns than the last two portfolios. However, the highest returns are concentrated in portfolio 4, instead of portfolio 5. Despite that the distribution of returns is not exactly as to be expected based on the relevant literature, the average difference in weekly returns between portfolio 1 and portfolio 5 is still 0.14%. Following a strategy where portfolio 1 would be sold and for the same amount of capital portfolio 5 would be purchased, and would be rebalanced once per week based on the put skew measure, delivers an abnormal excessive return of 14 bp per week. This equates to 7.55% on an annual basis, if transaction costs are not taken into account. Unfortunately, the FM-regression in the previous section shows that the put skew measure does not function as a reliable predictor for next week's stock returns in this sample. The t-statistic falls even further when company size is introduced as a control variable. The portfolios with lower put skew values also seem to be smaller in company size, than the portfolios with high put skew values. I cannot explain from the perspective of economic theory how and if these two variables are related. The portfolios with lower put skew values show slightly higher average values for historical stock volatility. An explanation may be that an increase in average historical stock volatility leads to a relatively bigger increase in ATM put options than it does in OTM put options. However, to draw conclusion would require additional research. The last column reports the average volatility premium. A clear trend is not visible; only the first and last portfolio have higher values.

Based on the FM-regression and the portfolio strategy both the put skew as well as the call skew do not show significant results. Although the t-statistic of the put skew measure comes close to significance, it shows a notable drop once the control variable company size is introduced in the equation. The portfolio construction strategy – reported in table 3 – suggests that the put skew measure may lead abnormal returns, but the FM-regression indicated that this effect was due to company size.

6. Conclusion and discussion

The Black-Scholes model (Black & Scholes, 1973; Merton, 1973) forms the very basis of option pricing. The model is used for calculation the price of a European option of which the underlying follows a geometric Brownian motion stochastic process. Since the exercise price does not influence the implied volatility of an option in this model, the implied volatility surface over the option exercise prices should be flat. However, Bates (1991) was among the first to report that this was not the case. Many other economists followed and showed that the volatility smile or volatility smirk was present among options with a wide variation of underlying assets. Gârleanu et al. (2009) construct an equilibrium model in which end-user demand affect an option's price. The implication of this is that the existence of a volatility smile may be explained due to informed traders. A group of investors who exploit their informational advantage by purchasing highly leveraged financial products. The magnitude of difference in implied volatility of ATM and OTM options may then function as a proxy for the presence of informed traders, which in turn means that this difference may function as a predictor for future stock returns. This thesis researches the predictability of the difference in implied volatility between OTM options and ATM on the stock returns on a weekly basis. Xing et al. (2010) report that the difference between the OTM put options and ATM call options functions as a significant predictor. Using similar techniques on a sample of options on stocks which make up the Euro Stoxx 50 I find that a similar predictability does not hold. A FM-regression indicates that the call skew measure has very little significance, whereas the reported t-statistic of the put skew measure is almost significant. However, the introduction of control variables drastically decreases the significance of the predictive power of the put skew. The reasons for the different results are not unambiguous. My sample runs over a different time period, which may have as a result that the effect has disappeared. Despite the time period difference, my best guess why results are not significant is related to firm-level noise. The portfolios in the paper by Xing et al. (2010) consist of 168 stocks each, whereas the portfolios in this thesis consist of 10 stocks each. Thus, the portfolios in the former research are more diversified. Smaller portfolios may cause firm-level noise to dominate the effect of the skew measures, even if the effect is actually present.

If the predictive effect of the implied volatility skew indeed exists, future research could indicate where the point comes where the effect of the implied volatility skew dominates

individual firm noise. Also at this point it is not yet clear what effect the portfolio size has on how long the predictability of the skew measure lasts.

References

- Alexander, C. (2001). Principal Component Analysis of Volatility Smiles and Skews. *EFMA 2001 Lugano Meetings*, 2000-2010.
- An, B.-J., Ang, A., Bali, T. G., & Cakici, N. (2014, October). The Joint Cross Section of Stocks and Options. *The Journal of Finance*, 2297-2328.
- Ang, A., Hodrick, R., Xing, Y., & Zhang, X. (2006). The Cross-Section of Volatility and Expected Returns. *Journal of Finance*, 259-299.
- Bakshi, G., & Kapadia, N. (2003). Delta-Hedged Gains and the Negative Market Volatility Risk Premium. *The Review of Financial Studies*, 527-566.
- Ball, R. J., & Brown, P. (1968). An empirical evaluation of accounting income numbers. *Journal of Accounting Research*, 159-178.
- Bates, D. S. (1991). The Crash of '87: Was It Expected? The Evidence from Options Markets. *The Journal of Finance*, 1009-1044.
- Bernard, V. L., & Thomas, J. K. (1989). Post-Earnings-Announcement Drift: Delayed Price Response or Risk Premium. *Journal of Accounting Research*, 1-48.
- Black, F. (1975). Fact and fantasy in use of options. *Financial Analysts Journal*, 36-41.
- Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *The journal of political economy*, 637-654.
- Bollen, N. P., & Whaley, R. E. (2004). Does Net Buying Pressure Affect the Shape of Implied Volatility Functions? *The Journal of Finance*, 711-753.
- Cao, C., Chen, Z., & Griffin, J. M. (2005). Informational Content of Option Volume Prior to Takeovers. *Journal of Business*, 1073-1109.
- Carr, P., & Wu, L. (2007). Stochastic Skew in Currency Options. *Journal of Financial Economics*, 213-247.
- Chalamandaris, G., & Tsekrekos, A. E. (2010). Predictable Dynamics in Implied Volatility Surfaces from OTC Currency Options. *Journal of Banking & Finance*, 1175-1188.
- Chan, W. (2003). Stock price reaction to news and no-news: drift and reversal after headlines. *Journal of Financial Economics*, 223-260.
- Conrad, J., Dittmar, R., & Ghysels, E. (2013). Ex Ante Skewness and Expected Stock Returns. *The Journal of Finance*, 85-124.
- Cremers, M., & Weinbaum, D. (2010). Deviations from Put-Call Parity and Stock Return Predictability. *Journal of Financial and Quantitative Analysis*, 335-367.

- Derman, E. (2003). Laughter in the Dark - the Problem of the Volatility Smile. *Lecture Notes, Master Program in Financial Engineering, Columbia University*, <http://www.ederman.com/new/-docs/laughter.html>.
- Doran, J. S., & Krieger, K. (2010). Implications for Asset Returns in the Implied Volatility Skew. *Financial Analysts Journal*, 65-76.
- Easley, D., O'Hara, M., & Srinivas, P. S. (1998). Option Volume and Stock Prices: Evidence on Where Informed Traders Trade. *The Journal of Finance*, 431-465.
- Feng, S., Zhang, Y., & Friesen, G. C. (2015). The relationship between the option-implied volatility smile, stock returns and heterogeneous beliefs. *International Review of Financial Analysis*, 62-73.
- Gârleanu, N., Pedersen, L. H., & Poteshman, A. M. (2009). Demand-Based Option Pricing. *Rev Financ Stud*, 4259-4299.
- Heston, S. (1993). A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *Journal of Finance*, 65-91.
- Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers—implications for stock-market efficiency. *The Journal of Finance*, 65-91.
- Kou, S. G. (2002). A Jump-Diffusion Model for Option Pricing. *Management Science*, 1086-1101.
- McDonald, R. L. (2014). Brownian Motion and Itô's Lemma. In R. L. McDonald, *Derivatives Markets: Pearson New International Edition* (pp. 617-640). Essex: Pearson Education Limited.
- McDonald, R. L. (2014). Volatility. In R. L. McDonald, *Derivatives Markets: Pearson International Edition* (pp. 731-761). Essex: Pearson Education Limited.
- Merton, R. C. (1973). Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 141-183.
- Natenberg, S. (1994). Volatility. In S. Natenberg, *Option Volatility & Pricing: Advanced Trading Strategies & Techniques* (pp. 51-80). United States of America: McGraw-Hill.
- Ofek, E., Richardson, M., & Whitelaw, F. (2004). Limited Arbitrage and Short Sales Restrictions: Evidence from the Options Markets. *Journal of Financial Economics*, 305-342.
- Pan, J., & Potesman, A. M. (2006). The Information in Option Volume for Future Stock Prices. *The Review of Financial Studies*, 871-908.
- Pena, I., Rubio, G., & Serna, G. (1999). Why do we smile? on the determinants of the implied volatility function. *Journal of Banking & Finance*, 1151-1179.

- Reid, S. (2015, April 7). *Asset Prices Simulated using Geometric Brownian Motion*. Retrieved May 3, 2017, from Turing Finance: <http://www.turingfinance.com/random-walks-down-wall-street-stochastic-processes-in-python/>
- Rubinstein, M. (1985). Nonparametric Tests of Alternative Option Pricing Models Using All Reported Trades and Quotes on the 30 Most Active CBOE Option Classes from August 23, 1976 Through August 31, 1978. *The Journal of Finance*, 455-480.
- Rubinstein, M. (1994). Implied Binomial Trees. *Journal of Finance*, 771-818.
- Sheikh, A. M. (1991). Transaction Data Tests of S&P 100 Call Option Pricing. *The Journal of Financial and Quantitative Analysis*, 459-475.
- Sitter, R. (2009, March 31). *Volatility Surface*. Retrieved from MathWorks: <https://nl.mathworks.com/matlabcentral/fileexchange/23316-volatility-surface>
- Sniper. (2002). *Stock market crash - Black Monday - October 1987*. Retrieved May 2, 2017, from Sniper: <http://www.sniper.at/stock-market-crash-of-1987.htm>
- Stoxx. (2017, July 13). *EURO STOXX 50*. Retrieved July 13, 2017, from Stoxx: <https://www.stoxx.com/index-details?symbol=SX5E>
- Trolle, A. B., & Schwartz, E. S. (2009). Unspanned Stochastic Volatility and the Pricing of Commodity Derivatives. *The Review of Financial Studies*, 4423-4461.
- Wallmeier, M., & Hafner, R. (2000). The Dynamics of DAX Implied Volatilities. *SSRN Electronic Journal*.
- Xing, Y., Zhang, X., & Zhao, R. (2010). What Does the Individual Option Volatility Smirk Tell Us About Future Equity Returns? *Journal of Financial and Quantitative Analysis*, 641-662.

Appendix

Alternative FM-Regressions

Fama-MacBeth Regression							
All companies with 1 missing value or more are filtered out, leaving 36 firms							
	Call Skew	Put Skew	Log Size	Lagged Return	Volatility Stock	Volatility Premium	
<i>Panel A. FM-regression with call skew</i>							
Coefficient	0.0214						
T-statistic	0.4810						
NW-adjusted	0.5024						
Coefficient	-0.0112		-0.0016	-0.0018	-0.0038	-0.0002	
T-statistic	-0.4065		-3.3644***	-0.4617	-0.3680	-0.0150	
NW-adjusted	-0.4119		-3.3784***	-0.4438	-0.3596	-0.0147	
<i>Panel B. FM-regression with put skew</i>							
Coefficient		-0.0341					
T-statistic		-1.2324					
NW-adjusted		-1.3606					
Coefficient		-0.0121	-0.0014	-0.0011	-0.0043	-0.0039	
T-statistic		-0.5268	-3.0414***	-0.2771	-0.4341	-0.3447	
NW-adjusted		-0.5463	-3.2941***	-0.2767	-0.4326	-0.3472	
<i>Panel C. FM-regression with call skew and put skew</i>							
Coefficient	0.0059	-0.0413					
T-statistic	0.1309	-1.5143					
NW-adjusted	0.1352	-1.7175					
Coefficient	-0.0173	-0.0156	-0.0015	-0.0016	-0.0054	-0.0037	
T-statistic	-0.6044	-0.6538	-3.1749***	-0.4116	-0.5317	-0.3268	
NW-adjusted	-0.5879	-0.6999	-3.3202***	-0.4021	-0.5345	-0.3356	

*, **, and *** indicate significance at the 10%, 5% and 1% confidence level respectively.

Table 4 – a representation of the coefficients and t-stats analyzed through a FM-regression on 260 weekly skews and 36 firms

Fama-MacBeth Regression

All companies with 5 missing value or more are filtered out, leaving 43 firms

	Call Skew	Put Skew	Log Size	Lagged Return	Volatility Stock	Volatility Premium
<i>Panel A. FM-regression with call skew</i>						
Coefficient	0.0001					
T-statistic	0.0020					
NW-adjusted	0.0020					
Coefficient	-0.0047		-0.0015	-0.0028	0.0011	0.0029
T-statistic	-0.2137		-3.6962***	-0.8540	0.1118	0.2911
NW-adjusted	-0.1834		-3.7149***	-0.8406	-0.1109	0.3035
<i>Panel B. FM-regression with put skew</i>						
Coefficient		-0.0191				
T-statistic		-0.8279				
NW-adjusted		-0.9382				
Coefficient		-0.0211	-0.0016	-0.0021	0.0004	-0.0002
T-statistic		-1.1249	-3.7215***	-0.6472	0.0430	-0.0231
NW-adjusted		-1.1196	-3.8337***	-0.6638	0.0446	-0.0246
<i>Panel C. FM-regression with call skew and put skew</i>						
Coefficient	-0.0120	-0.0179				
T-statistic	-0.3317	-0.7546				
NW-adjusted	-0.3258	-0.8623				
Coefficient	-0.0128	-0.0211	-0.0015	-0.0026	0.0003	-0.0003
T-statistic	-0.5601	-1.0714	-3.5731***	-0.7901	0.0261	-0.0255
NW-adjusted	-0.4922	-1.0997	-3.6694***	-0.7861	0.0266	-0.0272

*, **, and *** indicate significance at the 10%, 5% and 1% confidence level respectively.

Table 5 – a representation of the coefficients and t-stats analyzed through a FM-regression on 260 weekly skews and 43 firms

Fama-MacBeth Regression

All companies with 10 missing value or more are filtered out, leaving 45 firms

	Call Skew	Put Skew	Log Size	Lagged Return	Volatility Stock	Volatility Premium
<i>Panel A. FM-regression with call skew</i>						
Coefficient	0.0069					
T-statistic	0.1948					
NW-adjusted	0.1976					
Coefficient	-0.0100		-0.0012	-0.0017	-0.0043	0.0079
T-statistic	-0.4332		-2.0335**	-0.5332	-0.4663	0.7422
NW-adjusted	-0.3873		-1.8619*	-0.5571	-0.4409	0.7739
<i>Panel B. FM-regression with put skew</i>						
Coefficient		-0.0187				
T-statistic		-0.8189				
NW-adjusted		-0.9469				
Coefficient		-0.0164	-0.0012	-0.0011	-0.0031	0.0070
T-statistic		-0.8793	-2.0165***	-0.3604	-0.3295	0.6760
NW-adjusted		-0.8932	-1.8574*	-0.3900	-0.3358	0.7070
<i>Panel C. FM-regression with call skew and put skew</i>						
Coefficient	-0.0037	-0.0167				
T-statistic	-0.1025	-0.7029				
NW-adjusted	-0.1037	-0.8003				
Coefficient	-0.0176	-0.0145	-0.0011	-0.0015	-0.0038	0.0064
T-statistic	-0.7409	-0.7498	-1.8774*	-0.4727	-0.4013	0.6013
NW-adjusted	-0.6687	-0.7811	-1.7230***	-0.4987	-0.4019	0.6334

*, **, and *** indicate significance at the 10%, 5% and 1% confidence level respectively.

Table 6 – a representation of the coefficients and t-stats analyzed through a FM-regression on 260 weekly skews and 45 firms

MATLAB script for FM regression

FM-regression script

%%This script uses optiondata.mat and performs FamaMacBeth regressions on
%%set of Stock data with SCALL, SPUT and Control variables. In order to run
%%code just click Run from MATLAB Editor. One can change fnum parameter to
%%filter data as required. This file requires nwest.m file

%%Clear all data and load Option Data

clear all

load optiondata

clc

%%fnum defines number of blank data to filter data. Use 1 to drop all

%%stocks with at least 1 missing data while use 260 or more to use original

%%data. This section drops stocks from all variables and generates new data

fnum = 1;

filter = sum(Sput == 0) < fnum;

Lret = Lret(:,filter);

PVol = PVol(:,filter);

Scall = Scall(:,filter);

Sput = Sput(:,filter);

Size = Size(:,filter);

StockRet = StockRet(:,filter);

VolStock = VolStock(:,filter);

%%Define lag parameter for NW regressions.

lag = floor(4*(size(Sput,2)/100)^(2/9));

lag2 = floor(4*(259/100)^(2/9));

%%Define constant variables to use in FamaMacBeth regressions

onemat = ones(size(Sput,2),1);

onemat2 = ones(259,1);

%%Sample output variable to store all coefficients and tstats in one place

%%Copy and paste output and nwoutput to Excel

output = zeros(12,6);

nwoutput = zeros(12,6);

%%Fama Macbeth 1st step Regressions. i starts from week2 so that Y
%%(StockRet) is week 2(i) while other variables are week 1 (i-1)

```
for i=2:260
    %%StockRet vs SCALL
    Y = StockRet(i,:);
    X = [onemat Scall(i-1,:)];
    nwres = nwest(Y,X,lag);
    betaestonlycall(i-1,:) = nwres.beta';
end
```

```
for i=2:260
    %%StockRet vs SPUT
    Y = StockRet(i,:);
    X = [onemat Sput(i-1,:)];
    nwres = nwest(Y,X,lag);
    betaestonlyput(i-1,:) = nwres.beta';
end
```

```
for i=2:260
    %%StockRet vs SCALL and SPUT
    Y = StockRet(i,:);
    X = [onemat Scall(i-1,:) Sput(i-1,:)];
    nwres = nwest(Y,X,lag);
    betaestonlycallput(i-1,:) = nwres.beta';
end
```

```
for i=2:260
    %%StockRet vs SCALL and Control Variables
    Y = StockRet(i,:);
    X = [onemat Scall(i-1,:) Size(i-1,:) Lret(i-1,:) VolStock(i-1,:) PVol(i-1,:)];
    nwres = nwest(Y,X,lag);
    betaestcall(i-1,:) = nwres.beta';
end
```

end

for i=2:260

%%StockRet vs SPUT and Control Variables

Y = StockRet(i,:);

X = [onemat Sput(i-1,:) Size(i-1,:) Lret(i-1,:) VolStock(i-1,:) PVol(i-1,:)];

nwres = nwest(Y,X,lag);

betaestput(i-1,:) = nwres.beta';

end

for i=2:260

%%StockRet vs SCALL, SPUT and Control Variables

Y = StockRet(i,:);

X = [onemat Scall(i-1,:) Sput(i-1,:) Size(i-1,:) Lret(i-1,:) VolStock(i-1,:) PVol(i-1,:)];

nwres = nwest(Y,X,lag);

betaestcallput(i-1,:) = nwres.beta';

end

%%Mean of betas for each system

onlycallcoeff = mean(betaestonlycall);

onlyputcoeff = mean(betaestonlyput);

onlycallputcoeff = mean(betaestonlycallput);

callcoeff = mean(betaestcall);

putcoeff = mean(betaestput);

callputcoeff = mean(betaestcallput);

%%Tstat calculated by iid assumption.

onlycalltstat = mean(betaestonlycall)./std(betaestonlycall)*sqrt(259);

onlyputtstat = mean(betaestonlyput)./std(betaestonlyput)*sqrt(259);

onlycallputtstat = mean(betaestonlycallput)./std(betaestonlycallput)*sqrt(259);

calltstat = mean(betaestcall)./std(betaestcall)*sqrt(259);

puttstat = mean(betaestput)./std(betaestput)*sqrt(259);

callputtstat = mean(betaestcallput)./std(betaestcallput)*sqrt(259);

%%Store IID results to OUTPUT


```

output(1,1) = onlycallcoeff(2);
output(2,1) = onlycalltstat(2);
output(3,2) = onlyputcoeff(2);
output(4,2) = onlyputtstat(2);
output(5,1:2) = onlycallputcoeff(2:3);
output(6,1:2) = onlycallputtstat(2:3);
output(7,1) = callcoeff(2);
output(7,3:6) = callcoeff(3:6);
output(8,1) = calltstat(2);
output(8,3:6) = calltstat(3:6);
output(9,2) = putcoeff(2);
output(9,3:6) = putcoeff(3:6);
output(10,2) = puttstat(2);
output(10,3:6) = puttstat(3:6);
output(11,:) = callputcoeff(2:7);
output(12,:) = callputtstat(2:7);

```

%%IF not IID use NW regressions with LAG2

```

nwonlycall = nwest(betaestonlycall(:,2),onemat2,lag2);
nwonlyput = nwest(betaestonlyput(:,2),onemat2,lag2);
nwonlycallput(1) = nwest(betaestonlycallput(:,2),onemat2,lag2);
nwonlycallput(2) = nwest(betaestonlycallput(:,3),onemat2,lag2);

```

```

for i = 2:6

```

```

    nwcall(i-1) = nwest(betaestcall(:,i),onemat2,lag2);

```

```

    nwput(i-1) = nwest(betaestput(:,i),onemat2,lag2);

```

```

end

```

```

for i = 2:7

```

```

    nwcallput(i-1) = nwest(betaestcallput(:,i),onemat2,lag2);

```

```

end

```

%%Store non-IID results (NW regressions) to NWOUTPUT

```

nwoutput(1,1) = nwonlycall.beta;

```

```

nwoutput(2,1) = nwonlycall.tstat;

```

```

nwoutput(3,2) = nwonlyput.beta;
nwoutput(4,2) = nwonlyput.tstat;
nwoutput(5,1:2) = [nwonlycallput(1).beta nwonlycallput(2).beta];
nwoutput(6,1:2) = [nwonlycallput(1).tstat nwonlycallput(2).tstat];
nwoutput(7,1) = nwcall(1).beta;
nwoutput(7,3:6) = [nwcall(2).beta nwcall(3).beta nwcall(4).beta nwcall(5).beta];
nwoutput(8,1) = nwcall(1).tstat;
nwoutput(8,3:6) = [nwcall(2).tstat nwcall(3).tstat nwcall(4).tstat nwcall(5).tstat];
nwoutput(9,2) = nwput(1).beta;
nwoutput(9,3:6) = [nwput(2).beta nwput(3).beta nwput(4).beta nwput(5).beta];
nwoutput(10,2) = nwput(1).tstat;
nwoutput(10,3:6) = [nwput(2).tstat nwput(3).tstat nwput(4).tstat nwput(5).tstat];
nwoutput(11,:) = [nwcallput(1).beta nwcallput(2).beta nwcallput(3).beta nwcallput(4).beta
nwcallput(5).beta nwcallput(6).beta];
nwoutput(12,:) = [nwcallput(1).tstat nwcallput(2).tstat nwcallput(3).tstat nwcallput(4).tstat
nwcallput(5).tstat nwcallput(6).tstat];

```

Newey-West support script

```

function results=nwest(y,x,nlag)
% PURPOSE: computes Newey-West adjusted heteroscedastic-serial
%          consistent Least-squares Regression
%-----
% USAGE: results = nwest(y,x,nlag)
% where: y = dependent variable vector (nobs x 1)
%        x = independent variables matrix (nobs x nvar)
%        nlag = lag length to use
%-----
% RETURNS: a structure
%          results.meth = 'newlyw'
%          results.beta = bhat
%          results.tstat = t-stats
%          results.yhat = yhat
%          results.resid = residuals
%          results.sige = e'*e/(n-k)
%          results.rsqr = rsquared

```

```

%      results.rbar = rbar-squared
%      results.dw   = Durbin-Watson Statistic
%      results.nobs = nobs
%      results.nvar = nvars
%      results.y    = y data vector
% -----
% SEE ALSO: nwest_d, prt(results), plt(results)
%-----
% References: Gallant, R. (1987),
% "Nonlinear Statistical Models," pp.137-139.
%-----

% written by:
% James P. LeSage, Dept of Economics
% University of Toledo
% 2801 W. Bancroft St,
% Toledo, OH 43606
% % jlesage@spatial-econometrics.com


if (nargin ~= 3); error('Wrong # of arguments to nwest'); end;

[nobs nvar] = size(x);

results.meth = 'nwest';
results.y    = y;
results.nobs = nobs;
results.nvar = nvar;

xpxi = inv(x'*x);
results.beta = xpxi*(x'*y);
results.yhat = x*results.beta;
results.resid = y - results.yhat;
sigu = results.resid'*results.resid;

```

```

results.sige = sigu/(nobs-nvar);

% perform Newey-West correction
emat = [];
for i=1:nvar;
emat = [emat
        results.resid'];
end;

hhat=emat.*x';
G=zeros(nvar,nvar); w=zeros(2*nlag+1,1);
a=0;

while a~=nlag+1;
    ga=zeros(nvar,nvar);
    w(nlag+1+a,1)=(nlag+1-a)/(nlag+1);
    za=hhat(:,(a+1):nobs)*hhat(:,1:nobs-a)';
    if a==0;
        ga=ga+za;
    else
        ga=ga+za+za';
    end;
    G=G+w(nlag+1+a,1)*ga;
    a=a+1;
end; % end of while

V=xpxi*G*xpxi;
nwerr= sqrt(diag(V));

results.tstat = results.beta./nwerr; % Newey-West t-statistics
ym = y - ones(nobs,1)*mean(y);
rsqr1 = sigu;
rsqr2 = ym'*ym;
results.rsqr = 1.0 - rsqr1/rsqr2; % r-squared
rsqr1 = rsqr1/(nobs-nvar);

```

```
rsqr2 = rsqr2/(nobs-1.0);  
results.rbar = 1 - (rsqr1/rsqr2); % rbar-squared  
ediff = results.resid(2:nobs) - results.resid(1:nobs-1);  
results.dw = diag((ediff'*ediff)./(sigu))'; % durbin-watson
```

VBA code for option selection and portfolio construction

Initial selection code

Module: A_Globals

```
'=====
'GLOBAL VARIABLES USED IN INITIAL SELECTION MENU
'=====
Public G_Option_Type           As String    'Reads whether put or call is needed
Public G_Min_Mat               As Integer   'The minimum number of days to maturity
Public G_Max_Mat               As Integer   'The maximum number of days to maturity
Public G_Mon_Rat_Max_OTMC      As Double    'The required maximum moneyness ratio of
OTM Call option
Public G_Mon_Rat_Min_OTMC      As Double    'The required minimum moneyness ratio of
OTM Call option
Public G_Mon_Rat_Max_OTMP      As Double    'The required maximum moneyness ratio of
OTM Put option
Public G_Mon_Rat_Min_OTMP      As Double    'The required minimum moneyness ratio of
OTM Put option
Public G_Mon_Rat_Max_ATM       As Double    'The required maximum moneyness ratio of
ATM option
Public G_Mon_Rat_Min_ATM       As Double    'The required minimum moneyness ratio of
ATM option
Public G_Comp_Name             As String    'The company name of the options
Public G_Source_Path           As String    'Where to find source file
Public G_Source_Name           As String    'Where to find source file
Public G_Source                As String    'Name of source file
Public G_Report_Path           As String    'Where to find report template
Public G_Report_Name           As String    'Report Name
Public G_Report_Temp           As String    'Name of report file
Public G_Rep_Pre               As String    'Prefix used for saving reports
Public G_Report_Output         As String    'Path to output
Public G_Sheets                As Integer   'Number of sheets with codes
Public G_CPS                   As Integer   'Number of codes per sheet
Public G_CodeTab               As Integer   'Sheet containing all codes
'=====

'=====
'GLOBAL VARIABLES USED IN READING SOURCE FILE ||STATIC||
'=====
Public G_total_options         As Integer   '
Public G_option_code()         As String    'Option code as entered in to database
Public G_option_matdate()      As Date      'Option maturity date
Public G_option_name()         As String    'Option name
Public G_option_exerpr()       As Double    'Option exercise prise
Public G_option_exchange()     As String    'Exchange at which option is traded
Public G_put_call()            As String    'Contains whether option is a put or a call
'=====

'=====
'GLOBAL VARIABLES USED IN READING SOURCE FILE ||TIME SERIES||
'=====
Public G_total_dates           As Integer   'Total number of dates available in sample
Public G_dates()               As Date      'Used to store all dates available in sample
Public G_option_IV()           As Double    'Option implied volatility
Public G_option_OI()           As Double    'Option open interest
Public G_option_VM()           As Double    'Option volume traded
Public G_option_DZ()           As Double    'Option Delta
Public G_option_GM()           As Double    'Option Gamma
Public G_underlying()          As Double    'The price of the underlying asset
Public G_codes_last_sheet      As Integer   'Counter for number of options on last sheet
(loop purposes)
```

```

'=====
'
'=====
'GLOBAL VARIABLES USED FOR CONSTRUCTING NEW VARIABLES ||TIME SERIES||
'=====
Public G_time_to_mat()      As Integer      'Number that represents days left to
maturity
Public G_moneyness()        As Double      'Moneyness proxy for all dates for all
options
'=====

'=====
'GLOBAL VARIABLES USED FOR SELECTING OPTIONS FOR DATA SAMPLE
'=====
Public G_select_put()       As Boolean      'indicates whether option is selected or
not
Public G_select_call()      As Boolean      'indicates whether option is selected or
not
Public G_select_OTMP()      As Boolean      'indicates whether option is selected or
not
Public G_select_OTMC()      As Boolean      'indicates whether option is selected or
not
Public G_select_ATMP()      As Boolean      'indicates whether option is selected or
not
Public G_select_ATMC()      As Boolean      'indicates whether option is selected or
not
Public G_OTMP_mon()         As Double      'Contains closest OTM put moneyness
Public G_OTMC_mon()         As Double      'Contains closest OTM call moneyness
Public G_ATMP_mon()         As Double      'Contains closest ATM put moneyness
Public G_ATMC_mon()         As Double      'Contains closest OTM put moneyness
Public G_OTM_ATM()          As String      'Contains whether option is "OTM"
'=====

```

Module: A_Constants

```

'CONSTANTS USED FOR READING SOURCE FILE OF STATIC CODES
'=====
Public Const FIRST_DATA_ROW = 1
Public Const FIRST_DATA_COLUMN = 1
'=====

'CONSTANTS USED FOR READING SOURCE FILE, TS SECTION
'=====
Public Const FIRST_CODE_SHEET = 1
Public Const FIRST_TS_DATA_ROW = 3
Public Const FIRST_DATA_COLUMN_TS = 1
Public Const NO_DATA_COLUMN = 6
'=====

'CONSTANTS USED FOR REPORTING
'=====
Public Const NO_DATA_COLUMN_REP = 9
'=====

```

Module: A_Initial_Selection

```

'=====
'Performs all subroutines from reading selection menu to reporting the first selection
'=====
Sub Initial_Selection()
    Call Read_Menu_1

```

```

Call Read_Source_File_Static
Call Read_Source_File_TS
Call Construct_Mat_Time_and_Moneyness
Call SelectOptions1
Call SelectOptions2
Call WriteInitialRep

```

End Sub

Module: Read_Menu

```

'=====
'READS INITITAL SELECTION MENU
'=====
Sub Read_Menu_1()

    Sheets("Initial Selection").Select
    Range("A1").Select

    'Read in what type of option
    G_Option_Type = Trim(Range("PutCall"))

    'Read in maturity range in days
    G_Min_Mat = Trim(Range("MinMat"))           'Minimum maturity of options
    G_Max_Mat = Trim(Range("MaxMat"))           'Maximum maturity of options

    'Read in moneyness ratio for OTM/ATM options (=exercise prise/stock prise); <1 for
    put, >1 for call
    G_Mon_Rat_Max_OTMC = Trim(Range("MonRatMaxOTMC"))
    G_Mon_Rat_Min_OTMC = Trim(Range("MonRatMinOTMC"))
    G_Mon_Rat_Max_OTMP = Trim(Range("MonRatMaxOTMP"))
    G_Mon_Rat_Min_OTMP = Trim(Range("MonRatMinOTMP"))
    G_Mon_Rat_Max_ATM = Trim(Range("MonRatMaxATM"))
    G_Mon_Rat_Min_ATM = Trim(Range("MonRatMinATM"))

    'Reads in all information to find and open source file
    G_Comp_Name = Trim(Range("ComName"))           'Name of company
    G_Source_Path = Trim(Range("PathSource"))       'Find source path
    G_Source_Name = Trim(Range("NameSource"))
    G_Source = G_Source_Path & Trim(Range("NameSource")) 'Path + Source name

    'Reads in all information to find, open and save initial selection report
    G_Report_Path = Trim(Range("PathRepTemp1"))     'Find report path
    G_Report_Name = Trim(Range("RepTempName1"))
    G_Report_Temp = G_Report_Path & Trim(Range("RepTempName1")) 'Name of template
    report initial selection
    G_Rep_Pre = Trim(Range("RepPrefix"))             'Name of prefix used when saving
    reports
    G_Report_Output = Trim(Range("PathRepSave1"))    'Location where to save report

    'Reads in how many sheets with option codes there are and how many per page
    G_Sheets = Trim(Range("CodeSheets"))           'Number of sheets containing codes
    G_CPS = Trim(Range("CodesPS"))                 'Number of codes per sheet
    G_CodeTab = Trim(Range("CodeTab"))              'Sheet containing all codes

End Sub

```

Module: Read_Source

```

Dim j           As Long
Dim i2          As Long
Dim ir_ts       As Long
Dim ic_ts       As Long
Dim icol        As Long

```



```

Dim irow                As Long
Dim row_stop            As Long
Dim Last_TS_Sheet       As Integer
Dim isheet              As Integer
Dim row_stop_ts         As Long
Dim vlc                 As Long
Dim oic                 As Long
Dim vmc                 As Long
Dim dzc                 As Long
Dim oupc                As Long
Dim gmc                 As Long
Dim codes_last_sheet    As Long

```

```

'=====

```

```

'READS SOURCE FILE INTO VARIABLES ||STATIC||

```

```

'=====

```

```

Sub Read_Source_File_Static()

```

```

    'Open source file with name from menu and open first sheet

```

```

    Workbooks.Open Filename:=G_Source
    Sheets(G_CodeTab).Select
    Range("A1").Select
    j = 1

```

```

    'Start by counting number of options

```

```

    current_row = FIRST_DATA_ROW
    G_total_options = 0

```

```

    Do While Cells(current_row, FIRST_DATA_COLUMN).Value <> ""
        current_row = current_row + 1
        G_total_options = G_total_options + 1
    Loop

```

```

    row_stop = current_row - 1

```

```

    'Redimension the data arrays based on the number of total options in the source file

```

```

    ReDim G_option_code(G_total_options) As String
    ReDim G_option_matdate(G_total_options) As Date
    ReDim G_option_name(G_total_options) As String
    ReDim G_option_exerpr(G_total_options) As Double
    ReDim G_option_exchange(G_total_options) As String
    ReDim G_put_call(G_total_options) As String

```

```

    'Use function to read data of source file

```

```

    j = Read_Source_Data_Static(FIRST_DATA_ROW, row_stop, j)

```

```

End Sub

```

```

Function Read_Source_Data_Static(FIRST_DATA_ROW, row_stop, j)

```

```

    For i2 = FIRST_DATA_ROW To row_stop
        G_option_code(j) = Trim(Cells(i2, 1).Value)
        G_option_matdate(j) = Trim(Cells(i2, 4).Value)
        G_option_exchange(j) = Trim(Cells(i2, 5).Value)
        G_option_name(j) = Cells(i2, 6).Value
        G_option_exerpr(j) = Cells(i2, 3).Value

        If InStr(1, LCase(Cells(i2, 6)), LCase("put")) Then
            G_put_call(j) = "put"
        Else: G_put_call(j) = "call"
        End If

```

```

        j = j + 1
    Next i2

```

```

Next i2

Read_Portfolio_Data = j - 1

End Function

'=====
'READS SOURCE FILE INTO VARIABLES ||TIME SERIES||
'=====

Sub Read_Source_File_TS()

    'Open source file with name from menu and open first sheet

    Workbooks(G_Source_Name).Activate
    Sheets(FIRST_CODE_SHEET).Select
    Range("A1").Select

    'Start by counting the number of dates and write into report
    current_row_ts = FIRST_TS_DATA_ROW
    G_total_dates = 0

    Do While Cells(current_row_ts, FIRST_DATA_COLUMN_TS).Value <> ""
        current_row_ts = current_row_ts + 1
        G_total_dates = G_total_dates + 1
    Loop
    row_stop_ts = current_row_ts - 1

    'Redim required variables to contain information of all dates
    ReDim G_dates(G_total_dates) As Date
    ReDim G_option_IV(1 To G_total_dates, 1 To G_total_options) As Double
    ReDim G_option_OI(1 To G_total_dates, 1 To G_total_options) As Double
    ReDim G_option_VM(1 To G_total_dates, 1 To G_total_options) As Double
    ReDim G_option_DZ(1 To G_total_dates, 1 To G_total_options) As Double
    ReDim G_option_GM(1 To G_total_dates, 1 To G_total_options) As Double
    ReDim G_underlying(1 To G_total_dates, 1 To G_total_options) As Double

    'Read in dates into variable
    Sheets(FIRST_CODE_SHEET).Select
    ir_ts = 0

    For irow = FIRST_TS_DATA_ROW To row_stop_ts

        ir_ts = ir_ts + 1
        G_dates(ir_ts) = Cells(irow, FIRST_DATA_COLUMN).Value

    Next irow

    'Declare number of variables that are used for reading information of correct column
and row in loop
    row_stop_ts = FIRST_TS_DATA_ROW + G_total_dates
    Last_TS_Sheet = FIRST_CODE_SHEET + G_Sheets - 1
    G_codes_last_sheet = G_total_options - ((G_Sheets - 1) * G_CPS)

    ic_ts = 0

    'Three loops: outer goes through sheets, middle through columns on different pages
and inner through rows
    For isheet = FIRST_CODE_SHEET To Last_TS_Sheet
        Sheets(isheet).Select

        ir_ts = 0
        vlc = 2
        oic = 3

```

```

vmc = 4
dzc = 5
oupc = 6
gmc = 7

If isheet < Last_TS_Sheet Then

    For icol = 1 To G_CPS
        ic_ts = ic_ts + 1

        For irow = FIRST_TS_DATA_ROW To row_stop_ts
            ir_ts = ir_ts + 1

                If (IsNumeric(Trim(Cells(irow, vlc).Value))) Then G_option_IV(ir_ts,
ic_ts) = Cells(irow, vlc).Value
                If (IsNumeric(Trim(Cells(irow, oic).Value))) Then G_option_OI(ir_ts,
ic_ts) = Cells(irow, oic).Value
                If (IsNumeric(Trim(Cells(irow, vmc).Value))) Then G_option_VM(ir_ts,
ic_ts) = Cells(irow, vmc).Value
                If (IsNumeric(Trim(Cells(irow, dzc).Value))) Then G_option_DZ(ir_ts,
ic_ts) = Cells(irow, dzc).Value
                If (IsNumeric(Trim(Cells(irow, oupc).Value))) Then G_underlying(ir_ts,
ic_ts) = Cells(irow, oupc).Value
                If (IsNumeric(Trim(Cells(irow, gmc).Value))) Then G_option_GM(ir_ts,
ic_ts) = Cells(irow, gmc).Value

            Next irow

            vlc = vlc + NO_DATA_COLUMN
            oic = oic + NO_DATA_COLUMN
            vmc = vmc + NO_DATA_COLUMN
            dzc = dzc + NO_DATA_COLUMN
            oupc = oupc + NO_DATA_COLUMN
            gmc = gmc + NO_DATA_COLUMN
            ir_ts = 0

        Next icol

    Else

        For icol = 1 To G_codes_last_sheet
            ic_ts = ic_ts + 1

            For irow = FIRST_TS_DATA_ROW To row_stop_ts
                ir_ts = ir_ts + 1

                    If (IsNumeric(Trim(Cells(irow, vlc).Value))) Then G_option_IV(ir_ts,
ic_ts) = Cells(irow, vlc).Value
                    If (IsNumeric(Trim(Cells(irow, oic).Value))) Then G_option_OI(ir_ts,
ic_ts) = Cells(irow, oic).Value
                    If (IsNumeric(Trim(Cells(irow, vmc).Value))) Then G_option_VM(ir_ts,
ic_ts) = Cells(irow, vmc).Value
                    If (IsNumeric(Trim(Cells(irow, dzc).Value))) Then G_option_DZ(ir_ts,
ic_ts) = Cells(irow, dzc).Value
                    If (IsNumeric(Trim(Cells(irow, oupc).Value))) Then G_underlying(ir_ts,
ic_ts) = Cells(irow, oupc).Value
                    If (IsNumeric(Trim(Cells(irow, gmc).Value))) Then G_option_GM(ir_ts,
ic_ts) = Cells(irow, gmc).Value

                Next irow

                vlc = vlc + NO_DATA_COLUMN
                oic = oic + NO_DATA_COLUMN
                vmc = vmc + NO_DATA_COLUMN
                dzc = dzc + NO_DATA_COLUMN
                oupc = oupc + NO_DATA_COLUMN
                gmc = gmc + NO_DATA_COLUMN
                ir_ts = 0

            Next icol

        End If
    End If

```

```

        End If

    Next isheet

    ActiveWorkbook.Close False

End Sub

```

Module: Construct_Variables

```

'=====
'CONSTRUCTION OF NEW VARIABLES IS NEEDED FOR SELECTION PROCEDURE ||TIME SERIES||
'=====
Sub Construct_Mat_Time_and_Moneyiness()

'FIRST START WITH THE TIME LEFT TO MATURITY

'Redimension the variable to hold all information
ReDim G_time_to_mat(1 To G_total_dates, 1 To G_total_options) As Integer

'Construct a loop to calculate days left to maturity
For icol = 1 To G_total_options
    For irow = 1 To G_total_dates
        G_time_to_mat(irow, icol) = DateDiff("d", (G_dates(irow)),
        (G_option_matdate(icol)))
    Next irow
Next icol

'SECOND, CONSTRUCT MONEYNESS PROXY; EXERCISE PRISE/STOCK PRISE

'Redimension the variable to hold all information
ReDim G_moneyiness(1 To G_total_dates, 1 To G_total_options) As Double

'Construct a loop to calculate moneyiness for every option for every day
For icol = 1 To G_total_options
    For irow = 1 To G_total_dates
        G_moneyiness(irow, icol) = G_option_exerpr(icol) / G_underlying(irow, 1)
    Next irow
Next icol

End Sub

```

Module: Option_Selection

```

Sub SelectOptions1()

'First selection selects all options that pass initial criteria
ReDim G_select_put(1 To G_total_dates, 1 To G_total_options) As Boolean
ReDim G_select_call(1 To G_total_dates, 1 To G_total_options) As Boolean

For icol = 1 To G_total_options
    For irow = 1 To G_total_dates
        G_select_put(irow, icol) = True
        If G_time_to_mat(irow, icol) < G_Min_Mat Or _
            G_time_to_mat(irow, icol) > G_Max_Mat Or _
            UCase(G_put_call(icol)) <> UCase("PUT") Or _
            G_moneyiness(irow, icol) < G_Mon_Rat_Min_OTMP Or _
            G_moneyiness(irow, icol) > G_Mon_Rat_Max_ATM Or _
            G_option_IV(irow, icol) = 0 Then

```

```

        G_select_put(irow, icol) = False
    End If
Next irow
Next icol

```

```

For icol = 1 To G_total_options
    For irow = 1 To G_total_dates

```

```

        G_select_call(irow, icol) = True

```

```

        If G_time_to_mat(irow, icol) < G_Min_Mat Or _
            G_time_to_mat(irow, icol) > G_Max_Mat Or _
            UCase(G_put_call(icol)) <> UCase("CALL") Or _
            G_moneyness(irow, icol) > G_Mon_Rat_Max_OTMC Or _
            G_moneyness(irow, icol) < G_Mon_Rat_Min_ATM Or _
            G_option_IV(irow, icol) = 0 Then
            G_select_call(irow, icol) = False
        End If

```

```

    Next irow
Next icol

```

```

End Sub

```

```

Sub SelectOptions2()

```

```

    'Second selection leaves only 1 ATM option and 1 OTM option
    ReDim G_select_OTMP(1 To G_total_dates, 1 To G_total_options) As Boolean
    ReDim G_select_OTMC(1 To G_total_dates, 1 To G_total_options) As Boolean
    ReDim G_select_ATMP(1 To G_total_dates, 1 To G_total_options) As Boolean
    ReDim G_select_ATMC(1 To G_total_dates, 1 To G_total_options) As Boolean

```

```

    ReDim G_OTMP_mon(1 To G_total_dates) As Double
    ReDim G_OTMC_mon(1 To G_total_dates) As Double
    ReDim G_ATMP_mon(1 To G_total_dates) As Double
    ReDim G_ATMC_mon(1 To G_total_dates) As Double

```

```

    ReDim G_OTM_ATM(1 To G_total_dates, 1 To G_total_options) As String

```

```

    'Find the best fitting out of the money put option
    For irow = 1 To G_total_dates

```

```

        G_OTMP_mon(irow) = 1

```

```

        For icol = 1 To G_total_options

```

```

            G_select_OTMP(irow, icol) = True

```

```

            If G_select_put(irow, icol) = False Then
                G_select_OTMP(irow, icol) = False
            Else

```

```

                If G_moneyness(irow, icol) < G_Mon_Rat_Max_OTMP And _
                    (G_Mon_Rat_Max_OTMP - G_moneyness(irow, icol)) < G_OTMP_mon(irow) Then
                    G_OTMP_mon(irow) = G_Mon_Rat_Max_OTMP - G_moneyness(irow, icol)
                    G_OTM_ATM(irow, icol) = "OTM"
                Else: G_select_OTMP(irow, icol) = False
            End If

```

```

        End If
    Next icol
Next irow

```

```

For irow = 1 To G_total_dates
    For icol = 1 To G_total_options
        If (G_Mon_Rat_Max_OTMP - G_moneyness(irow, icol)) <> G_OTMP_mon(irow) Or _
            G_select_OTMP(irow, icol) = False Then
            G_select_OTMP(irow, icol) = False

```

```

        End If
    Next icol
Next irow

'Find the best fitting out of the money call option
For irow = 1 To G_total_dates

    G_OTMC_mon(irow) = 1

    For icol = 1 To G_total_options

        G_select_OTMC(irow, icol) = True

        If G_select_call(irow, icol) = False Then
            G_select_OTMC(irow, icol) = False
        Else
            If G_moneyness(irow, icol) > G_Mon_Rat_Min_OTMC And _
                (G_moneyness(irow, icol) - G_Mon_Rat_Min_OTMC) < G_OTMC_mon(irow) Then
                G_OTMC_mon(irow) = (G_moneyness(irow, icol) - G_Mon_Rat_Min_OTMC)
                G_OTM_ATM(irow, icol) = "OTM"
            Else: G_select_OTMC(irow, icol) = False
            End If
        End If

    Next icol
Next irow

For irow = 1 To G_total_dates
    For icol = 1 To G_total_options
        If (G_moneyness(irow, icol) - G_Mon_Rat_Min_OTMC) <> G_OTMC_mon(irow) Or _
            G_select_OTMC(irow, icol) = False Then
            G_select_OTMC(irow, icol) = False
        End If
    Next icol
Next irow

'Find the best fitting at of the money put option
For irow = 1 To G_total_dates

    G_ATMP_mon(irow) = 1

    For icol = 1 To G_total_options

        G_select_ATMP(irow, icol) = True

        If G_select_put(irow, icol) = False Then
            G_select_ATMP(irow, icol) = False
        Else
            If G_moneyness(irow, icol) < G_Mon_Rat_Max_ATM And _
                G_moneyness(irow, icol) > G_Mon_Rat_Min_ATM And _
                ((G_moneyness(irow, icol) - 1) * (G_moneyness(irow, icol) - 1)) <
G_ATMP_mon(irow) Then
                G_ATMP_mon(irow) = ((G_moneyness(irow, icol) - 1) * (G_moneyness(irow, icol) -
1))
                G_OTM_ATM(irow, icol) = "ATM"
            Else: G_select_ATMP(irow, icol) = False
            End If
        End If

    Next icol
Next irow

For irow = 1 To G_total_dates
    For icol = 1 To G_total_options
        If ((G_moneyness(irow, icol) - 1) * (G_moneyness(irow, icol) - 1)) <>
G_ATMP_mon(irow) Or _

```

```

        G_select_ATMP(irow, icol) = False Then
        G_select_ATMP(irow, icol) = False
    End If

    Next icol
Next irow

'Find the best fitting at the money call option
For irow = 1 To G_total_dates

    G_ATMC_mon(irow) = 1

    For icol = 1 To G_total_options

        G_select_ATMC(irow, icol) = True

        If G_select_call(irow, icol) = False Then
            G_select_ATMC(irow, icol) = False
        Else
            If G_moneyness(irow, icol) < G_Mon_Rat_Max_ATM And _
            G_moneyness(irow, icol) > G_Mon_Rat_Min_ATM And _
            ((G_moneyness(irow, icol) - 1) * (G_moneyness(irow, icol) - 1)) <
G_ATMC_mon(irow) Then
                G_ATMC_mon(irow) = ((G_moneyness(irow, icol) - 1) * (G_moneyness(irow, icol) -
1))
                G_OTM_ATM(irow, icol) = "ATM"
            Else: G_select_ATMC(irow, icol) = False
        End If
    End If

    Next icol
Next irow

```

```

For irow = 1 To G_total_dates
    For icol = 1 To G_total_options
        If ((G_moneyness(irow, icol) - 1) * (G_moneyness(irow, icol) - 1)) <>
G_ATMC_mon(irow) Or _
        G_select_ATMC(irow, icol) = False Then
            G_select_ATMC(irow, icol) = False
        End If
    Next icol
Next irow

```

End Sub

Module: Write_Report

```

Sub WriteInitialRep()

    'Open report

    Workbooks.Open Filename:=G_Report_Temp
    Sheets("Assumptions").Select
    Cells(4, 5).Value = G_total_options
    Cells(6, 5).Value = G_total_dates

    Sheets("Selection Calls").Select

    For irow = 1 To G_total_dates
        Cells((irow + 1), 1).Value = G_dates(irow)
        Cells((irow + 1), 2).Value = G_underlying(irow, 1)
    Next irow

```

```

Sheets("Selection Puts").Select

For irow = 1 To G_total_dates
    Cells((irow + 1), 1).Value = G_dates(irow)
    Cells((irow + 1), 2).Value = G_underlying(irow, 1)
Next irow

'Write the selection of calls into the report || ATM ||
Sheets("Selection Calls").Select

For irow = 1 To G_total_dates

    code = 3
    pc = 4
    vlc = 5
    oic = 6
    vmc = 7
    dzc = 8
    gmc = 9
    mat = 10
    moneyiness = 11

    For icol = 1 To G_total_options

        If G_select_ATMC(irow, icol) = True Then

            Cells((irow + 1), code).Value = G_option_name(icol)
            Cells((irow + 1), pc).Value = G_OTM_ATM(irow, icol) & " " &
UCase(G_put_call(icol))
            Cells((irow + 1), vlc).Value = G_option_IV(irow, icol)
            Cells((irow + 1), oic).Value = G_option_OI(irow, icol)
            Cells((irow + 1), vmc).Value = G_option_VM(irow, icol)
            Cells((irow + 1), dzc).Value = G_option_DZ(irow, icol)
            Cells((irow + 1), gmc).Value = G_option_GM(irow, icol)
            Cells((irow + 1), mat).Value = G_time_to_mat(irow, icol)
            Cells((irow + 1), moneyiness).Value = G_moneyiness(irow, icol)

            code = code + NO_DATA_COLUMN_REP
            pc = pc + NO_DATA_COLUMN_REP
            vlc = vlc + NO_DATA_COLUMN_REP
            oic = oic + NO_DATA_COLUMN_REP
            vmc = vmc + NO_DATA_COLUMN_REP
            dzc = dzc + NO_DATA_COLUMN_REP
            gmc = gmc + NO_DATA_COLUMN_REP
            mat = mat + NO_DATA_COLUMN_REP
            moneyiness = moneyiness + NO_DATA_COLUMN_REP
        End If

    Next icol

Next irow

'Write the selection of calls into the report || OTM ||

For irow = 1 To G_total_dates

    code = 12
    pc = 13
    vlc = 14
    oic = 15
    vmc = 16
    dzc = 17
    gmc = 18
    mat = 19
    moneyiness = 20

    For icol = 1 To G_total_options

        If G_select_OTMC(irow, icol) = True Then

```



```

        Cells((irow + 1), code).Value = G_option_name(icol)
        Cells((irow + 1), pc).Value = G_OTM_ATM(irow, icol) & " " &
UCase(G_put_call(icol))
        Cells((irow + 1), vlc).Value = G_option_IV(irow, icol)
        Cells((irow + 1), oic).Value = G_option_OI(irow, icol)
        Cells((irow + 1), vmc).Value = G_option_VM(irow, icol)
        Cells((irow + 1), dzc).Value = G_option_DZ(irow, icol)
        Cells((irow + 1), gmc).Value = G_option_GM(irow, icol)
        Cells((irow + 1), mat).Value = G_time_to_mat(irow, icol)
        Cells((irow + 1), moneyness).Value = G_moneyness(irow, icol)

```

```

        code = code + NO_DATA_COLUMN_REP
        pc = pc + NO_DATA_COLUMN_REP
        vlc = vlc + NO_DATA_COLUMN_REP
        oic = oic + NO_DATA_COLUMN_REP
        vmc = vmc + NO_DATA_COLUMN_REP
        dzc = dzc + NO_DATA_COLUMN_REP
        gmc = gmc + NO_DATA_COLUMN_REP
        mat = mat + NO_DATA_COLUMN_REP
        moneyness = moneyness + NO_DATA_COLUMN_REP
    End If

```

```
Next icol
```

```
Next irow
```

```

'Write the selection of puts into the report || ATM ||
Sheets("Selection Puts").Select

```

```
For irow = 1 To G_total_dates
```

```

    code = 3
    pc = 4
    vlc = 5
    oic = 6
    vmc = 7
    dzc = 8
    gmc = 9
    mat = 10
    moneyness = 11

```

```
For icol = 1 To G_total_options
```

```
    If G_select_ATMP(irow, icol) = True Then
```

```

        Cells((irow + 1), code).Value = G_option_name(icol)
        Cells((irow + 1), pc).Value = G_OTM_ATM(irow, icol) & " " &
UCase(G_put_call(icol))
        Cells((irow + 1), vlc).Value = G_option_IV(irow, icol)
        Cells((irow + 1), oic).Value = G_option_OI(irow, icol)
        Cells((irow + 1), vmc).Value = G_option_VM(irow, icol)
        Cells((irow + 1), dzc).Value = G_option_DZ(irow, icol)
        Cells((irow + 1), gmc).Value = G_option_GM(irow, icol)
        Cells((irow + 1), mat).Value = G_time_to_mat(irow, icol)
        Cells((irow + 1), moneyness).Value = G_moneyness(irow, icol)

```

```

        code = code + NO_DATA_COLUMN_REP
        pc = pc + NO_DATA_COLUMN_REP
        vlc = vlc + NO_DATA_COLUMN_REP
        oic = oic + NO_DATA_COLUMN_REP
        vmc = vmc + NO_DATA_COLUMN_REP
        dzc = dzc + NO_DATA_COLUMN_REP
        gmc = gmc + NO_DATA_COLUMN_REP
        mat = mat + NO_DATA_COLUMN_REP
        moneyness = moneyness + NO_DATA_COLUMN_REP
    End If

```

```
End If
```

```

Next icol

Next irow

'Write the selection of puts into the report || OTM ||
For irow = 1 To G_total_dates

    code = 12
    pc = 13
    vlc = 14
    oic = 15
    vmc = 16
    dzc = 17
    gmc = 18
    mat = 19
    moneyiness = 20

    For icol = 1 To G_total_options

        If G_select_OTMP(irow, icol) = True Then

            Cells((irow + 1), code).Value = G_option_name(icol)
            Cells((irow + 1), pc).Value = G_OTM_ATM(irow, icol) & " " &
UCase(G_put_call(icol))
            Cells((irow + 1), vlc).Value = G_option_IV(irow, icol)
            Cells((irow + 1), oic).Value = G_option_OI(irow, icol)
            Cells((irow + 1), vmc).Value = G_option_VM(irow, icol)
            Cells((irow + 1), dzc).Value = G_option_DZ(irow, icol)
            Cells((irow + 1), gmc).Value = G_option_GM(irow, icol)
            Cells((irow + 1), mat).Value = G_time_to_mat(irow, icol)
            Cells((irow + 1), moneyiness).Value = G_moneyiness(irow, icol)

            code = code + NO_DATA_COLUMN_REP
            pc = pc + NO_DATA_COLUMN_REP
            vlc = vlc + NO_DATA_COLUMN_REP
            oic = oic + NO_DATA_COLUMN_REP
            vmc = vmc + NO_DATA_COLUMN_REP
            dzc = dzc + NO_DATA_COLUMN_REP
            gmc = gmc + NO_DATA_COLUMN_REP
            mat = mat + NO_DATA_COLUMN_REP
            moneyiness = moneyiness + NO_DATA_COLUMN_REP

        End If

    Next icol

Next irow

ActiveWorkbook.SaveAs Filename:=G_Report_Output & G_Rep_Pre & G_Comp_Name
ActiveWorkbook.Close

End Sub

```

Report Transformer code

Module: A_Globals

```
'USED FOR READING MENU
Public G_SourcePath      As String
Public G_CompanyName     As String
Public G_SourcePrefix    As String
Public G_SavePath        As String
Public G_SavePrefix      As String

Public G_TotalSource     As String
Public G_SourceFile      As String
Public G_TotalSave       As String
Public G_ReportName      As String
Public G_TotalReport     As String

Public G_Equity_Path     As String
Public G_Equity_Rep_Name As String
Public G_Equity_Total    As String
Public G_Rep_Sheet       As Integer

'USED FOR READING SOURCE - R1_ Reports
Public G_Total_Dates As Integer
Public G_Dates() As Date

Public G_ATMC_Vol()      As Double
Public G_OTMC_Vol()      As Double
Public G_ATMP_Vol()      As Double
Public G_OTMP_Vol()      As Double

'USED FOR READING SOURCE - EQUITY REPORT
Public G_Total_Dates_EQ As Integer
Public G_Dates_EQ()     As Date
Public G_Stock_Price()  As Double
Public G_LN_MC()        As Double

'USED FOR CALCULATIONS
Public G_Call_Skew()     As Double
Public G_Put_Skew()      As Double

Public G_Stock_Ret()     As Double
```

Module: A_Constants

```
'USED FOR READING SOURCE FILE
Public Const FIRST_DATA_ROW = 2

'USED FOR WRITING THE REPORT
Public Const DATA_COLUMNS = 13
```

Module: A_RunModel

```
Sub RunModel()

Call ReadMenu
Call ReadSource
Call ReadSource2
```

```

Call CalculationsR1
Call WriteReport

End Sub

```

Module: ReadMenu

```

Sub ReadMenu()

'Read in variables from menu
G_SourcePath = Trim(Range("SourcePath"))
G_SourcePrefix = Trim(Range("SourcePrefix"))
G_CompanyName = Trim(Range("CompanyName"))
G_ReportName = Trim(Range("ReportName"))
G_SavePrefix = Trim(Range("SavePrefix"))
G_SavePath = Trim(Range("SavePath"))

G_Equity_Path = Trim(Range("EquityPath"))
G_Equity_Rep_Name = Trim(Range("EquityReport"))
G_Rep_Sheet = Trim(Range("ReqSheet"))

G_Equity_Total = G_Equity_Path & G_Equity_Rep_Name
G_TotalSource = G_SourcePath & G_SourcePrefix & G_CompanyName
G_TotalSave = G_SavePath & G_SavePrefix & G_CompanyName
G_TotalReport = G_SavePath & G_ReportName

End Sub

```

Module: Read_Source

```

'=====R1_REPORTS=====
Sub ReadSource()

Workbooks.Open Filename:=G_TotalSource
    Sheets(1).Select
    Range("A1").Select

i_date = FIRST_DATA_ROW
G_Total_Dates = 0

'READ IN DATES
Do While Cells(i_date, 1) <> ""
    i_date = i_date + 1
    G_Total_Dates = G_Total_Dates + 1
Loop

ReDim G_Dates(G_Total_Dates) As Date
ReDim G_ATMC_Vol(G_Total_Dates) As Double
ReDim G_OTMC_Vol(G_Total_Dates) As Double
ReDim G_ATMP_Vol(G_Total_Dates) As Double
ReDim G_OTMP_Vol(G_Total_Dates) As Double

For i = 1 To G_Total_Dates
    Sheets(1).Select

    G_Dates(i) = Cells((i + 1), 1).Value

    G_ATMC_Vol(i) = Cells((i + 1), 5).Value
    G_OTMC_Vol(i) = Cells((i + 1), 14).Value

    Sheets(2).Select
    G_ATMP_Vol(i) = Cells((i + 1), 5).Value
    G_OTMP_Vol(i) = Cells((i + 1), 14).Value

```

```
Next i
```

```
ActiveWorkbook.Close False
```

```
End Sub
```

Module: Calculations

```
Sub CalculationsR1()
```

```
'Calculate Skews on daily basis
```

```
ReDim G_Call_Skew(G_Total_Dates) As Double
```

```
ReDim G_Put_Skew(G_Total_Dates) As Double
```

```
For i = 1 To G_Total_Dates
```

```
If G_OTMC_Vol(i) <> 0 And G_ATMC_Vol(i) <> 0 Then  
    G_Call_Skew(i) = G_OTMC_Vol(i) - G_ATMC_Vol(i)
```

```
Else
```

```
    G_Call_Skew(i) = 0
```

```
End If
```

```
If G_OTMP_Vol(i) <> 0 And G_ATMP_Vol(i) <> 0 Then  
    G_Put_Skew(i) = G_OTMP_Vol(i) - G_ATMP_Vol(i)
```

```
Else
```

```
    G_Put_Skew(i) = 0
```

```
End If
```

```
Next i
```

```
Calculate Daily Stock Returns
```

```
ReDim G_Stock_Ret(G_Total_Dates_EQ) As Double
```

```
For i = 1 To G_Total_Dates_EQ
```

```
    If i < 2 Then
```

```
        G_Stock_Ret(i) = 0.01
```

```
    Else
```

```
        G_Stock_Ret(i) = (G_Stock_Price(i) / G_Stock_Price(i - 1)) - 1
```

```
    End If
```

```
Next i
```

```
End Sub
```

Module: Write_Rep

```
Sub WriteReport()
```

```
Workbooks.Open Filename:=G_TotalReport
```

```
    Sheets(1).Select
```

```
    Range("A1").Select
```

```
cn = 3 + (G_Rep_Sheet - 2) * DATA_COLUMNS
```

```
sc = 4 + (G_Rep_Sheet - 2) * DATA_COLUMNS
```

```
sp = 5 + (G_Rep_Sheet - 2) * DATA_COLUMNS
```

```
spr = 6 + (G_Rep_Sheet - 2) * DATA_COLUMNS
```

```
sr = 7 + (G_Rep_Sheet - 2) * DATA_COLUMNS
```

```
mc = 8 + (G_Rep_Sheet - 2) * DATA_COLUMNS
```

```
bm = 9 + (G_Rep_Sheet - 2) * DATA_COLUMNS
```

```
vp = 10 + (G_Rep_Sheet - 2) * DATA_COLUMNS
```

```
Row = 2
```

```
Row_eq = 2
```

```
For i = 89 To 1392

    Cells(Row, cn).Value = G_CompanyName
    Cells(Row, sc).Value = G_Call_Skew(i)
    Cells(Row, sp).Value = G_Put_Skew(i)

    Row = Row + 1
Next i

For i = 262 To 1565

    Cells(Row_eq, spr).Value = G_Stock_Price(i)
    Cells(Row_eq, sr).Value = G_Stock_Ret(i)
    Cells(Row_eq, mc).Value = G_LN_MC(i)

    Row_eq = Row_eq + 1
Next i

ActiveWorkbook.Close True

End Sub
```

Analysis File report code

Module: A_Globals

'USED FOR READING MENU

```
Public G_SourcePath As String
Public G_SourceName As String
Public G_Source As String
Public G_NoComp As Integer
```

```
Public G_RepLoc As String
Public G_RepName As String
Public G_Rep As String
```

'USED FOR READING SOURCE

```
Public G_TotalDates As Integer
Public G_Dates() As Date
Public G_Day() As String
Public G_Tues As Integer
```

```
Public G_Com_Name() As String
Public G_Skew_P() As Double
Public G_Skew_C() As Double
Public G_IV_P() As Double
Public G_IV_C() As Double
Public G_Stock_V() As Double
Public G_Stock() As Double
Public G_Return() As Double
Public G_LRet() As Double
Public G_LN_MC() As Double
```

'USED FOR CALCULATIONS

```
Public G_SWC() As Double
Public G_SWP() As Double
Public G_PVOL() As Double
Public G_WR() As Double
```

Module: A_Constants

'READ SOURCE

```
Public Const FDR = 2
Public Const FDC = 3
Public Const DATA_COL = 13
```

'WRITE REP

```
Public Const REP_COL = 8
```

Module: A_RunModel

```
Sub RunModel1()
```

```
    Call ReadMenu1
    Call ReadSource1
    Call CalculationsSkew
    Call CalculationRet
    Call Write_Rep
```

```
End Sub
```

Module: ReadMenu

```
Sub ReadMenu1()  
  
    G_SourcePath = Trim(Range("SourcePath"))  
    G_SourceName = Trim(Range("SourceName"))  
  
    G_Source = G_SourcePath & G_SourceName  
  
    G_NoComp = Trim(Range("NoComp"))  
  
    G_RepLoc = Trim(Range("LocRep"))  
    G_RepName = Trim(Range("RepName"))  
  
    G_Rep = G_RepLoc & G_RepName  
  
End Sub
```

Module: ReadSource

```
Sub ReadSource1()  
  
Workbooks.Open Filename:=G_Source  
Sheets(1).Select  
Range("A1").Select  
  
'Start reading dates and days  
i = 2  
G_TotalDates = 0  
G_Tues = 0  
  
Do While Cells((i + 1), 1) <> ""  
    G_TotalDates = G_TotalDates + 1  
    i = i + 1  
Loop  
  
ReDim G_Dates(G_TotalDates) As Date  
ReDim G_Day(G_TotalDates) As String  
  
ReDim G_Com_Name(1 To G_TotalDates, 1 To G_NoComp) As String 'check  
ReDim G_Skew_P(1 To G_TotalDates, 1 To G_NoComp) As Double 'check  
ReDim G_Skew_C(1 To G_TotalDates, 1 To G_NoComp) As Double 'check  
ReDim G_IV_P(1 To G_TotalDates, 1 To G_NoComp) As Double  
ReDim G_IV_C(1 To G_TotalDates, 1 To G_NoComp) As Double  
ReDim G_Stock_V(1 To G_TotalDates, 1 To G_NoComp) As Double  
ReDim G_Stock(1 To G_TotalDates, 1 To G_NoComp) As Double  
ReDim G_Return(1 To G_TotalDates, 1 To G_NoComp) As Double  
ReDim G_LRet(1 To G_TotalDates, 1 To G_NoComp) As Double  
ReDim G_LN_MC(1 To G_TotalDates, 1 To G_NoComp) As Double  
  
For i = 1 To G_TotalDates  
    G_Dates(i) = Cells((i + 1), 1).Value  
    G_Day(i) = Cells((i + 1), 2).Value  
  
    If G_Day(i) = "TUES" Then G_Tues = G_Tues + 1  
  
    For c = 1 To G_NoComp  
  
        cn = 3 + (c - 1) * DATA_COL  
        sc = 4 + (c - 1) * DATA_COL  
        sp = 5 + (c - 1) * DATA_COL  
        ivc = 6 + (c - 1) * DATA_COL  
        ivp = 7 + (c - 1) * DATA_COL  
        vs = 8 + (c - 1) * DATA_COL  
        'pvol = 9 + (c - 1) * DATA_COL  
        stpr = 10 + (c - 1) * DATA_COL
```



```

        sr = 11 + (c - 1) * DATA_COL
        lr = 12 + (c - 1) * DATA_COL
        lnmc = 13 + (c - 1) * DATA_COL

        G_Com_Name(i, c) = Cells((i + 1), cn).Value
        If IsNumeric(Cells((i + 1), sp).Value) Then G_Skew_P(i, c) = Cells((i + 1),
sp).Value
        If IsNumeric(Cells((i + 1), sc).Value) Then G_Skew_C(i, c) = Cells((i + 1),
sc).Value
        If IsNumeric(Cells((i + 1), ivp).Value) Then G_IV_P(i, c) = Cells((i + 1),
ivp).Value
        If IsNumeric(Cells((i + 1), ivc).Value) Then G_IV_C(i, c) = Cells((i + 1),
ivc).Value
        If IsNumeric(Cells((i + 1), vs).Value) Then G_Stock_V(i, c) = Cells((i + 1),
vs).Value
        'If IsNumeric(Cells((i + 1), pvol).Value) Then G_PVOL(i, c) = Cells((i + 1),
pvol).Value
        If IsNumeric(Cells((i + 1), strp).Value) Then G_Stock(i, c) = Cells((i + 1),
strp).Value
        If IsNumeric(Cells((i + 1), sr).Value) Then G_Return(i, c) = Cells((i + 1),
sr).Value
        If IsNumeric(Cells((i + 1), lr).Value) Then G_LRet(i, c) = Cells((i + 1),
lr).Value
        If IsNumeric(Cells((i + 1), lnmc).Value) Then G_LN_MC(i, c) = Cells((i + 1),
lnmc).Value

    Next c
Next i

For i = 1 To 260
    For c = 1 To 46
        Cells((1500 + i), c).Value = G_Com_Name(i, c)
    Next c
Next i

ActiveWorkbook.Close False

End Sub

```

Module: Calculations

```

Sub CalculationsSkew()

ReDim G_SWC(1 To G_Tues, 1 To G_NoComp) As Double
ReDim G_SWP(1 To G_Tues, 1 To G_NoComp) As Double
ReDim G_PVOL(1 To G_Tues, 1 To G_NoComp) As Double

'Calculate Average weekly call skew
For c = 1 To G_NoComp
    Csc = 0
    d = 0
    w = 0

    For i = 1 To G_TotalDates

        If G_Skew_C(i, c) <> 0 Then

            Csc = Csc + G_Skew_C(i, c)
            d = d + 1

        End If

        If UCase(G_Day(i)) = UCase("TUES") Then
            w = w + 1
            If d <> 0 Then G_SWC(w, c) = Csc / d Else G_SWC(w, c) = 0
            Csc = 0
            d = 0
        End If
    Next i
Next c

End Sub

```

```

        End If
    Next i
Next c

'Calculate weekly PVOL
For c = 1 To G_NoComp
    w = 0

    For i = 1 To G_TotalDates

        If UCase(G_Day(i)) = UCase("TUES") Then
            w = w + 1
            If G_IV_C(i, c) = 0 Then G_PVOL(w, c) = 0 Else G_PVOL(w, c) = G_IV_C(i, c) -
G_Stock_V(i, c)
            End If
        End If
    Next i
Next c

'Calculate Average weekly put skew
For c = 1 To G_NoComp
    Csc = 0
    d = 0
    w = 0

    For i = 1 To G_TotalDates

        If G_Skew_P(i, c) <> 0 Then

            Csc = Csc + G_Skew_P(i, c)
            d = d + 1

        End If

        If UCase(G_Day(i)) = UCase("TUES") Then
            w = w + 1
            If d <> 0 Then G_SWP(w, c) = Csc / d Else G_SWP(w, c) = 0
            Csc = 0
            d = 0
        End If
    Next i
Next c

End Sub

```

```

Sub CalculationRet()

ReDim G_WR(1 To G_Tues, 1 To G_NoComp) As Double

'Calculate Weekly returns
For c = 1 To G_NoComp
    Wr = 1
    w = 0

    For i = 1 To G_TotalDates

        Wr = Wr * (1 + G_Return(i, c))

        If UCase(G_Day(i)) = UCase("TUES") Then
            w = w + 1
        End If
    Next i
Next c

End Sub

```

```

        G_WR(w, c) = Wr - 1
        Wr = 1
    End If
Next i
Next c

End Sub

```

Module: WriteReport

```

Sub Write_Rep()

Workbooks.Open Filename:=G_Rep
Sheets(1).Select
Range("A1").Select

'Report T in weeks; starting with tuesday

w = 0

For i = 1 To 260
    w = w + 1
    Cells((i + 1), 1) = w
Next i

For c = 1 To 46

    cn = 2 + ((c - 1) * REP_COL)
    sr = 3 + ((c - 1) * REP_COL)
    sc = 4 + ((c - 1) * REP_COL)
    sp = 5 + ((c - 1) * REP_COL)
    pvcl = 9 + ((c - 1) * REP_COL)

    For i = 1 To 260

        Cells((i + 1), cn).Value = c
        Cells((i + 1), sr).Value = G_WR(i, c)
        Cells((i + 1), sc).Value = G_SWC(i, c)
        Cells((i + 1), sp).Value = G_SWP(i, c)
        Cells((i + 1), pvcl).Value = G_PVOL(i, c)

    Next i
Next c

For c = 1 To G_NoComp

    ls = 6 + ((c - 1) * REP_COL)
    lret = 7 + ((c - 1) * REP_COL)
    sv = 8 + ((c - 1) * REP_COL)

    w = 0

    For i = 1 To G_TotalDates

        If UCase(G_Day(i)) = UCase("TUES") Then
            w = w + 1
            Cells(w, lret).Value = G_LRet(i, c)
            Cells(w, ls).Value = G_LN_MC(i, c)
            Cells(w, sv).Value = G_Stock_V(i, c)

        End If

        If i = G_TotalDates Then
            w = w + 1

```

```
Cells(w, lret).Value = G_LRet(i, c)
Cells(w, ls).Value = G_LN_MC(i, c)
Cells(w, sv).Value = G_Stock_V(i, c)

End If

Next i

Next c

End Sub
```

Portfolio construction code

Module: A_Globals

```
'GLOBALS USED FOR READING SOURCE
Public G_ID()           As Integer
Public G_Stockret()     As Double
Public G_Scall()        As Double
Public G_Sput()         As Double
Public G_Size()         As Double
Public G_Lret()         As Double
Public G_Volstock()     As Double
Public G_Pvol()         As Double

'GLOBALS USED FOR PORTFOLIO CONSTRUCTION
Public G_P1_Stockret    As Double
Public G_P1_Scall       As Double
Public G_P1_Sput        As Double
Public G_P1_Size        As Double
Public G_P1_Lret        As Double
Public G_P1_Volstock    As Double
Public G_P1_Pvol        As Double

Public G_P2_Stockret    As Double
Public G_P2_Scall       As Double
Public G_P2_Sput        As Double
Public G_P2_Size        As Double
Public G_P2_Lret        As Double
Public G_P2_Volstock    As Double
Public G_P2_Pvol        As Double

Public G_P3_Stockret    As Double
Public G_P3_Scall       As Double
Public G_P3_Sput        As Double
Public G_P3_Size        As Double
Public G_P3_Lret        As Double
Public G_P3_Volstock    As Double
Public G_P3_Pvol        As Double

Public G_P4_Stockret    As Double
Public G_P4_Scall       As Double
Public G_P4_Sput        As Double
Public G_P4_Size        As Double
Public G_P4_Lret        As Double
Public G_P4_Volstock    As Double
Public G_P4_Pvol        As Double

Public G_P5_Stockret    As Double
Public G_P5_Scall       As Double
Public G_P5_Sput        As Double
Public G_P5_Size        As Double
Public G_P5_Lret        As Double
Public G_P5_Volstock    As Double
Public G_P5_Pvol        As Double

'VARIABLES USED FOR FILTER MACRO
Public G_Range          As String
Public G_Crit           As String
Public G_WS             As Integer
```

Module: A_Constants

```
Const WEEKS = 260
Const FIRMS = 46
```

Module: A_RunModel

```
Sub RunModel1()  
    Call ReadSource1  
    'Call PortfolioSelect1  
    'Call PortfolioSelect_P1C  
    'Call PortfolioSelect_P2C  
    'Call PortfolioSelect_P3C  
    'Call PortfolioSelect_P4C  
    'Call PortfolioSelect_P5C  
    Call PortfolioSelect2  
    Call PortfolioSelect_P1P  
    Call PortfolioSelect_P2P  
    Call PortfolioSelect_P3P  
    Call PortfolioSelect_P4P  
    Call PortfolioSelect_P5P  
End Sub
```

Module: ReadSource

```
Sub ReadSource1()  
    ReDim G_ID(1 To 260, 1 To 46) As Integer  
    ReDim G_Stockret(1 To 260, 1 To 46) As Double  
    ReDim G_Scall(1 To 260, 1 To 46) As Double  
    ReDim G_Sput(1 To 260, 1 To 46) As Double  
    ReDim G_Size(1 To 260, 1 To 46) As Double  
    ReDim G_Lret(1 To 260, 1 To 46) As Double  
    ReDim G_Volstock(1 To 260, 1 To 46) As Double  
    ReDim G_Pvol(1 To 260, 1 To 46) As Double  
    Worksheets(1).Select  
    For c = 1 To 46  
        For i = 1 To 260  
            x = (c - 1) * 260 + (i + 1)  
            G_ID(i, c) = Cells(x, 2).Value  
            G_Stockret(i, c) = Cells(x, 3).Value  
            G_Scall(i, c) = Cells(x, 4).Value  
            G_Sput(i, c) = Cells(x, 5).Value  
            G_Size(i, c) = Cells(x, 6).Value  
            G_Lret(i, c) = Cells(x, 7).Value  
            G_Volstock(i, c) = Cells(x, 8).Value  
            G_Pvol(i, c) = Cells(x, 9).Value  
        Next i  
    Next c  
End Sub
```

Module: CallPortfolio

```
Sub PortfolioSelect1()  
    G_WS = 3
```

```
Worksheets(G_WS).Select
```

```
For week = 1 To 260
```

```
    y = 1 + (week - 1) * 50
```

```
    Cells(y, 1) = "ID"  
    Cells(y, 2).Value = "STOCKET"  
    Cells(y, 3).Value = "SCALL"  
    Cells(y, 4).Value = "SPUT"  
    Cells(y, 5).Value = "Size"  
    Cells(y, 6).Value = "LRET"  
    Cells(y, 7).Value = "VOLSTOCK"  
    Cells(y, 8).Value = "PVOL"
```

```
For Row = 1 To 46
```

```
    x = Row + 1 + (week - 1) * 50
```

```
    Cells(x, 1) = G_ID(week, Row)  
    Cells(x, 2).Value = G_Stockret(week, Row)  
    Cells(x, 3).Value = G_Scall(week, Row)  
    Cells(x, 4).Value = G_Sput(week, Row)  
    Cells(x, 5).Value = G_Size(week, Row)  
    Cells(x, 6).Value = G_Lret(week, Row)  
    Cells(x, 7).Value = G_Volstock(week, Row)  
    Cells(x, 8).Value = G_Pvol(week, Row)
```

```
Next Row
```

```
Next week
```

```
For k = 1 To 260
```

```
    p = 1 + (k - 1) * 50  
    q = 47 + (k - 1) * 50
```

```
    G_Range = "A" & p & ":" & "H" & q  
    G_Crit = "C" & p & ":" & "C" & q
```

```
    If k = 1 Then Call Filter1  
    If k > 1 Then Call Filter2
```

```
Next k
```

```
End Sub
```

```
Sub PortfolioSelect_P1C()
```

```
    G_P1_Stockret = 0  
    G_P1_Scall = 0  
    G_P1_Sput = 0  
    G_P1_Size = 0  
    G_P1_Lret = 0  
    G_P1_Volstock = 0  
    G_P1_Pvol = 0
```

```
Dim r As Range, v As Variant
```

```
For k = 1 To 260
```

```
    p = 38 + (k - 1) * 50
```

```

q = 47 + (k - 1) * 50

P1_Range_1 = "B" & p & ":" & "B" & q

Set r = Sheets(3).Range(P1_Range_1)
v = Application.WorksheetFunction.Sum(r)

G_P1_Stockret = G_P1_Stockret + v

If k = 260 Then Cells(1, 10).Value = G_P1_Stockret / 2600

Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50

    P1_Range_2 = "C" & p & ":" & "C" & q

    Set r = Sheets(3).Range(P1_Range_2)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Scall = G_P1_Scall + v

    If k = 260 Then Cells(1, 11).Value = G_P1_Scall / 2600

Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_3 = "D" & p & ":" & "D" & q

    Set r = Sheets(3).Range(P1_Range_3)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Sput = G_P1_Sput + v

    If k = 260 Then Cells(1, 12).Value = G_P1_Sput / 2600

Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_4 = "E" & p & ":" & "E" & q

    Set r = Sheets(3).Range(P1_Range_4)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Size = G_P1_Size + v

    If k = 260 Then Cells(1, 13).Value = G_P1_Size / 2600

Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

```



```

P1_Range_5 = "F" & p & ":" & "F" & q

Set r = Sheets(3).Range(P1_Range_5)
v = Application.WorksheetFunction.Sum(r)

G_P1_Lret = G_P1_Lret + v

If k = 260 Then Cells(1, 14).Value = G_P1_Lret / 2600
Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_6 = "G" & p & ":" & "G" & q

    Set r = Sheets(3).Range(P1_Range_6)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Volstock = G_P1_Volstock + v

    If k = 260 Then Cells(1, 15).Value = G_P1_Volstock / 2600
Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(3).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Pvol = G_P1_Pvol + v

    If k = 260 Then Cells(1, 16).Value = G_P1_Pvol / 2600
Next k

End Sub

Sub PortfolioSelect_P2C()

    G_P2_Stockret = 0
    G_P2_Scall = 0
    G_P2_Sput = 0
    G_P2_Size = 0
    G_P2_Lret = 0
    G_P2_Volstock = 0
    G_P2_Pvol = 0

    Dim r As Range, v As Variant

    For k = 1 To 260

        p = 28 + (k - 1) * 50
        q = 37 + (k - 1) * 50

```

```

P1_Range_1 = "B" & p & ":" & "B" & q

Set r = Sheets(3).Range(P1_Range_1)
v = Application.WorksheetFunction.Sum(r)

G_P2_Stockret = G_P2_Stockret + v

If k = 260 Then Cells(2, 10).Value = G_P2_Stockret / 2600

Next k

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50

    P1_Range_2 = "C" & p & ":" & "C" & q

    Set r = Sheets(3).Range(P1_Range_2)
    v = Application.WorksheetFunction.Sum(r)

    G_P2_Scall = G_P2_Scall + v

    If k = 260 Then Cells(2, 11).Value = G_P2_Scall / 2600

Next k

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50

    P1_Range_3 = "D" & p & ":" & "D" & q

    Set r = Sheets(3).Range(P1_Range_3)
    v = Application.WorksheetFunction.Sum(r)

    G_P2_Sput = G_P2_Sput + v

    If k = 260 Then Cells(2, 12).Value = G_P2_Sput / 2600

Next k

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50

    P1_Range_4 = "E" & p & ":" & "E" & q

    Set r = Sheets(3).Range(P1_Range_4)
    v = Application.WorksheetFunction.Sum(r)

    G_P2_Size = G_P2_Size + v

    If k = 260 Then Cells(2, 13).Value = G_P2_Size / 2600

Next k

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50

    P1_Range_5 = "F" & p & ":" & "F" & q

    Set r = Sheets(3).Range(P1_Range_5)
    v = Application.WorksheetFunction.Sum(r)

```

```

    G_P2_Lret = G_P2_Lret + v

    If k = 260 Then Cells(2, 14).Value = G_P2_Lret / 2600
Next k

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_6 = "G" & p & ":" & "G" & q

    Set r = Sheets(3).Range(P1_Range_6)
    v = Application.WorksheetFunction.Sum(r)

    G_P2_Volstock = G_P2_Volstock + v

    If k = 260 Then Cells(2, 15).Value = G_P2_Volstock / 2600
Next k

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(3).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

    G_P2_Pvol = G_P2_Pvol + v

    If k = 260 Then Cells(2, 16).Value = G_P2_Pvol / 2600
Next k

End Sub

Sub PortfolioSelect_P5C()

    G_P5_Stockret = 0
    G_P5_Scall = 0
    G_P5_Sput = 0
    G_P5_Size = 0
    G_P5_Lret = 0
    G_P5_Volstock = 0
    G_P5_Pvol = 0

    Dim r As Range, v As Variant

    For k = 1 To 260

        p = 2 + (k - 1) * 50
        q = 11 + (k - 1) * 50

        P1_Range_1 = "B" & p & ":" & "B" & q

        Set r = Sheets(3).Range(P1_Range_1)
        v = Application.WorksheetFunction.Sum(r)

```

```

G_P5_Stockret = G_P5_Stockret + v

If k = 260 Then Cells(5, 10).Value = G_P5_Stockret / 2600

Next k

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50

    P1_Range_2 = "C" & p & ":" & "C" & q

    Set r = Sheets(3).Range(P1_Range_2)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Scall = G_P5_Scall + v

    If k = 260 Then Cells(5, 11).Value = G_P5_Scall / 2600

Next k

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50

    P1_Range_3 = "D" & p & ":" & "D" & q

    Set r = Sheets(3).Range(P1_Range_3)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Sput = G_P5_Sput + v

    If k = 260 Then Cells(5, 12).Value = G_P5_Sput / 2600

Next k

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50

    P1_Range_4 = "E" & p & ":" & "E" & q

    Set r = Sheets(3).Range(P1_Range_4)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Size = G_P5_Size + v

    If k = 260 Then Cells(5, 13).Value = G_P5_Size / 2600

Next k

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50

    P1_Range_5 = "F" & p & ":" & "F" & q

    Set r = Sheets(3).Range(P1_Range_5)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Lret = G_P5_Lret + v

    If k = 260 Then Cells(5, 14).Value = G_P5_Lret / 2600

Next k

```

```

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50

    P1_Range_6 = "G" & p & ":" & "G" & q

    Set r = Sheets(3).Range(P1_Range_6)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Volstock = G_P5_Volstock + v

    If k = 260 Then Cells(5, 15).Value = G_P5_Volstock / 2600

```

```

Next k

```

```

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(3).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Pvol = G_P5_Pvol + v

    If k = 260 Then Cells(5, 16).Value = G_P5_Pvol / 2600

```

```

Next k

```

```

End Sub

```

```

Sub PortfolioSelect_P4C()

```

```

    G_P4_Stockret = 0
    G_P4_Scall = 0
    G_P4_Sput = 0
    G_P4_Size = 0
    G_P4_Lret = 0
    G_P4_Volstock = 0
    G_P4_Pvol = 0

```

```

    Dim r As Range, v As Variant

```

```

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_1 = "B" & p & ":" & "B" & q

    Set r = Sheets(3).Range(P1_Range_1)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Stockret = G_P4_Stockret + v

    If k = 260 Then Cells(4, 10).Value = G_P4_Stockret / 2600

```

```

Next k

```

```

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_2 = "C" & p & ":" & "C" & q

    Set r = Sheets(3).Range(P1_Range_2)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Scall = G_P4_Scall + v

    If k = 260 Then Cells(4, 11).Value = G_P4_Scall / 2600

Next k

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_3 = "D" & p & ":" & "D" & q

    Set r = Sheets(3).Range(P1_Range_3)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Sput = G_P4_Sput + v

    If k = 260 Then Cells(4, 12).Value = G_P4_Sput / 2600

Next k

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_4 = "E" & p & ":" & "E" & q

    Set r = Sheets(3).Range(P1_Range_4)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Size = G_P4_Size + v

    If k = 260 Then Cells(4, 13).Value = G_P4_Size / 2600

Next k

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_5 = "F" & p & ":" & "F" & q

    Set r = Sheets(3).Range(P1_Range_5)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Lret = G_P4_Lret + v

    If k = 260 Then Cells(4, 14).Value = G_P4_Lret / 2600

Next k

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

```

```

P1_Range_6 = "G" & p & ":" & "G" & q

Set r = Sheets(3).Range(P1_Range_6)
v = Application.WorksheetFunction.Sum(r)

G_P4_Volstock = G_P4_Volstock + v

If k = 260 Then Cells(4, 15).Value = G_P4_Volstock / 2600

Next k

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(3).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Pvol = G_P4_Pvol + v

    If k = 260 Then Cells(4, 16).Value = G_P4_Pvol / 2600

Next k

End Sub

```

```

Sub PortfolioSelect_P3C()

G_P3_Stockret = 0
G_P3_Scall = 0
G_P3_Sput = 0
G_P3_Size = 0
G_P3_Lret = 0
G_P3_Volstock = 0
G_P3_Pvol = 0

Dim r As Range, v As Variant

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_1 = "B" & p & ":" & "B" & q

    Set r = Sheets(3).Range(P1_Range_1)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Stockret = G_P3_Stockret + v

    If k = 260 Then Cells(3, 10).Value = G_P3_Stockret / 2600

Next k

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_2 = "C" & p & ":" & "C" & q

```

```

Set r = Sheets(3).Range(P1_Range_2)
v = Application.WorksheetFunction.Sum(r)

G_P3_Scall = G_P3_Scall + v

If k = 260 Then Cells(3, 11).Value = G_P3_Scall / 2600
Next k

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_3 = "D" & p & ":" & "D" & q

    Set r = Sheets(3).Range(P1_Range_3)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Sput = G_P3_Sput + v

    If k = 260 Then Cells(3, 12).Value = G_P3_Sput / 2600
Next k

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_4 = "E" & p & ":" & "E" & q

    Set r = Sheets(3).Range(P1_Range_4)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Size = G_P3_Size + v

    If k = 260 Then Cells(3, 13).Value = G_P3_Size / 2600
Next k

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_5 = "F" & p & ":" & "F" & q

    Set r = Sheets(3).Range(P1_Range_5)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Lret = G_P3_Lret + v

    If k = 260 Then Cells(3, 14).Value = G_P3_Lret / 2600
Next k

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_6 = "G" & p & ":" & "G" & q

    Set r = Sheets(3).Range(P1_Range_6)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Volstock = G_P3_Volstock + v

```



```

    If k = 260 Then Cells(3, 15).Value = G_P3_Volstock / 2600
Next k

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(3).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Pvol = G_P3_Pvol + v

    If k = 260 Then Cells(3, 16).Value = G_P3_Pvol / 2600
Next k

End Sub

```

Module: PutPortfolio

```

Sub PortfolioSelect2()

    G_WS = 4

    Worksheets(G_WS).Select

    For week = 1 To 260

        y = 1 + (week - 1) * 50

        Cells(y, 1) = "ID"
        Cells(y, 2).Value = "STOCKET"
        Cells(y, 3).Value = "SCALL"
        Cells(y, 4).Value = "SPUT"
        Cells(y, 5).Value = "Size"
        Cells(y, 6).Value = "LRET"
        Cells(y, 7).Value = "VOLSTOCK"
        Cells(y, 8).Value = "PVOL"

        For Row = 1 To 46

            x = Row + 1 + (week - 1) * 50

            Cells(x, 1) = G_ID(week, Row)
            Cells(x, 2).Value = G_Stockret(week, Row)
            Cells(x, 3).Value = G_Scall(week, Row)
            Cells(x, 4).Value = G_Sput(week, Row)
            Cells(x, 5).Value = G_Size(week, Row)
            Cells(x, 6).Value = G_Lret(week, Row)
            Cells(x, 7).Value = G_Volstock(week, Row)
            Cells(x, 8).Value = G_Pvol(week, Row)

            Next Row

        Next week

    For k = 1 To 260

        p = 1 + (k - 1) * 50
        q = 47 + (k - 1) * 50
    
```

```

G_Range = "A" & p & ":" & "H" & q
G_Crit = "D" & p & ":" & "D" & q

If k = 1 Then Call Filter1
If k > 1 Then Call Filter2

Next k

End Sub

Sub PortfolioSelect_P1P()

G_P1_Stockret = 0
G_P1_Scall = 0
G_P1_Sput = 0
G_P1_Size = 0
G_P1_Lret = 0
G_P1_Volstock = 0
G_P1_Pvol = 0

Dim r As Range, v As Variant

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50

    P1_Range_1 = "B" & p & ":" & "B" & q

    Set r = Sheets(4).Range(P1_Range_1)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Stockret = G_P1_Stockret + v

    If k = 260 Then Cells(1, 10).Value = G_P1_Stockret / 2600

Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50

    P1_Range_2 = "C" & p & ":" & "C" & q

    Set r = Sheets(4).Range(P1_Range_2)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Scall = G_P1_Scall + v

    If k = 260 Then Cells(1, 11).Value = G_P1_Scall / 2600

Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_3 = "D" & p & ":" & "D" & q

    Set r = Sheets(4).Range(P1_Range_3)
    v = Application.WorksheetFunction.Sum(r)

```

```

G_P1_Sput = G_P1_Sput + v

If k = 260 Then Cells(1, 12).Value = G_P1_Sput / 2600
Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_4 = "E" & p & ":" & "E" & q

    Set r = Sheets(4).Range(P1_Range_4)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Size = G_P1_Size + v

    If k = 260 Then Cells(1, 13).Value = G_P1_Size / 2600
Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_5 = "F" & p & ":" & "F" & q

    Set r = Sheets(4).Range(P1_Range_5)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Lret = G_P1_Lret + v

    If k = 260 Then Cells(1, 14).Value = G_P1_Lret / 2600
Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_6 = "G" & p & ":" & "G" & q

    Set r = Sheets(4).Range(P1_Range_6)
    v = Application.WorksheetFunction.Sum(r)

    G_P1_Volstock = G_P1_Volstock + v

    If k = 260 Then Cells(1, 15).Value = G_P1_Volstock / 2600
Next k

For k = 1 To 260

    p = 38 + (k - 1) * 50
    q = 47 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(4).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

```

```

    G_P1_Pvol = G_P1_Pvol + v

    If k = 260 Then Cells(1, 16).Value = G_P1_Pvol / 2600
Next k

End Sub


Sub PortfolioSelect_P2P()

G_P2_Stockret = 0
G_P2_Scall = 0
G_P2_Sput = 0
G_P2_Size = 0
G_P2_Lret = 0
G_P2_Volstock = 0
G_P2_Pvol = 0


Dim r As Range, v As Variant

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50

    P1_Range_1 = "B" & p & ":" & "B" & q

    Set r = Sheets(4).Range(P1_Range_1)
    v = Application.WorksheetFunction.Sum(r)

    G_P2_Stockret = G_P2_Stockret + v

    If k = 260 Then Cells(2, 10).Value = G_P2_Stockret / 2600

Next k

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50

    P1_Range_2 = "C" & p & ":" & "C" & q

    Set r = Sheets(4).Range(P1_Range_2)
    v = Application.WorksheetFunction.Sum(r)

    G_P2_Scall = G_P2_Scall + v

    If k = 260 Then Cells(2, 11).Value = G_P2_Scall / 2600

Next k

For k = 1 To 260

    p = 28 + (k - 1) * 50
    q = 37 + (k - 1) * 50

    P1_Range_3 = "D" & p & ":" & "D" & q

    Set r = Sheets(4).Range(P1_Range_3)
    v = Application.WorksheetFunction.Sum(r)

    G_P2_Sput = G_P2_Sput + v

    If k = 260 Then Cells(2, 12).Value = G_P2_Sput / 2600

```

Next k

For k = 1 To 260

p = 28 + (k - 1) * 50

q = 37 + (k - 1) * 50

P1_Range_4 = "E" & p & ":" & "E" & q

Set r = Sheets(4).Range(P1_Range_4)

v = Application.WorksheetFunction.Sum(r)

G_P2_Size = G_P2_Size + v

If k = 260 Then Cells(2, 13).Value = G_P2_Size / 2600

Next k

For k = 1 To 260

p = 28 + (k - 1) * 50

q = 37 + (k - 1) * 50

P1_Range_5 = "F" & p & ":" & "F" & q

Set r = Sheets(4).Range(P1_Range_5)

v = Application.WorksheetFunction.Sum(r)

G_P2_Lret = G_P2_Lret + v

If k = 260 Then Cells(2, 14).Value = G_P2_Lret / 2600

Next k

For k = 1 To 260

p = 28 + (k - 1) * 50

q = 37 + (k - 1) * 50

d = 1 + (k - 1) * 50

P1_Range_6 = "G" & p & ":" & "G" & q

Set r = Sheets(4).Range(P1_Range_6)

v = Application.WorksheetFunction.Sum(r)

G_P2_Volstock = G_P2_Volstock + v

If k = 260 Then Cells(2, 15).Value = G_P2_Volstock / 2600

Next k

For k = 1 To 260

p = 28 + (k - 1) * 50

q = 37 + (k - 1) * 50

d = 1 + (k - 1) * 50

P1_Range_7 = "H" & p & ":" & "H" & q

Set r = Sheets(4).Range(P1_Range_7)

v = Application.WorksheetFunction.Sum(r)

G_P2_Pvol = G_P2_Pvol + v

If k = 260 Then Cells(2, 16).Value = G_P2_Pvol / 2600

Next k

End Sub

Sub PortfolioSelect_P5P()

```
G_P5_Stockret = 0
G_P5_Scall = 0
G_P5_Sput = 0
G_P5_Size = 0
G_P5_Lret = 0
G_P5_Volstock = 0
G_P5_Pvol = 0
```

Dim r As Range, v As Variant

For k = 1 To 260

```
p = 2 + (k - 1) * 50
q = 11 + (k - 1) * 50
```

P1_Range_1 = "B" & p & ":" & "B" & q

```
Set r = Sheets(4).Range(P1_Range_1)
v = Application.WorksheetFunction.Sum(r)
```

G_P5_Stockret = G_P5_Stockret + v

If k = 260 Then Cells(5, 10).Value = G_P5_Stockret / 2600

Next k

For k = 1 To 260

```
p = 2 + (k - 1) * 50
q = 11 + (k - 1) * 50
```

P1_Range_2 = "C" & p & ":" & "C" & q

```
Set r = Sheets(4).Range(P1_Range_2)
v = Application.WorksheetFunction.Sum(r)
```

G_P5_Scall = G_P5_Scall + v

If k = 260 Then Cells(5, 11).Value = G_P5_Scall / 2600

Next k

For k = 1 To 260

```
p = 2 + (k - 1) * 50
q = 11 + (k - 1) * 50
```

P1_Range_3 = "D" & p & ":" & "D" & q

```
Set r = Sheets(4).Range(P1_Range_3)
v = Application.WorksheetFunction.Sum(r)
```

G_P5_Sput = G_P5_Sput + v

If k = 260 Then Cells(5, 12).Value = G_P5_Sput / 2600

Next k

For k = 1 To 260

```
p = 2 + (k - 1) * 50
```

```

q = 11 + (k - 1) * 50

P1_Range_4 = "E" & p & ":" & "E" & q

Set r = Sheets(4).Range(P1_Range_4)
v = Application.WorksheetFunction.Sum(r)

G_P5_Size = G_P5_Size + v

If k = 260 Then Cells(5, 13).Value = G_P5_Size / 2600
Next k

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50

    P1_Range_5 = "F" & p & ":" & "F" & q

    Set r = Sheets(4).Range(P1_Range_5)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Lret = G_P5_Lret + v

    If k = 260 Then Cells(5, 14).Value = G_P5_Lret / 2600
Next k

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50

    P1_Range_6 = "G" & p & ":" & "G" & q

    Set r = Sheets(4).Range(P1_Range_6)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Volstock = G_P5_Volstock + v

    If k = 260 Then Cells(5, 15).Value = G_P5_Volstock / 2600
Next k

For k = 1 To 260

    p = 2 + (k - 1) * 50
    q = 11 + (k - 1) * 50
    d = 1 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(4).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

    G_P5_Pvol = G_P5_Pvol + v

    If k = 260 Then Cells(5, 16).Value = G_P5_Pvol / 2600
Next k

End Sub

Sub PortfolioSelect_P4P()

```

```
G_P4_Stockret = 0
G_P4_Scall = 0
G_P4_Sput = 0
G_P4_Size = 0
G_P4_Lret = 0
G_P4_Volstock = 0
G_P4_Pvol = 0
```

```
Dim r As Range, v As Variant
```

```
For k = 1 To 260
```

```
    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50
```

```
    P1_Range_1 = "B" & p & ":" & "B" & q
```

```
    Set r = Sheets(4).Range(P1_Range_1)
    v = Application.WorksheetFunction.Sum(r)
```

```
    G_P4_Stockret = G_P4_Stockret + v
```

```
    If k = 260 Then Cells(4, 10).Value = G_P4_Stockret / 2600
```

```
Next k
```

```
For k = 1 To 260
```

```
    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50
```

```
    P1_Range_2 = "C" & p & ":" & "C" & q
```

```
    Set r = Sheets(4).Range(P1_Range_2)
    v = Application.WorksheetFunction.Sum(r)
```

```
    G_P4_Scall = G_P4_Scall + v
```

```
    If k = 260 Then Cells(4, 11).Value = G_P4_Scall / 2600
```

```
Next k
```

```
For k = 1 To 260
```

```
    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50
```

```
    P1_Range_3 = "D" & p & ":" & "D" & q
```

```
    Set r = Sheets(4).Range(P1_Range_3)
    v = Application.WorksheetFunction.Sum(r)
```

```
    G_P4_Sput = G_P4_Sput + v
```

```
    If k = 260 Then Cells(4, 12).Value = G_P4_Sput / 2600
```

```
Next k
```

```
For k = 1 To 260
```

```
    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50
```

```
    P1_Range_4 = "E" & p & ":" & "E" & q
```

```
    Set r = Sheets(4).Range(P1_Range_4)
    v = Application.WorksheetFunction.Sum(r)
```



```

    G_P4_Size = G_P4_Size + v

    If k = 260 Then Cells(4, 13).Value = G_P4_Size / 2600
Next k

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_5 = "F" & p & ":" & "F" & q

    Set r = Sheets(4).Range(P1_Range_5)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Lret = G_P4_Lret + v

    If k = 260 Then Cells(4, 14).Value = G_P4_Lret / 2600
Next k

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_6 = "G" & p & ":" & "G" & q

    Set r = Sheets(4).Range(P1_Range_6)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Volstock = G_P4_Volstock + v

    If k = 260 Then Cells(4, 15).Value = G_P4_Volstock / 2600
Next k

For k = 1 To 260

    p = 12 + (k - 1) * 50
    q = 21 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(4).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

    G_P4_Pvol = G_P4_Pvol + v

    If k = 260 Then Cells(4, 16).Value = G_P4_Pvol / 2600
Next k

End Sub

Sub PortfolioSelect_P3P()

    G_P3_Stockret = 0
    G_P3_Scall = 0
    G_P3_Sput = 0
    G_P3_Size = 0
    G_P3_Lret = 0
    G_P3_Volstock = 0

```

```
G_P3_Pvol = 0
```

```
Dim r As Range, v As Variant
```

```
For k = 1 To 260
```

```
    p = 22 + (k - 1) * 50
```

```
    q = 27 + (k - 1) * 50
```

```
    P1_Range_1 = "B" & p & ":" & "B" & q
```

```
    Set r = Sheets(4).Range(P1_Range_1)
```

```
    v = Application.WorksheetFunction.Sum(r)
```

```
    G_P3_Stockret = G_P3_Stockret + v
```

```
    If k = 260 Then Cells(3, 10).Value = G_P3_Stockret / 2600
```

```
Next k
```

```
For k = 1 To 260
```

```
    p = 22 + (k - 1) * 50
```

```
    q = 27 + (k - 1) * 50
```

```
    P1_Range_2 = "C" & p & ":" & "C" & q
```

```
    Set r = Sheets(4).Range(P1_Range_2)
```

```
    v = Application.WorksheetFunction.Sum(r)
```

```
    G_P3_Scall = G_P3_Scall + v
```

```
    If k = 260 Then Cells(3, 11).Value = G_P3_Scall / 2600
```

```
Next k
```

```
For k = 1 To 260
```

```
    p = 22 + (k - 1) * 50
```

```
    q = 27 + (k - 1) * 50
```

```
    P1_Range_3 = "D" & p & ":" & "D" & q
```

```
    Set r = Sheets(4).Range(P1_Range_3)
```

```
    v = Application.WorksheetFunction.Sum(r)
```

```
    G_P3_Sput = G_P3_Sput + v
```

```
    If k = 260 Then Cells(3, 12).Value = G_P3_Sput / 2600
```

```
Next k
```

```
For k = 1 To 260
```

```
    p = 22 + (k - 1) * 50
```

```
    q = 27 + (k - 1) * 50
```

```
    P1_Range_4 = "E" & p & ":" & "E" & q
```

```
    Set r = Sheets(4).Range(P1_Range_4)
```

```
    v = Application.WorksheetFunction.Sum(r)
```

```
    G_P3_Size = G_P3_Size + v
```

```
    If k = 260 Then Cells(3, 13).Value = G_P3_Size / 2600
```

```
Next k
```

```

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_5 = "F" & p & ":" & "F" & q

    Set r = Sheets(4).Range(P1_Range_5)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Lret = G_P3_Lret + v

    If k = 260 Then Cells(3, 14).Value = G_P3_Lret / 2600

Next k

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_6 = "G" & p & ":" & "G" & q

    Set r = Sheets(4).Range(P1_Range_6)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Volstock = G_P3_Volstock + v

    If k = 260 Then Cells(3, 15).Value = G_P3_Volstock / 2600

Next k

For k = 1 To 260

    p = 22 + (k - 1) * 50
    q = 27 + (k - 1) * 50

    P1_Range_7 = "H" & p & ":" & "H" & q

    Set r = Sheets(4).Range(P1_Range_7)
    v = Application.WorksheetFunction.Sum(r)

    G_P3_Pvol = G_P3_Pvol + v

    If k = 260 Then Cells(3, 16).Value = G_P3_Pvol / 2600

Next k

End Sub

```

Module: Filters

```

Sub Filter1()

    Range(G_Range).Select
    Selection.AutoFilter
    ActiveWorkbook.Worksheets(G_WS).AutoFilter.Sort.SortFields.Clear
    ActiveWorkbook.Worksheets(G_WS).AutoFilter.Sort.SortFields.Add Key:=Range( _
        G_Crit), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
        xlSortNormal
    With ActiveWorkbook.Worksheets(G_WS).AutoFilter.Sort
        .Header = xlYes
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With

```

End Sub

Sub Filter2()

```
    Range(G_Range).Select
    Selection.AutoFilter
    Selection.AutoFilter
    ActiveWorkbook.Worksheets(G_WS).AutoFilter.Sort.SortFields.Clear
    ActiveWorkbook.Worksheets(G_WS).AutoFilter.Sort.SortFields.Add Key:=Range( _
        G_Crit), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
        xlSortNormal
    With ActiveWorkbook.Worksheets(G_WS).AutoFilter.Sort
        .Header = xlYes
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
```

End Sub

