Calendar Seasonals in Equity Option Markets

October 2011

Master Thesis Financial Economics
Erasmus University Rotterdam
Erasmus School of Economics
Author: A.C. de Jong
Supervisor: Dr. G. Baltussen
Co-Reader: Dr. S. van Bekkum

Acknowledgment

I would like to thank my supervisor Guido Baltussen for his support and feedback during the thesis trajectory process.

Furthermore, I want to use this opportunity to thank family for their moral support during the completion of my study. Last but not least, I want to thank Janneke Loof for spell checking my thesis.

Arjen de Jong

Abstract

Calendar anomalies are recurring patterns in returns on assets which cannot be explained by underlying fundamental factors. A lot of research has been done on these seasonals in the stock markets, but there is hardly any research on seasonal effects in the equity option markets. Therefore, this research focuses on six well-known calendar anomalies, namely the Halloween, Holiday, Monday, January, Turn-of-the-Month and Triple Witch effect, and whether these anomalies manifest themselves in the equity option market. This is tested by using excess returns and delta hedge excess returns. A negative Monday and a positive Turn-of-the-Month effect is found in both excess returns and delta hedge returns, indicating that the seasonal effect found is not generated by the underlying asset. Some evidence is found that the January effect is an anomaly as well, but it is less convincing. When it is tested whether the calendar seasonals are not a proxy of each other, the results hold. A number of robustness checks are performed and no evidence is found that the results are due to illiquidity, moneyness, or maturity of the option. Finally, it is tested whether the excess returns can be explained by a change in implied volatility or risk neutral distribution by creating a beta neutral straddle and skewness asset. When it comes to the Monday and Turn-of-the-Month effect, both portfolios explain a part of the excess returns. For the January effect, the implied volatility explains a part of the excess return.

Table of Contend

1. Introduction

Calendar anomalies in the stock market have been documented for almost over a century (see for example Fields, 1937) in the financial literature. However, there is hardly any research done on seasonal effects in the equity option markets. Maloney and Rogalski (1989) look at the January effect in equity options and more recently, Shemesh and Jones (2011) look at the Monday effect. There is no academic literature that looks into the six major calendar seasonals that exist in the stock market, the Halloween, Holiday, Monday, Turn-of-the-Month and Triple Witch effect, and whether they manifest themselves in the option market. This research starts to fill this gap in the financial literature.

The central research question in this thesis is:

Do the Halloween, Holiday, Monday, January, Turn-of-the-Month and Triple Witch effects manifest themselves in the equity option market?

To test this hypothesis a dataset is constructed, which follows the S&P 500 and contains over 26 million options with a maturity up to 52 days over a period of 15 years. By constructing two portfolios, one of excess returns and one with delta excess returns, the influence of seasonality in the underlying stock is removed. Using a dummy regression, it is tested whether there are any calendar anomalies in the option markets and whether they are robust calendar seasonals in their own right or a proxy of another seasonal effect. A test is also provided to see whether other properties of the option may explain the seasonal effect found in the options, such as liquidity, moneyness and maturity. Finally, it is investigated whether the implied volatility and risk neutral distribution can explain a part of the excess returns found.

The remainder of this thesis is organized as follows; chapter 2 provides a literature overview of the six calendar anomalies studied, chapter 3 describes the data used and chapter 4 describes the methodology applied. Chapter 5 describes the results of the research and chapter 6 provides the robustness tests for the results found in chapter 5. Chapter 7 explains part of the found excess returns by the change in implied volatility and risk neutral distribution by creating a beta neutral straddle and a skewness asset. The conclusion and suggestions for further research are given in chapter 8. A list of references and the appendix can be found at the end of the paper.

2. Literature Review

In this chapter will provide an overview of the literature on the efficient market hypothesis (EMH) and the six calendar anomalies researched in this study. The calendar seasonals, which are the Monday, January, Holiday, Turn-of-the-Month, Triple Witch and Halloween effect are treated in order of seniority.

2.1. Efficient Market Hypothesis

The efficient market hypothesis (EMH) is a well-known hypothesis about how asset prices reflect all available information and follow a random walk in capital markets. The EMH was found both by Samuelson (1965) and Fama (1965) at about the same time and is the leading theory in asset pricing. The EMH implies that when new information reaches the market, stock prices will change and reflect this new information. The more efficient a market is, the more the changes in prices are random, because information is priced in a couple of seconds and nobody knows if the next piece of information is positive of negative.

In a perfectly efficient market there is no mispricing. Therefore, there is no possibility to generate abnormal returns. Abnormal returns are actual returns minus normal returns. Normal returns are calculated using asset pricing models such as the Capital asset pricing model (Sharpe, 1964), the arbitrage pricing theory (Ross, 1980) and the three factor model (Fama and French, 1993). There are many other asset pricing models taking risk and other fundamental factors into account. The problem with the EMH is that if it is tested empirically a joint testing hypothesis is done. Not only the EMH is tested, but also the asset pricing model used to test the EMH (Fama, 1970) (For more information on the efficient market hypothesis see the overview article of Lo (2007).

2.2 Anomalies

"Anomalies are empirical results that seem to be inconsistent with maintained theories of asset-pricing behavior. They indicate either market inefficiency (profit opportunities) or inadequacies in the underlying asset-pricing model".

- Schwert (2002, P.2)

If the return of an asset has a regular pattern which cannot be explained by fundamental factors, it is called an anomaly. According to the EMH this should not happen, because prices should reflect all available information. If this is not the case, investors would make use of this information by trading on it and therefore the pattern would disappear. Calendar anomalies are cyclical anomalies in returns, where the cycle is based on the calendar year.

2.3 Data Mining

In the financial literature there seems to be a tendency for researchers to focus on anomalies and other unusual findings. This probably is because there is higher chance of getting published by finding an anomaly rather than by re-establishing the obvious. That is why the risk exists that researchers just run huge regression on their data sets (S&P 500, NYSE etc.) until they find some sort of relationship between

variables. Through sheer coincidence they are bound to find a relation between variables. When academics find a pattern, they try to come up with a theory to explain the pattern they found, which can be called data mining. Sullivan, Timmerman and White (1999) argue that a lot of calendar seasonals are due to data mining. Lo and MacKinlay (1988) show as well that the more studies done in a certain field, the higher the chance of data mining is.

A way to minimize the chance that findings are due to data mining is to first construct a theory and then start looking for empirical evidence, instead of the other way around. Other ways to minimize the chance that an anomaly is just an artifact of data mining are for example out-of-sample testing, the use of different data set, the use of a multiple countries and an increase in significances levels (Lakonishok and Smith, 1988). Nevertheless, the most useful control for data mining is the notion is that results with no clear connection should require more evidence.

2.4 The Monday Effect

Fields (1931) was the first to write about one of the most puzzling empirical findings in finance; the Monday effect, which implies that on average, there are negative stock returns on Monday while stock returns are positive for the rest of the week.

In modern academic financial literature French (1980) is the first to stumble across the Monday effect, also known as the weekend effect in the literature. He tries to explain if stock returns are continuously generated or only during trading days. He finds that, instead of a positive return, Monday returns have a significant negative return while the other four trading days have a positive return

After French's studies, a lot of academics started to examine the Monday effect in the US, such as Gibbons and Hess (1981), Keim and Stambaugh (1984), Lakonishok and Smidt (1988) and Bessembinder and Hertzel (1993). Hence for the US market there is a lot of evidence for the Monday effect.

Academics also researched non-US markets and find evidence for the Monday effect as well, such as Jaffe and Westerfield (1985); Hindmarch, Jentsch and Drew (1987); Aggarwal and Rivoili (1989) and Dubois and Loevet (1996).

2.4.1 Decline of the Monday Effect

Connolly (1989) finds that the Monday effect is stronger in an equal-weighted (EW) index than in a value-weighted (VW) index, giving rise to the idea that small firms generate the Monday effect. Kamara (1997), Brusa, Liu and Schulman (2000) and Mehdian and Perry (2001) all find that the Monday effect has disappeared for large caps, while the Monday effect still is significant for small caps. Mehdian and Perry (2001) even find that for large companies the Monday returns are significantly higher than the average returns of the rest of the week.

2.4.2 Settlement

An explanation for the Monday effect is the settlement hypothesis (Gibbons and Hess, 1981; Lakonishok and Levi, 1982). This means that the settlement of a stock in the US takes 5 business days (since 1968). Therefore, in a normal week with no holidays, this means that payment is exactly a week later (5

business days plus two days of weekend). It takes the bank one day to transfer to money from the buyer to seller; therefore a normal transaction in a normal week takes eight days. However, if stocks are purchased on Friday it takes 10 day for the entire process: two weekend days, 5 days of settlement, two weekend days and one day to clear the check. Buyers keep their money for two more days in their account. Consequently, they get two more days interest on their money than if they had bought their stocks on a non-Friday. Sellers on the other hand have to wait two more days for their money and therefore forgo two days of interest. The sellers want to be compensated for the loss of two days of interest and demand a premium on the stock they sell on Friday, equal to two days of interest. On Monday this premium should be deducted from the stock because clearing now takes only eight days once again.

Lakonishok and Levi (1982) test the settlement hypothesis empirically and find that only 17 percent of the Monday effect is explained through delays in trading and settlement of the stock. Dyl and Martin (1985) show that the settlement procedure cannot be the reason for the Monday effect, because before 1968 there was a 7 days clearing and settlement procedure instead of 8. Therefore, if the 8 days settlement procedure is the reason for the Monday effect, no Monday effect should have been found before 1968, but Dyl and Martin (1985) do find a significant Monday effect.

Keim and Stambaugh (1984) compared the difference of size the Monday effect over a one or two day weekend (if there was Saturday trading or not) and find that there is no significant difference between a one or two day weekend. This contradict the settlement hypothesis because when there is Saturday trading, there is one less day of interest to forgo and therefore the Monday effect should be smaller. They also find a stronger Monday effect in the sub period 1928-1952 where the interest was lower in comparison to the period 1953-1982. This should be the other way around if the settlement procedure would be the reason for the Monday effect, because with a high interest premium you forgo more interest if investors sell on Friday, therefore investors should demand a higher premium in periods with higher interest.

2.4.3 News

Penman (1987) studied the effect of news announcements on the Monday effect and finds that bad news is more likely to reach the market on Monday and Friday than on any other day of the week. Fishe, Gosnell and Lasser (1993) find support for Penman (1987) by discovering that bad news during the weekend is highly correlated with the negative Monday effect. Damodaran (1989) finds that earnings and dividends announcements on Friday are more negative than announcements on other days of the week. In addition, most of the Friday news announcements are after the closure of the market, so negative abnormal returns spill over to the next trading day in the case of a Friday to Monday.

Pettengill and Buster (1994) find that the news announcements in the weekend are too small to cause a weekend effect. Another reason that the Monday effect is probably not generated by news arriving on Friday or in the weekend is that Sullivan and Liano (2003) find that the Monday effect is due to a broad spectrum of stock that decline a little and not because of a few stocks who decline a lot. This argues

against the corporate news announcements theory because then there would be a few stocks declining a lot.

2.4.4 Information Processing Hypothesis

Miller (1988) forms the information processing hypothesis; it is costly for individual investors to gather and process information during the weekday when individual investors have a “normal” job. Therefore in the weekend they gather and process the information and make investment decisions and pass these through to their broker on Monday. Additionally, during the week time brokerage firms contact their clients with more buy recommendations than sell recommendations (Groth et al., 1979; Diefenbach, 1972; Dimson and Marsh, 1986). Consequently, buy decisions will be made during the weekdays while sell decisions are made during the weekends. Therefore, more sell transactions than buy transactions are seen on Monday from individual investors.

Lakonishok and Maberly (1990) and later Abraham and Ikenberry (1994) prove this by finding that on Monday, there is a lower trading volume on the New York Stock Exchange (NYSE) due to lower transactions of institutional investors. There are however higher amounts of transactions from individual investors. The transactions are also skewed to more sell transaction than buy transaction. Odd-lot (an amount of shares that is lower than normal for the kind of transaction, indicating an individual investor) sell transaction are 29 percent higher than odd-lot sell transaction on the other trading days (Brockman and Mickayluk, 1998).

2.4.5 Institutional Investors versus Individual Investors

Kamara (1997) find evidence for the declining Monday effect in large caps and continuing Monday effect is small caps. Transaction cost for small caps are the same for institutional and individual investors (Keim and Madhavan, 1995) while for the large caps the transaction costs are a lot lower for institutional investors than individual investors (Kawaller, 1991).Therefore, it is easier for institutions to make use of arbitrage trade on large stocks than small stocks and as a result the Monday effect disappears for large stocks.

Brockman and Mickayluk (1998) find a significant correlation between index returns from Friday to Monday, but no significant correlation between stock returns from Friday to Monday. Finding this correlation they argue that Individual investors sell and buy individual stocks and institutional investors buy and sell portfolios of stock. Therefore the Monday effect is created on index level and not on stock level, which makes the institutional investors the driving force behind the Monday effect. Sias and Stark (1995) find that stocks with large institutional investors seem to have a stronger Monday effect than stocks held by individual investors, while Petengill and Wingender (2003) find the opposite.

Brooks and Kim (1997) look at the intraday trade size and find that on Monday, there are a lot more small size transactions and less large transactions than on other trading days. In addition, the small transactions are for the largest part selling transactions, which is in line with the information processing hypothesis. If the small transactions stand for individual investors and the big transactions for institutional investors then the Monday effect is directly generated by individual investors who are selling and indirectly by institutional investors who take liquidity out of the market by trading less than

on other days. These findings are supported by Wang and Walker (2000) who find the same results in the Asian market.

2.4.6 Blue Monday Hypothesis

Orborne (1962) finds that people are less optimistic on Monday than on other days of the week and that people are the most optimistic on Friday. Therefore, investments made in the optimistic light of Friday may seem not as risky, but may seem very risky after the weekend on a pessimistic Monday. Therefore investors would be more inclined to sell stock on Monday then to buy stock. Pettengill (1993) finds that on Friday the investors chose a significant more risky portfolio than on Monday. Gondhalekar and Mehdian (2003) argue that if the Monday effect is created by pessimistic investors on Monday, it is a source of non-diversifiable risk.

2.4.7 Short Seller Closing Position

The latest explanation for the Monday effect comes from Chen and Singal (2003); short sellers do not want to keep their short position over the weekend because they cannot close their position and therefore have unlimited downside risk. Therefore short sellers buy the stocks they sold short on Friday to close their short position, driving the prices up. On Monday short seller open their short position again by selling stock, driving the stock prices down. After the introduction of put options the Monday effect becomes weaker because short sellers can use put options to take a similar position as a short position, with the advantage that their downside risk is limited to 100 percent of their investment instead a unlimited down side risk. Pettengill, Wingender and Kohli (2003) disagree with Chen and Signal (2003) arguing that put options are a riskier investment than shorting stocks. Therefore, the disappearance of the Monday effect cannot be explained by short sellers who switch from shorting stock moved to put options, because they are less risky according to Pettengill, Wingender and Kohli. (2003).

2.5 The January Effect

Wachtel (1942) was the first to write about the January effect, which implies that stock returns are up until eight times higher in January than during other months of the year. Other academics who find the January effect in US and non US markets are: Officer (1975); Gultekin and Gultekin (1983); Van den Berg and Wessels (1985); The Berges, McConnell and Schlarbaum. (1985); Reinganum and Shapiro (1987); Comolli and Zemba (2000); Ho (1990); Wong, et al. (1990) and Haugen and Jorion (1996). There is a lot empirical evidence that the January effect is a worldwide phenomenon.

2.5.1 Shifting of the January Effect

Mehdian and Perry (2002) and Gu (2003) claim that the January effect has disappeared. Both use monthly data to re-examine the January effect in the US. They find for the month January excessive positive returns after 1987, but these returns are not significant different from zero.

Moller and Zilca (2008) find the research of Mehdian and Perry (2002) and Gu (2003) limited because they use monthly data instead if daily data. In their opinion daily data gives a better picture of the January effect because two of the most supported explanations for the January effect, tax-loss selling

and the window dressing hypothesis, predict that that the majority of the effect should appear early in the month. Moller and Zilca (2008) find that the January effect has not declined. Instead, the January effect has become stronger in the first half of January and less strong in the second half. The higher returns in the beginning of the month offset the lower returns at the end of the month and overall the January effect keeps the same size. This

2.5.2 Small Stock versus Large Stock

Lakonishok and Smidt (1986) do not find the January effect for large caps, they only find it for small caps. Keim (1982) finds that the small firm effect, as introduced by Banz (1981) about how small firms earn higher risk adjusted returns on average than larger firm, is generated for 50 percent in January. From this 50 percent, 26 percent is even generated during the first week of January and 11 percent during the first day. Reinganum (1983) notes that the January returns are only abnormally high for small stock whose prices have declined in the previous period. The high excess return found in the first week by Keim (1983) is not present in small stock that gained in the last period.

2.5.3 Tax Loss Selling

One of the most cited explanations for the January effect is tax loss selling. Wachtel (1942) cited this reason already. Dyl (1977) and Reinganum (1983) suggest that US investors will take their losses in December to create a tax shield for their income taxes at the end of the year and wait until January to take the profits on their investments.

The tax loss selling hypothesis is underwritten by Constantinides and Ingersoll (1984) who look at the timing option investors have with their tax on realized capital gains to influence their cash flows, by selling their losing stocks in December and winning stocks in January. It is optimal for investors to take their losses in December and retrieve their winnings in January.

Roll (1983) argues against tax loss selling; if other investors are aware of tax loss selling they would buy stocks in December and take their excess returns in January. Although Roll argues against tax loss selling, he finds that stocks with a negative return last year have a higher average return in January then stock who did not have a negative return last year.

Schultz (1985) and Jones, Lee (Jones, Lee, & Apenbrink, 1991)et al. (1991) find that there is no January effect before 1917. An income tax bill was passed in 1917 that increased the maximum income tax rate from 15 percent to 67 percent[1]. After 1917, when the income tax was increased, they both find a January effect.

Eakins and Sewell (1993) find that individual investors possess more small stock and institutional investors possess more large stock. They also look at large stock with a high individual ownership and find that these stocks also manifest the January effect. In combination with the finding that the January effect is a small cap effect led Eakins and Sewell (1993) to the conclusion that individual investors create the Monday effect, probably through tax loss selling.

[1] Schultz (1985)

Chen, Noronha and Singal (2004) test the four main reasons given for the January effect; tax loss selling, window dressing, information hypothesis and bid- ask spread. They find the most evidence for the tax loss selling being the driving force behind the January effect. They additionally find that investors postpone selling their winning stocks in January and therefore they defer their capital gains tax for a year. Their findings are in line with Constantinides and Ingersoll (1984).

The tax loss selling hypothesis is hard to uphold when looking at international evidence. In Japan, where there is no capital gains tax or tax shield exist, there is a January effect (Kato and Schallheim, 1985). The same goes for the Netherlands (Van den Berg and Wessels, 1985). The English tax year for individual investors does not end 31st December like the US, but on 5 April. However, there still is a January effect, and also an April effect (Reinganum and Shapiro, 1987). They same goes for Australia (Brown, et al, 1983). It is possible that the tax loss selling from the US affected foreign markets, but that would mean that the international capital markets are entwined. Entwined capital markets would argue against the tax loss selling hypothesis, because investors who live in countries where there is no capital gain tax or have another end of their tax year should arbitrage the January effect away.

2.5.4 Window Dressing

Window dressing is an idea by Haugen and Lakonishok (1987) and Lakonishok, et al. (1991) that fund managers change their holding when they have to report their holdings when their performance is reviewed. Fund managers sell their losing stock and buy winning stock to have a better appearance.

Lakonishok et al. (1991) look at 769 pension funds which stock they sell and buy. They find that, every quarter funds sell disproportional stocks that underperform and buy back well performing stocks. Selling of poor performing stock and replacing them with well performing stock is especially strong in the 4 quarter, at the end of the year. This is exactly what the window dressing hypothesis stands for.

2.5.5 Bid – Ask Spread

Daily prices of stocks in databases are closing prices, the price of the last transaction of that day. The last transaction of the day on many exchanges is filled by a market maker and does not represent the true price, but the bid price (the price the market maker pays), or ask price (the price for which the market maker sells the stock), instead of a transaction between participants not involving a market maker. If there is a pattern where the market closes with a bid or with an ask price, measurement errors may occur. Measurement errors are relatively larger for small illiquid stock, since their bid-ask spread is a higher percentage of the stock price and the bid – ask spread for less liquid stock is higher than for liquid stock. Keim (1989) finds that in December more closing prices were bid prices, while ask prices were dominant at the start of January, resulting in a positive January return even if the stock prices did not change. Bhardwaj and Brook (1992) find the same; more closing bid prices dominate the end of December and more ask closing prices dominate at the beginning of January. Lamoureux and Sanger (1989) and Fortin (1990) find that relative bid-ask spreads are constant during December – January

2.6 The Holiday Effect

The holiday effect, which is also known as the pre-holiday effect, is about how equity returns are on average higher the day before a long weekend or a holiday[2] than on a normal day. Fields (1934) was the first to found that the DJIA had higher return on a day before a holiday than on a normal day.

Other researchers who find the Holiday effect in the US market are: Merill (1966); Fosback (1976); Lakonishok and Smidt (1988); Petengill (1989); Brockman and Michayluk (1998); Wilson and Jones (1993); Fabozzi et al. (1994) and Vergin and McGennis (1999). In international markets, Ziemba (1991) finds the holiday effect in the Japanese stock market. Mills and Coutts (1995) and Arsad and Coutts (1997) find the pre-holiday effect in the UK. Arumugam (1999) finds the holiday effect in India. Coutts et al (2000) find the holiday effect for the Greece market. Kim and Park (1994) find that stock markets in the United Kingdom and Japan show a holiday effect independent of the US. Cadsby and Ratner (1992) find a significant Holiday effect in the United States, Canada, Japan, Hong Kong and Australia. However, they did not find a significant effect in the UK, Italy, Switzerland, West Germany and France, which are all European countries.

All the countries who exhibit a holiday effect show it for their own specific holidays. The only country which exhibits a holiday effect and a US pre-holiday effect is Hong Kong. Nevertheless, in all countries that exhibit a holiday effect, the strongest pre-holiday return is found in a holiday joined by a US holiday. That non-US countries exhibit a holiday effect suggests that the holiday is not an artifact of data mining in the US. However, the non-universality of the effect suggests that the holiday effect is linked to local institutions and not a universal factor.

2.6.1 Explanations and Disappearance Holiday Effect

A lot of academics have tried to explain the Holiday effect, such as Pettengill (1989) who tries to explain the Holiday effect by reasoning that the market is closed on the next day, the same as on a Friday. Keim (1989) looks at bid – ask spreads, Ariel (1987) at market specialist and measurement errors. Liano et al. (1992) look at over-the-counter (OTC) markets and investigate whether other calendar seasonals create the Holiday effect such as the turn-of-the-month (TOM), weekend effect and January effect. Kim and Park (1994) try to explain the holiday effect by controlling for size, institutional factors, trading methods, clearing mechanism, settlement procedure and bid - ask spreads. However, all fail to find empirical evidence for the explanation.

Fabozzi et al (1994) find evidence for the inventory adjustment hypothesis as they find lower volume turn-over pre-holiday and a higher volume turn-over post holiday, as traders re-balance their portfolio. The inventory adjustment implies that traders do not like to hold position over non-trading days, especially short positions which have an infinite loss potential. That is why traders are less inclined to

[2] Most studies use the following holidays for the US: New Year's Day, President Day (formerly Washington's Birthday, Good Friday, Memorial Day, July 4th, Labor Day, Thanksgiving and Christmas.

hold short positions over non-trading days. Therefore, there is less selling pressure on a day before a holiday. Another reason Fabozzi et al (1994) suggest is that people are more positive before the weekend (Deldin et al., 1986) and therefore are also more positive a day before a holiday.

Chong et al. (2005) find a declining holiday effect in the US, UK and the Hong Kong market. Returns become even negative on average pre-holiday, which is the opposite of the holiday effect. McGuinnes (2005) find in the Hang Seng index that the Holiday effect disappears in 1995. This is consequent with a decline US holiday effect.

2.7 Turn-of-the-Month effect

The first to write about the turn-of-the-month (TOM) is Ariel (1987). He finds that positive returns occur on the days just before and during the first half of the month.

Jaffe, Keim and Westerfield (1989) copied the study of Ariel (1987) and find the TOM effect in Australia, but not in the UK, Japan and Canada. Boudreaux (1995) extend their research further to Denmark, France, Germany, Norway, Singapore/Malaysia, Spain and Switzerland. They find the TOM effect for Danish, Norwegian and German markets. Barone (1990) finds the TOM effect in the Italian market, Ziemba (1991) in the Japanese market. Cadsby and Ratner (1992) find the TOM effect for the United States, Canada, the United Kingdom, Australia, Switzerland and West Germany. However, they find no evidence of the TOM effect for Japan, Hong Kong, Italy or France. Van de Sar (2003) finds the TOM effect in the Netherlands. Agrawal and Tandon (1994) find the TOM effect in 14 out of 18 non US markets. Kunkel et al. (2003) look at 19 countries for a TOM effect after the market crash 1987 and find it in 16 countries[3]. Finding of the TOM effect in all of these countries indicates that it is a universal effect and not just a spillover from the US. In addition, most of these authors argue that the TOM effect is not through data mining but has a connection with local institutions and practices.

2.7.1 Liquidity

Ogden (1990) forms the TOM liquidity hypothesis. Implying that in the US the payment of wages, dividends, interest, principals and other liabilities of firms is standardized and paid at the end of each month, especially at the turn of the year. In order to reduce transaction cost investors wait until they have a certain amount of cash/liquid investment before they purchase assets. Since the liquid position of an investor is the best at the TOM, demand for stocks will be greater around that time. When investors receive money, they do not want to use it for consumption but for investment, therefore they invest it right away (Ritter, 1988 and Ziemba, 1989). Following this argument during a month with a lot of liquidity, investors will reinvest a lot and stock prices will rise, while during a month with low liquidity, investors cannot reinvest their money and stock prices do not rise. If liquidity is the reason, then the monetary policy of the FED also has a large influence on the TOM effect. If the monetary policy is expanding more liquid profits will be made and if it is stringent, less liquid profits will be made and

[3] Austria, Belgium, Denmark, France, Germany, the Netherlands, Switzerland, UK, Australia, Japan, New Zealand, Singapore, Canada, United States, Mexico, and South Africa

therefore the TOM effect will be smaller. Ogden (1990) tests his hypothesis and finds empirical evidence for it.

Other academics find more evidence for the liquidity hypothesis of Ogden (1990). Ziemba (1991) find that the TOM effect in Japan is earlier than in the United States and that this is because most salaries are paid in Japan on the 25th of the month. Booth et al. (2001) find that in the Finnish market, the TOM returns are highly positively correlated with volume, share volume and number of trades. These are all indicators of increased trading around the end/beginning of the month.

2.7.2 News Announcements

Penman (1987) tries to explain the TOM effect by the effect of the arrival of corporate news announcements. He finds that positive corporate news announcements are given in the first two weeks of the month, while negative news announcements are mostly given in the last two weeks of the month, reasoning that firms want to give good news as soon as possible and delay bad news. The positive news creates a positive returns at the start of the month and more negative returns at the end of the month.

Nikkinen, et al. (2007) also link macro news announcement to the TOM effect. When Nikkinen et al. (2007) correct the TOM effect for the arrival of macro economical news, the TOM effect disappears.

2.7.3 Other Explanations

McConnell and Xu (2008) combine their study with the one of Lakonishok and Smidt (1988) and create a study that covers over 109 years of data (from 1897 to 2005). On average, all positive returns on the stock occurred in the TOM period. McConnell and Xu (2008) find that the TOM effect is not due to a number of the following control variables: small stock, low price stock, the January effect or high volatility at the end of the month, increased risk rate or risk free rate at the end of the month. It is also not only US based, it is found in 30 of the 34 non-US based countries they consider. McConnell and Xu (2008) find no difference between buying pressure and liquidity during the end of the month. This is the opposite of what Booth et al. (2001) say and against the Ogden's liquidity hypothesis (1990).

2.8 Triple Witch Hour

The Triple Witch hour refers to the last trading hour on the 3rd Friday of March, June, September and December on which stock index futures, index options and individual options (the three witches) all expire. In the last hour of the day, arbitrageurs have to unwind their hedge stock/index position because their futures and options expire. This creates a huge order imbalance and increase volatility and trading volume.

On request of the major option exchanges Stoll and Whaley (1986, 1987, and 1990) examine the influence of expiration days and program trading[4] on the stock market. Stoll and Whaley find that on the last trading hour of the expiration dates, there is a huge increase of volatility and trading volume compared to the normal last trading hour on Friday. This is confirmed by Day and Lewis (1988).

[4] "*The purchase or sale of a portfolio of stocks pursuant to a single order is called program trading*."(Stoll and Whaley 1987, P2)

The Triple Witch effect is found as well in non-US markets such as the Canadian market (Chamberlain, Cheung and Kwan, 1989), the UK (Pope and Yadav, 1992), the German market (Pope and Yadav, 1992), the Spanish market (Illueca and lafuente, 2006) and the Indian market (Bhaumik and Bose, 2007).

Nevertheless, there is also a lot of international evidence against the Triple Witch effect. Bacha and Vila (1994) study the Nikkei 225 and find that stock prices on expiration days are not more volatile than non-expiration days. Stoll and Whaley (1997) find no significant evidence for a price effect in the Australian market. Corredor et al. (2001) study the Spanish market and do not find a significant Triple Witch effect. Felixson (2002) find no significant Triple Witch effect in the Finish market. Alkebäck and Hagelin (2004) look at the Swedish market and find no proof for price distortions. All the markets where no evidence was found for the Triple Witch effect have other settlement systems than the US market.

2.8.1 Moving Expiration Hour from Closing to Opening Hour

The Chicago Mercantile Exchange (CME), NYSE and New York Future Exchange (NYFE) changed the expiration hour of index futures and options from the last trading hour to the opening hour in June 1987. They hereby made an effort to lessen the abnormal returns created on the triple witching hour due to program trading. Stoll and Whaley (1991) find that the expiration effect (increased volatility and volume) has only shifted from the closing of the market to the opening of the market. Chen and Williams (1994) also study the change in expiration time and conclude that mean return and standard deviation of stock returns on Triple Witching Fridays and non-Triple Witching Fridays are not significantly different.

2.8.2. Index Arbitrage

Index futures are priced according to an arbitrage argument. If the future pays out dividend, interest rates are non-stochastic, markets are perfect and there are no taxes, a future is priced according the following formula:

$$F(t, T) = S(t)e^{r(T-t)} - D(t, T) \quad (1)$$

where:

- F(t, T) equals the future's price at time t for a contract that matures at time T
- S(t) equals the spot index value at time t
- D(t, T) equals the time T value of dividends paid on the component stocks between t and T
- r(T - t) equals the risk-free interest rate spanning the period from t to T

There are significant deviations from this arbitrage equation in the future market, which gives arbitrageurs the possibility to profit from. Of all mispriced arbitrage opportunities, 70 percent are gone within a day and are probably created due to noise in the market (Chung, 1991).

Index arbitrage is a strategy used by large, institutional investors who make profit using the spread between the prices in spot and future market. For example, an investor could buy a stock portfolio duplicating the S&P 100 index in the spot market and sell the S&P 100 in the future market and make a profit from the differences in prices between the exchanges.

2.8.3 Cash Settlement

Another reason given for the triple witch hour effect by Stoll and Whaley (1997) is that index futures are cash settled. When the future contract expires and is settled with the counterparty through a cash transaction, the stock portfolio that hedges the future still has to be liquidated in the market. An arbitrageur who is long (short) in the index future is short (long) in the spot market and has to buy (sell) the stocks in the spot market to close out his position. If he can buy or sell his stocks against the same price that is used to calculate the index future, there is no problem. But if this is not the case, he runs a basic risk. If a lot of arbitrageurs unwind at the same moment in the same direction, price effects are possible (Stoll and Whaley, 1997).

2.8.4 Market Procedures

If markets are deep, meaning that there has to be a really large order to change the stock price, and suppliers of liquidity respond quickly to buying or selling pressure, the price effect of large orders will be small. If traders know in advance that non-fundamental price changes were to happen due to arbitrageurs unwinding their position, they would buy (sell) underpriced (overpriced) stock and limit the price effect. In markets where these mechanisms are not very well designed, a sudden shift in order balance can lead to price effects (Stoll and Whaley, 1997).

2.8.5 Settlement Procedure

Stoll and Whaley (1997) discuss the two different ways of settling index future contracts: single price setting and average price setting. The single price setting will center all trading to one moment, which leads to debt of trading and prevents the prices from being pushed out of equilibrium. It also prevents basis risk, because arbitrageurs can unwind their future and stock position at the same moment against the same price. However, a disadvantage is that the market can be steered to a particular level for a short period by manipulators (Kumar and Seppi, 1992).The single price settlement can happen at market opening and closure. An advantage of settlement at opening is that the opening can be postponed until supply and demand are in balance. It also can be a disadvantage, because it creates confusion among investors. Settlement at closure has the advantage of the index settlement value and the regularly disseminated index to be the same. However, research shows that settlement of index futures at opening or closing of the market seems to make no difference for the expiration effect (Herbst and Maberly, 1990 and Stoll and Whaley 1991).

The average price settlement takes the average price of an index over a certain period as settlement price for an index future. A disadvantage of the average price is that it creates basis risk for a trader, because he cannot unwind his future position and stock position for the same price. He has to try to sell or buy stock during the entire day to match the settlement of the future contract. Even if he succeeds in trading the entire day on the right time, he has to make additional transaction costs, which he would not have to make if he unloaded his position at one moment in time. Hillion and Suominen (2004) suggest that a closing auction reduces price manipulation and brings the closing price to his fundamental price and that price manipulation is harder using value-weighted price setting (average price setting) than single price setting.

2.9 Halloween Effect

One of the newer seasonal puzzles is the Halloween effect. Bouman and Jacobsen (2002) test if the old market saying is true: "sell in May and go away, but remember to come back in September"[5]. This means that you sell your stocks in May and move your money to a money account and return to the stock market in September. Bouman and Jacobsen (2002) analyze 37 countries and find in 36 of them the saying is actually true. Stock returns are lower during the period of May until September than during November until April. The effect is the most profound in European countries.

A simple trading strategy, buying the market portfolio in October, sell the portfolio in May and buy short term government bonds from April through October, outperforms the market portfolio as found by Bouman and Jacobsen (2002). This effect is not created by data mining, cross correlation between markets, risk factors, the January effect and it cannot be explained by changing interest rates during the summer and winter. The Halloween effect is not only found in developed economies, but also in emerging countries and does not seem to decline. Bouman and Jacobsen (2002) find a relation between the Halloween effect and vacations, which they argue can be a reason for risk aversion due to the need of liquidity. In countries where the summer vacation is important, the effect is the strongest.

Jacobsen et al. (2005) find that the Halloween effect is unrelated to the size effect and B/M and there is also no connection between E/P and CF/P and the Halloween effect. The only thing they find is that the Halloween effect is stronger in low dividend portfolios. The main conclusion from Jacobsen et al (2005) is that the Halloween effect is a worldwide effect affecting all investors driven by macro factors.

2.9.1 Disappearance of the Halloween Effect

Maberly and Pierce (2004) re-examine the research done by Bouman and Jacobsen (2002) and find that the Halloween effect disappears after they adjust for outliers. Lucey and Zhao (2008) find only weak evidence for an independent Halloween effect in the US market and that its effect is not steady over time for a value-weighted index.

Haggard and Witte (2010) redo the study of Maberly and Pierce (2004), using different statistical methods and find in the opposite; a significant Halloween effect just like Bouman and Jacobsen (2002) and suggest that outliers do not drive the results of Bouman and Jacobsen (2002).

2.9.2 SAD and Weather

Kamstra et al. (2003) find a relation between season affective disorder (SAD), a major depressive disorder caused by less daylight during the fall, and stock returns. Kamstra et al. (2003) argue as follows: psychology trials have proven that depression leads to a lower risk appetite. Therefore, investors who suffer from SAD are depressed during the fall when days becomes shorter and therefore lose their risk appetite and become more positive during the winter when days become longer again. Therefore during the fall they demand a higher risk premium causing stock prices to drop. As days become longer again investors become more optimistic again and risk appetite increases. A lower risk premium is demanded

[5] The is another ending to the saying, *"but buy back on St. Leger Day"*

and stock prices increase. Garrett et al. (2005) link SAD to the CAPM. They find that the SAD effect is fully captured by a model that captures both time variation in market risk premium and the market price of risk.

Jacobsen and Marquering (2008) comment on the studies of Kamstra et al. (2003) and Cao and Wei (2005). They argue that there is not enough evidence to link temperature induced mood changes to the Halloween effect. Jacobsen and Marquering (2008) show that other variables with a strong connection to the seasons explain the Halloween effect as well or even better, such as ice-cream production and air travel. Parker and Tavassoli (2000) suggest the opposite of what SAD suggests. People in sad mood become risk seeking, while people in good mood become risk averse. Goetzmann and Zhu (2005) and Theissen (2007) show that there is no connection between the weather and the buy and sell behavior of investors. Jacobsen and Visaltanachoti (2009) show that the Halloween effect is a seasonal effect and can explained by al kinds of variables as long as these variables are connected with the seasons.

2.9.3 Optimism Hypothesis

Another theory is the optimism hypothesis by Doeswijk (2008). People are over-optimistic (Weinstein 1980), and they make expectation in the last quarter of the year for the next year that are too positive. During the New Year, reality catches up with them and they update their positive expectations downwards. Therefore, stock prices increase during the first part of the year due to positive expectations and when investors find out their expectations were too positive, stock price declines.

2.10 Options and Calendar Seasonals

There is a lot of literature about seasonality in the stock market, whilst less is written about seasonality in the option market. Maloney and Rogalski (1989) look at call options at around the turn of the year. They argue that stock have a higher return around the turn of the year and expect call options therefore to also have a higher return. Maloney and Rogalski (1989) find that in the last 6 weeks of the calenderer year implied volatility (IV) increases and after the turn of the year decreases again. Jones and Shemesh (2011) try to find more evidence for the theory of Chen and Singal (2003) which is about the Monday effect and how it is due to short sellers closing out their position before the weekend. Jones and Shemesh (2011) find that over the weekend, call and put option have lower returns. They use delta hedge and non-hedged option portfolios, different maturities and liquidity checks. However, they find a lower return on Monday than on other days in all cases.

3. Data

In order to test the hypothesis, data is used from the Ivy DB dataset from Optionmetrics, which is the most comprehensive option database publicly available. Daily data is used from 4 January 1996 until 29 October 2010. Optionmetrics covers all US options on equities and indexes. The focus in this thesis will be on the S&P 500 single stock options with a maturity up to 52 days. The focus lies on single equity options instead of index options, because an index option is an option on a portfolio instead of a portfolio of options. Therefore, it is affected by correlation and volatilities, but it is less sensitive for systematic movements of idiosyncratic volatility than individual option portfolios (Jones and Shemesh, 2011). The difference between single equity options and index options will be further explored in chapter 7.3. The data used includes daily closing bid –ask quotes, open interest, implied volatility (IV), option delta and vega calculated by Optionmetrics. For the individual American equity options, Optionmetrics uses the binomial tree model from Cox et al. (1979) to compute the option IV. The IV surface is constructed of estimated volatilities and adjusted by a kernel smoothing technique[6]. Battalio and Schultz (2006) note in their paper that Optionmetrics´ closing option quotes and closing stock prices are obtained at different points in time and hereby lead to violation of the put – call parity and other arbitrage bounds. Therefore, the put – call parity will not be used in this thesis. This is not an issue for the study, because the tests require no perfect coordinated trading data of the stock and the option market. The daily stock price data is from CRSP and stock beta is from Datastream.

The Optionmetrics database does not cover option prices, but includes end of the day bid – ask quotes. In order to calculate the option prices, the standard method of averaging bid – ask pricing is used. The portfolios´ returns are measured on equal- and value weighted average of the returns. Options have a net zero supply, so they cannot be value-weighed in the traditional way. Therefore, dollar open interest is used for value-weighting the options. Using value weighted option has two advantages. It emphasizes options that are more liquid and more representative for the entire option market and should reduce the measurement error bias pointed out by Blume and Stambaugh (1983).

For simplicity reasons, stocks going ex-dividend are not taken in to account. This influences the portfolios with returns. The other portfolios all are delta neutral and the decrease/increase of the value of the option will be offset by the increase/decrease value of the delta stock.

The strategy of early exercise is not held into account. With American call options on non-dividend paying stock, this is not an issue. Merton (1973) shows that it is never optimal to exercise an American call option on a non-dividend paying stock early. On the other hand, for American call options on dividend paying stocks, it may be optimal to exercise the option before the option goes ex-dividend. When it comes to American put options, early exercise may be optimal on dividend and non-dividend paying stocks (Engström and Nordén, 2000). For put options this is not a major issue, because early exercise is used to speed up the cash flow forward in time. In the long run, this may cause a large

[6] For more information see Optionmetrics manual.

difference in returns, but on a day to day basis the influence should be small. To not exercise a call option for a large dividend may make a huge difference for the day to day returns (Jones and Shemesh, 2011).

Optionmetrics does not give an IV for all the data in the sample and therefore also no delta and vega. No IV is given by Optionmetrics when one or more of the following conditions are met; the option has a non-standard settlement, the midpoint of the bid-ask price is below intrinsic value, the vega of the option is below 0.5 and finally, the implied volatility calculation fails to converge or the underlying price is not available. Prior researchers simply omitted the data from their sample (see for example Bali and Hovakimian, 2009). However, Duarte and Jones (2008) show that if the options without IV are systematically omitted from the dataset, the result will be the systematical removal of options with an observed price below their value. This will in turn lead to a downward bias in the options returns. Duarte and Jones (2008) suggest that it is better to fill in the missing IV with IV's of similar contracts. For example if an IV of a call (put) contract is missing, then the IV from a put (call) contract written on the same firm with the same strike price and maturity is used. If both call and put IV are missing, then the IV is used from the same contract on the previous day or even further back if necessary.

For the contracts with the missing IV, the delta and vega is calculated after finding their IV by using the Black and Scholes (1987) formula with a zero dividend yield.

$$\text{Delta Call} = \Phi\left[\frac{\ln\frac{s}{X}+\left(r-\lambda+\frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}\right] \tag{2}$$

$$\text{Delta Put} = -\Phi\left[-\frac{\ln\frac{s}{X}+\left(r-\lambda+\frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}\right] \tag{3}$$

where:

- s is the price of the underlying asset
- X is the strike price of the option
- r is the risk free rate
- σ is the volatility of the underlying asset
- t is the time to maturity
- ϕ [·] the standard normal probability density function
- λ is the dividend yield of the underlying asset.

$$\text{Vega Call/Put} = Se^{-qt}N\left[-\frac{\ln\frac{s}{X}+\left(r-\lambda+\frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}\right]\sqrt{t} \tag{4}$$

where N [·] is the standard normal cumulative distribution function.

The Optionmetrics zero yield curve with the shortest maturity for each day is used as a riskless rate in order to calculate the missing deltas and vegas. The zero yield curve contains the zero coupon interest rate curve, which is calculated from a collection of continuously compounded zero-coupon interest rates at various maturities.

If the delta and vega used to calculate the portfolios return is incorrect, the resulting returns still represent a viable investment strategy. The use of removing the stock price movements' influence on an option by delta hedging is standard in academic research and industry and has shown to be quiet precise (Duarte and Jones, 2008). Hull and Suo (2002) claim that the pricing and hedging of a plain vanilla instrument is generally model independent.

To filter out noise and large errors of the data set, all options with bid prices of 998 and ask prices of 999 are removed because Optionmetrics use these codes to show missing data. Options with large return reversals are also omitted from the dataset, as well as large bid – ask spreads, because they contain a lot of noise and can be market makers who do not want to trade and therefore post non-competitive quotes (which contain no information). Therefore the following rules taken from Jones and Shemesh (2011) are used:

- day t − 1 or t bid-ask spread is more than $5.00 or 200% of the midpoint
- day t − 1 or t ask price is less than the bid or more than twice the price of the underlying stock

To eliminate the most illiquid option contracts which are most of the time deep out the money, the next two rules are used:

- day t − 2 bid price is less than $0.50 or 0.1% of the price of the underlying stock
- day t − 2 bid-ask spread is more than 25% of the midpoint

3.1 Summary Statistics

Table 1 shows a summary of the statistics of the option portfolios, which are averaged per day with a maturity between 1 and 52 days. The call and put options have surprisingly high positive returns. This can be explained by two factors. First, more than 75 percent of the sample exists of ITM options with a delta greater than 0.6 (-.6 for put options), while 11 percent of the data sample consists of OTM options (delta < .4 call; - delta < put). Coval and Shumway (2001) show that daily option returns increase as their moneyness increases, so the majority of the dataset exists of relatively high returns options. To control for skewed distribution of the moneyness of the option and to compare the summary statistics to those of Coval and Shumway (2001), the summary statistics of ATM options (0.4 < delta < 0.6 call; -.6 < delta < -.4 put) are calculated. The ATM call options have a daily return of .61 percent (see part II of table 1) which is in line with Coval and Shumway (2001) and Jones and Shemesh (2011). These high daily returns are better understood when one knows that options have market betas between 20 and 55 (Coval and Shumway (2001).

The average put option return is still high (.27 percent) compared to Coval and Shumway (2001), where the put options have a negative return. This is explained by the average daily returns per annum (see appendix 1). A put option had a return of 5.06 percent during the financial crisis in 2008, while during the dot.com crisis in 2000 it had a return of 2.04 percent per day. When the year 2008 is excluded from the sample, the average put option return is - .24 percent. When the year 2000 is also excluded, the put option return is - .52 percent, which corresponds with Coval and Shumway (2001). The negative put option premium exists, because the put options hedge the negative jump risk premium. It is also known

that risk-averse investors are willing to overpay for put options (Chowdhury et al., 2011). The excess returns are lower than the "normal" returns. This is because the risk less rate is subtracted. The delta hedge returns are negative on average, which is in line with the literature (see for example Cao and Han, 2009).

The returns and excess returns are positively skewed as expected and the delta hedge returns are heavily negatively skewed due to the underlying stock. The skewness of the returns give a distorted mean, therefore the medians are also given. The returns have a median of exactly zero and the delta hedge return has a median of around -.55.

Table 1: Summary Statistics

This table reports the summary statistics of the returns, excess returns, delta hedge and excess delta hedge returns averaged per day. The mean and median are in percentage. The second half of the table shows the summary statistics for the ATM options. .4 < delta < .6 for the call options and -.6< delta < -.4 for the put options. Data is from Optionmetrics from 4 January 1996 until 29 October 2010.

Summarize Statistics						
	Mean	Median	Standard Deviation	Skewness	Excess Kurtosis	Observations
Call & Put Return	1.09	0.00	0.2282	27	7662	26031624
Call Return	1.33	0.00	0.2274	21	4895	13436176
Put Return	0.84	0.00	0.2290	33	10535	12595448
Call & Put Excess Return	1.09	-0.01	0.2282	27	7699	25946708
Call Excess Return	1.31	-0.01	0.2274	21	4923	13395665
Put Excess Return	0.84	-0.02	0.2291	33	10574	12551043
Call & Put Delta Hedge Return	-2.35	-0.55	0.1404	-301	356358	22087954
Call Delta Hedge Return	-2.42	-0.57	0.1516	-394	500254	11055449
Put Delta Hedge Return	-2.27	-0.53	0.1282	-138	45968	11032505
Excess Call & Put Delta Hedge Return	-2.35	-0.55	0.1402	-302	358559	22061108
Excess Call Delta Hedge Return	-2.32	-0.51	0.1509	-400	510013	11043312
Excess Put Delta Hedge Return	-2.38	-0.60	0.1286	-137	45631	11017796
Summarize Statistics ATM options						
Call & Put Return	0.44	-3.64	0.3135	5.75	287	2827034
Call Return	0.61	-4.14	0.3266	3.97	74	1418860
Put Return	0.27	-3.17	0.3004	7.88	565	1405652
Call & Put Excess Return	0.43	-3.64	0.3134	5.76	288	2817812
Call Excess Return	0.59	-4.16	0.3265	3.96	74	1414582
Put Excess Return	0.28	-3.19	0.3005	7.91	568	1400696

3.2 Portfolios

There are four different portfolios created to test whether there are calendar seasonals in the option market. The next section describes how these portfolios are realized.

3.2.1 Naked Call and Put Option Returns

From the option data, different types of portfolios are created. First, to find out whether there are any seasonal effects in option returns, portfolios of naked calls and puts returns and excess returns are created. To control for the case that the excess returns are found due to the way the riskless rate is computed or because the riskless rate was taken for the weekend or Holiday effect over multiple days, "normal" returns are also calculated.

Call and put returns:

$$R_t = \frac{C_t - C_{t-1}}{C_{t-1}} \tag{5}$$

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{6}$$

Call and put excess returns:

$$ER_t = \frac{C_t - C_{t-1}}{C_{t-1}} - ND_{t-1,t} r_{t-1} \tag{7}$$

$$ER_t = \frac{P_t - P_{t-1}}{P_{t-1}} - ND_{t-1,t} r_{t-1} \tag{8}$$

where:

- C (P) is the call (put) option midpoint
- r the riskless return per day
- ND is the number of calendar days between t-1 and t.

3.2.2 Delta Hedge Call/Put Option Returns

Delta hedge portfolios are created to remove any exposure to the underlying stock (delta neutral position). It is hereby prevented that the seasonal abnormal return is attributable to the underlying stock movement.

Delta hedge call and put return

$$R_t = \frac{C_t - C_{t-1}}{C_{t-1}} - -\frac{\Delta S_{t-1}}{C_{t-1}} \frac{S_t - S_{t-1}}{S_{t-1}} \tag{9}$$

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} - -\frac{\Delta S_{t-1}}{C_{t-1}} \frac{S_t - S_{t-1}}{S_{t-1}} \tag{10}$$

Delta hedge call and put excess returns

$$ER_t = \frac{C_t - C_{t-1}}{C_{t-1}} - ND_{t-1,t}r_{t-1} - \left(\frac{\Delta S_{t-1}}{C_{t-1}}\frac{S_t - S_{t-1}}{S_{t-1}} - ND_{t-1,t}r_{t-1}\right) \quad (11)$$

$$ER_t = \frac{P_t - P_{t-1}}{P_{t-1}} - ND_{t-1,t}r_{t-1} - \left(\frac{\Delta S_{t-1}}{C_{t-1}}\frac{S_t - S_{t-1}}{S_{t-1}} - ND_{t-1,t}r_{t-1}\right) \quad (12)$$

The position taken in the underlying position in stock and S the underlying stock price is shown by Δ.

3.2.3 Beta Neutral Straddle

To examine whether the abnormal returns are generated due to changes in volatility (vega) on the specific calendar days and not by any other characteristics, an ATM straddle portfolio is created. The straddle is created on a beta neutral base to avoid any influence of the underlying stock. When the stock price of the underlying asset increases or decreases a lot, the straddle becomes a long/short position in that asset instead of a straddle. This can be explained as follows: if the call (put) option has a delta of nearly 1 (-1) and the put (call) option a delta of approximately 0, the change in value of the call and put option do not cancel out each other's change in values when the underlying asset changes in value and everything else held equal. If for example the underlying stock increases $1 in value, the call option value increase roughly by $1, but the put option does not decrease by $1. To minimize the effect of straddles turning into a long/short positions, straddles are created one month from maturity using ATM calls and puts options with the same strike price. For the call option, a delta close to 0.5 is taken as ATM and for the put option a delta of -0.5. An advantage of using deltas instead of strike prices that are close to the spot price is that a lot of stocks pay out dividend. Therefore the spot price may not be close to the mean of the distribution of the stock price at expatriation. The deltas given by Optionmetrics are calculated following Cox et al. (1979), which implies using a binomial tree model which does not only take dividends into account, but also the possibility of early exercise (Bali and Murray, 2011).

Following Black and Scholes (1973) a beta of a call option is as follows:

$$\beta_c = \frac{s}{C} N\left[\frac{\ln\frac{s}{X} + \left(r - \lambda + \frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}\right]\beta_s \quad (13)$$

where:

- s is the price of the underlying asset
- C is price of the call option
- X is the strike price of the option
- r is the risk free rate
- σ is the volatility of the underlying asset
- t is the time to maturity
- N [·] is the cumulative normal distribution
- β_s is the underlying asset's beta and λ is the dividend yield of the underlying asset. Then the Beta of a put can be calculated:

The beta of a put option:

$$\beta_p = \frac{c}{P}\left(N\left[\frac{\ln\frac{s}{x}+\left(r-\lambda+\frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}\right]-1\right)\beta_s \tag{14}$$

Where P is the price of the put option. The beta of a call option is positive and the beta of a put is negative as long as the beta of the underlying asset is positive.

A beta neutral straddle now can be made by matching calls and puts with the same strike price, maturity, and underlying asset in proportion of their betas. Therefore, the effective beta of the asset becomes zero because the beta from the call cancels out the beta of the put (and vice versa).

Following Coval and Shumway (2001), the zero beta straddle can be constructed solving the next equation:

$$\begin{aligned} R_v &= \theta r_c + (1-\theta) r_p \\ \theta\beta_c + (1-\theta)\beta_p &= 0 \end{aligned} \tag{15}$$

where:

- r_v is the straddle return
- θ is the fraction of the straddle's value in call options
- β_c and β_p are the betas of the call and put
- r_c and r_p are the returns of call and put, respectively.

This problem is solved by the weight function:

$$\theta = \frac{-\beta_p}{\beta_c - \beta_p} \tag{16}$$

Substituting equation (15) in (14), the straddle return is calculated as follows

$$R_v = \frac{-\beta_p}{\beta_c - \beta_p} r_c + \frac{-\beta_p}{\beta_c - \beta_p} r_p \tag{17}$$

The call options beta β_c and the put option's beta β_p will be computed as the Black-Scholes beta given by equation (12) and (13). Coval and Shumway (2001) only use the call option betas and assume a put - call parity to calculate the put option beta. However, the closing prices from Optionmetrics are not taken at the exact same time and therefore the put – call parity does not hold. Therefore, both put and call option betas are used to construct the beta-neutral straddles.

3.2.4 Skewness Asset

Finally, a skewness asset is created to measure how much of the abnormal return is due to the increase or decrease of the skewness. The skewness asset is created using the method of Bali and Murray (2011). To create the skewness asset, only exposure to changes in skewness of the asset is included and the exposure to the underlying stock (delta neutral) and the volatility (vega neutral) is removed. Following Bali and Murray (2011), the skewness asset is created each month on the second trading day after

expiration, using out of the money (OTM) call and put options and a short position in the underlying stock. The OTM options contracts are contracts with a delta close to 0.1 (cal) or -0.1 (put). The skewness asset is created each month and is followed until expiration and the return is calculated each day by calculating the price changes of the combined assets.

The skewness asset consist of an OTM call contract, a position of $Pos_{p,\,otm} = -v_{c,\,otm}/v_{p,\,otm}$ in a OTM put contract and a position of $-(Pos_{c,\,otm}\Delta_{c,\,otm} + Pos_{p,\,otm}\Delta_{p,\,otm})$ in the underlying stock.

Therefore the price of a skewness asset is:

$$P = C_{otm} + \frac{-\upsilon_{c,otm}}{\upsilon_{p,otm}} P_{otm} - \left(Pos_{c,otm}\Delta_{c,otm} + Pos_{p,otm}\Delta_{p,otm}\right)S \tag{18}$$

where:

- C_{otm} is an OTM call contract
- v is the vega of the option
- P_{otm} is an OTM put contract
- $Pos_{c,\,otm}$ is the size of the position of the call contract (one)
- $\Delta_{c,\,otm}$ is the delta of the OTM call contract
- $Pos_{p,\,otm}$ is the position in the OTM put contract ($-v_{c,\,otm}/v_{p,\,otm}$)
- S is the underlying stock

The vega position of the put contract makes the skewness asset vega neutral and the delta position in the underlying stock removes the exposure for the movement of the stock price.

The return of a skewness asset is:

$$R_t = \frac{\text{Skewness asset}_t - \text{Skewness asset}_{t-1}}{\text{Skewness asset}_{t-1}} \tag{19}$$

If the risk neutral density function of the underlying stock of the skewness asset increases in a way that the right tail increases, but the left tail stays the same, then the OTM call option will increase in value and the value of the OTM put option will be unchanged. Therefore, the value of the skewness asset will increase with the value increase of the skewness of the risk neutral density. Now arguing vice versa; an increase in the left tail, while the right tail stays the same, causes the OTM put to increase in value, while the OTM call will keep the same value. The OTM put is shorted, which causes the skewness asset to lose value. To recapitulate, a positive return of the skewness asset is due to the increase of the right side of the risk-neutral- density function and a decrease in value of the skewness asset is an increase of the left sight of the risk neutral density function.

4. Methodology

4.1 Panel Data

The dataset has two dimensions; a cross-sectional dimension (subscript i) and a time series dimension (subscript t). Therefore, it is used as unbalanced panel data/longitudinal data, where the time variable is the date and the panel ID is the option identifier. The advantages of combining both differences between individual groups and inside individual data groups over using it as time-series or cross-sectional data are:

- The model parameters are more accurate because the data contains more degrees of freedom and more sample variability and thereby increasing the efficiency of the estimates (Hsiao et al. 1995).
- Panel data controls for the influence of omitted variables and uncovers dynamic relationships by the reduced collinearity and measurement errors (Hsiao, 2007 and Baltagi, 2007).

A problem with panel data is unobserved heterogeneity. Unobserved heterogeneity can either be in the form of a random variable or a fixed parameter or a combination of both. In the first case, a random effect model can solve the unobserved heterogeneity. For a fixed parameter a fixed effect model can be used. If there is a combination of random and fixed variable, there is a mixed effect model.

4.2 Random Effect Estimator

The random effect assumes that α_i is random distributed with a common mean and is independent of x_{it} (uncorrelated). The advantages of the random model are that when the sample size increases, the number of parameters stay the same and derivation of efficient estimators that use inter and intra (group) variation is allowed. The random model can also estimate the impact of time invariant variables.

$$y_{it} = \alpha_i + x_{it}\beta + z_i + \varepsilon_{it}, \tag{20}$$

where y_{it} is the dependent variable observed for individual i at time t, x_{it} is the time-variant regressor, z_i are time-invariant regressor, α_i are random individual specific effect which are random and uncorrelated with the regressor (uncorrelated with x_{it} or z_i), and ε_{it} is an idiosyncratic error term.

The major drawback of the random model is that the conditional density of α_i has to be given while it is unknown**.** If α_i is correlated with x_{it} the random model is not correct and as a result the estimator is biased (Hsiao, 2007).

4.2.1 Fixed Effect Estimator

The benefits of a fixed model are that it allows individual (i) or time specific (t) effects to be correlated to x_{it} and these correlation patterns do not have to be modeled. The drawbacks from the fixed model include that the number of unknown parameters increases with the number of sample observations. It also cannot measure time in-invariant (variables that do not vary overt time) variable because those are absorbed in the fixed effect (Hsiao, 2006).

$$y_{it} = \alpha_i + x_{it}\beta + z_i + \varepsilon_{it} \quad (21)$$

α_i is not independent of x_{it} or z_i, which allows for a limit form of endogeneity.

To summarize, both models are the opposite from each other; the drawback of the random error model is the advantage of the fixed effect model and vice versa.

4.2.2 Random or Fixed Model and the Hausman Test

Hausman (1978) notes that independent of α_i being fixed or random, the fixed effect estimator is consistent. However, the random estimator is efficient only when α_i is uncorrelated with x_{it}. Hausman creates a test to determine whether the random or fixed model has to be used. The null hypothesis states that the random effect model is consistent and the alternative hypothesis states that the random effect is inconsistent (the fixed model is always consistent). Although the Hausman test is commonly used (Baltagi et al. 2003) to assess whether the fixed or random model has to be used, it has some serious shortcomings. It assumes that one of the estimators is efficient and has minimal asymptotic variance. This is violated, for example, if your observations are clustered or the model is ill-specified (Cameron and Trivedi, 2009). Because the dataset used is clustered and likely heterogeneous, a Hausman test is invalid. Therefore, in line with the literature (see for example Arellano, 1993 and Hayansi, 2000), a user-made generalized Hausman test is used, which allows for heteroscedasticity and clustered errors and generates non-negative test statistics[7] (Schaffer and Stillman, 2010). The generalized Hausman test gives a significant P-value at a 5 percent level. Therefore, a fixed effect estimator will be used for the regressions.

4.3 Robust Standard Errors

Petersen (2009) points out that many researchers do not adjust their standard errors for dependence of their residuals. If standard errors are not adjusted for correlation between observations, standard errors may become smaller than they should be. Small standard errors get large t-statistics that lead to statistics that are significant, while they are in fact not, which leads to distortion of the analysis. Robust standard errors are a tradeoff between a decrease in bias, which improves the performance of the test statistic, and in an increase variance which increases the chance of finding significance even when there is none (Thompson, 2010). Stock and Watson (2008) show that normal robust standard errors are inconsistent when a fixed effect estimator is used. Therefore the standard errors are made robust by clustering them by date as will be described in the next section.

4.3.1 Robust Clustered Standard Errors

Thompson (2010) argues that double clustering is necessary in datasets that meet the following criteria:

- the regression error includes a significant time and firm component;
- the independent variable changes over time and cross section;
- the number of firm and time periods is about to be equal.

[7] Stata command – xtoverid - see for more information the xtoverid help file

There are a lot more option clusters than date clusters in the dataset, while there are only time specific independent variables in the regression; therefore the dataset is clustered along one dimension. The standard errors are clustered by time, since it is wiser to cluster along the dimension with the least observations in general, because this has a larger effect. An additional reason is that the regressors vary only by time and not by firm. The standard errors are also adjusted for white standard errors to account for possible correlations between clusters. Cluster errors are an Eicker-Huber-White-Robust treatment of errors.

The standard errors are clustered by date on an individual-level. The robust estimator of the variance-covariance estimator is a natural robust estimator when sampling vectors at the individual-level instead of the observation level. The individual-level robust estimator is preferred over the observation level, because sampling took place at the individual level. This individual level robust estimator is consistent, even in the presence of heteroscedasticity and autocorrelation (Cameron and Trivedi, 2005 and Wooldridge, 2002).

The sample size is large enough, as Kézdi (2004) shows that 50 clusters (with about the same cluster size) is often close enough to infinity for accurate inference.

4.4 Dummy Regression

To examine whether there are any seasonal patterns in the option markets, six seasonal dummies are regressed independently on the option (excess) returns

$$r_{it} = \alpha_i + \beta_{it} D_t^{Seasonal} + \varepsilon_t \quad (22)$$

where:

- r is the daily (excess) return for option contract i at date t;
- D is the seasonal dummy which turns to one on the day an abnormal return is expected due to a calendar effect and zero otherwise;
- β is the difference in return on a day when there is a calendar effect compared to a “normal” day;
- ε_{it} is an idiosyncratic error term .

Note that β is not the total excess return for that calendar day, but only the difference between a “special” calendar day and a “normal” one.

The six calendar effects are:

Effect	Seasonal Dummy (D = 1)	Non Seasonal Dummy (D = 0)
Halloween	November 1 – April 30	May 1 - October 31
January	January 1 – January 31	February 1 – December 31
January 2	January 1 – January 5	January 6 – December 31
Turn-of-the-month	Last trading day of the month (t-1) until fourth trading day of month (t +4)	Fifth trading day of the month (t + 4) until second last trading day of the moth (t – 2)
Holiday	Trading day before either New Year's Day, Presidents' Day, Good Friday, Memorial Day, July Fourth, Labor Day, Thanksgiving, or Christmas	All non Holiday trading days/all other days
Monday	Monday	Tuesday till Friday
Triple Witch	3 Friday of March, June, September and December	All other trading days

The academic community is not unanimous on the definition of all calendar effects. The definitions above are taken from Grimbacherm et al. (2010), with the exceptions of the Monday effect where Grimbacherm et al. take Friday as the dummy variable and they do not examine the Triple Witch effect. Two January dummies are created because as seen in the literature, the January effect has migrated to the first five days of the month (Moller and Zilca, 2008). If there is a January effect in the option market, it can be in both forms, as depicted in the list above. Consequently, both variants will be examined separately.

4.4.1 Interaction Calendar Anomalies

Some of the calendar anomalies have interaction with each other because they occur simultaneously on the same day. In order to examine if all the calenderer effects are independent effects or just a proxy of another calendar anomaly, a multivariate panel regression is run to control for all calendar anomalies simultaneously.

$$r_{it} = \alpha_i + \sum \beta_{it} D_t^{Seasonal} + \varepsilon_t \quad (23)$$

The calendar anomalies are not linearly related to each other; therefore interaction dummies are constructed to measure all individual calendar effects in combination with each other. This is not possible for the January effect; the January effect is always part of the Halloween effect. In total there are 63 different possible interaction combinations. However, there are 22 of them in the dataset, therefore 22 dummies are created.

5 Results

5.1Primary Results

Table 2A and 2B show the excess returns and delta hedge excess returns of equation (22) for the combined call & put portfolio, the call portfolio and the put portfolio. The first observation that strikes attention is that the Monday effect is significant for the call & put combination and for the call portfolio and the put portfolio on respectively a 1 and 5 percent level and it has a negative coefficient with nearly the exact same size for all portfolios of -.87 percent. Finding a significant negative Monday effect is in line with the findings of Jones and Shemesh (2011), who also find negative Monday returns in the option market. The negative Monday return also corresponds with the literature on the Monday effect in the stock market (see chapter 2 for the literature review).

The Halloween dummy shows a different sign for the coefficient of the dummy for the call portfolio than for the put portfolio. Although the Halloween dummy is not significant for the put portfolio, it has a positive coefficient. For the call portfolio, it has a negative coefficient and it is significant. This can be an indication that the Halloween effect is derived from excess negative stock returns on Halloween days. Therefore, when the stock price goes down the call price also goes down, but the put price goes up, ceteris paribus. This is contradicting to the Halloween effect in the stock market, where stocks earn positive excess returns during the Halloween period.

The same line of arguing can be used for the Holiday dummy, which also shows opposites signs in the call and put portfolio, while both are not significant. The Monday, January, TOM and Triple Witch dummies all have the same sign for their coefficients for the call and put portfolio. Therefore, their excess returns are probably not totally explained through the underlying stock movement.

Both January dummies are significant for the combined call & put portfolio. The January dummy has a lower excess positive return than the January2 dummy. This excess positive January return corresponds with the literature found on the January effect and the higher returns in the beginning of the month are in line with Moller and Zilca (2008).

The TOM effect is significant at a 1 percent level for the call & put options, and significant at a 5 percent level for the call portfolio. It has positive excess returns, in line with the effect seen in the stock market. The Triple Witch dummy is not significant in any of the portfolios.

In most regressions, the combined call & put portfolio is significant or significant on a higher level than the separate call and put portfolio. When the call and put portfolio are combined, the number of groups in the regression doubles. When the number of groups increases, it decreases the variance in the standard errors. The variance of the standard errors is estimated by averaging across groups. Including more groups means a greater number of terms in the average estimator and less in the estimation error. The number of groups increases the number of terms and the average standard error declines, so by doubling the number of information in the cross sectional dimension the variance of declines.

Table 2A: Results regression excess returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of excess returns.The excess returns are the dependent variables. Halloween, Holiday, Monday, January, TOM and Triple Witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

	Excess Return								
	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	1.26	0.0008	16.76	1.86	0.0032	5.84	0.62	0.0036	1.75
Halloween	-0.41	0.0015	-2.74***	-1.28	0.0065	-1.96**	0.54	0.0073	0.74
Constant	1.10	0.0003	33.73	1.29	0.0016	8.27	0.89	0.0018	5.07
Holiday	-0.34	0.0014	-2.40**	0.67	0.0086	0.78	-1.44	0.0091	-1.59
Constant	1.24	0.0003	36.68	1.46	0.0017	8.64	1.00	0.0019	5.30
Monday	-0.87	0.0009	-9.96***	-0.84	0.0041	-2.07**	-0.90	0.0045	-1.98**
Constant	1.02	0.0003	33.00	1.28	0.0016	8.17	0.75	0.0018	4.18
January	0.74	0.0015	4.82***	0.38	0.0061	0.63	1.13	0.0064	1.76*
Constant	1.06	0.0003	35.23	1.29	0.0015	8.49	0.81	0.0017	4.73
January2	2.11	0.0070	3.02***	1.99	0.0229	0.87	2.25	0.0183	1.23
Constant	0.98	0.0003	29.60	1.11	0.0017	6.40	0.84	0.0019	4.35
TOM	0.40	0.0008	4.93***	0.74	0.0036	2.04**	0.02	0.0041	0.06
Constant	1.09	0.0003	33.74	1.32	0.0016	8.45	0.84	0.0017	4.86
Triple Witch	-0.03	0.0014	-0.19	-0.03	0.0087	-0.04	-0.02	0.0095	-0.02

Table 2B: Results regression delta hedge excess returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of delta hedge excess returns.The delta hedge excess returns are the dependent variables. Halloween, Holiday, Monday, January, TOM and Triple Witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 until 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

Delta Hedge Excess Return

	Call & Put			**Call**			**Put**		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	-2.40	0.0006	-41.55	-2.42	0.0009	-27.37	-2.38	0.0006	-40.19
Halloween	0.13	0.0012	1.05	0.24	0.0019	1.25	0.02	0.0013	0.17
Constant	-2.35	0.0002	-98.11	-2.32	0.0004	-57.71	-2.38	0.0002	-96.04
Holiday	0.20	0.0011	1.78*	0.21	0.0021	1.03	0.19	0.0012	1.56
Constant	-2.24	0.0003	-88.71	-2.24	0.0004	-51.34	-2.23	0.0003	-85.17
Monday	-0.63	0.0006	-10.16***	-0.44	0.0010	-4.56***	-0.82	0.0006	-12.81***
Constant	-2.32	0.0002	-99.06	-2.28	0.0004	-59.31	-2.36	0.0003	-89.30
January	-0.35	0.0015	-2.39**	-0.48	0.0025	-1.95*	-0.21	0.0011	-1.89*
Constant	-2.33	0.0002	105.97	-2.29	0.0004	-62.64	-2.37	0.0002	-97.14
January2	-1.13	0.0057	-1.99**	-1.53	0.0100	-1.53	-0.72	0.0026	-2.75***
Constant	-2.44	0.0003	-93.12	-2.40	0.0004	-54.08	-2.48	0.0003	-84.67
TOM	0.34	0.0005	6.26***	0.30	0.0009	3.34***	0.37	0.0005	7.35***
Constant	-2.35	0.0002	-98.83	-2.32	0.0004	-58.00	-2.37	0.0002	-96.46
Triple Witch	0.01	0.0011	0.09	0.05	0.0018	0.30	-0.04	0.0013	-0.29

For the delta excess returns (table 2B) all the constants are significant on a 1 percent level and negative. The negative constants are caused by the fact that delta hedge returns are on average negative (Cao and Han, 2009). The lower robust standard errors and therefore the higher t statistics are due to the fact that the options are hedged by the underlying stock and are therefore less volatile.

As expected from the excess return regression, the Halloween dummy is not significant for the delta hedge excess returns. The delta hedge excess returns are corrected for the underlying stock price and therefore the excess negative returns on the stock on Halloween days do not affect the return of the options anymore. The significant Halloween effect in the excess return portfolio was apparently induced by the underlying stock. The same goes for the Holiday dummy.

The Monday dummy is significant at a 1 percent level for all three portfolios and shows the highest t-statistics of all dummies. Its coefficient is a bit smaller than the coefficient of the excess return portfolio. However, for the call portfolio it halves; probably a small part of the Monday negative excess returns is due to the underlying asset and now removed, but the larger part is clearly generated by another source. That is probably the reason why the excess call returns halves; it loses the influence of the negative excess returns on the stock on Monday, while the put option only seems to have less influence of the stock return and loses only a tenth of its excess negative return.

Both January dummies are significant at a 5 percent level and negative for the call & put combination portfolio, while the put portfolio for the January2 effect is significant at a 1 percent level. The excess positive returns, seen in table 2A, turn to excess negative returns for the delta hedge portfolio. The excess return of the Januart2 dummy is again larger than the excess return of the January dummy while it has the opposite sign as compared to the excess return regression. It does not seem that the positive excess return was induced by the underlying stock. The call and put option are both positive in the excess return regression, though not significant. That would not be the case if the January excess returns would be induced primarily by the positive excess stock return in January.

The TOM dummy is significant at a 1 percent level for all three portfolios. Its coefficient is nearly the same as for the excess return regression. The call coefficient is halved, as seen by the Monday effect. Because it is a delta hedge return it loses the positive excess stock return. The put portfolio is insignificant so therefore it cannot be verified if the size of the put coefficient increase.

The Triple Witch effect is not significant at all, as seen before in the excess return.

The finding of a significant Monday and TOM dummy and to a lesser extend significant January dummies in the delta hedge portfolio suggests that both effects are not entirely the result of seasonality in the underlying stock. If the seasonals in the options would be induced through the seasonality in the underlying stock, the delta hedge would eliminate most if not all of the excess return and this would decrease its significance. The calendar effects observed in the delta hedge excess returns are more clearly observed than in the excess returns and not driven by the underlying stock results. The literature on calendar anomalies in stocks is huge, therefore the main focus will be on the delta hedge excess returns.

5.2 Interaction Effect

At first hand, it seems that two of the six calendar seasonals from the stock market also manifest themselves clearly in the option market, which are the Monday and TOM effect. In addition, there also seems to be a January effect, but by far not as convincing as the Monday and TOM effect. To test if there are any interaction effects between the seasonals and whether the Monday, Tom and January effects are not explained by other seasonal effects, equation (22) is run with all six calendar dummies at once and it is ran twice: once with the January dummy for the entire month January and once with the January dummy for the first 5 days (January2). See appendix 2 for the results.

The most interesting observation is that all dummies except the Triple Witch dummy seem significant at a 1 percent level for the excess return call & put portfolio. Secondly, the TOM dummy shows significant negative returns for call and put portfolio while in the singular delta hedge and excess return portfolio (table 2A and 2B) it has positive returns.

The problem with this multiple regression is that it is based on the assumption that the seasonals are linearly related to each other. The multivariate regression does not assume interaction between the seasonals. For example, the Holiday effect may be stronger when there is a Monday effect. Therefore, to correct for non-linear interaction between dummies regression (23) is run.

Table 3A and 3B show the outcomes of regression (23) for excess returns and delta hedge excess returns. The first part of the table shows the effects of the seasonals when they are the only seasonal on a specific day. The other part of the table shows all possible combinations. In total, there are 22 of the possible 63 combinations of dummies that occur. For example, January always coincides with the Halloween effect, meaning that there is no January effect alone and consequently there is no independent dummy for the January effect. The January2 dummy is omitted because it never occurs on its own and is always accompanied with the Halloween dummy. All other combinations always include the Halloween effect and occur at most onetime per year. Therefore January2 has a low frequency too derive any conclusion from it.

In the interaction excess return portfolio (table 3A), the independent Halloween effect is not significant, where the Halloween effect is significant in the independent regression and the multiple regressions. The reason for a significant Halloween effect in singular and multiple regressions can be found in the second part of table 3A. When the Halloween effect coincides with the Monday effect, the combination is significant for the call & put portfolio and the call portfolio at respectively a 1 and 10 percent level. The Monday effect is independently significant for the call & put and call portfolio as seen in the upper part of table 3A. When the Halloween effect coincides with the Monday effect, its coefficient becomes negative through the negative Monday excess returns. This evidence leads to the conclusion that the Halloween effect is not a seasonal in its own, but a proxy of the Monday effect.

The Holiday dummy is significantly independent for the put portfolio. Nevertheless, it has a frequency of .40 percent, meaning that on average it has one observation per year and 15 observations through the entire dataset which is too low to come up with a conclusion about the independent Holiday effect.

The independent Monday dummy has a lower significance level for the call portfolio and is insignificant for the put portfolio, where the TOM dummy is positive significant for call & put portfolio.

The combined Halloween and January dummy is significant at a 10 percent level for the call & put portfolio and insignificant for the other two portfolios. The Halloween effect is also always combined with the January effect in the singular regression, therefore the combined Halloween – January dummy is the closest dummy to come to an independent January effect. The insignificance of this combination indicates that the January dummy on its own is not a seasonal.

For the third and fourth part of table 3A, the combination of the Halloween, Holiday and Triple Witch effect is striking. It has a huge positive return for the call portfolio 13.23 percent, and a huge negative return -12.28 percent for the put portfolio; however it occurs 0.04 percent of time in the sample. Meaning it is one day (21 December 2007). The huge excess returns are due to the underlying stock. This also becomes clear in table 3B where the returns are delta hedged and the huge excess returns are hedged away and become -.66 percent for the call and -0.15 for the put portfolio.

Table 3B shows the same regression as table 3A, but it does so for the delta hedge excess returns. The independent Halloween dummy is significant at a 5 percent level for the call & put portfolio, while it is insignificant in the singular regression on delta hedge excess returns (table 2B). An explanation for significant Halloween dummy may be the influence the Monday effect has on the Halloween effect. When the Monday and Halloween effect occur on the same day, the negative Monday effect evens out the positive Halloween effect. The Halloween and Monday effect occur together 4.12 percent of the time, which is around 20 percent of all the Halloween observations. The Halloween defect independently has an excess return of 0.15 percent, where the combination of the Halloween and Monday dummy has an excess return of -.72 percent. The same is true for the combination of the TOM and Halloween dummies which occur even more often; 7.34 percent of the time. When the Halloween and TOM effect are combined, they are significantly positive for all three portfolios. The Halloween effect seems to be a plaything of the Monday and TOM effect. While the Halloween effect is heavily influenced by the Monday and TOM effect, it is not significant when joined by the January effect.

The Holiday effect independently has a significant call portfolio, but as mentioned before occurs only 0.43 percent of the time in the delta hedge excess return what makes it negligible. The significant Holiday in the singular delta hedge excess return is probably due to the combination of the Holiday and TOM effect that occurs 1.22 percent of time and has a positive return.

The Monday and TOM dummies are both significant for all portfolios at a 1 percent level, just as in the singular delta hedge excess return regression.

Wrapping up the first results, it can be observed that there are calendar seasonals in the option market and these are the best visible in the delta hedge returns. The seasonals are not all derived through the price movement of their underlying asset; otherwise they would not be clearer in the delta hedge excess return than the excess return. The Monday effect is significant at a 1 percent level for both the excess return and delta hedge excess return. When tested independently of other seasonal effects, it shows the

same strong results. The TOM effect is less visible in the excess return portfolio than in the delta hedge excess return portfolio. This is probably caused by the delta hedge portfolio being less volatile. When tested independently of other seasonals, it shows the same significant results as seen in the singular regression. The January effect seems significant, but not as obvious as the Monday and TOM effect. Unfortunately, it cannot be tested independently, because it always coincides with the Halloween effect. However, when tested in combination with the Halloween effect it is insignificant. The Halloween effect found is due to the Monday and TOM effect as shown in the independent regression. The fact that the Halloween effect is so heavily influenced by the Monday and TOM effect is not strange, because about 25 percent of occurrences of the Halloween effect coincide with the Monday and TOM effect. It seems that the Holiday effect is not an anomaly in the option market. Although it is significant in the delta hedge excess return portfolio, it is not significant when measured without influence of other calendar seasonals. The Holiday dummy occurs more often in combination with another seasonal than on its own. For the Triple Witch effect, there was no evidence found at all.

Table 3A: Results regression excess returns on interaction seasonal dummies This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of excess returns.The excess returns are the dependent variables. The dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. A 1 in the left collom means that the dummy is activated a zero means it is inactive. Standard errors are calculted by clustering by date. The dummy coefficient indicates the additional positive/negative excess return received during that period. Coeffinient are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

						Excess Return									
						Call & Put			Call			Put			
Halloween	Holiday	Monday	January	TOM	Triple Witch	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	Frequencies
1	0	0	0	0	0	-0.11	0.0008	-1.33	-0.30	0.0043	-0.71	0.10	0.0048	0.20	18.10%
0	1	0	0	0	0	-0.17	0.0054	-0.32	3.91	0.0270	1.45	-4.56	0.0191	-2.39[2]	0.40%
0	0	1	0	0	0	-1.18	0.0012	-9.93[1]	-1.31	0.0058	-2.25[2]	-1.04	0.0066	-1.59	7.49%
0	0	0	0	1	0	0.50	0.0011	4.59[1]	0.80	0.0053	1.52	0.18	0.0062	0.29	12.15%
0	0	0	0	0	1	-0.11	0.0021	-0.55	0.79	0.0121	0.65	-1.10	0.0133	-0.83	0.84%
1	1	0	0	0	0	-0.40	0.0024	-1.64	1.89	0.0160	1.18	-2.96	0.0162	-1.82[3]	0.72%
1	0	1	0	0	0	-1.04	0.0012	-8.84[1]	-1.27	0.0075	-1.68[3]	-0.79	0.0080	-0.99	4.12%
1	0	0	1	0	0	0.28	0.0016	1.73[3]	-0.49	0.0069	-0.72	1.14	0.0077	1.48	5.05%
1	0	0	0	1	0	0.18	0.0011	1.61	0.53	0.0060	0.88	-0.20	0.0071	-0.28	6.88%
1	0	0	0	0	1	0.03	0.0020	0.17	-1.53	0.0107	-1.43	1.85	0.0118	1.57	0.80%
0	1	0	0	1	0	-0.33	0.0021	-1.61	0.55	0.0135	0.41	-1.25	0.0154	-0.81	1.09%
0	0	1	0	1	0	0.07	0.0026	0.25	0.39	0.0124	0.31	-0.28	0.0123	-0.22	2.55%
1	1	1	0	0	0	-1.83	0.0015	-11.91[1]	-0.79	0.0098	-0.81	-2.73	0.0087	-3.14[1]	0.02%
1	1	0	1	0	0	0.53	0.0052	1.00	-2.27	0.0263	-0.86	3.55	0.0306	1.16	0.52%
1	1	0	0	1	0	-0.95	0.0026	-3.69[1]	-2.88	0.0134	-2.15[2]	1.37	0.0168	0.81	0.41%
1	1	0	0	0	1	0.53	0.0017	3.14[1]	13.23	0.0107	12.38[1]	-12.28	0.0125	-9.81[1]	0.04%
1	0	1	1	0	0	-0.96	0.0024	-3.93[1]	-0.24	0.0110	-0.22	-1.76	0.0127	-1.38	0.89%
1	0	1	0	1	0	-0.15	0.0039	-0.38	-0.57	0.0122	-0.47	0.30	0.0174	0.17	1.79%
1	0	0	1	1	0	1.51	0.0048	3.11[1]	1.62	0.0162	1.00	1.38	0.0136	1.01	2.08%
1	1	1	0	1	0	-1.17	0.0013	-8.88[1]	-4.14	0.0190	-2.18[2]	1.81	0.0167	1.09	0.08%
1	0	1	1	1	0	0.29	0.0036	0.82	2.93	0.0201	1.46	-2.56	0.0200	-1.28	0.47%
Constant						-0.32	0.00082	-3.951	-0.15	0.0040	-0.38	-0.51	0.0044	-1.14	33.52%
Total															100.00%

Table 3B: Results regression excess returns on interaction seasonal dummies This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of delta hedge excess returns.The delta hedge excess returns are the dependent variables. The dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. A 1 in the left collom means that the dummy is activated a zero means it is inactive. Standard errors are calculted by clustering by date. The dummy coefficient indicates the additional positive/negative excess return received during that period. Coeffiecients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

						Delta Hedge Excess Return								
						Call & Put			Call			Put		
Halloween	Holiday	Monday	January	TOM	Triple Witch	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	Frequencies
1	0	0	0	0	0	0.15	0.0007	2.09[2]	0.17	0.0012	1.43	0.13	0.0008	1.64	17.71%
0	1	0	0	0	0	-0.39	0.0029	-1.36	-0.79	0.0040	-1.98[2]	0.00	0.0024	0.00	0.43%
0	0	1	0	0	0	-0.82	0.0011	-7.71[1]	-0.57	0.0017	-3.31[1]	-1.06	0.0010	-10.89[1]	7.40%
0	0	0	0	1	0	0.47	0.0007	6.83[1]	0.39	0.0012	3.42[1]	0.54	0.0007	8.26[1]	12.32%
0	0	0	0	0	1	-0.03	0.0019	-0.17	0.09	0.0029	0.31	-0.16	0.0019	-0.83	0.81%
1	1	0	0	0	0	0.06	0.0023	0.25	-0.33	0.0045	-0.73	0.45	0.0015	2.97[1]	0.59%
1	0	1	0	0	0	-0.72	0.0010	-7.12[1]	-0.48	0.0015	-3.22[1]	-0.96	0.0010	-9.91[1]	4.11%
1	0	0	1	0	0	0.03	0.0010	0.34	-0.02	0.0017	-0.14	0.09	0.0010	0.86	4.64%
1	0	0	0	1	0	0.47	0.0008	5.95[1]	0.43	0.0013	3.31[1]	0.52	0.0008	6.20[1]	7.34%
1	0	0	0	0	1	0.08	0.0012	0.63	0.00	0.0018	-0.03	0.10	0.0016	0.63	0.75%
0	1	0	0	1	0	0.38	0.0018	2.10[2]	0.50	0.0037	1.37	0.26	0.0021	1.26	1.22%
0	0	1	0	1	0	-0.11	0.0014	-0.76	0.00	0.0018	-.03	-0.21	0.0017	-1.21	2.52%
1	1	1	0	0	0	-0.77	0.0013	-6.02[1]	-0.93	0.0015	-6.13[1]	-0.62	0.0015	-4.27[1]	0.02%
1	1	0	1	0	0	-0.24	0.0035	-0.68	0.04	0.0036	0.11	-0.50	0.0068	-0.74	0.29%
1	1	0	0	1	0	0.89	0.0023	3.95[1]	1.36	0.0044	3.12[1]	0.39	0.0025	1.57	0.36%
1	1	0	0	0	1	-0.40	0.0012	-3.20[1]	-0.66	0.0020	-3.30[1]	-0.15	0.0019	-0.77	0.04%
1	0	1	1	0	0	-0.43	0.0016	-2.71[1]	-0.48	0.0026	-1.85[3]	-0.39	0.0017	-2.32[2]	0.90%
1	0	1	0	1	0	-0.16	0.0013	-1.22	-0.09	0.0023	-0.39	-0.23	0.0018	-1.24	1.92%
1	0	0	1	1	0	-0.39	0.0039	-1.01	-0.52	0.0067	-0.77	-0.26	0.0018	-1.42	2.15%
1	1	1	0	1	0	0.63	0.0029	2.13[2]	1.46	0.0048	3.07[1]	-0.19	0.0018	-1.02	0.08%
1	0	1	1	1	0	-0.70	0.0039	-1.80[3]	-1.00	0.0068	-1.47	-0.40	0.0037	-1.06	0.49%
Constant						-0.02	0.000594	-0.25	0.09	0.0010	0.93	-0.12	0.0006	- 2.00[2]	33.91%
Total															100%

6. Robustness Checks

6.1 Normal Returns

To test whether the Monday, TOM and January effect are due to the way the riskless rate is calculated, equation (22) is run on “normal” option returns and “normal” delta hedge returns. These returns are the same as the excess returns, however, the riskless return is not subtracted from the returns. The tables can be found in appendix 3A and 3B.

The only difference between the return regression and excess return regression is that in the call portfolio for the Monday dummy the significance level drops from 5 percent to 10 percent. Between the delta hedge returns and delta hedge excess returns there is no difference in the level of significance of the dummies. Concluding, none of the seasonals changes in significance levels when returns, excess returns, delta returns and delta hedge excess returns are compared. Therefore, it can be said that the findings are not due to the way the riskless rate is calculated.

6.2 Liquidity

Stock options can be extremely illiquid. To test whether the results are not through illiquid option contracts, a value-weighted regression (VW) is run. The options are value-weighted by their value of open interest (price * open interest). In this way, the options that are most traded and thus have a representative price have the highest impact. In addition, because the price of the option is taken, options with a high maturity, volatility and moneyness get a higher impact.

In appendix 4A and 4B the results of the excess and delta hedges excess returns of the VW regressions are shown. The excess return table gives a total different view than seen before. Especially the Monday dummy insignificance is striking. There are no results which are significant on a 5 percent level.

The VW regressions for the delta hedge excess returns give a more familiar picture. The Monday dummy is significant for the call & put portfolio and the put portfolio, but insignificant for the call portfolio. The TOM dummy is significant for all three portfolios and the Halloween dummy is also significant on all levels, only one level lower than the TOM does. The January dummy is also significant in all portfolios. The January2 is a lot weaker.

A reason for the shift in calendar seasonal in the VW regressions might be that the seasonals are created by low priced and/or illiquid options. In the VW regressions, the highest priced and most liquid options have the most influence. To test whether this is the case, three separate tests for liquidity, moneyness and maturity are run.

To test if illiquid options are the reason for the existence of calendar seasonals in the option market, a liquid sub sample is created. From the original dataset the options from the more liquid S&P100 are kept and all options with a zero open interest are deleted. Therefore, all options in the new data set are traded in the secondary market. By taking these measures, a more liquid data set is obtained. The results for this dataset can be found in appendix 5A and 5B. In the excess return regression, a small drop

in significance levels is visible for the Halloween, Holliday and Monday effect compared to the original regression in table 2A. For the delta hedge excess returns this is not the case. The Holiday dummy even is significant for the put portfolio. The regression the S&P 100 proves that for the excess return regression, liquid options are slightly less influenced by the calendar seasonals, while for the delta hedge excess returns the liquid options show the same effect. Therefore, the calendar anomalies are not specific to illiquid options.

6.3 Moneyness

To test whether the seasonals are correlated with the moneyness of the option, the original dataset is divided in five portfolios based on delta of the option, of which the results can be found in appendix 6A and 6B. First thing that is striking is that more than 60 percent of contracts in the dataset have a delta of .8 or larger (-.8 of smaller for put options), while the most OTM delta buckets exist out of only 2.81 percent of the total options in the dataset. The two ITM delta buckets are responsible for more than 75 percent of data in the sample. Therefore, the results found so far are heavily influenced by these ITM options.

The Halloween and Holiday effect are only present in the ITM and ATM options and not in the OTM options. They are probably significant in the singular regression because of the OTM options. The Monday effect is significant for all buckets except the highest ITM money bucket. The excess returns become smaller when the moneyness increases. Next to the decreasing economical significance of the Monday effect, the statistical significance also declines with the increase of the delta for the put portfolio. The economical effect of decreasing excess returns with increasing maturity is also visible for both January dummies and the TOM dummy. Notable is also the Triple Witch dummy, which is significant in all buckets, except the most OTM delta bucket.

For the delta hedge excess returns the Monday, January, January2 and TOM effect show the same economical pattern and the Monday effect also shows the statistical pattern. The Halloween dummy is insignificant for all delta buckets and portfolios. The Holiday dummy has two significant put buckets between -8< delta < -.4 as the call & put portfolio.

The moneyness seems to have an enormous influence in the regression because of the skewed distribution of the options in the database. The Halloween and Holiday effect seem to be extra influenced by this, because only the ITM options show the seasonals. For the Monday, January, January2 and TOM effect, it seems that the skewed distribution only pushes the size of the effect down and for the already statistical strong Monday effect the skewed distribution also pushes its statistical significance down, because the dataset is heavily skewed in the direction of ITM options.

6.4 Maturity

The existing of calendar seasonals may differ per maturity. To test whether the maturity of an option has an influence on the presence of calendar anomalies, the data sample is divided in three different sub samples. Sample one is closest to expiration and has a maturity from 1 to 10 days. The options in the

second sample have a maturity between 11 and 31 days and the third sample has a maturity from 32 till 52 days. The results are shown in appendix 7A and 7B.

For the January2 dummy there are no options with a maturity between 1 and 10 days, because the earliest expiration date in January is the third Friday in January and that is always more than 10 days away. For the lowest maturity basket, the Halloween and January dummy suffer from collinearity when regressed with a fixed estimator regressor. The Halloween and January dummy suffer from collinearity because they relatively have too many cross-sectional observations and therefore too many dummy variable needs to be specified. Nevertheless, in order to get results for the Halloween and January effect a normal panel regression with clustered standard errors is run without the effect the fixed effect estimator.

The results are stated in appendix 7A and 7B. The distribution of the options is quite even. The maturity ranges from 1 to 10 days and has only half of the observations of that of the other two maturity ranges. It is due to the fact that it also has half of the total days the other observations have.

The excess returns per maturity bucket results (appendix 7A) do not show special results; all results are in line with the expectations. For the Monday and TOM dummy the lowest maturity bucket is significant, while for all other dummies it is insignificant. For the TOM call & put portfolio, the 11 till 31 days maturity bucket insignificant.

For the delta hedge excess returns (appendix 7B) the lowest maturity overall is also insignificant or has the lowest t value of the three buckets for the portfolio. Calendar seasonals seem to be weaker when an option is close to expiration. For the January effect the sign even changes from negative to positive when an option enters the closed bucket. The TOM dummy shows a negative significant sign in the 11 till 31 days bucket, while it is significantly positive for all other buckets.

6.5 Sub Samples

To test whether the results found are due to datamining, equitation (22) is run per annum on the excess returns and delta hedge excess return portfolio. If the calendar seasonals returns have consistent robust results, it is confirmation that the results are not from datamining. The results are in Appendix 8A, 8b, and 8C for the excess results and 9A, 9B and 9C for the delta hedge excess returns.

For the excess returns (appendix 8) the Halloween, Holiday, January2 and Triple Witch dummies all show significant results in some years, but most of the time they show insignificant results. The significant results within a portfolio show different signs for their coefficient as well. This indicates that the significant excess returns are not generated by a consequent underlying source and it is more likely that the significant results are a coincidence.

The Monday dummy is significant 9 out of 15 years at a 1 percent level for the call & put portfolio and it is respectively 2 and 4 years out of 15 years significant for the call and put portfolio. All the significant Monday returns are negative. The TOM dummy is positively significant 8 out of the 15 years for the call & put portfolio, 2 years for the call portfolio and 2 years for the put portfolio. For the put portfolio, it is

negatively significant as opposed to the positive results for the call & put and call portfolio. This is because the positive excess returns generated by the underlying stock are creating a negative significant excess return for the put option, ceteris paribus. The January dummy has 8 positive significant returns for the call & put portfolio and 1 for the put portfolio.

For the delta hedge excess returns (appendix 9) the Halloween and Holiday effect show a clearer picture. All coefficients, except one, have the same sign, but still have a low number of significant years.

The Monday dummy is 10, 9 and 13 times out of the 15 times significant respectively for the call & put, call and put portfolio. There is a clear declining trend in the size and statistical significance of the Monday effect. Especially from 2007 and onwards the Monday effect seems to disappear. The Monday effect is also stronger in the put portfolio than the call portfolio. If the results are compared to the results of Jones and Shemesh (2011) who also do a per annum regression with delta hedge excess returns for the Monday effect, the results match with exception of the years 2006 and 2007. Jones and Shemesh (2011) hereby find significant negative results while insignificant negative returns are found for the combined portfolio. This may be explained by the skewness of the dataset used, as shown in chapter 6.3, where the Monday effect is less present in the high ITM options which consist of 60 percent of the dataset. The observed downward trend therefore manifests itself earlier in the data. Jones and Shemesh (2011) also find a weaker Monday effect in the call portfolio than in the put portfolio. The TOM effect is not as consequent as the Monday effect, but still is significant 10 out of the 15 years for both the call & put portfolio and put portfolio but only 5 times for the call portfolio. All significant results are positive.

The January effect shows an interesting feature. Where it was negative for most of the delta excess returns regression it now is positive for all but one coefficient. This result cannot be driven by a negative outlier, since there is only one negative coefficient, which is -1.01 percent for the 1996 call portfolio. This is far too small to change the sign of 14 years of positive returns. The January dummy showed already positive returns for the options with a maturity up to 10 days (see section 6.4). However, the maturity range 1 till 10 days represents 18.54 percent from the January return, while the 11 till 31 range and 32 till 52 range represent 41.17 and 40.67 percent, respectively.

After all seasonals are tested with all kinds of robustness checks, there can be concluded that the Monday and TOM effect are seasonals in their own right and cannot be explained by the way the riskless rate is calculated. The effects are also not due to illiquid options. When it comes to moneyness, the Monday and TOM effect are visible in all level of moneyness, except the highest ITM options, which explains why the Monday dummy was not significant in the VW regression. A remarkable fact is that the TOM dummy changes signs in the maturity ranges of 11 till 31 days and becomes negative. The per annum regression gives the idea that the Monday effect is declining and fading away.

The January effect is not consequently significant for the separate call and put portfolios in the robustness check. However, it shows remarkable significant consequent returns in the per annum regression and changes signs from negative to positive for the delta hedge excess returns.

7. Seasonals in Straddle, Skewness Asset and Index Returns

The underlying stock explains the Monday, January and TOM effect only partly, as all three seasonals are significantly visible in the delta hedge excess return regression (table 2B). Therefore, straddle and skewness asset portfolios are created to test whether a part of their excess returns can be explained by volatility and/or skewness of the asset.

7.1 Straddle Excess Return

The straddle portfolio is created as described in the methodology section 3.2.3. Note that the Triple Witch dummy is absent, because the straddles a created one day after expiration day for a month. During that month the daily returns are calculated from the created straddle. Therefore there are no returns on expiration day because then the straddle expires and consequently no returns on the Triple Witch effect.

The excess return straddle portfolio shows that the Monday, January, January2 and TOM dummies are all significant (table 4). The Monday dummy has the highest t statistic (-17.67) and largest excess return of -3.74 percent, implying that the volatility declines a lot on Monday in comparison to the other coefficients. The drop in volatility explains part of the drop in option prices on Monday, for both call and put options. The decline in volatility is also in line with the findings of Jones and Shemesh (2011), who find a lower volatility on Mondays than any other days of the week. A reason for the decrease in volatility on Monday is that options lose volatility when moving closer to their expiration date. Between Friday and Monday, an option has two days extra before the expiration date than between “normal” trading days. Therefore on Monday, the IV decreases more than on other days because it has to make up for the weekend. Although this reasoning sounds clear, French and Roll (1986) show that the amount of decline in volatility in the underlying asset is the same over the weekend as during “normal” trading days and therefore the decline in implied volatility on Monday is not in line with the pattern in realized volatility.

Both January dummies are positively significant for the excess straddle returns. This may explain the positive returns found in the regression per annum, but is not in line with the negative returns found in the other regressions. The excess positive straddle returns found in January means either that the realized volatility is higher in January or that the IV is lower or a combination of those two.

The TOM dummy is significant as well, but has a small coefficient (0.76%) in comparison with the other significant dummies. The increase in volatility around the TOM is partly the explanation for the positive excess TOM returns.

7.2 Skewness asset

When it comes to the skewness asset, the Triple Witch dummy is absent as well. This is for the same reason as for the straddle portfolio, where the skewness asset is created the second day after expiration day for one month and expiries the next expiration day so consequently there is no Triple Witch data.

For the skewness asset, the Monday and TOM effect are significant, both on a 1 percent level. The Monday dummy has a negative coefficient of -3.7 percent, meaning that on Monday, an increase in the left side of the risk- neutral- density function is observed. The market in general expects on Monday that there can be larger negative returns than they expected before. For the TOM dummy, an increase in the right side of the risk neutral density curve of 1.19 percent is found. Around the turn of the month, investors have a positive expectation of the market and expect possible positive returns that are higher than before. Both the January dummies are insignificant and these excess returns are apparently not due to a change of the risk- neutral- density function. All constants are negative, because on average the skewness asset has a negative return (see for example Bali and Murray, 2010).

Table 4: This table reports the results of the fixed pooled equal weighted (ew) regression on the excess return of a straddle portfolio and the excess return of a skewness asset portfolio.The straddle or skewness excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

Excess Straddle and Excess Skewness Asset Return

	Excess Staddle			**Excess Skewness Asset**		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	1.57	0.0023	6.84	-1.65	0.0037	-4.49
Halloween	-0.91	0.0049	-1.84*	-0.80	0.0081	-0.99
Constant	1.23	0.0010	12.70	-2.00	0.0016	-12.40
Holiday	-0.64	0.0048	-1.35	0.47	0.0071	0.65
Constant	1.98	0.0010	18.96	-1.39	0.0017	-8.34
Monday	-3.74	0.0021	-17.67***	-3.70	0.0045	-8.19***
Constant	1.04	0.0010	10.56	-2.00	0.0017	-11.99
January	1.83	0.0040	4.54***	0.22	0.0075	0.29
Constant	1.17	0.0009	12.42	-1.98	0.0016	-12.62
January2	3.73	0.0137	2.71***	-0.15	0.0190	-0.08
Constant	0.99	0.0011	8.82	-2.37	0.0018	-13.08
TOM	0.76	0.0020	3.74***	1.19	0.0036	3.35***

7.3 S&P 500 Index Options

At first hand, it is expected that index options show the same seasonal anomalies as the equity options, because they are both driven by the same (risk) factors. However there are some differences between index options and single equity options. Index options are European options, cash settled and have a continuous dividend stream[8], while equity options are American options, physically delivered and have a non-continuous dividend stream[9]. However, the main difference is the underlying asset of the index/equity options. An equity option has a single stock as underlying asset, where an index option is an option on a portfolio instead of a portfolio of options. Index options are an option on a portfolio; it is influenced differently by correlation and variance than equity options.

Index options are influenced by correlation while single equity options are not. It is known that correlation changes over time and increases when returns are low. With increased correlation, the variance of investor portfolios also increases because diversification benefits drop. A hedge against the increase of correlation is an index option. Driessen, Meanhout and Volkov (2009) show that correlation risk is priced; an asset that has a relatively high pay-off when the correlations are higher than expected, earns on average a negative excess return. This negative risk premium can be seen as an insurance premium against increasing correlation. Therefore, index options are relatively expensive where equity options relatively cheap (see for example Driessen, Meanhout and Volkov, 2009; Schürhoff and Ziegler, 2010). This high price for index options can be seen as an insurance premium against the loss of diversification benefits, whereas equity options do not show this premium.

Bakshi, Kapadia and Madan (2003) show that equity options are slightly skewed to the left or positively skewed, as in this research (see table 1), while index options are negatively skewed. This means that for equity options, the risk-neutral density function is less negative than the market skewness. Schűrhoff and Ziegler (2010) find that the market prices of the common idiosyncratic variance risk factors are positive. Portfolio of equity options are mainly driven by average idiosyncratic variance, next to industry and market variance, whereas index options are mainly driven by aggregate volatility (Campbell et al., 2001) and not by average idiosyncratic volatility.

If the calendar seasonals are driven by systematical movement in average idiosyncratic volatility, they will be visible in the equity options, but not in the index options. The result of the regression on the S&P500 index options can be found in appendix 10A and 10B. The results of the excess returns show that the Monday dummy is significant for the combined call & put portfolio and the put portfolio, both with a lower t-value than for the equity options (see table 2A). For the less volatile delta hedge excess returns, the Monday and TOM dummy are significant at a 1 percent level for all portfolios of index options, but with a lower t-value. The Triple Witch dummy also shows significant results. Concluding, the Monday and TOM effect are not entirely due to idiosyncratic variance of their underlying stock.

[8] S&P 500 index options http://www.cboe.com/Products/EquityOptionSpecs.aspx

[9] Equity options http://www.cboe.com/products/indexopts/spx_spec.aspx

8. Conclusion

Out of the six calendar anomalies tested over a period of 15 years, where data was retrieved from 5 January 1996 to 29 October 2010, two anomalies show a strong significant presence in the option market; the Monday and TOM effect. The finding of the negative Monday effect in the option market is in line with the financial literature. There is no literature to verify the findings of a positive TOM effect with. It seems that the Monday effect is declining in the option market, where the TOM effect keeps its size. The straddle and skewness asset confirm the excess returns found for the Monday and TOM effect and the direction.

The January effect show fickle evidence, while the combined call & put portfolio regression is always significant at a level of at least 5 percent. The separate call and put are insignificant most of the time. However, in the delta hedge per annum regression it shows positive significant robust returns, where the returns in the singular regression are negative.

The January, Monday and TOM effect all have in common that they indicate a start of a time period. The Monday effect starts the work week, the TOM starts the month and January starts the year. It seems that the smaller the time unit, the better the effect is visible in the option market.

The Halloween effect seems to be a proxy of the Monday and TOM effect as seen in the interaction regression (chapter 5.2). The same goes for the Holiday effect that occurs more often in combination with other seasonal effects than independently. In the robustness checks, they both fail to show robust significant results; therefore they are probably not calendar seasonals in the option market.

The Triple Witch effect is insignificant in all the regressions. It may be due to the way this research was done by using end-of-the-day closing bid-ask prices and therefore the effect of expiration was less visible. Nevertheless, as far as this study concerns, it is not a calendar seasonal in the option market.

Many regressions have been done in this study; therefore table 6 is created to give an overview of all tables in this study.

While the research shows evidence that there are calendar seasonals in the option market, it has its shortcomings and further research is needed. Research on options with a higher maturity than 52 days, tests for bid – ask spreads, interest, non-US markets, and Fama and French factors all necessary. It is interesting that the three calendar effects found have in common that they occur in the change of a time period; the week, month and year. Perhaps they have commonalities in psychological, intuitional or macro economical factors. In addition, it is known that dividends have a major impact on option prices, especially super dividends. These dividends follow a seasonal pattern as well and therefore anomalies found can be a proxy for stocks going ex-dividend. More research could be done on these matters.

This study is a start in the field of calendar anomalies in equity option markets. Hopefully more research will be done to explore the field and fill in the gaps.

Table 6: Summary of all regressions done

The + or – sign indicates whether the significant coefficient was positive or negative. +++, ++, + or ---, --, - signs indicate how t statistics are statistically significant at the 1%, 5%, or 10% levels, respectively. For the regressions with buckets within the different portfolios, the three numbers stand for the number of buckets significant at a 5 percent level.

(Excess) Returns

	Halloween			Holiday			Monday			January			January2			TOM			Triple Witch		
	C&P	*Call*	*Put*	*C&P*	*Call*	*Put*	*C&P*	*Call*	*Put*	*C&P*	*Call*	*Put*	*C&P*	*Call*	*Put*	*C&P*	*Call*	*Put*	*C&P*	*Call*	*Put*
Return	---	--		--			---	-	--	+++		+	+++			+++	++				
Excess Return	---	--		--			---	--	--	+++		+	+++			+++	++				
Multi Variate	---	-		---			---	-		+++		-	+++		-	+++	--	--			
Interaction						--	---	--			N.A.			N.A.		+++					
VW						+				+		+									
S&P 100	--	-					---		-	+++			+++								
Delta (5)	-2	0	0	-3	0	-2	-4	-4	-3	+5	-2	+1	+5	0	+3	+5	+3	0	-4	-1/+1	0
Maturity (3)	-2	-1	0	-1	0	-1	-3	-1	-1	+2	0	0	+2	0	0	+2	+1	0	-1	0	0
Per Annum (15)	-2/+1	0	+1	-2/+1	-1	-2	-9	-2	-4	+8	0	+1	-1/+3	-1/+3	-3/+4	+8	+2	-2	+2	-1/+1	-1/+1
SP500 Index Opties							---		--										++		

Delta Hedge (Excess) Returns

Return				+			---	---	---	--	-	-	--		---	+++	+++	+++			
Excess Return				+			---	---	---	--	-	-	--		---	+++	+++	+++			
Multi Variate					-	---	---	---	---	--			--			+++	---	---			
Interaction	++				--		---	---	---		N.A.			N.A.		+++	+++	+++			
VW	++	+	++				--		--	---	--	--	-		---	+++	++	+++			
S&P 100				+		++	---	---	---	--	--		--	-	--	+++	+++	+++			
Delta (5)	0	0	0	+2	0	+2	-5	-4	-5	-5	+1	-1	-4	0	+4	+4	+5	+4	-1/+1	-1	-1/+1
Maturity (3)	0	+2	0	-1/+1	-1	0	-3	-2	-3	-2/+1	-2/+1	-1	-1	-1	-1	-1/+2	-1/+1	-1/+2	0	0	0
Per Annum (15)	+3	+4	-1/+1	+1	+1	+1	-10	-9	-13	+12	-1/+8	+10	-1/+2	-1/+3	+4	+10	+5	+10	-1	-2/+1	-3/+1
SP500 Index Opties			+				---	---	---						--	+++	+++	+++	++		+++
Straddle		-						---			+++			+++			+++			N.A.	
Skew								---									+++			N.A.	

Reference

Abraham, A., & Ikenberry, D. (1994). The Individual Investor and the Weekend Effect. *Journal of Financial and Quantitative Analysis (June 1994)* , 263-277.

Aggarwal, R., & Rivoli, P. (1989). Seasonal and Day-Of-The-Week Effects in Four Emerging Stock Markets. *Financial Review (November 1989)* , 541-550.

Aggarwal, R., & Rivoli, P. (1989). Seasonal and Day-Of-The-Week Effects in Four Emerging Stock Markets. *Financial Review (November 1989)* , 541-550.

Agrawal, A., & Tandon, K. (1994). Anomalies or illusions? Evidence from stock markets in eighteen countries. *Journal of International Money and Finance 13* , 83-106.

Alkebäck, P., & Hagelin, N. (2004). Expiration day effects of index futures and options: Evidence from a market with a long settlement period. *Applied Financial Economics 14 (2004)* , 385-396.

Arellano, M. (2002). On the testing of correlated effects with panel data. *Journal of Econometrics, Volume 59, Issues 1-2* , 87-97.

Ariel, R. (1990). High Stock Returns before Holidays: Existence and Evidence on Possible Causes. *The Journal of Finance Vol. 45, No. 5 (Dec., 1990)* , 1611-1626.

Arsad, Z., & Coutts, J. (1997). Security price anomalies in the London international stock exchange: a 60 year perspective. *Applied Financial Economics 7 (1997) (5)* , 455-464.

Arumugam, S. (1999). Focus on high stock returns before holidays: new evidence from India. *Journal of Financial Management and Analysis 12 (1999) (2)* , 69-84.

Bacha, O., & Vila, A. (1994). Futures markets, regulation and volatility: the case of the Nikkei stock index futures markets. *Pac-Basin Finance J 2* , 201-225.

Bakshi, G., Kapadia, N., & Madan, D. (2003). Stock return characteristics, skew laws, and the differential pricing of individual equity options. *Review of Financial Studies (2003) 16 (1)* , 101-143.

Bali, T., & A. Hovakimian. (2009). Volatility Spreads and Expected Stock Returns. *Management Science, Forthcoming. Available at SSRN: http://ssrn.com/abstract=1443848* .

Balo, T., & Murray, S. (n.d.). Implied Risk-Neutral Skewness and the Cross-Section of Option Return Returns. *Working Paper, 2010, Baruch College* .

Baltagi, B., Bresson, G., & Pirotte, A. (2003). Fixed effects, random effects or Hausman–Taylor?: A pretest estimator. *Economics Letters, Volume 79, Issue 3* , 361-369.

Baltagi, B., Bresson, G., & Pirotte, A. (2007). Panel unit root tests and spatial dependence. *Journal of Applied Econometrics, Volume 22, Issue 2* , 339-360.

Banz, R. (1981). The relationship between return and market value of common stocks. *Journal of Financial Economics Volume 9, Issue 1, March 1981* , 3-18.

Barone, E. (1990). The Italian Stock Market: Efficiency and Calendar Anomalies. *Working paper. Centre for Research in Finance. IMI Group No 1* .

Battalio, R., & Schultz, P. (2006). Options and the Bubble. *he Journal of Finance, 61* , 2071-2102.

Berg, W. V., & Wessels, R. (1985). Stock market seasonality and taxes: An examination of the tax loss selling hypothesis. *Journal of Business Finance & Accounting 12 (1985)* , 515-530.

Berges, A., McConnell, J., & Schlarbaum, G. (1984). The turn of the year in Canada. *Journal of Finance 39* , 185-192.

Bessembinder, H., & Hertzel, M. (1993). Return Autocorrelation around Nontrading Days. *Review of Financial Studies (Spring 1993)* , 155-189.

Bhardwaj, R. K., & Brooks, L. (1992). The January Anomaly: Effects of Low Share Price, Transaction Costs, and Bid-Ask Bias. *The Journal of Finance Vol. 47, No. 2 (Jun., 1992)* , 553-575.

Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy Vol. 81, No. 3 (May - Jun., 1973)* , 637-654.

Blume, M., & Stambaugh, R. (1983). Bias in computed returns: An application to the size effect. *Journal of Financial Economics 12* , 387-553.

Booth, T., Kallunki, J., & Martikainen, T. (2001). Liquidity and the turn-of-the-month effect: evidence from Finland. *Journal of International Financial Markets. Institutions and Money, 11* , 137-146.

Boudreaux, D. (1995). The monthly effect in international stock markets: evidence and implications. *Journal of Financial and Strategic Decisions 8* , 15-20.

Bouman, S., & Jacobsen, B. (2002). The Halloween Indicator, "Sell in May and Go Away": Another Puzzle. *The American Economic Review Vol. 92, No. 5 (Dec., 2002)* , 1618-1635.

Brockman, P., & Michayluk, D. (1998). Individual Versus Institutional Investors and the Weekend Effect. *Journal of Economics and Finance (Spring 1998)* , 71-85.

Brockman, P., & Michayluk, D. (1998). The persistence holiday effect: additional evidence. *Applied Economics Letters 5* , 205-209.

Brooks, R., & Kim, H. (1997). The Individual Investor and the Weekend Effect: A Reexamination with Intraday Data. *Quarterly Review of Economics and Finance (Fall 1997)* , 725-737.

Brown, P., Keim, D., Kleidon, A., & Marsh, T. (1983). Stock return seasonalities and the tax-loss selling hypothesis: Analysis of the arguments and Australian evidence. *Journal of Financial Economics 12* , 105-128.

Brusa, J., Liu, P., & Schulman, C. (2000). The Weekend Effect, 'Reverse' Weekend Effect, and Firm Size. *Journal of Business Finance and Accounting (June/July 2000)* , 555-574.

C. Jacobsen, & Marquering, W. (2008). Is it the weather? *Journal of Banking and Finance 32 (2008)* , 526-540.

Cadsby, C., & Ratner, M. (1992). Turn-of-month and pre-holiday effects on stock returns: some international evidence. *Journal of Banking and Finance 16 (1992) (3)* , 497-509.

Cameron, A., & Trivedi, P. K. (2005). *Microeconometrics: Methods and.* Microeconometrics: Methods and.

Cameron, A., & Trivedi, P. (2009). *Microeconometrics Using Stata.* Stata Press.

Cao, J., & Han, B. (2009). Cross-section of Stock Option Returns and Stock Volatility Risk Premium. *Working Paper, http://ssrn.com/abstract=1446143* .

Cao, M., & Wei, j. (2005). Stock market returns: a note on temperature anomaly. *Journal of Banking and Finance 29 (2005)* , 1559-1573.

Chen, C., & Williams, J. (1994). Triple-witching hour, the change in expiration timing, and stock market reaction. *Journal of Futures Markets, 14* , 275-292.

Chen, H., & Signal, V. (2003). Role of Speculative Short Sales in Price Formation: Case of the Weekend Effect. *Journal of Finance (April 2003)* , 685-705.

Chen, H., Noronha, G., & Singal, V. (2004). The Price Response to S&P 500 Index Additions and Deletions: Evidence of Asymmetry and a New Explanation. *Journal of Finance 59* , 1901-1929.

Chong, R., Hudson, R., Keasey, K., & Littler, L. (2005). Pre-holiday effects: International evidence on the decline and reversal of a stock market anomaly. *Journal Internation Money Finance 24* , 1226-1236.

Chowdhury, M., Lung, P., & Diltz, J. (2011). Cashing in on Fear: Implied Jump Risk Premium and the Cross-Section of Option Returns. *Available at SSRN: http://ssrn.com/abstract=1868605* .

Chung, Y. (1991). A Transactions Data Test of Stock Index Futures Market Efficiency and Index Arbitrage Profitability. *The Journal of Finance Vol. 46, No. 5 (Dec., 1991)* , 1791-1809.

Condoyanni, L., O'Hanlon, J., & Ward, C. (1987). Day of the Week Effects on Stock Returns: International Evidence. *Journal of Business Finance and Accounting (Summer 1987)* , 159-174.

Connolly, R. (1989). An Examination of the Robustness of the Weekend Effect. *Journal of Financial and Quantitative Analysis (June 1989)* , 133-170.

Constantinides, G., & Ingersoll, J. E. (1984). optimal Bond Trading with personal taxes. *Journal of Financial Economics 13 (September 1984)* , 299-355.

Corredor, P., Lechón, P., & Santamaría, R. (2001). Option-Expiration Effects in Small Markets: The Spanish Stock Exchange. *Journal of Futures Markets, 21* , 905-928.

Coutts, J., Kaplanidis, C., & Roberts, J. (2000). Security price anomalies in an emerging market: the case of the Athens stock exchange. *Applied Financial Economics 10 (2000) (5)* , 561-571.

Coval, J., & Shumway, T. (2001). Expected Option Returns. *The Journal of Finance, Volume 56, Issue 3 (June 2001)* , 983-1009.

Cox, J., Ross, S., & Rubinstein, M. (1979). Option pricing: A simplified approach. *Journal of Financial Economics, Volume 7, Issue 3* , 229-263.

Damodaran, A. (1989). The Weekend Effect in Information Releases: A Study of Earnings and Dividend Announcements. *Review of Financial Studies (Winter 1989)* , 607-623.

Day, T., & Lewis, C. (1988). The behavior of the volatility implicit in option prices. *Journal of Financial Economics 22(1988)* , 103-122.

Deldin, P., Levin, J., & Irwin, P. (1986). The effect of mood induction in a risky decision-making task. *Bulletin of the Psychonomic Society 24 (1986)* , 4-6.

Diefenbach, R. E. (1972). How Good Is Institutional Brokerage Research? *Financial Analysts Journal Vol. 28, No. 1 (Jan. - Feb., 1972)* , 54-60.

Dimson, E., & Marsh, P. (1986). Event study methodologies and the size effect: The case of UK press recommendations. *Journal of Financial Economics 17* , 113-142.

Doeswijk, R. (2008). The optimism cycle: sell in May. *De Economist 156 (2) (2008)* , 175-200.

Driessen, J., Maenhout, P., & Volkov, G. (2005). Option-implied correlations and the price of correlation risk. *Working Paper* .

Duarte, J., & Jones, C. (2008). The price of market volatility risk. *The price of market volatility risk* .

DuBois, M., & Louvet, P. (1996). The Day-of-the-Week Effect: The International Evidence. *Journal of Banking and Finance (November 1996)* , 1463-1484.

Dyl, A. (1977). Capital Gains Taxation and Year-End Stock Market Behavior. *The Journal of Finance Vol. 32, No. 1 (Mar., 1977)* , 165-175.

Dyl, E., & Martin, S. (1985). Weekend Effects on Stock Returns: A Comment. *Journal of Finance (March 1985)* , 347-349.

Eakins, S., & Sewell, S. (1993). Tax Loss Selling, Institutional Investors, and the January Effect: a Note. *Journal of Financial Reseach 16* , 377-384.

Engström, M., & Nordén, L. (2000). The early exercise premium in American put option prices. *Journal of Multinational Financial Management, Volume 10, Issues 3-4* , 461-479.

Fabozzi, F., Ma, C., & Briley, J. (1994). Holiday trading in futures markets. *Journal of Finance 44 (1994) (1)* , 307-324.

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance Vol. 25, No. 2, Papers and Proceedings of the Twenty-Eighth Annual Meeting of the American Finance Association New York, N.Y. December, 28-30, 1969 (May, 1970)* , 383-417 .

Fama, E. (1965). The Behavior of Stock Market Prices. *Journal of Business 38* , 34-105.

Fama, E., & French, K. (1993). Common Risk Factors in the Returns on Bonds and Stocks. *Journal of Financial Economics 33 (1993)* , 3-53.

Felixson, K. (2002). The expiration day effect of index options and index futures on the underlying shares. *[working paper] Swedish School of Economics and Business Administration (473)* .

Fields, M. (1931). Stock Prices: A Problem in Verification. *Journal of Business (Oct., 1931)* , 415-418.

Fishe, R., Gosnell, T., & Lasser, D. (1993). Good News, Bad News, Volume and the Monday Effect. *Journal of Business Finance and Accounting (November 1993)* , 881-892.

Fortin, R. (1990). Transaction costs and day-of-the-week effects in the OTC/NASDAQ equity market. *Journal of Financial Res. 8* , 243-248.

Fosback, N. (1976). Stock Market Logic. *Fort Lauderdale* , 157.

French, K. (1980). Stock Returns and the Weekend Effect. *Journal of Financial Economics (March1980)* , 55-69.

French, K., & Roll, R. (1986). Stock return variances : The arrival of information and the reaction of traders. *Journal of Financial Economics 17* , 5-26.

Garrett, I., Kamstra, M., & Kramer, L. (2005). Winter blues and time variation in the price of risk. *Journal of Empirical Finance 12 (2005)* , 291-316.

Gibbons, M., & Hess, P. (1981). Day of the Week Effects and Asset Returns. *Journal of Business (October 1981)* , 579-596.

Goetzmann, W., & Zhu, N. (2005). Rain or shine: Where is the weather effect? *European Financial Management 11 (5) (2005)* , 559-578.

Gondhalekar, V., & Mehdian, S. (2003). The Blue-Monday Hypothesis: Evidence Based on Nasdaq Stocks, 1971-2000. *Quarterly Journal of Business and Economics, 42, nos. 3 and 4 (2003)* , 73-92.

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

Grimbacher, S., Swinkels, L., & Vliet, P. V. (2010). An anatomy of calendar effects. *Available at SSRN: http://ssrn.com/abstract=1593770* .

Groth, J., Lewellen, W., Schlarbaum, G., & Lease, R. (1979). An Analysis of Brokerage House Securities Recommendations. *Financial Analysts Journal Vol. 35, No. 1 (Jan. - Feb., 1979)* , 32-40.

Gu, A. (2003). The Reversing Weekend Effect: Evidence from the U.S. Equity Markets. *Review of Quantitative Finance and Accounting* , 5-14.

Gultekin, M., & Gultekin, N. (1983). Stock market seasonality: International evidence. *Journal of Financial Economics 12* , 469-482.

Haggard, K., & Witte, H. (2010). The Halloween effect: Trick or treat? *International Review of Financial Analysis 19* , 37-387.

Haugen, R., & Jorion, P. (1996). The January effect: still here after all these years. *Financial Analysts Journal (1996)* , 27-31.

Haugen, R., & Lakonishok, J. (1987). *Only in January. An Investo'rs Guide to the Unsolved Mystery of the Stock Market. The Incredible January Effect.* Homewood, IL: Dow Jones-Irwin, 1987, forthcoming.

Hausmann, J., & Taylor, W. (1978). Specification tests in econometrics. *Econometrica Vol. 46, No. 6 (Nov., 1978)* , 1251-1271.

Hayashi, F. (2000). *Econometrics.* Econometrics.

Herbst, A., & Maberly, E. (1990). Stock Index Futures, Expiration Day Volatility, and the 'Special' Friday Opening: A Note. *The Journal of Futures Markets, 10* , 323-325.

Hillion, P., & Suominen, M. (2004). The manipulation of closing prices. *Journal of Financial Markets 7 (2004)* , 351-375.

Hindmarch, S., Jentsch, D., & Drew, D. (1984). A Note on Canadian Stock Returns and the Weekend Effect. *Journal of Business Administration (vol. 14, 1984)* , 163-172.

Ho, Y. (1990). Stock return seasonalities in Asia Pacific markets. *Journal of International Financial Management and Accounting 2 (1990)* , 47-77.

Hsiao, C. (2007). Panel data analysis—advantages and challenges . *TEST 16* , 1-22.

Hull, J., & Suo, W. (2002). A Methodology for Assessing Model Risk and its Application to the Implied Volatility Function Model. *Journal of Financial and Quantitative Analysis (2002), 37* , 397-318.

Illueca, M., & LaFuente, J. (2006). New evidence on expiration-day effects using realized volatility: An intraday analysis for the Spanish stock exchange. *Journal of Futures Markets, 26* , 923-938.

J.Nikkinen, Sahlström, P., & Äijö, J. (2007). Turn-of-the-month and intramonth effects: Explanation from the important macroeconomic news announcements. *Journal of Futures Markets 27(2)* , 105-126.

Jacobsen, B., & Visaltanachoti, N. (2009). The Halloween effect in US sectors. *Financial Review 44 (2009)* , 437-459.

Jacobsen, B., Mamun, A., & Visaltanachoti, N. (2005). Seasonal, size and value anomalies. *SSRN Working Paper* .

Jaffe, J., & Westerfield, R. (1985). The Weekend Effect in Common Stock Returns: The International Evidence. *Journal of Finance (June 1985)* , 433-454.

Jaffe, J., Keim, D., & Westerfield, R. (1989). Earnings Yields, Market Values, and Stock Returns. *The Journal of Finance Vol. 44, No. 1 (Mar., 1989)* , 135-148.

Jones, C. S., & Shemesh, J. (2011). The Weekend Effect in Equity Option Returns. *Atlanta Meetings Paper; Marshall School of Business Working Paper No. FBE 03-10* , Available at SSRN: http://ssrn.com/abstract=1364721.

Jones, S. L., Lee, W., & Apenbrink, R. (1991). New Evidence on The January Effect Before Personal Income Taxes. *The Journal of Finance Vol. 46, No. 5 (Dec., 1991)* , 1909-1924.

Kamara, A. (1997). New Evidence on the Monday Seasonal in Stock Returns. *Journal of Business (January 1997)* , 63-84.

Kamstra, M., Kramer, L., & Levi, M. (2003). Winter Blues: A SAD Stock Market Cycle. *The American Economic Review Vol. 93, No. 1 (Mar., 20030* , 324-343.

Kato, K., & Schallheim, J. S. (1985). Seasonal and Size Anomalies in the Japanese Stock Market. *Journal of Financial and Quantitative Analysis, 20* , 243-260.

Keim, D. (1982). Further Evidence on Size Effects and Yield Effects: The Implications of Stock Return Seasonality. *Working Paper, Graduate School of Business, University of Chicago , April 1982* .

Keim, D. (1989). Trading patterns, bid-ask spreads, and estimated security returns: The case of common stocks at calendar turning points. *Journal of Financial Economics 25* , 75-98.

Keim, D., & Madhavan, A. (1996). Execution costs and investment performance: An empirical analysis of institutional equity trades. *Unpublished working paper. Rodney L. White Center, The Wharton School, Philadelphia.*

Keim, D., & Stambaugh, R. (1984). A Further Investigation of the Weekend Effect in Stock Returns. *Journal of Finance (July 1984)* , 819-835.

Keim, D., & Ziemba, T. (2000). *Security Market Imperfections In Worldwide Equity Markets.* Cambridge University Press.

Kézdi, G. (2004). Robust Standard Error Estimation in Fixed-Effects Panel Models. *Hungarian Statistical Review special number 9 (2004)* , 95-116.

Kim, B., & Park, J. (1994). Holiday effects and stock returns: further evidence. *Journal of Financial and Quantitative Analysis 29 (1994) (1)* , 145-157.

Kumar, P., & Seppi, D. (1992). Futures Manipulation with "Cash Settlement". *The Journal of Finance Vol. 47, No. 4 (Sep., 1992)* , 1485-1502.

Kunkel, R., Compton, W., & Beyer, S. (2003). The turn-of-the-month effect still lives: The international evidence. *International Review of Financial Analysis 12 (2003)* , 207-221.

Lakonishok, J., & Levi, M. (1982). Weekend Effects on Stock Returns: A Note. *Journal of Finance (June 1982)* , 883-889.

Lakonishok, J., & Maberly, E. (1990). The Weekend Effect: Trading Patterns of Individual and Institutional Investors. *Journal of Finance (March 1990)* , 231-243.

Lakonishok, J., & Smidt, S. (1988). Are Seasonal Anomalies Real? A Ninety Year Perspective. *Review of Financial Studies (Winter 1988)* , 403-425.

Lakonishok, J., & Smidt, S. (1986). Volume for Winners and Losers: Taxation and Other Motives for Stock Trading. *The Journal of Finance Vol. 41, No. 4 (Sep., 1986)* , 951-974.

Lakonishok, J., Shleifer, A., & R.W.Vishny. (1989). The Impact of Institutional Trading on Stock Prices. *Journal of Financial Vol. 44, No. 5 (Dec., 1989)* , 1219-1245.

Lamoureux, C., & Sanger, G. (1989). Firm Size and Turn-of-the-Year Effects in the OTC/NASDAQ Market. *The Journal of Finance Vol. 44, No. 5 (Dec., 1989)* , 1219-1245.

Liano, K., Marchand, P., & Huang, G. (1992). The holiday effect in stock returns: evidence from the OTC market. *Review of Financial Economics 2 (1992) (1)* , 45-54.

Lo, A. (2007). Efficient Markets Hypothesis. *A DICTIONARY OF ECONOMICS, L. Blume, S. Durlauf, eds., 2nd Edition, Palgrave Macmillan Ltd., 2007* , Available at SSRN: http://ssrn.com/abstract=991509.

Lo, A. W., & MacKinlay, A. C. (1988). Stock market prices do not follow random walks: evidence from a simple specification test. *Rev. Financ. Stud. (1988) 1* , 41-66.

Lucey, M., & Zhao, S. (2008). Halloween or January? Yet another puzzle. *International Review of Financial Analysis 17 (2008)* , 1055-1069.

Maberly, E., & Pierce, R. (2004). Stock market efficiency withstands another challenge: Solving the “sell in May/buy after Halloween" puzzle. *Econ Jounal Watch1(1)* , 29-46.

Maloney, K. J., & Rogalski, R. J. (1989). Call-Option Pricing and the Turn of the Year. *The Journal of Business Vol. 62, No. 4 (Oct., 1989)* , 539-552.

McConnell, J., & Xu, W. (2008). Equity Returns at the Turn of the Month. *Financial Analysts Journal 64 (2)* , 49-64.

Mehdian, S., & Perry, M. (2002). Anomalies in US equity markets: A re-examination of the January effect. *Applied Financial Economics 12 (2002)* , 141-145.

Mehdian, S., & Perry, M. (2001). The Reversal of the Monday Effect: New Evidence from US Equity Markets. *Journal of Business Finance and Accounting (September/October 2001)* , 1043-1065.

Merill, A. (1966). *Behavior of Prices on Wall Street.* The Analysis Press, Chappaqua, NY (1966).

Merton, R. (1973). Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science Vol. 4, No. 1 (Spring, 1973)* , 141-183.

Miller, E. (1988). Why a Weekend Effect? *Journal of Portfolio Management (Summer 1988)* , 43-48.

Mills, T., & Coutts, J. (1995). Calendar effects in the London Stock Exchange FTSE indices. *The European Journal of Finance 1 (1995) (1)* , 73-93.

Moller, N., & Zilca, S. (2008). The evolution of the January effect. *Journal of Banking and Finance 32 (2008)* , 447-457.

Moller, N., & Zilca, S. (2008). The evolution of the January effect. *Journal of Banking and Finance 32 (2008)* , 447-457.

Officer, R. (1975). Seasonality in Australian capital markets: Market efficiency and empirical issues. *Journal of Financial Economics 2 (1975)* , 29-51.

Ogden, J. (1990). Turn-of-month evaluations of liquid profits and stock returns: a common explanation for the monthly and January effects. *Journal of Finance 45* , 1259-1272.

Osborne, M. (1962). Periodic Structure in the Brownian Motion of Stock Prices. *Operations Research (May/June 1962)* , 345-379.

Parker, P., & Tavassoli, N. (2000). Homeostasis and consumer behavior across cultures. *International Journal of Research in Marketing 17 (2000)* , 33-53.

Penman, S. (1987). The Distribution of Earnings News Over Time and Seonalties in Aggregate Stock return. *Journal of Financial Economics, 18* , 199-228.

Petersen, M. (2009). Estimating standard errors in finance panel data sets: Comparing approaches. *Review of Financial Studies (2009) 22 (1)* , 435-480.

Pettengil, G., Wingender, J., & Kohli, R. (2003). Arbitrage, institutional investors, and the Monday effect. *Quarterly Journal of Business and Economics 42 (2003)* , 49-64.

Pettengill, G. (1993). An Experimental Study of the 'Blue Monday' Hypothesis. *Journal of SocioEconomics (Fall 1993)* , 241-257.

Pettengill, G. (1989). Holiday Closings and Security Returns. *Journal of Financial Research, 1989, vol. 12, issue 1* , 57-67.

Pettengill, G., & Buster, D. (1994). Variations in Return Signs: Announcements and the Weekday Anomaly. *Quarterly Journal of Business and Economics (Summer 1994)* , 81-91.

Pettengill, G., Wingender, J., & Kohl, R. (2003). Arbitrage, institutional investors, and the Monday effect. *Quarterly Journal of Business and Economics 42 (2003)* , 49-64.

Pope, P., & Yadav, P. (1992). HE IMPACT OF OPTION EXPIRATION ON UNDERLYING STOCKS: THE UK EVIDENCE. *Journal of Business Finance & Accounting, 19* , 329-344.

Reinganum, M. (1983). The anomalous stock market behavior of small firms in January: Empirical tests for tax-loss selling effects. *Journal of Financial Economics 12* , 89-104.

Reinganum, M., & Shapiro, A. (1987). Taxes and stock return seasonality: Evidence from the London stock exchange. *Journal of Business 60 (1987)* , 281-295.

Ritter, J. (1988). The buying and selling behavior of individual investors at the turn of the year. *Journal of Finance 43* , 701-717.

Roll, R. (1983). On computing mean returns and the small firm premium. *Journal of Financial Economics 12 (1983)* , 371-386.

Roll, R., & Ross, S. A. (1980). An Empirical Investigation of the Arbitrage Pricing Theory. *The Journal of Finance Vol. 35, No. 5 (Dec., 1980)* , 1073-1103.

Samuelson, P. A. (1965). Proof That Properly Anticipated Prices Fluctuate Randomly. *Industrial Management Review 6 (Spring, 1965)* , 41-90.

Sar, N. V. (2003). Calendar effects on the Amsterdam stock exchange. *The Economist, 2003, 151(3)* , 271-293.

Schaffer, M., & Stillman, S. (2010). XTOVERID: Stata module to calculate tests of overidentifying restrictions after xtreg, xtivreg, xtivreg2, xthtaylor. *Available at http://ideas.repec.org/c/boc/bocode/s456779.html.*

Schultz, P. (1985). Personal Income Taxes and the January Effect: Small Firm Stock Returns Before the War Revenue Act of 1917: A Note. *The Journal of Finance Vol. 40, No. 1 (Mar., 1985)* , 333-343.

Schürhoff, N., & Ziegler, A. (2011). Variance risk, financial intermediation, and the cross-section of expected option returns. *EPR Discussion Paper No. DP8268. Available at SSRN: http://ssrn.com/abstract=1782562* .

Schwert, G. (2002). *Anomalies and market efficiency.* in George Constantinides: Milton Harris, and Rene Stulz, eds. Handbook of the Economics of Finance (North-Holland, Amster-dam).

Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *The Journal of Finance Vol. 19, No. 3 (Sep., 1964)* , 425-442 .

Sias, R., & Starks, L. (1995). The Day-of-the-Week Anomaly: The Role of Institutional Investors. *Financial Analysts Journal (May/June 1995)* , 58-37.

Stock, J., & Watson, M. (2008). Heteroskedasticity-Robust Standard Errors for Fixed. *Econometrica, 76,* , 155-174.

Stol, H., & Whaley, R. (1986-3). Expiration day effects of index options and futures. *Monograph Series in Finance and Economics* .

Stoll, H., & Whaley, R. (1997). Expiration-day effects of the all ordinaries share price index futures: empirical evidence and alternative settlement procedures. *Australian Journal of Management 22 (1997)* , 139-174.

Stoll, H., & Whaley, R. (1991). Expiration-Day Effects: What Has Changed? *Financial Analysts Journal Vol. 47, No. 1 (Jan. - Feb., 1991)* , 58-72.

Stoll, H., & Whaley, R. (1987). Program trading and expiration day effects. *Financial Analysts Journal 43 (March/April)1* , 6-28.

Stoll, H., & Whaley, R. (1990). Program Trading and Individual Stock Returns: Ingredients of the Triple-Witching Brew. *The Journal of Business Vol. 63, No. 1, Part 2* , 165-192.

Sullivan, J., & Liano, K. (2003). Market Breadth and the Monday Seasonal in Stock Returns. *Quarterly Journal of Business and Economics, 42, nos. 3 and 4 (2003)* , 65-72.

Sullivan, R., Timmerman, A., & White, H. (1999). Data-Snooping, Technical Trading Rule Performance, and the Bootstrap. *The Journal of Finance Vol. 54, No. 5 (Oct., 1999)* , 1647-1691.

T.W. Chamberlain, C. C., & Kwan, C. (1989). Expiration-Day Effects of Index Futures and Options: Some Canadian Evidence. *Financial Analysts Journal Vol. 45, No. 5 (Sep. - Oct., 1989)* , 67-71.

Theissen, E. (2007). An analysis of private investors' stock market return forecasts. *Applied Financial Economics 17 (1) (2007)* , 35-43.

Thompson, S. (2010). Simple formulas for standard errors that cluster by both firm and time. *Journal of Financial Economics, Volume 99, Issue 1* , 1-10.

Vergin, R., & McGinnis, J. (1999). Revisiting the holiday effect: Is it on holiday. *Applied Financial Economics 9* , 477-482.

Wachtel, S. (1942). Certain Observations on Seasonal Movements in Stock Prices. *The Journal of Business of the University of Chicago Vol. 15, No. 2 (Apr., 1942)* , 184-193.

Wang, Y., & Walker, M. (2000). An Empirical Test of Individual and Institutional Trading Patterns in Japan, Hong Kong and Taiwan. *Journal of Economics and Finance (Summer 2000)* , 178-194.

Weinistein, N. (1980). Unrealistic optimism about future life events. *Journal of Personality and Social Psychology, Vol 39(5), Nov 1980* , 806-820.

Wilson, J., & Jones, C. (1993). Comparison of seasonal anomalies across major equity markets: A Note. *Financial Review 28 (February)* , 107-115.

Wong, P., Neoh, S., Lee, K., & Thjong, T. (1990). easonality in the Malaysian stock market. *Asia Pacific Journal of Management 7 (1990)* , 43-63.

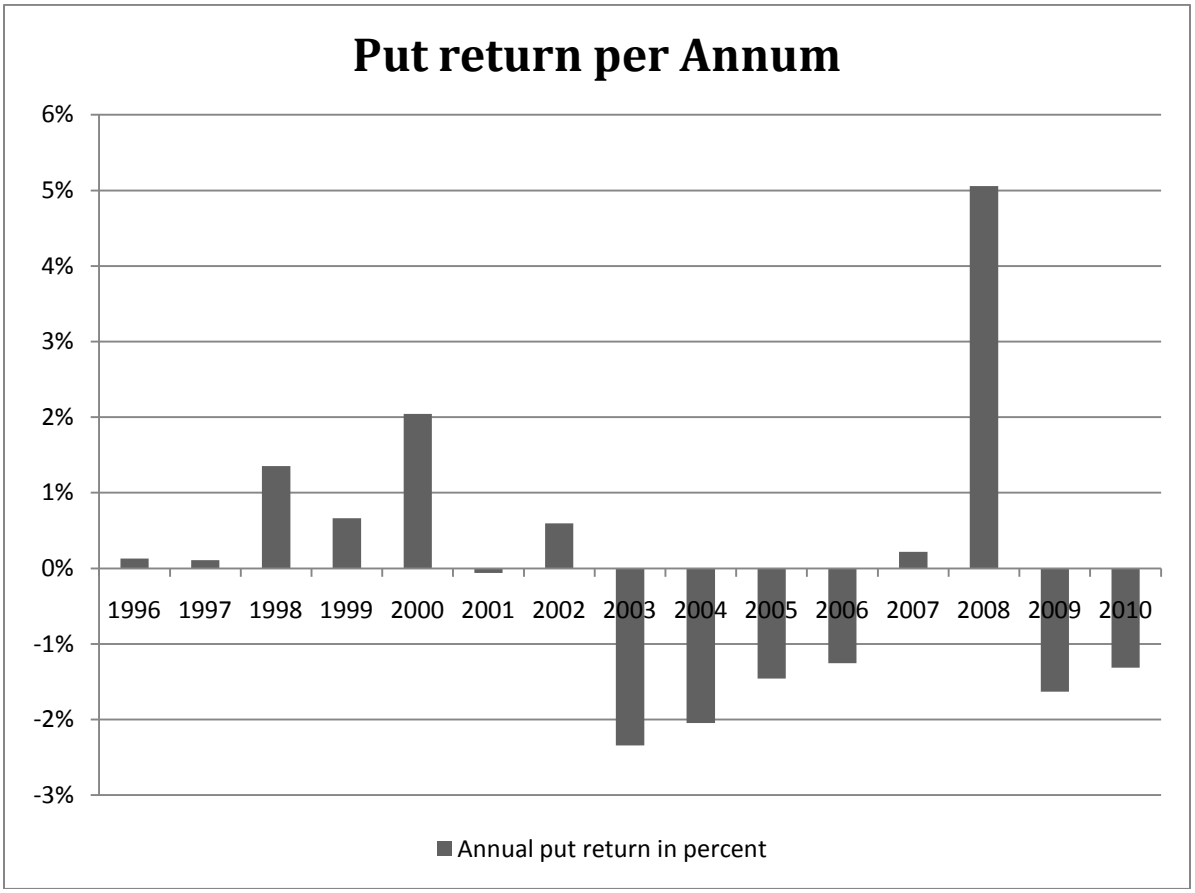
Wooldridge, J. (2002). *Econometric Analysis of Cross Section and Panel Data.* Cambridge, MA: MIT Press.

Ziemba, W. (1991). Japanese security market regularities: Monthly, turn of the month and year, holiday and Golden Week effects. *Japan and the World Economy 3* , 119-146.

Ziemba, W. (1989). *Seasonality effects in Japanese futures markets.* North-Holland, Amsterdam.: Rhee and Chang, Editors, 1989. Research on Pacific Basin Security Markets.

Appendix

Appendix 1: Averaged annual put return per annum in percentage. Data is daily from 5 January 1996 through 29 October 2010



Appendix 2A:Results multivariate regression excess returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of excess returns.The excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

Excess Return

	Call & Put			**Call**			**Put**		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Halloween	-0.39	0.0015	-2.61***	-1.24	0.0065	-1.92*	0.55	0.0073	0.76
Holiday	-0.55	0.0015	-3.75***	0.38	0.0061	0.62	1.07	0.0065	1.66*
Monday	-0.88	0.0009	-10.31***	0.74	0.0037	1.99**	0.09	0.0041	0.22
January	0.70	0.0015	4.70***	0.36	0.0088	0.41	-1.53	0.0091	-1.67*
TOM	0.42	0.0008	5.24***	-0.82	0.0041	-2.03**	-0.94	0.0046	-2.07**
Triple Witch	-0.01	0.0015	-0.07	0.11	0.0087	0.13	-0.14	0.0094	-0.15
Constant	1.24	0.0008	16.22	1.73	0.0034	5.10	0.71	0.0038	1.88

Appendix 2B: Results multivariate regression excess returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of excess returns.The excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

Excess Return

	Call & Put			**Call**			**Put**		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Halloween	-0.39	0.0015	-2.60***	-1.24	0.0065	-1.92*	0.56	0.0073	0.76
Holiday	-0.49	0.0015	-3.36***	1.56	0.0231	0.67	2.19	0.0185	1.18
Monday	-0.89	0.0009	-10.40***	0.65	0.0036	1.80*	-0.02	0.0042	-0.05
January2	1.86	0.0070	2.67***	0.41	0.0087	0.47	-1.47	0.0092	-1.60*
TOM	0.33	0.0007	4.41***	-0.83	0.0041	-2.04**	-0.95	0.0046	-2.09**
Triple Witch	-0.06	0.0015	-0.40	0.09	0.0087	0.10	-0.21	0.0094	-0.22
Constant	1.31	0.0008	17.22	1.77	0.0034	5.23	0.80	0.0037	2.14

Appendix 2C: Results multivariate regression delta hedge excess returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of delta hedge excess returns.The delta hedge excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coeficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

Delta Hedge Excess Return

	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Halloween	0.15	0.0012	1.26	0.25	0.0019	1.34	0.0539	0.0013	0.42
Holiday	0.01	0.0011	0.12	-0.49	0.0024	-2.00**	-0.2330	0.0012	-1.99**
Monday	-0.64	0.0006	-10.33***	0.30	0.0009	3.35***	0.3725	0.0005	7.40***
January	-0.36	0.0015	-2.47**	0.05	0.0020	0.25	-0.0250	0.0013	-0.20
TOM	0.34	0.0005	6.29***	-0.45	0.0010	-4.58***	-0.8245	0.0006	-13.12***
Triple Witch	-0.04	0.0011	-0.34	0.02	0.0018	0.11	-0.0980	0.0013	-0.77
Constant	-2.36	0.0006	-39.99	-2.39	0.0009	-25.92	-2.3367	0.0006	-37.1

Appendix 2D: Results multivariate regression delta hedge excess returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of delta hedge excess returns.The delta hedge excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coeficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

Delta Hedge Excess Return

	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Halloween	0.15	0.0012	1.25	0.25	0.0019	1.34	0.05	0.0013	0.42
Holiday	-0.03	0.0011	-0.30	-1.81	0.0100	-1.81*	-1.03	0.0028	-3.68***
Monday	-0.63	0.0006	-10.26***	0.39	0.0008	5.03***	0.42	0.0005	8.46***
January2	-1.42	0.0057	-2.49**	0.00	0.0020	-0.02	-0.06	0.0013	-0.47
TOM	0.41	0.0005	8.72***	-0.44	0.0010	-4.52***	-0.82	0.0006	-13.10***
Triple Witch	-0.02	0.0011	-0.18	0.04	0.0018	0.25	-0.09	0.0013	-0.68
Constant	-2.39	0.0006	-41.33	-2.43	0.0009	-26.86	-2.36	0.0006	-38.22

Appendix 3A: Results regression returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of returns.The returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

	Return								
	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	1.26	0.0008	16.68	1.92	0.0032	6.08	0.55	0.0036	1.56
Halloween	-0.40	0.0015	-2.63***	-1.39	0.0064	-2.16**	0.69	0.0072	0.95
Constant	1.11	0.0003	33.91	1.31	0.0016	8.39	0.89	0.0018	5.04
Holiday	-0.35	0.0014	-2.43**	0.67	0.0086	0.78	-1.44	0.0091	-1.59
Constant	1.25	0.0003	36.80	1.47	0.0017	8.67	1.02	0.0019	5.38
Monday	-0.87	0.0009	-9.98***	-0.76	0.0041	-1.87*	-0.99	0.0046	-2.14**
Constant	1.03	0.0003	33.18	1.30	0.0016	8.29	0.74	0.0018	4.16
January	0.74	0.0015	4.83***	0.38	0.0061	0.63	1.13	0.0064	1.76*
Constant	1.06	0.0003	35.38	1.30	0.0015	8.60	0.81	0.0017	4.70
January2	2.12	0.0069	3.08***	2.00	0.0228	0.88	2.25	0.0183	1.23
Constant	0.98	0.0003	29.74	1.13	0.0017	6.52	0.83	0.0019	4.29
TOM	0.41	0.0008	5.08***	0.74	0.0036	2.04**	0.05	0.0041	0.12
Constant	1.09	0.0003	33.92	1.33	0.0016	8.56	0.84	0.0017	4.83
Triple Witch	-0.02	0.0014	-0.16	-0.03	0.0086	-0.04	-0.01	0.0094	-0.01

Appendix 3B:Results regression delta hedge returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of delta hedge returns.The delta hedge returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

	Delta Hedg Return								
	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	-2.42	0.0006	-40.78	-2.57	0.0009	-28.16	-2.26	0.0006	-38.22
Halloween	0.15	0.0012	1.17	0.29	0.0019	1.52	0.00	0.0013	-0.00
Constant	-2.36	0.0002	-94.62	-2.46	0.0004	-57.67	-2.27	0.0002	-90.82
Holiday	0.21	0.0011	1.81*	0.25	0.0021	1.21	0.16	0.0012	1.33
Constant	-2.24	0.0003	-87.91	-2.34	0.0004	-52.33	-2.14	0.0003	-80.14
Monday	-0.65	0.0007	-9.26***	-0.63	0.0011	-5.47***	-0.68	0.0006	-10.65***
Constant	-2.33	0.0002	-95.39	-2.41	0.0004	-58.95	-2.24	0.0003	-84.53
January	-0.34	0.0015	-2.36**	-0.47	0.0025	-1.91*	-0.22	0.0011	-1.94*
Constant	-2.34	0.0002	101.54	-2.43	0.0004	-62.03	-2.25	0.0002	-91.78
January2	-1.12	0.0057	-1.97**	-1.56	0.0100	-1.55	-0.67	0.0025	-2.67***
Constant	-2.45	0.0003	-88.86	-2.54	0.0005	-53.48	-2.36	0.0003	-80.19
TOM	0.34	0.0006	6.23***	0.32	0.0010	3.41***	0.36	0.0005	7.24***
Constant	-2.36	0.0002	-95.21	-2.45	0.0004	-57.90	-2.26	0.0002	-91.21
Triple Witch	0.02	0.0011	0.15	0.11	0.0018	0.59	-0.08	0.0012	-0.60

Appendix 4A: Results value weighted regression excess returns on seasonal dummies

This table reports the results of the fixed pooled value weighted (vw; weighted by value of openintrest) regression on a call & put, call or put portfolio of excess returns. The excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively

	VW Excess Return								
	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	5.11	0.0050	10.17	5.09	0.0083	6.10	5.25	0.0094	5.60
Halloween	-1.46	0.0077	-1.89*	-2.33	0.0128	-1.81*	0.13	0.0159	0.08
Constant	4.34	0.0015	29.00	3.76	0.0022	16.75	5.42	0.0041	13.33
Holiday	-0.96	0.0060	-1.58	-0.07	0.0119	-0.06	-2.81	0.0147	-1.91*
Constant	4.27	0.0015	28.02	3.81	0.0025	15.40	5.13	0.0040	12.74
Monday	0.16	0.0042	0.38	-0.34	0.0052	-0.65	1.05	0.0122	0.86
Constant	4.20	0.0015	27.12	3.72	0.0022	16.71	5.09	0.0043	11.85
January	0.78	0.0042	1.84*	0.24	0.0068	0.35	2.15	0.0125	1.73*
Constant	4.27	0.0015	29.14	3.72	0.0022	17.02	5.30	0.0040	13.30
January2	1.02	0.0112	0.92	1.03	0.0199	0.52	1.00	0.0307	0.33
Constant	4.14	0.0016	25.46	3.59	0.0026	13.89	5.15	0.0042	12.16
TOM	0.62	0.0033	1.86*	0.62	0.0048	1.29	0.61	0.0095	0.64
Constant	4.31	0.0015	28.93	3.74	0.0022	16.71	5.35	0.0040	13.34
Triple Witch	-0.14	0.0059	-0.23	0.66	0.0136	0.49	-1.78	0.0203	-0.87

Appendix 4AB: Results value weighted regression delta hedge excess returns on seasonal dummies This table reports the results of the fixed pooled value weighted (vw; weighted by value of openintrest) regression on a call & put, call or put portfolio of delta hedge excess returns.The delta hedge excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

VW Delta Hedge Excess Return

	Call & Put			**Call**			**Put**		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	-3.00	0.0039	-7.62	-3.08	0.0061	-5.03	-2.87	0.0015	-19.26
Halloween	1.24	0.0062	2.01**	1.58	0.0094	1.68*	0.65	0.0026	2.54**
Constant	-2.34	0.0008	-28.51	-2.22	0.0012	-17.90	-2.54	0.0005	-48.34
Holiday	0.00	0.0015	0.00	-0.07	0.0019	-0.35	0.13	0.0021	0.62
Constant	-2.30	0.0009	-25.02	-2.23	0.0014	-15.97	-2.42	0.0005	-44.70
Monday	-0.24	0.0011	-2.08**	0.00	0.0016	0.00	-0.64	0.0015	-4.16***
Constant	-2.30	0.0008	-28.30	-2.18	0.0012	-17.76	-2.51	0.0005	-46.62
January	-0.41	0.0014	-2.98***	-0.44	0.0018	-2.39**	-0.34	0.0014	-2.38**
Constant	-2.32	0.0008	-28.81	-2.20	0.0012	-18.04	-2.53	0.0005	-48.31
January2	-0.75	0.0044	-1.69*	-0.80	0.0057	-1.4	-0.61	0.0019	-3.16***
Constant	-2.44	0.0010	-23.52	-2.34	0.0016	-14.80	-2.61	0.0006	-42.12
TOM	0.35	0.0012	2.99***	0.40	0.0017	2.32**	0.26	0.0009	2.89***
Constant	-2.34	0.0008	-28.74	-2.22	0.0012	-18.05	-2.54	0.0005	-48.15
Triple Witch	-0.02	0.0020	-0.09	-0.09	0.0025	-0.37	0.14	0.0021	0.64

Appendix 5A: Results regression excess returns on the S&P100 seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of excess returns.The returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

Exccess Retuns S&P100

	Call & Put			**Call**			**Put**		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	1.20	0.0012	9.89	1.74	0.0039	4.48	0.54	0.0051	1.05
Halloween	-0.58	0.0024	-2.39**	-0.15	0.0080	-1.93*	0.63	0.0106	0.60
Constant	0.96	0.0005	21.04	1.05	0.0018	5.93	0.85	0.0024	3.61
Holiday	-0.22	0.0020	-1.08	0.67	0.0101	0.67	-1.31	0.0123	-1.06
Constant	1.12	0.0005	23.27	1.21	0.0019	6.25	1.01	0.0025	3.98
Monday	-0.94	0.0012	-7.84***	-0.78	0.0045	-1.72	-1.14	0.0062	-1.86*
Constant	0.87	0.0004	19.64	1.01	0.0018	5.66	0.70	0.0024	2.90
January	0.92	0.0018	5.14***	0.66	0.0066	1.00	1.25	0.0082	1.53
Constant	0.92	0.0004	21.25	1.02	0.0017	5.92	0.79	0.0023	3.42
January2	2.35	0.0073	3.22***	3.25	0.0227	1.43	1.16	0.0222	0.52
Constant	0.81	0.0005	17.01	0.87	0.0020	4.35	0.75	0.0026	2.88
TOM	0.50	0.0011	4.62***	0.74	0.0041	1.83*	0.21	0.0054	0.38
Constant	0.95	0.0005	21.13	1.07	0.0018	6.06	0.81	0.0023	3.49
Triple Witch	-0.06	0.0021	-0.30	0.18	0.0116	0.15	-0.37	0.0142	-0.26

Appendix 5A: Results regression excess returns on the S&P100 seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of excess returns.The returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

Delta Hedge Excess Return S&P 100

	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	-2.39	0.0012	-24.31	-2.72	0.0019	-14.44	-3.03	0.0009	-32.46
Halloween	0.23	0.0025	0.93	0.31	0.0039	0.77	0.15	0.0020	0.74
Constant	-2.78	0.0004	-75.12	-2.60	0.0006	-46.11	-2.98	0.0003	-87.22
Holiday	0.27	0.0017	1.62*	0.20	0.0024	0.81	0.35	0.0016	2.19**
Constant	-2.65	0.0004	-66.20	-2.52	0.0006	-40.29	-2.79	0.0004	-79.33
Monday	-0.67	0.0009	-7.46***	-0.39	0.0013	-3.10***	-0.99	0.0009	-10.52***
Constant	-2.74	0.0004	-74.84	-2.55	0.0005	-46.30	-2.95	0.0004	-83.65
January	-0.37	0.0016	-2.24**	-0.50	0.0025	-2.03**	-0.22	0.0014	-1.55
Constant	-2.75	0.0004	-78.14	-2.57	0.0005	-47.91	-2.96	0.0003	-88.77
January2	-1.18	0.0057	-2.08**	-1.50	0.0087	-1.72*	-0.80	0.0035	-2.32**
Constant	-2.91	0.0004	-68.27	-2.73	0.0007	-40.57	-3.12	0.0004	-81.52
TOM	0.51	0.0008	6.63***	0.50	0.0011	4.42***	0.53	0.0007	7.15***
Constant	-2.76	0.0004	-75.71	-2.59	0.0006	-46.46	-2.97	0.0003	-87.83
Triple Witch	-0.17	0.0018	-0.96	-0.24	0.0028	-0.87	-0.09	0.0016	-0.58

Appendix 6A: Results regression excess returns per delta bucket for excess return portfolios seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of excess returns.The portfolio is divided on delta basis.The excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1, 2, 3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

	Excess Returns per Delta Bucket														
	Call & Put					Call					Put				
	0 <> .2 -.2<> 0	.2 <> .4 -.4<>-.2	.4 <> .6 -.6<>-.4	.6 <>.8 -.8<>-.6	.8 <> 1 -1<>-.8	0 <> .2 -.2<> 0	.2 <> .4 -.4<>-.2	.4 <> .6 -.6<>-.4	.6 <>.8 -.8<>-.6	.8 <> 1 -1<>-.8	-.2<> 0 0 <> .2	-.4<>-.2 .2 <> .4	-.6<>-.4 .4 <> .6	-.8<>-.6 .6 <>.8	-1<>-.8 .8 <> 1
Halloween	2.02	-0.24	-0.83	-0.63	-0.43	1.90	0.89	0.55	0.02	0.06	4.45	1.99	1.15	0.63	0.05
	(1.41)	(-0.37)	(-2.22[1])	(-1.83[3])	(-3.15[1])	(1.39)	(0.98)	(1.05)	(0.06)	(1.00)	(1.83[3])	(1.02)	(0.75)	(0.55)	(0.10)
Holiday	0.53	-0.20	-0.62	-0.55	-0.36	1.74	0.93	0.42	0.26	0.08	0.17	-1.59	-2.23	-2.23	-1.31
	(0.39)	(-0.35)	(-2.64[1])	(-2.45[2])	(-2.39[2])	(1.15)	(1.39)	(0.96)	(0.94)	(0.92)	(0.07)	(-0.94)	(-1.67[3])	(-1.99[2])	(-2.17[2])
Monday	-5.78	-3.94	-2.22	-1.11	-0.04	-1.13	-1.81	-1.53	-0.77	-0.08	-5.00	-3.17	-1.99	-1.02	-0.20
	(-11.51[1])	(-17.87[1])	(-16.44[1])	(-8.11[1])	(-0.47)	(-1.93[3])	(-5.83[1])	(-7.31[1])	(-5.90[1])	(-2.16[2])	(-5.25[1])	(-4.32[1])	(-3.12[1])	(-1.86[3])	(-0.67)
January	5.89	3.91	2.22	1.33	0.51	-10.23	-4.68	-1.33	-0.26	-0.05	4.35	2.55	2.31	1.97	0.94
	(3.24[1])	(3.78[1])	(4.98[1])	(4.78[1])	(5.07[1])	(-1.85[3])	(-2.39[2])	(-2.25[2])	(-1.04)	(-1.12)	(1.77[3])	(1.51)	(1.78[3])	(1.93[3])	(2.21[2])
January2	13.38	9.53	5.16	2.81	0.83	-19.00	-8.94	-2.75	-0.64	-0.21	9.69	7.73	5.78	3.80	1.16
	(3.72[1])	(3.91[1])	(3.70[1])	(3.36[1])	(2.89[1])	(-1.68[3])	(-1.91[3])	(-1.68[3])	(-1.04)	(-1.84[3])	(1.97[2])	(2.27[2])	(2.10[2])	(1.73[3])	(1.08)
TOM	2.23	1.27	1.14	1.00	0.36	0.00	0.87	0.83	0.64	0.01	1.50	0.75	0.66	0.54	0.12
	(3.74[1])	(4.84[1])	(7.71[1])	(8.09[1])	(5.58[1])	(0.00)	(2.25[2])	(4.37[1])	(6.20[1])	(0.27)	(1.54)	(1.02)	(1.05)	(1.03)	(0.46)
Triple Witch	-2.02	-1.20	-1.56	-1.92	-0.76	0.14	0.32	-0.33	-0.92	0.24	-2.11	-0.34	-0.63	-1.24	-0.58
	(-1.94[3])	(-2.89[1])	(-6.20[1])	(-8.20[1])	(-4.61[1])	(0.10)	(0.38)	(-0.82)	(-3.84[1])	(2.26[2])	(-1.03)	(-0.23)	(-0.52)	(-1.05)	(-0.78)
Frequencies	2.81%	8.23%	10.86%	14.44%	63.46%	1.89%	7.09%	10.56%	14.69%	65.78%	3.72%	9.38%	11.16%	14.20%	61.54%

Appendix 6B: Results regression excess returns per delta bucket for the delta hedged portfolios seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of delta hedge excess returns.The portfolio is divided on delta basis..The delta hedge excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2], [3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

Delta Excess Returns per Delta Bucket

	Call & Put					**Call**					**Put**				
	0 <> .2 -.2<> 0	.2 <> .4 -.4<>-.2	.4 <> .6 -.6<>-.4	.6 <>.8 -.8<>-.6	.8 <> 1 -1<>-.8	0 <> .2	.2 <> .4	.4 <> .6	.6 <>.8	.8 <> 1	-.2<> 0	-.4<>-.2	-.6<>-.4	-.8<>-.6	-1<>-.8
Halloween	-0.14	-0.09	0.23	0.10	0.05	-3.41	-3.24	-2.98	-1.96	-0.90	-1.05	-0.82	-0.07	0.18	0.04
	(-0.15)	(-0.16)	(0.65)	(0.50)	(1.55)	(-1.49)	(-1.66[3])	(-1.89[3])	(-1.54)	(-1.86[3])	(-0.96)	(-1.59)	(-0.21)	(0.94)	(0.76)
Holiday	0.76	0.65	0.54	0.37	0.05	1.32	1.78	1.11	1.03	0.53	0.32	0.45	0.66	0.49	0.02
	(1.18)	(1.82[3])	(2.25[2])	(2.41[2])	(1.72[3])	(0.51)	(1.14)	(0.79)	(0.86)	(0.82)	(0.59)	(1.49)	(3.48[1])	(3.68[1])	(0.21)
Monday	-2.30	-2.20	-1.66	-0.87	-0.17	-7.46	-4.98	-2.46	-1.19	0.12	-2.85	-2.49	-1.78	-0.97	-0.27
	(-7.61[1])	(-12.10[1])	(-13.04[1])	(-11.20[1])	(-9.62[1])	(-8.09[1])	(-7.37[1])	(-3.87[1])	(-2.09[2])	(0.38)	(-10.10[1])	(-16.42[1])	(-17.08[1])	(-14.12[1])	(-7.75[1])
January	-4.91	-2.44	-0.91	-0.38	-0.05	8.65	5.73	2.12	0.71	0.09	-1.95	-0.77	-0.52	-0.51	-0.05
	(-2.19[2])	(-2.55[2])	(-2.46[2])	(-2.27[2])	(-2.78[1])	(1.63)	(2.04[2])	(1.35)	(0.62)	(0.21)	(-1.94[3])	(-1.66[3])	(-1.71[3])	(-2.92[1])	(-1.34)
January2	-9.13	-5.22	-2.11	-0.65	-0.20	19.79	11.84	4.52	1.89	0.53	-3.46	-2.33	-1.50	-0.66	-0.19
	(-2.05[2])	(-2.36[2])	(-2.34[2])	(-1.76[3])	(-2.86[1])	(1.54)	(1.58)	(1.01)	(0.64)	(0.43)	(-1.90[3])	(-2.91[1])	(-3.30[1])	(-2.94[1])	(-2.41[2])
TOM	1.01	1.02	0.83	0.63	0.01	3.90	2.01	1.64	1.44	0.58	1.45	1.13	0.83	0.62	0.01
	(2.23[2])	(5.23[1])	(7.49[1])	(10.23[1])	(0.70)	(2.56[2])	(2.53[2])	(2.57[2])	(2.74[1])	(2.24[2])	(5.61[1])	(8.80[1])	(9.72[1])	(10.92[1])	(0.37)
Triple Witch	0.02	0.21	-0.22	-1.00	0.20	-1.85	-2.36	-2.55	-2.58	-0.92	-0.04	0.14	-0.12	-1.10	0.16
	(0.03)	(0.44)	(-0.81)	(-5.41[1])	(4.41[1])	(-1.06)	(-1.64[3])	(-1.95[3])	(-2.12[2])	(-1.13)	(-0.07)	(0.45)	(-0.54)	(-4.39[1])	(1.99[2])
Frequencies	2.81%	8.23%	10.86%	14.44%	63.46%	1.89%	7.09%	10.56%	14.69%	65.78%	3.72%	9.38%	11.16%	14.20%	61.54%

Appendix 7A: Results regression excess returns maturity portfolios seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of excess returns.The portfolio is divided onmaturity ranges.The excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2], [3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

	Excess Return Per Maturity Bucket								
	Call & Put			Call			Put		
	1 - 10	11 - 31	32 - 52	1 - 10	11 - 31	32 - 52	1 - 10	11 - 31	32 - 52
Halloween	-0.13	-0.64	-0.83	-0.46	-1.09	-3.90	0.24	-0.16	2.33
	(-0.49)	(-2.96[1])	(-2.94[1])	(-0.48)	(-1.13)	(-2.69[1])	(0.19)	(-0.14)	(1.50)
Holiday	0.57	-0.66	-0.15	3.37	0.41	-0.02	-2.46	-1.84	-0.29
	(0.99)	(-4.45[1])	(-0.87)	(1.23)	(0.53)	(-0.02)	(-0.96)	(-2.16[2])	(-0.25)
Monday	-1.26	-0.86	-0.86	-0.97	-1.05	-0.66	-1.58	-0.65	-1.07
	(-8.67[1])	(-7.51[1])	(-9.09[1])	(-1.30)	(-2.16[2])	(-1.52)	(-1.93[3])	(-1.17)	(-2.15[2])
January	-0.51	0.99	1.07	-1.40	0.91	1.53	0.54	1.09	0.59
	(-1.54)	(4.30[1])	(6.07[1])	(-1.12)	(0.96)	(1.52)	(0.31)	(1.12)	(0.55)
January2	N.A.	2.01	1.99	N.A.	1.35	1.89	N.A.	2.81	2.09
	N.A.	(3.34[1])	(3.56[1])	N.A.	(0.64)	(0.88)	N.A.	(1.54)	(1.20)
TOM	1.30	-0.03	0.74	2.53	0.50	0.99	-0.16	-0.61	0.49
	(4.03[1])	(-0.37)	(9.67[1])	(1.67)	(1.25)	(2.68[1])	(-0.10)	(-1.36)	(1.19)
Triple Witch	0.02	-0.20	-0.70	-0.25	0.71	-1.83	0.32	-1.18	0.59
	(0.09)	(-1.24)	(-4.51[1])	(-0.25)	(0.60)	(-1.88[3])	(0.31)	(-0.87)	(0.58)
Frequencies	19.82%	40.10%	40.10%	20.20%	40.04%	39.76%	19.42%	39.96%	40.62%

Appendix 7A: Results regression excess returns maturity portfolios seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of delta hedge excess returns.The portfolio is divided onmaturity ranges.The delta hedge excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2],[3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

Delta Hedge Excess Return Per Maturity Bucket

	Call & Put			Call			Put		
	1 - 10	11 - 31	32 - 52	1 - 10	11 - 31	32 - 52	1 - 10	11 - 31	32 - 52
Halloween	-0.10	-0.11	0.01	0.17	0.45	0.66	0.04	0.17	0.33
	(-0.25)	(-0.71)	(0.05)	(0.54)	(1.96[2])	(2.18[2])	(0.16)	(1.14)	(1.71[3])
Holiday	-0.16	0.21	0.07	-1.19	0.33	0.32	-0.65	0.27	0.19
	(-0.24)	(1.50)	(0.54)	(-2.42[2])	(1.57)	(1.12)	(-1.97[2])	(2.30[2])	(1.22)
Monday	-0.70	-1.13	-0.80	-0.11	-0.64	-0.59	-0.41	-0.88	-0.69
	(-4.27[1])	(-14.18[1])	(-12.22[1])	(-0.56)	(-5.49[1])	(-5.35[1])	(-3.61[1])	(-12.52[1])	(-9.39[1])
January	0.82	-0.33	-0.17	0.66	-0.87	-0.49	0.74	-0.60	-0.33
	(1.81[3])	(-2.06[2])	(-1.35)	(2.02[2])	(-2.30[2])	(-2.19[2])	(2.70[1])	(-2.71[1])	(-2.54[2])
January2	N.A.	-1.66	-0.38	N.A.	-2.10	-0.98	N.A.	-1.88	-0.68
	N.A.	(-5.33[1])	(-1.94[3])	N.A.	(-2.13[2])	(-1.59)	N.A.	(-3.32[1])	(-1.87[3])
TOM	0.51	-0.34	0.32	0.71	-0.32	0.29	0.61	-0.33	0.30
	(2.37[2])	(-5.67[1])	(6.41[1])	(1.86[3])	(-2.95[1])	(3.55[1])	(3.19[1])	(-5.28[1])	(5.72[1])
Triple Witch	0.01	0.29	-0.23	0.24	0.12	0.10	0.13	0.20	-0.07
	(0.04)	(1.49)	(-1.60)	(1.06)	(0.52)	(0.68)	(0.98)	(1.19)	(-0.62)
Frequencies	15.86%	40.68%	43.47%	15.77%	40.78%	43.44%	15.94%	40.57%	43.49%

Appendix 8A: Results regression call & put options excess return per annum

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put portfolio of excess returns.The regression are run per annum, noted an the horizontal axes of the table.The excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2,], [3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

Call & Put Excess Return Per Annum

Year

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Halloween	-0.08	-1.00	-1.32	-1.46	-0.63	-0.49	-0.15	0.22	0.00	0.12	0.16	-0.61	-0.48	0.18	1.02
	(-0.24)	(-1.30)	(-2.41[2])	(-2.02[2])	(-1.04)	(-0.95)	(-0.70)	(1.42)	(-0.01)	(0.83)	(1.02)	(-1.44)	(-0.76)	(0.42)	(6.08[1])
Holiday	-0.33	0.30	-0.91	-0.20	-1.19	-1.12	0.33	0.20	0.16	0.17	0.19	0.26	-0.02	-0.37	0.36
	(-0.64)	(0.49)	(-1.46)	(-0.35)	(-2.38[2])	(-6.38[1])	(1.37)	(0.84)	(2.04[2])	(1.74[3])	(1.17)	(0.85)	(-0.05)	(-0.94)	(0.78)
Monday	-0.87	-0.76	-1.08	-1.83	-1.99	-1.84	-0.79	-0.56	-0.41	-0.24	-0.08	-0.15	0.05	-0.28	-0.71
	(-3.15[1])	(-1.80[3])	(-1.99[2])	(-8.09[1])	(-6.65[1])	(-8.92[1])	(-5.23[1])	(-5.65[1])	(-3.80[1])	(-3.25[1])	(-0.82)	(-1.03)	(0.15)	(-1.44)	(-1.84[3])
January	1.11	0.53	0.10	0.25	0.33	0.02	0.53	0.37	0.56	0.16	0.96	0.61	2.26	1.31	1.19
	(1.69)	(0.66)	(0.17)	(0.36)	(0.50)	(0.02)	(1.97[2])	(1.96[2])	(3.35[1])	(1.10)	(3.95[1])	(4.67[1])	(6.43[1])	(3.30[1])	(5.22[1])
January2	0.30	4.76	-1.01	0.34	1.47	7.66	0.28	-0.36	0.09	0.10	-0.19	0.26	1.14	0.97	-0.23
	(0.74)	(16.14[1])	(-1.98[2])	(0.20)	(2.48[2])	(1.70[3])	(1.05)	(-0.49)	(0.26)	(0.42)	(-0.18)	(0.67)	(3.75[1])	(0.92)	(-0.34)
TOM	0.11	-0.08	0.57	0.22	0.24	0.70	0.42	0.32	0.25	0.23	0.26	0.41	0.80	0.61	0.23
	(0.55)	(-0.29)	(1.58)	(0.96)	(0.90)	(1.40)	(3.51[1])	(4.63[1])	(3.87[1])	(4.79[1])	(2.94[1])	(2.97[1])	(3.10[1])	(4.86[1])	(0.94)
Triple Witch	0.38	-0.49	-0.22	1.03	0.60	0.91	0.02	-0.38	-0.05	-0.03	0.02	-0.09	-1.59	-0.01	0.20
	(1.22)	(-1.63[3])	(-0.29)	(5.92[1])	(1.14)	(2.24[2])	(0.10)	(-1.63[3])	(-0.75)	(-0.25)	(0.13)	(-0.37)	(-1.72[3])	(-0.05)	(1.47)

Appendix 8B: Results regression call option excess return per annum

This table reports the results of the fixed pooled equal weighted (ew) regression on a cal portfolio of excess returns.The regression are run per annum, noted an the horizontal axes of the table.The excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2], [3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

Call Excess Return Per Annum

Year

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Halloween	0.04	-0.77	-2.17	-3.55	-1.42	-1.77	1.21	-2.49	0.35	1.47	-0.64	-2.62	-5.46	-0.25	0.64
	(0.03)	(-0.26)	(-1.25)	(-1.46)	(-0.56)	(-0.64)	(0.63)	(-1.57)	(0.28)	(0.89)	(-0.54)	(-1.42)	(-1.56)	(-0.09)	(0.37)
Holiday	-1.09	-0.18	1.53	4.28	0.49	-0.95	2.99	1.37	-1.68	-0.74	0.75	3.40	0.15	1.73	-1.64
	(-0.44)	(-0.05)	(0.57)	(1.11)	(0.16)	(-0.41)	(1.52)	(0.72)	(-2.18[2])	(-0.83)	(0.61)	(1.49)	(0.04)	(0.39)	(-0.53)
Monday	0.03	1.31	-2.81	-1.97	-0.79	-2.63	-1.83	0.05	-0.07	0.41	-1.13	-1.40	-3.68	-0.24	1.84
	(0.03)	(0.97)	(-1.82[3])	(-1.81[3])	(-0.55)	(-2.10[2])	(-1.07)	(0.04)	(-0.07)	(0.57)	(-1.23)	(-1.31)	(-2.06[2])	(-0.13)	(1.09)
January	2.23	3.01	-3.07	4.12	-2.42	3.48	0.99	0.39	1.22	-1.06	3.24	0.31	2.75	-2.10	-3.65
	(1.12)	(1.09)	(-1.29)	(1.22)	(-0.65)	(1.85[3])	(0.36)	(0.12)	(0.79)	(-0.59)	(1.76[3])	(0.19)	(0.64)	(-0.61)	(-1.45)
January2	-1.36	13.91	1.06	4.62	-10.65	4.85	7.11	7.96	1.18	-5.98	6.27	-1.94	-4.90	12.70	8.32
	(-0.68)	(12.27[1])	(0.43)	(1.05)	(-1.89[3])	(0.32)	(3.16[1])	(1.12)	(0.32)	(-4.59[1])	(1.51)	(-1.20)	(-1.26)	(2.13[2])	(3.32[1])
TOM	0.24	2.47	-0.70	2.50	2.17	1.75	-0.60	1.15	1.32	0.76	0.08	0.71	-1.44	0.43	1.37
	(0.25)	(2.31[2])	(-0.49)	(2.13[2])	(1.55)	(0.96)	(-0.40)	(1.21)	(1.77[3])	(0.95)	(0.10)	(0.70)	(-0.84)	(0.31)	(0.86)
Triple Witch	3.57	-2.29	-0.13	0.95	-3.90	-3.58	3.74	1.54	-1.89	0.75	-0.21	4.32	4.18	-1.41	-2.62
	(2.60[1])	(-1.45)	(-0.08)	(0.28)	(-2.02[2])	(-1.10)	(1.13)	(0.45)	(-1.07)	(0.43)	(-0.22)	(1.43)	(0.49)	(-0.67)	(-1.50)

Appendix 8C: Result regression put option excess return per annum

This table reports the results of the fixed pooled equal weighted (ew) regression on a put portfolio of excess returns.The regression are run per annum, noted an the horizontal axes of the table.The excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2],[3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

Put Excess Return Per Annum

Year

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Halloween	-0.26	-1.22	0.09	1.38	0.20	0.66	-2.68	2.45	-1.39	-2.22	-0.01	4.61	3.16	-0.71	-0.17
	(-0.16)	(-0.33)	(0.05)	(0.77)	(0.08)	(0.28)	(-1.42)	(1.51)	(-0.98)	(-1.39)	(-0.01)	(2.00[2])	(1.11)	(-0.24)	(-0.06)
Holiday	0.76	0.98	-3.79	-5.85	-2.88	-1.28	-3.24	-2.01	1.37	-0.29	-2.47	-4.44	-0.62	-1.01	1.23
	(0.29)	(0.21)	(-1.73[3])	(-1.69[3])	(-1.07)	(-0.63)	(-2.58[1])	(-1.19)	(1.35)	(-0.32)	(-1.88[3])	(-1.97[2])	(-0.18)	(-0.23)	(0.36)
Monday	-2.12	-3.72	0.91	-1.68	-3.23	-1.11	-0.12	-0.75	-0.90	-1.86	0.37	-0.52	1.98	0.20	-4.00
	(-1.48)	(-1.96[2])	(0.39)	(-1.55)	(-2.62[1])	(-0.90)	(-0.08)	(-0.53)	(-0.82)	(-2.26[2])	(0.36)	(-0.41)	(0.95)	(0.11)	(-2.16[2])
January	-1.01	-3.51	4.99	-4.12	2.77	-3.58	-0.42	0.98	0.43	1.54	-1.64	0.26	-0.44	3.42	5.77
	(-0.47)	(-1.27)	(2.01[2])	(-1.35)	(0.82)	(-1.51)	(-0.14)	(0.35)	(0.24)	(0.86)	(-0.91)	(0.13)	(-0.11)	(1.02)	(1.68[3])
January2	2.76	-10.06	-3.94	-6.47	15.57	10.43	-4.86	-4.07	-0.30	10.36	-3.85	5.72	9.66	-7.78	-5.15
	(1.06)	(-7.76[1])	(-1.22)	(-1.92[3])	(2.25[2])	(1.76[3])	(-2.98[1])	(-0.90)	(-0.07)	(4.46[1])	(-1.24)	(3.39[1])	(2.05[2])	(-1.84[3])	(-2.03[2])
TOM	-0.07	-3.75	2.01	-2.46	-1.75	-0.28	1.24	-0.68	-0.91	-0.22	0.19	-0.55	1.82	1.09	0.52
	(-0.06)	(-3.01[1])	(1.13)	(-2.23[2])	(-1.36)	(-0.24)	(0.91)	(-0.71)	(-1.07)	(-0.26)	(0.22)	(-0.40)	(1.00)	(0.74)	(0.25)
Triple Witch	-4.15	2.36	-0.33	1.12	5.30	4.25	-2.60	-1.55	1.80	-1.12	0.05	-5.00	-2.58	1.00	1.14
	(-3.67[1])	(1.08)	(-0.12)	(0.28)	(5.28[1])	(1.54)	(-0.96)	(-0.55)	(0.78)	(-0.63)	(0.05)	(-1.85[3])	(-0.41)	(0.38)	(0.40)

Appendix 9A: Results regression call & put options delta hedge excess return per annum

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put portfolio of delta hedge excess returns.The regression are run per annum, noted an the horizontal axes of the table.The delta hedge excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2], [3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

Call & Put Delta Hedge Excess Return Per Annum

Year

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Halloween	0.16	-0.20	0.43	0.16	0.54	1.05	-0.15	0.22	0.00	0.12	0.16	-0.61	-0.48	0.18	1.02
	(0.63)	(-0.56)	(2.09[2])	(0.54)	(0.92)	(3.03[1])	(-0.70)	(1.42)	(-0.01)	(0.83)	(1.02)	(-1.44)	(-0.76)	(0.42)	(6.08[1])
Holiday	0.21	0.11	0.44	-0.40	-0.24	-0.09	0.33	0.20	0.16	0.17	0.19	0.26	-0.02	-0.37	0.36
	(1.25)	(0.67)	(1.11)	(-0.74)	(-0.49)	(-0.32)	(1.37)	(0.84)	(2.04[1])	(1.74)	(1.17)	(0.85)	(-0.05)	(-0.94)	(0.78)
Monday	-1.21	-1.41	-1.30	-1.28	-1.42	-0.93	-0.79	-0.56	-0.41	-0.24	-0.08	-0.15	0.05	-0.28	-0.71
	(-13.10[1])	(-9.85[1])	(-9.86[1])	(-10.35[1])	(-8.03[1])	(-5.35[1])	(-5.23[1])	(-5.65[1])	(-3.80[1])	(-3.25[1])	(-0.82)	(-1.03)	(0.15)	(-1.44)	(-1.84[3])
January	0.09	0.47	0.31	1.64	1.55	1.55	0.53	0.37	0.56	0.16	0.96	0.61	2.26	1.31	1.19
	(0.35)	(2.27[2])	(1.24)	(4.06[1])	(4.22[1])	(2.82[1])	(1.97[2])	(1.96[2])	(3.35[1])	(1.10)	(3.95[1])	(4.67[1])	(6.43[1])	(3.30[1])	(5.22[1])
January2	1.26	-0.68	-0.23	-0.26	1.10	-2.15	0.28	-0.36	0.09	0.10	-0.19	0.26	1.14	0.97	-0.23
	(7.38[1])	(-2.00[2])	(-0.79)	(-0.78)	(1.25)	(-0.83)	(1.05)	(-0.49)	(0.26)	(0.42)	(-0.18)	(0.67)	(3.75[1])	(0.92)	(-0.34)
TOM	0.17	0.13	0.22	0.40	0.52	0.01	0.42	0.32	0.25	0.23	0.26	0.41	0.80	0.61	0.23
	(1.69[3])	(0.96)	(1.41)	(2.65[2])	(2.55[2])	(0.05)	(3.51[1])	(4.63[1])	(3.87[1])	(4.79[1])	(2.94[1])	(2.97[1])	(3.10[1])	(4.86[1])	(0.94)
Triple Witch	-0.29	0.36	0.34	0.01	-0.26	0.63	0.02	-0.38	-0.05	-0.03	0.02	-0.09	-1.59	-0.01	0.20
	(-2.26[2])	(1.78[3])	(1.94[3])	(0.11)	(-1.29)	(1.68[3])	(0.10)	(-1.63)	(-0.75)	(-0.25)	(0.13)	(-0.37)	(-1.72[3])	(-0.05)	(1.47)

Appendix 9B: Results regression call option delta hedge excess return per annum

This table reports the results of the fixed pooled equal weighted (ew) regression on a cal portfolio of delta hedge excess returns.The regression are run per annum, noted an the horizontal axes of the table.The delta hedge excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2,], [3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

Call Delta Hedge Excess Return Per Annum

Year

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Halloween	0.52	0.15	0.84	0.59	1.02	1.51	0.10	0.18	0.39	0.67	0.34	-1.21	-0.74	-0.26	0.17
	(1.51)	(0.29)	(3.16[1])	(1.37)	(1.33)	(3.16[1])	(0.37)	(0.73)	(2.24[2])	(2.37[2])	(1.45)	(-1.68[3])	(-0.57)	(-0.39)	(0.49)
Holiday	0.23	-0.16	0.14	-0.78	-0.80	0.01	0.28	0.13	0.14	0.25	0.10	0.23	0.11	-0.26	0.93
	(0.78)	(-0.77)	(0.30)	(-0.93)	(-1.05)	(0.01)	(0.79)	(0.40)	(1.09)	(2.28[2])	(0.42)	(0.43)	(0.08)	(-0.39)	(0.87)
Monday	-1.18	-1.22	-0.97	-1.06	-1.28	-0.94	-0.66	-0.53	-0.32	-0.11	0.16	0.20	0.86	-0.14	-0.69
	(-7.70[1])	(-6.58[1])	(-4.87[1])	(-5.16[1])	(-3.93[1])	(-4.13[1])	(-3.12[1])	(-4.19[1])	(-2.74[1])	(-0.90)	(1.13)	(0.73)	(1.72[3])	(-0.46)	(-1.12)
January	-1.01	0.12	-0.48	1.62	1.46	1.34	0.47	1.12	0.06	-0.44	0.80	0.21	2.40	2.01	1.74
	(-2.70[1])	(0.37)	(-1.32)	(2.63[1])	(2.49[2])	(2.78[1])	(1.41)	(3.15[1])	(0.23)	(-1.95[3])	(2.11[2])	(0.82)	(2.91[1])	(2.66[1])	(3.06[1])
January2	1.47	-2.18	-0.02	-0.83	2.18	-3.89	0.26	-0.82	-0.07	0.42	-1.53	0.34	3.03	0.98	-1.92
	(4.42[1])	(-12.65[1])	(-0.04)	(-1.39)	(2.71[1])	(-0.84)	(0.55)	(-0.55)	(-0.13)	(1.93[3])	(-0.90)	(1.66[3])	(5.01[1])	(0.47)	(-1.41)
TOM	0.24	-0.04	0.20	0.22	0.38	-0.10	0.57	0.22	0.28	0.26	0.19	0.32	1.45	0.46	0.02
	(1.77[3])	(-0.22)	(0.89)	(0.96)	(1.25)	(-0.21)	(3.25[1])	(2.00[2])	(3.41[1])	(2.74)	(1.31)	(1.27)	(3.03[1])	(2.05[2])	(0.04)
Triple Witch	-0.65	0.20	0.58	0.13	0.29	0.37	-0.53	-0.25	0.08	0.02	0.10	-0.38	-2.16	-0.19	0.96
	(-2.24[2])	(0.63)	(1.72[3])	(0.59)	(1.17)	(0.78)	(-2.16[2])	(-0.99)	(0.52)	(0.07)	(0.44)	(-1.14)	(-1.29)	(-0.26)	(2.32[2])

Appendix 9C: Result regression put option delta hedge excess return per annum

This table reports the results of the fixed pooled equal weighted (ew) regression on a put portfolio of delta hedge excess returns.The regression are run per annum, noted an the horizontal axes of the table.The delta hedge excess returns are the dependent variables. Halloween, holiday, Monday, January, TOM and triple witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date and are between parentheses. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. The [1], [2,], [3] superscript indicate t statistics are statistically significant at 10%, 5%, a 1% levels, respectively.

Put Delta Hedge Excess Return Per Annum

Year

	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Halloween	-0.27	-0.53	-0.22	-0.40	0.04	0.64	-0.42	0.27	-0.43	-0.38	-0.02	-0.04	-0.30	0.62	2.14
	(-1.09)	(-1.74[3])	(-0.63)	(-0.99)	(0.06)	(1.75[3])	(-1.46)	(1.01)	(-1.66[3])	(-2.11[2])	(-0.08)	(-0.09)	(-0.58)	(1.49)	(4.44[1])
Holiday	0.20	0.48	0.78	0.04	0.32	-0.19	0.37	0.27	0.18	0.08	0.28	0.29	-0.12	-0.48	-0.21
	(0.76)	(1.67[3])	(1.48)	(0.12)	(0.87)	(-0.66)	(1.85[3])	(1.73[3])	(1.37)	(0.45)	(2.11[2])	(1.37)	(-0.17)	(-1.20)	(-0.45)
Monday	-1.24	-1.67	-1.67	-1.54	-1.55	-0.92	-0.90	-0.59	-0.51	-0.36	-0.31	-0.49	-0.58	-0.41	-0.73
	(-10.55[1])	(-9.09[1])	(-8.53[1])	(-12.31[1])	(-6.87[1])	(-4.81[1])	(-6.34[1])	(-5.35[1])	(-2.86[1])	(-4.13[1])	(-2.87[1])	(-2.94[1])	(-1.74[3])	(-1.84[3])	(-2.48[2])
January	1.86	1.03	1.50	1.67	1.63	1.77	0.58	-0.18	1.16	0.75	1.11	1.06	2.16	0.79	0.63
	(5.90[1])	(2.93[1])	(6.28[1])	(3.04[1])	(3.63[1])	(1.84[3])	(1.85[3])	(-0.79)	(5.04[1])	(4.81[1])	(4.02[1])	(5.07[1])	(5.44[1])	(1.46)	(1.25)
January2	1.02	1.53	-0.50	0.56	-0.04	-0.50	0.30	0.09	0.30	-0.24	1.25	0.17	-0.34	0.96	1.50
	(4.75[1])	(1.60)	(-0.77)	(0.72)	(-0.03)	(-0.52)	(1.53)	(0.67)	(1.88[3])	(-0.46)	(3.30[1])	(0.28)	(-0.80)	(2.57[1])	(10.07[1])
TOM	0.07	0.36	0.24	0.61	0.65	0.11	0.29	0.44	0.22	0.21	0.32	0.49	0.30	0.76	0.44
	(0.62)	(2.01[2])	(1.28)	(4.11[1])	(3.12[1])	(0.76)	(2.35[2])	(5.40[1])	(2.11[2])	(2.98[1])	(3.37[1])	(3.44[1])	(1.14)	(5.38[1])	(1.95[3])
Triple Witch	0.17	0.59	0.07	-0.13	-0.81	0.81	0.50	-0.55	-0.20	-0.08	-0.07	0.19	-1.16	0.17	-0.77
	(0.90)	(1.38)	(0.32)	(-0.40)	(-3.20[1])	(2.16[2])	(1.62)	(-1.99[2])	(-0.76)	(-0.19)	(-0.29)	(0.80)	(-2.83[1])	(0.51)	(-1.52)

Appendix 10A: Results regression S&P500 index options excess returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of S&P 500 index options excess returns.The excess returns are the dependent variables. Halloween, Holiday, Monday, January, TOM and Triple Witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

S&P 500 Index Option's Excess Return

	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	1.71	0.0028	6.02	2.46	0.0062	3.95	0.69	0.0094	0.74
Halloweer	-0.52	0.0058	-0.90	-2.03	0.0132	-1.54	1.57	0.0194	0.81
Constant	1.50	0.0014	10.78	1.58	0.0030	5.19	1.41	0.0049	2.90
Holiday	-0.32	0.0062	-0.52	1.16	0.0175	0.66	-2.12	0.0235	-0.90
Constant	1.72	0.0015	11.31	1.64	0.0033	4.94	1.82	0.0053	3.43
Monday	-1.28	0.0032	-3.96***	-0.13	0.0077	-0.17	-2.66	0.0117	-2.27**
Constant	1.45	0.0014	10.03	1.65	0.0031	5.30	1.20	0.0051	2.38
January	0.56	0.0057	0.98	-0.55	0.0122	-0.45	2.04	0.0215	0.95
Constant	1.47	0.0014	10.82	1.61	0.0030	5.37	1.31	0.0048	2.74
January2	2.47	0.0183	1.35	0.84	0.0407	0.21	4.59	0.0566	0.81
Constant	1.48	0.0016	9.17	1.41	0.0035	4.06	1.57	0.0055	2.87
TOM	0.04	0.0030	0.13	0.75	0.0068	1.10	-0.80	0.0109	-0.74
Constant	1.51	0.0014	10.97	1.64	0.0030	5.42	1.35	0.0048	2.81
Triple Wit	-1.45	0.0063	-2.31**	-2.40	0.0192	-1.25	-0.36	0.0266	-0.14

Appendix 10B: Results regression S&P500 index options delta hedge excess returns on seasonal dummies

This table reports the results of the fixed pooled equal weighted (ew) regression on a call & put, call or put portfolio of S&P 500 index options delta hedge excess returns.The delta hedge excess returns are the dependent variables. Halloween, Holiday, Monday, January, TOM and Triple Witch dummies are the indepent variables. A dummy is assigned a value of one on the specific calendar day they represent and is assigned a value of zero otherwise. Standard errors are calculted by clustering by date. The constant represents the average return outside the period of the calendar dummy, the dummy coefficient indicates the additional positive/negative excess return received during that period. Coefficients are in percentage. Data is daily from 5 January 1996 through 29 October 2010. *, ** and *** stars indicate t statistics are statistically significant at the 10%, 5%, a 1% levels, respectively.

S&P 500 Index Option's Delta Hedge Excess Return

	Call & Put			Call			Put		
	Coeficient	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*	*Coeficient*	*Robust Std. Err.*	*T*
Constant	-6.78	0.0030	-22.27	-6.59	0.0043	-15.35	-7.03	0.0037	-18.87
Halloweer	0.96	0.0067	1.43	0.67	0.0095	0.71	1.36	0.0081	1.68*
Constant	-6.38	0.0011	-55.96	-6.31	0.0018	-36.06	-6.45	0.0013	-49.94
Holiday	0.10	0.0067	0.14	0.30	0.0085	0.36	-0.16	0.0096	-0.17
Constant	-5.90	0.0013	-47.12	-5.85	0.0019	-30.76	-5.95	0.0014	-41.68
Monday	-2.60	0.0026	-10.12***	-2.45	0.0041	-6.00***	-2.77	0.0031	-8.82***
Constant	-6.35	0.0012	-54.12	-6.24	0.0017	-36.07	-6.47	0.0015	-44.44
January	-0.41	0.0068	-0.61	-0.87	0.0102	-0.85	0.12	0.0087	0.14
Constant	-6.34	0.0011	-57.65	-6.26	0.0017	-37.64	-6.43	0.0013	-49.71
January2	-3.99	0.0255	-1.57	-4.14	0.0403	-1.03	-3.77	0.0186	-2.03**
Constant	-6.66	0.0013	-49.43	-6.62	0.0021	-32.12	-6.71	0.0016	-42.75
TOM	0.99	0.0024	4.04***	1.11	0.0038	2.93***	0.84	0.0027	3.13***
Constant	-6.39	0.0011	-56.58	-6.31	0.0017	-36.54	-6.48	0.0013	-50.18
Triple Wit	1.48	0.0068	2.17**	0.60	0.0090	0.66	2.42	0.0085	2.85***

Appendix 11A: Creating Option Prices, Add Stock Prices

```
local year "1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006
      2007 2008 2009 2010"

cd "E:\Data New\"

set more off

foreach y in `year'{

*** Sort Call, Put, stock beta and zeroyield file
use `y', clear
      preserve
            keep if cp_flag == "C"
                  sort optionid cusip date exdate strike_price
                        save `y'call, replace
            restore
            preserve
                  keep if cp_flag == "P"
                        sort optionid cusip date exdate strike_price
                              save `y'put, replace
                        restore



*** Generate Option Prices
      generate price = (best_bid + best_offer)/2
            egen ncp_flag = group( cp_flag)

      sort date exdate strike_price ncp_flag optionid
                  save `y', replace

      rename date DATE
      rename price PRICEb

*** Generate Option Prices -t
      foreach i in 1 2 3  {
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag optionid
                        merge date exdate strike_price ncp_flag optionid
using `y', keep(price)
      drop  _merge
      rename date date`i'
      rename price price`i'
      compress
}
      foreach i in 1 2 3  {
            replace date`i' =. if missing(price`i')
      }

      foreach i in 2 3   {
            replace date1 = date`i' if missing(date1)
      }

foreach i in 2 3   {
      replace price1 = price`i' if missing(price1)
      }
```

```
	rename DATE date
	rename PRICE price
	rename price1 price_t
	rename date1 date_t
	drop date2
	drop date3

	drop price2
	drop price3

	*** Stock to option

	sort cusip date
		merge cusip date using `y'stock, keep(prc)
			drop if missing(secid)
			drop  _merge



	rename date DATE
	rename prc PRC

	foreach i in 1 2 3 4 {
		generate date = DATE - `i'
			sort cusip date
				merge cusip date using `y'stock, keep(prc)

	drop if missing(secid)
	drop date
	drop  _merge

	rename prc prc`i'
	}

	foreach i in 2 3 4  {
	replace prc1 = prc`i' if missing(prc1)
	}


	rename DATE date
	rename prc1 prc_t
	rename PRC prc
	drop prc2
	drop prc3
	drop prc4


sort date exdate strike_price ncp_flag optionid
	duplicates drop optionid date, force

save `y'total, replace

}
```

Appendix 11B: Creating December files

```
local year "1996 1997  1998  1999  2000  2001  2002  2003  2004  2005  2006
      2007  2008  2009  2010"
cd "E:\Data New\"

foreach y in `year' {
      use `y', clear
            keep if date >= d(0112`y')
                  sort  date exdate strike_price ncp_flag optionid
                        save december`y', replace
            }

local year "     1998  1999  2000  2001  2002  2003  2004  2005  2006  2007
      2008  2009  2010"
```

Appendix 11C: Option Prices t-1 December =- January

```
cd "E:\Data New\"
set more off


foreach y in `year' {

      local x = `y' - 1

      use `y'total, clear
      rename price PRICE
      rename date DATE


            foreach i in 1 2 3 4 5 6 7  {
                  generate date = DATE - `i'
                        sort  date exdate strike_price ncp_flag optionid
                              merge date exdate strike_price ncp_flag
optionid using december`x', keep(price)
                  drop  _merge
                  rename date date`i'
                  rename price price`i'
                  compress



                        }


                  foreach i in 1 2 3 4 5 6 7  {
                  replace date`i' =. if missing(price`i')
                  }


                  foreach i in  1 2 3 4 5 6 7   {
                  replace date_t = date`i' if missing(date_t)
                  }
```

```
                foreach i in 1 2 3 4 5 6 7   {
                replace price_t = price`i' if missing(price_t)
                }


                rename DATE date
                rename PRICE price
                drop  date1 price1 date2 price2 date3 price3 date4
price4 date5 price5 date6 price6 date7 price7
                drop if missing(secid)
                duplicates drop optionid date, force
                save `y'total, replace

      }
```

Appendix 11D: Stock Prices t-1 December =- January

```
local year "1997 1998  1999  2000  2001  2002  2003  2004  2005  2006  2007
      2008  2009  2010"

cd "E:\Data New\"
set more off


foreach y in `year' {

      local x = `y' - 1

      use `y'total, clear
      rename prc PRC
      rename date DATE


            foreach i in 1 2 3 4 5  {
                  generate date = DATE - `i'
                        sort  cusip date optionid
                              merge cusip date using `x'stock, keep(prc)
                  drop  _merge
                  rename date date`i'
                  rename prc prc`i'
                  compress


                        }


                  foreach i in 1 2 3 4 5    {
                  replace prc_t = prc`i' if missing(prc_t)
                  }


                  rename DATE date
                  rename PRC prc
```

```
                drop  date1 prc1 date2 prc2 date3 prc3 date4 prc4 date5
prc5
                drop if missing(secid)
                duplicates drop optionid date, force
                save `y'total, replace
      }
```

Appendix 11E: Missing IV 1996

```
local y "1996"

cd "E:\Data New\"

set more off

use `y'call, clear
      sort cusip date exdate strike_price
            save `y'call, replace

use `y'put
      sort cusip date exdate strike_price
            save `y'put, replace

use `y'total, clear
      rename impl_volatility IV

***IV step 1 for Calls
      sort cusip date exdate strike_price
            merge cusip date exdate strike_price using `y'put,
keep(impl_volatility)
                replace  IV = impl_volatility if missing(IV) & ncp_flag
== 1
                      drop impl_volatility
                      drop  _merge

***IV step 1 for Puts
      sort cusip date exdate strike_price
            merge cusip date exdate strike_price using `y'call,
keep(impl_volatility)
                replace  IV = impl_volatility if missing(IV) & ncp_flag
== 2
                      drop impl_volatility
                      drop  _merge

*** Rename impl_volatility IV, Sort and Save File
      rename IV impl_volatility
                sort optionid cusip date exdate strike_price

cd "E:\Data New\IV DATA"
      save `y'IV, replace



*** Rename date to DATE and IV to impl_volatility
      rename date DATE
      rename impl_volatility IV
```

```
***IV Step 2 for Calls, goes back 10 calendar days
      foreach i in 1 2 4 5 6 7 8 9 10 {
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price
                        merge optionid cusip date exdate strike_price
using `y'IV, keep(impl_volatility)

      drop if missing(secid)

      replace  IV =  impl_volatility if missing(IV)
      drop impl_volatility
      drop date
      drop  _merge
      }

rename DATE date
rename IV impl_volatility
save `y'IV, replace
```

Appendix 11F: Missing IV

```
local year " 19971998  1999  2000  2001  2002  2003  2004  2005  2006  2007
      2008  2009  2010"

cd "E:\Data New\"

set more off

foreach y in `year' {

cd "E:\Data New\"

local x = `y' - 1

use `y'call, clear
      sort cusip date exdate strike_price
            save `y'call, replace

use `y'put, clear
      sort cusip date exdate strike_price
            save `y'put, replace

use `y'total, clear
      rename impl_volatility IV

***IV step 1 for Calls
      sort cusip date exdate strike_price
            merge cusip date exdate strike_price using `y'put,
keep(impl_volatility)
                  replace  IV = impl_volatility if missing(IV) & ncp_flag
== 1
                        drop impl_volatility
                        drop  _merge

***IV step 1 for Puts
      sort cusip date exdate strike_price
```

```
            merge cusip date exdate strike_price using `y'call,
keep(impl_volatility)
                  replace  IV = impl_volatility if missing(IV) & ncp_flag
== 2
                        drop impl_volatility
                        drop  _merge

*** Rename impl_volatility IV, Sort and Save File
      rename IV impl_volatility
                  sort optionid cusip date exdate strike_price

cd "E:\Data New\IV DATA"
      save `y'IV, replace



*** Rename date to DATE and IV to impl_volatility
      rename date DATE
      rename impl_volatility IV


***IV Step 2 for Calls, goes back 10 calendar days
      foreach i in 1 2 4 5 6 7 8 9 10 {
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price
                        merge optionid cusip date exdate strike_price
using `y'IV, keep(impl_volatility)

      drop if missing(secid)

      replace  IV =  impl_volatility if missing(IV)
      drop impl_volatility
      drop date
      drop  _merge
      }

      *IV Step 2 for Calls, goes back 10 calendar days
      foreach i in 1 2 4 5 6 7 8 9 10 {
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price
                        merge optionid cusip date exdate strike_price
using `x'IV, keep(impl_volatility)

      drop if missing(secid)

      replace  IV =  impl_volatility if missing(IV)
      drop impl_volatility
      drop date
      drop  _merge
      }



rename DATE date
rename IV impl_volatility
sort optionid cusip date exdate strike_price
save `y'IV, replace
```

```
}
```

Appendix 11G: Data Filter

```
local year "1996 1997 1998 1999 2000    2001  2002  2003  2004  2005  2006
      2007  2008  2009  2010"

cd "E:\Data New\IV DATA"

set more off

foreach y in `year'{

cd "E:\Data New\IV DATA"
      use `y'IV, clear


*** Create variable Spread, Spread of Midpoint, Bid Price in Percent of
Stock, Offer price of Stock Price
      generate Spread = best_offer - best_bid
      generate SpreadofMidPoint = (best_offer - best_bid)/(( best_bid+
best_offer)/2)
      generate BidPriceofStock =  best_bid /prc
      generate OfferPriceofStock = best_offer/prc

cd "E:\Data New\Gefilterde Data"
*** Save and Sort
                  sort optionid cusip date exdate strike_price ncp_flag
                        compress
                              save `y'filter, replace

*** Rename date DATE and best_bid BestBid
      rename date DATE
      rename best_bid BestBid


*** Day t-2 Bid Price is less than $0.50, drop
      foreach i in 2 3 4 5{
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price ncp_flag
                        merge optionid cusip date exdate strike_price
ncp_flag using `y'filter, keep(best_bid)

      drop if missing(secid)
      drop date
      drop  _merge

      recode best_bid .= 999
      rename best_bid best_bid`i'
}

      drop if best_bid2 <=.50
      drop if best_bid3 <=.50 & best_bid2 ==999
      drop if best_bid4 <=.50 & best_bid2 ==999 & best_bid3 ==999
      drop if best_bid5 <=.50 & best_bid2 ==999 & best_bid3 ==999 &
best_bid4 ==999

      drop best_bid2
      drop best_bid3
      drop best_bid4
```

```
      drop best_bid5

*** Day t-2 Bid Price is 0.1% or less of the price of the underlying
stock, drop
      foreach i in 2 3 4 5{
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price ncp_flag
                        merge optionid cusip date exdate strike_price
ncp_flag using `y'filter, keep(BidPriceofStock)

      drop if missing(secid)
      drop date
      drop  _merge

      recode BidPriceofStock .= 999
      rename BidPriceofStock BidPriceofStock`i'
}
      drop if BidPriceofStock2 <=.001
      drop if BidPriceofStock3 <=.001 & BidPriceofStock2 ==999
      drop if BidPriceofStock4 <=.001 & BidPriceofStock2 ==999 &
BidPriceofStock3 ==999
      drop if BidPriceofStock5 <=.001 & BidPriceofStock2 ==999 &
BidPriceofStock3 ==999 & BidPriceofStock5 ==999


      drop BidPriceofStock2
      drop BidPriceofStock3
      drop BidPriceofStock4
      drop BidPriceofStock5

*** Day t-2 Spread is more than 25% of the midpoint, drop
      foreach i in 2 3 4 5{
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price ncp_flag
                        merge optionid cusip date exdate strike_price
ncp_flag using `y'filter, keep(SpreadofMidPoint)

      drop if missing(secid)
      drop date
      drop  _merge

      recode SpreadofMidPoint .= -999
      rename SpreadofMidPoint SpreadofMidPoint`i'
}

      drop if SpreadofMidPoint2 >=.25
      drop if SpreadofMidPoint3 >=.25 & SpreadofMidPoint2 ==-999
      drop if SpreadofMidPoint4 >=.25 & SpreadofMidPoint2 ==-999 &
SpreadofMidPoint3 ==-999
      drop if SpreadofMidPoint5 >=.25 & SpreadofMidPoint2 ==-999 &
SpreadofMidPoint3 ==-999 & SpreadofMidPoint4 ==-999

      drop SpreadofMidPoint2
      drop SpreadofMidPoint3
      drop SpreadofMidPoint4
      drop SpreadofMidPoint5
```

```
***Day t-1 Offer price is less than Bid price, drop
      foreach i in 1 2 3 4{
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price ncp_flag
                        merge optionid cusip date exdate strike_price
ncp_flag using `y'filter, keep(Spread)

      drop if missing(secid)
      drop date
      drop  _merge

      recode Spread .= 999
      rename Spread Spread`i'
}

      drop if Spread1 <0
      drop if Spread2 <0 & Spread1 ==999
      drop if Spread3 <0 & Spread1 ==999 & Spread2 ==999
      drop if Spread4 <0 & Spread1 ==999 & Spread2 ==999 & Spread3 ==999

      drop Spread1
      drop Spread2
      drop Spread3
      drop Spread4

*** Day t Offer price is less than Bid price, drop
      generate Spread =  best_offer - BestBid
            drop if Spread <0
                  drop Spread

*** Day t-1 Offer Price is twice the Stock Price
      foreach i in 1 2 3 4{
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price ncp_flag
                        merge optionid cusip date exdate strike_price
ncp_flag using `y'filter, keep(OfferPriceofStock)

      drop if missing(secid)
      drop date
      drop  _merge

      recode OfferPriceofStock .= -999
      rename OfferPriceofStock OfferPriceofStock`i'

}
      drop if OfferPriceofStock1 >2
      drop if OfferPriceofStock2 >2 & OfferPriceofStock1 ==-999
      drop if OfferPriceofStock3 >2 & OfferPriceofStock1 ==-999 &
OfferPriceofStock2 ==-999
      drop if OfferPriceofStock3 >2 & OfferPriceofStock1 ==-999 &
OfferPriceofStock2 ==-999 & OfferPriceofStock3 ==-999


      drop OfferPriceofStock1
      drop OfferPriceofStock2
      drop OfferPriceofStock3
      drop OfferPriceofStock4
```

```
*** Dat t Offer Price twice the Stock Price
      generate OfferPriceofStock = best_offer / prc
            drop if OfferPriceofStock >2
                  drop OfferPriceofStock

*** Day t-1 Spread is more than $5
      foreach i in 1 2 3 4{
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price ncp_flag
                        merge optionid cusip date exdate strike_price
ncp_flag using `y'filter, keep(Spread)

      drop if missing(secid)
      drop date
      drop  _merge

      recode Spread .= -999
      rename Spread Spread`i'
}
      drop if Spread1 >=5
      drop if Spread2 >=5 & Spread1 ==-999
      drop if Spread3 >=5 & Spread1 ==-999 & Spread2 ==-999
      drop if Spread3 >=5 & Spread1 ==-999 & Spread2 ==-999 & Spread3 ==-
999

      drop Spread1
      drop Spread2
      drop Spread3
      drop Spread4

*** Day t Spread is more than $5
      generate Spread =  best_offer - BestBid
            drop if Spread >=5
                  drop Spread

*** Day t-1 Spread 200% of midpoint
      foreach i in 1 2 3 4{
            generate date = DATE - `i'
                  sort optionid cusip date exdate strike_price ncp_flag
                        merge optionid cusip date exdate strike_price
ncp_flag using `y'filter, keep(SpreadofMidPoint)

      drop if missing(secid)
      drop date
      drop  _merge

      recode SpreadofMidPoint .= -999
      rename SpreadofMidPoint SpreadofMidPoint`i'
}

      drop if SpreadofMidPoint1 >=2
      drop if SpreadofMidPoint2 >=2 & SpreadofMidPoint1 ==-999
      drop if SpreadofMidPoint3 >=2 & SpreadofMidPoint1 ==-999 &
SpreadofMidPoint2 ==-999
      drop if SpreadofMidPoint3 >=2 & SpreadofMidPoint1 ==-999 &
SpreadofMidPoint2 ==-999 & SpreadofMidPoint3 ==-999
```

```
      drop SpreadofMidPoint1
      drop SpreadofMidPoint2
      drop SpreadofMidPoint3
      drop SpreadofMidPoint4

*** Day t Spread 200% of midpoint
      generate SpreadofMidPoint = (BestBid - best_offer)/((best_offer +
BestBid )/2)
            drop if SpreadofMidPoint >=2
                  drop SpreadofMidPoint




      save `y'filter, replace

      }
```

Appendix 11H: Zero Yield

```
local year "1997 1998  1999  2000  2001  2002  2003  2004  2005  2006  2007
      2008  2009  2010"

cd "E:\Data New\Gefilterde Data\"

set more off

foreach y in `year' {
use `y'filter, clear
            rename DATE date
                  sort date
                  merge date using "E:\Data New\zeroyield", keep(rate)
                  drop if missing(secid)
                        save `y'filter, replace

            }
```

Appendix 11I: Return and Option Greeks

```
local year "1996 1997  1998  1999  2000  2001  2002  2003  2004  2005  2006
      2007  2008  2009  2010"

cd "E:\Data New\Gefilterde Data\"

set more off

foreach y in `year' {

cd "E:\Data New\Gefilterde Data\"


use `y'filter, clear

*** Generate Greeks, replace them and drop the variable
      generate P = (exdate - date)/365
```

```
      generate DeltaCall = normal((ln(prc/(strike_price/1000))+(rente+
impl_volatility^2)*P)/(impl_volatility*(P^0.5))) if  ncp_flag == 1
      generate DeltaPut = -1 +
normal((ln(prc/(strike_price/1000))+(rente+
impl_volatility^2)*P)/(impl_volatility*(P^0.5))) if  ncp_flag == 2

      replace  delta = DeltaCall if missing(delta) & ncp_flag == 1
      replace  delta = DeltaPut if missing(delta) & ncp_flag == 2

      drop DeltaCall
      drop DeltaPut

      generate Vega =
prc*normalden(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)

      replace vega = Vega if missing(vega)

      drop Vega
      drop T

*** generate rerurn
      generate return = (price - price_t)/price_t


*** Excess Returns
      generate T =  date - date_t
      generate excess_return = (price - price_t)/price_t -T*(rate/1000)


***Delta hedgde excess returns
      generate delta_return = (price - price_t)/price_t -
((delta*prc_t)/price_t)*((prc - prc_t)/prc_t)
      generate delta_excess_return = ((price - price_t)/price_t -
(T*rate/1000))-(((delta*prc_t)/price_t)*((prc - prc_t)/prc_t) -
T*(rate/1000))

      drop _merge
      drop if missing(secid)
      duplicates drop optionid date, force
      compress

      cd "E:\Data New\Dummy\"
      save `y'dummy, replace
      }
```

Appendix 11J: Dummies

```
set more off

foreach i in halloween Jan Jan2 TOM Holiday Monday triplewitch{
      generate `i' = 0
}

************************************************************
*** Halloween effect dummy
      ****
```

```
*****************************************************************

replace halloween = 1 if date >=d(01111995) & date <d(01041996)
replace halloween = 1 if date >=d(01111996) & date <d(01041997)
replace halloween = 1 if date >=d(01111997) & date <d(01041998)
replace halloween = 1 if date >=d(01111998) & date <d(01041999)
replace halloween = 1 if date >=d(01111999) & date <d(01042000)
replace halloween = 1 if date >=d(01112000) & date <d(01042001)
replace halloween = 1 if date >=d(01112001) & date <d(01042002)
replace halloween = 1 if date >=d(01112002) & date <d(01042003)
replace halloween = 1 if date >=d(01112003) & date <d(01042004)
replace halloween = 1 if date >=d(01112004) & date <d(01042005)
replace halloween = 1 if date >=d(01112005) & date <d(01042006)
replace halloween = 1 if date >=d(01112006) & date <d(01042007)
replace halloween = 1 if date >=d(01112007) & date <d(01042008)
replace halloween = 1 if date >=d(01112008) & date <d(01042009)
replace halloween = 1 if date >=d(01112009) & date <d(01042010)
replace halloween = 1 if date >=d(01112010) & date <d(01042011)

*****************************************************************
*** January effect dummy
     ****
*****************************************************************

replace Jan = 1 if date >=d(01011996) & date <=d(31011996)
replace Jan = 1 if date >=d(01011997) & date <=d(31011997)
replace Jan = 1 if date >=d(01011998) & date <=d(31011998)
replace Jan = 1 if date >=d(01011999) & date <=d(31011999)
replace Jan = 1 if date >=d(01012000) & date <=d(31012000)
replace Jan = 1 if date >=d(01012001) & date <=d(31012001)
replace Jan = 1 if date >=d(01012002) & date <=d(31012002)
replace Jan = 1 if date >=d(01012003) & date <=d(31012003)
replace Jan = 1 if date >=d(01012004) & date <=d(31012004)
replace Jan = 1 if date >=d(01012005) & date <=d(31012005)
replace Jan = 1 if date >=d(01012006) & date <=d(31012006)
replace Jan = 1 if date >=d(01012007) & date <=d(31012007)
replace Jan = 1 if date >=d(01012008) & date <=d(31012008)
replace Jan = 1 if date >=d(01012009) & date <=d(31012009)
replace Jan = 1 if date >=d(01012010) & date <=d(31012010)

*****************************************************************
*** January2 effect dummy
     ****
*****************************************************************

replace Jan2 = 1 if date >=d(01011996) & date <=d(05011996)
replace Jan2 = 1 if date >=d(01011997) & date <=d(05011997)
replace Jan2 = 1 if date >=d(01011998) & date <=d(05011998)
replace Jan2 = 1 if date >=d(01011999) & date <=d(05011999)
replace Jan2 = 1 if date >=d(01012000) & date <=d(05012000)
replace Jan2 = 1 if date >=d(01012001) & date <=d(05012001)
replace Jan2 = 1 if date >=d(01012002) & date <=d(05012002)
replace Jan2 = 1 if date >=d(01012003) & date <=d(05012003)
replace Jan2 = 1 if date >=d(01012004) & date <=d(05012004)
replace Jan2 = 1 if date >=d(01012005) & date <=d(05012005)
replace Jan2 = 1 if date >=d(01012006) & date <=d(05012006)
replace Jan2 = 1 if date >=d(01012007) & date <=d(05012007)
```

```
replace Jan2 = 1 if date >=d(01012008) & date <=d(05012008)
replace Jan2 = 1 if date >=d(01012009) & date <=d(05012009)
replace Jan2 = 1 if date >=d(01012010) & date <=d(05012010)

*****************************************************************
*** TOM  effect dummy 1996
      ****
*****************************************************************

replace TOM = 1 if date >=d(31121995) & date <=d(06011996)
replace TOM = 1 if date >=d(31011996) & date <=d(07021996)
replace TOM = 1 if date >=d(29021996) & date <=d(07031996)
replace TOM = 1 if date >=d(29031996) & date <=d(07041996)
replace TOM = 1 if date >=d(30041996) & date <=d(07051996)
replace TOM = 1 if date >=d(31051996) & date <=d(07061996)
replace TOM = 1 if date >=d(28061996) & date <=d(09071996)
replace TOM = 1 if date >=d(31071996) & date <=d(07081996)
replace TOM = 1 if date >=d(30081996) & date <=d(08091996)
replace TOM = 1 if date >=d(30091996) & date <=d(06101996)
replace TOM = 1 if date >=d(31101996) & date <=d(07111996)
replace TOM = 1 if date >=d(29111996) & date <=d(07121996)

*****************************************************************
*** TOM  effect dummy 1997
      ****
*****************************************************************

replace TOM = 1 if date >=d(31121996) & date <=d(07011997)
replace TOM = 1 if date >=d(31011997) & date <=d(06021997)
replace TOM = 1 if date >=d(28021997) & date <=d(06031997)
replace TOM = 1 if date >=d(31031997) & date <=d(04041997)
replace TOM = 1 if date >=d(30041997) & date <=d(06051997)
replace TOM = 1 if date >=d(30051997) & date <=d(05061997)
replace TOM = 1 if date >=d(30061997) & date <=d(07071997)
replace TOM = 1 if date >=d(31071997) & date <=d(06081997)
replace TOM = 1 if date >=d(29081997) & date <=d(06091997)
replace TOM = 1 if date >=d(30091997) & date <=d(06101997)
replace TOM = 1 if date >=d(31101997) & date <=d(06111997)
replace TOM = 1 if date >=d(28111997) & date <=d(04121997)

*****************************************************************
*** TOM effect dummy 1998
      ****
*****************************************************************

replace TOM = 1 if date >=d(31121997) & date <=d(07011998)
replace TOM = 1 if date >=d(30011998) & date <=d(05021998)
replace TOM = 1 if date >=d(27021998) & date <=d(05031998)
replace TOM = 1 if date >=d(31031998) & date <=d(06041998)
replace TOM = 1 if date >=d(30041998) & date <=d(06051998)
replace TOM = 1 if date >=d(29051998) & date <=d(04061998)
replace TOM = 1 if date >=d(30061998) & date <=d(07071998)
replace TOM = 1 if date >=d(31071998) & date <=d(06081998)
replace TOM = 1 if date >=d(31081998) & date <=d(04091998)
replace TOM = 1 if date >=d(30091998) & date <=d(06101998)
replace TOM = 1 if date >=d(30101998) & date <=d(05111998)
replace TOM = 1 if date >=d(30111998) & date <=d(04121998)
```

```
*****************************************************************
*** TOM  effect dummy 1999
      ****
*****************************************************************

replace TOM = 1 if date >=d(31121998) & date <=d(07011999)
replace TOM = 1 if date >=d(29011999) & date <=d(04021999)
replace TOM = 1 if date >=d(26021999) & date <=d(04031999)
replace TOM = 1 if date >=d(31031999) & date <=d(06041999)
replace TOM = 1 if date >=d(30041999) & date <=d(06051999)
replace TOM = 1 if date >=d(28051999) & date <=d(04061999)
replace TOM = 1 if date >=d(30061999) & date <=d(07071999)
replace TOM = 1 if date >=d(30071999) & date <=d(05081999)
replace TOM = 1 if date >=d(31081999) & date <=d(07091999)
replace TOM = 1 if date >=d(30091999) & date <=d(06101999)
replace TOM = 1 if date >=d(29101999) & date <=d(04111999)
replace TOM = 1 if date >=d(30111999) & date <=d(06121999)

*****************************************************************
*** TOM  effect dummy 2000
      ****
*****************************************************************

replace TOM = 1 if date >=d(31121999) & date <=d(06012000)
replace TOM = 1 if date >=d(31012000) & date <=d(04022000)
replace TOM = 1 if date >=d(29022000) & date <=d(04032000)
replace TOM = 1 if date >=d(31032000) & date <=d(06042000)
replace TOM = 1 if date >=d(28042000) & date <=d(04052000)
replace TOM = 1 if date >=d(31052000) & date <=d(06062000)
replace TOM = 1 if date >=d(30062000) & date <=d(07072000)
replace TOM = 1 if date >=d(31072000) & date <=d(04082000)
replace TOM = 1 if date >=d(31082000) & date <=d(07092000)
replace TOM = 1 if date >=d(29092000) & date <=d(05102000)
replace TOM = 1 if date >=d(31102000) & date <=d(06112000)
replace TOM = 1 if date >=d(30112000) & date <=d(06122000)

*****************************************************************
*** TOM effect dummy 2001
      ****
*****************************************************************

replace TOM = 1 if date >=d(29122000) & date <=d(05012001)
replace TOM = 1 if date >=d(31012001) & date <=d(06022001)
replace TOM = 1 if date >=d(28022001) & date <=d(06032001)
replace TOM = 1 if date >=d(30032001) & date <=d(05042001)
replace TOM = 1 if date >=d(30042001) & date <=d(04052001)
replace TOM = 1 if date >=d(31052001) & date <=d(06062001)
replace TOM = 1 if date >=d(29062001) & date <=d(06072001)
replace TOM = 1 if date >=d(31072001) & date <=d(06082001)
replace TOM = 1 if date >=d(31082001) & date <=d(07092001)
replace TOM = 1 if date >=d(28092001) & date <=d(04102001)
replace TOM = 1 if date >=d(31102001) & date <=d(06112001)
replace TOM = 1 if date >=d(30112001) & date <=d(06122001)

*****************************************************************
```

```
*** TOM effect dummy 2002
      ****
*****************************************************************

replace TOM = 1 if date >=d(31122001) & date <=d(07012002)
replace TOM = 1 if date >=d(31012002) & date <=d(06022002)
replace TOM = 1 if date >=d(28022002) & date <=d(06032002)
replace TOM = 1 if date >=d(29032002) & date <=d(04042002)
replace TOM = 1 if date >=d(30042002) & date <=d(06052002)
replace TOM = 1 if date >=d(31052002) & date <=d(06062002)
replace TOM = 1 if date >=d(28062002) & date <=d(05072002)
replace TOM = 1 if date >=d(31072002) & date <=d(06082002)
replace TOM = 1 if date >=d(30082002) & date <=d(06092002)
replace TOM = 1 if date >=d(30092002) & date <=d(04102002)
replace TOM = 1 if date >=d(31102002) & date <=d(06112002)
replace TOM = 1 if date >=d(29112002) & date <=d(05122002)

*****************************************************************
*** TOM effect dummy 2003
      ****
*****************************************************************

replace TOM = 1 if date >=d(31122002) & date <=d(07012003)
replace TOM = 1 if date >=d(31012003) & date <=d(06022003)
replace TOM = 1 if date >=d(28022003) & date <=d(06032003)
replace TOM = 1 if date >=d(28032003) & date <=d(04042003)
replace TOM = 1 if date >=d(30042003) & date <=d(06052003)
replace TOM = 1 if date >=d(30052003) & date <=d(05062003)
replace TOM = 1 if date >=d(30062003) & date <=d(07072003)
replace TOM = 1 if date >=d(31072003) & date <=d(06082003)
replace TOM = 1 if date >=d(29082003) & date <=d(05092003)
replace TOM = 1 if date >=d(30092003) & date <=d(06102003)
replace TOM = 1 if date >=d(31102003) & date <=d(06112003)
replace TOM = 1 if date >=d(28112003) & date <=d(04122003)

*****************************************************************
*** TOM effect dummy 2004
      ****
*****************************************************************

replace TOM = 1 if date >=d(31122003) & date <=d(07012004)
replace TOM = 1 if date >=d(30012004) & date <=d(05022004)
replace TOM = 1 if date >=d(27022004) & date <=d(04032004)
replace TOM = 1 if date >=d(27032004) & date <=d(06042004)
replace TOM = 1 if date >=d(30042004) & date <=d(06052004)
replace TOM = 1 if date >=d(28052004) & date <=d(04062004)
replace TOM = 1 if date >=d(30062004) & date <=d(07072004)
replace TOM = 1 if date >=d(30072004) & date <=d(05082004)
replace TOM = 1 if date >=d(31082004) & date <=d(07092004)
replace TOM = 1 if date >=d(30092004) & date <=d(06102004)
replace TOM = 1 if date >=d(29102004) & date <=d(04112004)
replace TOM = 1 if date >=d(30112004) & date <=d(06122004)

*****************************************************************
*** TOM effect dummy 2005
      ****
*****************************************************************
```

```
replace TOM = 1 if date >=d(30122004) & date <=d(06012005)
replace TOM = 1 if date >=d(31012005) & date <=d(04022005)
replace TOM = 1 if date >=d(28022005) & date <=d(04032005)
replace TOM = 1 if date >=d(31032005) & date <=d(06042005)
replace TOM = 1 if date >=d(29042005) & date <=d(05052005)
replace TOM = 1 if date >=d(31052005) & date <=d(06062005)
replace TOM = 1 if date >=d(30062005) & date <=d(06072005)
replace TOM = 1 if date >=d(29072005) & date <=d(04082005)
replace TOM = 1 if date >=d(31082005) & date <=d(07092005)
replace TOM = 1 if date >=d(30092005) & date <=d(06102005)
replace TOM = 1 if date >=d(31102005) & date <=d(04112005)
replace TOM = 1 if date >=d(30112005) & date <=d(06122005)

*****************************************************************
*** TOM effect dummy 2006
      ****
*****************************************************************

replace TOM = 1 if date >=d(30122005) & date <=d(06012006)
replace TOM = 1 if date >=d(31012006) & date <=d(06022006)
replace TOM = 1 if date >=d(28022006) & date <=d(06032006)
replace TOM = 1 if date >=d(31032006) & date <=d(06042006)
replace TOM = 1 if date >=d(28042006) & date <=d(04052006)
replace TOM = 1 if date >=d(31052006) & date <=d(06062006)
replace TOM = 1 if date >=d(30062006) & date <=d(07072006)
replace TOM = 1 if date >=d(31072006) & date <=d(04082006)
replace TOM = 1 if date >=d(31082006) & date <=d(07092006)
replace TOM = 1 if date >=d(29092006) & date <=d(05102006)
replace TOM = 1 if date >=d(31102006) & date <=d(06112006)
replace TOM = 1 if date >=d(30112006) & date <=d(06122006)

*****************************************************************
*** TOM effect dummy 2007
      ****
*****************************************************************

replace TOM = 1 if date >=d(29122006) & date <=d(05012007)
replace TOM = 1 if date >=d(31012007) & date <=d(06022007)
replace TOM = 1 if date >=d(28022007) & date <=d(06032007)
replace TOM = 1 if date >=d(30032007) & date <=d(05042007)
replace TOM = 1 if date >=d(30042007) & date <=d(04052007)
replace TOM = 1 if date >=d(31052007) & date <=d(06062007)
replace TOM = 1 if date >=d(29062007) & date <=d(06072007)
replace TOM = 1 if date >=d(31072007) & date <=d(06082007)
replace TOM = 1 if date >=d(31082007) & date <=d(07092007)
replace TOM = 1 if date >=d(28092007) & date <=d(04102007)
replace TOM = 1 if date >=d(31102007) & date <=d(06112007)
replace TOM = 1 if date >=d(30112007) & date <=d(06122007)

*****************************************************************
*** TOM effect dummy 2008
      ****
*****************************************************************

replace TOM = 1 if date >=d(31122007) & date <=d(07012008)
replace TOM = 1 if date >=d(31012008) & date <=d(06022008)
```

```
replace TOM = 1 if date >=d(29022008) & date <=d(06032008)
replace TOM = 1 if date >=d(31032008) & date <=d(04042008)
replace TOM = 1 if date >=d(30042008) & date <=d(06052008)
replace TOM = 1 if date >=d(30052008) & date <=d(05062008)
replace TOM = 1 if date >=d(30062008) & date <=d(07072008)
replace TOM = 1 if date >=d(31072008) & date <=d(06082008)
replace TOM = 1 if date >=d(29082008) & date <=d(05092008)
replace TOM = 1 if date >=d(30092008) & date <=d(06102008)
replace TOM = 1 if date >=d(31102008) & date <=d(06112008)
replace TOM = 1 if date >=d(28112008) & date <=d(04122008)

*****************************************************************
*** TOM effect dummy 2009
      ****
*****************************************************************

replace TOM = 1 if date >=d(31122008) & date <=d(07012009)
replace TOM = 1 if date >=d(30012009) & date <=d(05022009)
replace TOM = 1 if date >=d(27022009) & date <=d(05032009)
replace TOM = 1 if date >=d(31032009) & date <=d(06042009)
replace TOM = 1 if date >=d(29042009) & date <=d(06052009)
replace TOM = 1 if date >=d(31052009) & date <=d(04062009)
replace TOM = 1 if date >=d(30062009) & date <=d(07072009)
replace TOM = 1 if date >=d(31072009) & date <=d(06082009)
replace TOM = 1 if date >=d(31082009) & date <=d(04092009)
replace TOM = 1 if date >=d(30092009) & date <=d(06102009)
replace TOM = 1 if date >=d(30102009) & date <=d(05112009)
replace TOM = 1 if date >=d(30112009) & date <=d(04122009)

*****************************************************************
*** TOM  effect dummy 2010
      ****
*****************************************************************

replace TOM = 1 if date >=d(31122009) & date <=d(07012010)
replace TOM = 1 if date >=d(29012010) & date <=d(04022010)
replace TOM = 1 if date >=d(26022010) & date <=d(04032010)
replace TOM = 1 if date >=d(31032010) & date <=d(06042010)
replace TOM = 1 if date >=d(30042010) & date <=d(06052010)
replace TOM = 1 if date >=d(28052010) & date <=d(04062010)
replace TOM = 1 if date >=d(30062010) & date <=d(07072010)
replace TOM = 1 if date >=d(30072010) & date <=d(05082010)
replace TOM = 1 if date >=d(31082010) & date <=d(07092010)
replace TOM = 1 if date >=d(30092010) & date <=d(06102010)
replace TOM = 1 if date >=d(29102010) & date <=d(04112010)
replace TOM = 1 if date >=d(30112010) & date <=d(06122010)
replace TOM = 1 if date >=d(30122010)


*****************************************************************
*** Holliday dummy New Year's Day                              ****
*****************************************************************

replace Holiday = 1 if date ==d(31121996)
replace Holiday = 1 if date ==d(31121997)
replace Holiday = 1 if date ==d(31121998)
replace Holiday = 1 if date ==d(31121999)
```

```
replace Holiday = 1 if date ==d(29122000)
replace Holiday = 1 if date ==d(31122001)
replace Holiday = 1 if date ==d(31122002)
replace Holiday = 1 if date ==d(31122003)
replace Holiday = 1 if date ==d(31122004)
replace Holiday = 1 if date ==d(30122005)
replace Holiday = 1 if date ==d(29122006)
replace Holiday = 1 if date ==d(31122007)
replace Holiday = 1 if date ==d(31122008)
replace Holiday = 1 if date ==d(31122009)
replace Holiday = 1 if date ==d(31122010)

*****************************************************************
*** Holliday dummy President Day                                ****
*****************************************************************

replace Holiday = 1 if date ==d(16011996)
replace Holiday = 1 if date ==d(14011997)
replace Holiday = 1 if date ==d(13011998)
replace Holiday = 1 if date ==d(12011999)
replace Holiday = 1 if date ==d(18012000)
replace Holiday = 1 if date ==d(16012001)
replace Holiday = 1 if date ==d(15012002)
replace Holiday = 1 if date ==d(14012003)
replace Holiday = 1 if date ==d(13012004)
replace Holiday = 1 if date ==d(18012005)
replace Holiday = 1 if date ==d(17012006)
replace Holiday = 1 if date ==d(16012007)
replace Holiday = 1 if date ==d(15012008)
replace Holiday = 1 if date ==d(13012009)
replace Holiday = 1 if date ==d(12012010)

*****************************************************************
*** Holliday dummy Good Friday
      ****
*****************************************************************

replace Holiday = 1 if date ==d(04041996)
replace Holiday = 1 if date ==d(27031997)
replace Holiday = 1 if date ==d(09041998)
replace Holiday = 1 if date ==d(01041999)
replace Holiday = 1 if date ==d(20042000)
replace Holiday = 1 if date ==d(12042001)
replace Holiday = 1 if date ==d(28032002)
replace Holiday = 1 if date ==d(17042003)
replace Holiday = 1 if date ==d(08042004)
replace Holiday = 1 if date ==d(24032005)
replace Holiday = 1 if date ==d(13042006)
replace Holiday = 1 if date ==d(05042007)
replace Holiday = 1 if date ==d(20032008)
replace Holiday = 1 if date ==d(09042009)
replace Holiday = 1 if date ==d(01042010)

*****************************************************************
*** Holliday dummy Memorial Day
      ****
*****************************************************************
```

```
replace Holiday = 1 if date ==d(24051996)
replace Holiday = 1 if date ==d(23051997)
replace Holiday = 1 if date ==d(22051998)
replace Holiday = 1 if date ==d(28051999)
replace Holiday = 1 if date ==d(26052000)
replace Holiday = 1 if date ==d(25052001)
replace Holiday = 1 if date ==d(24052002)
replace Holiday = 1 if date ==d(23052003)
replace Holiday = 1 if date ==d(28052004)
replace Holiday = 1 if date ==d(27052005)
replace Holiday = 1 if date ==d(26052006)
replace Holiday = 1 if date ==d(25052007)
replace Holiday = 1 if date ==d(23052008)
replace Holiday = 1 if date ==d(22052009)
replace Holiday = 1 if date ==d(28052010)

*****************************************************************
*** Holliday dummy July the 4
      ****
*****************************************************************

replace Holiday = 1 if date ==d(03071996)
replace Holiday = 1 if date ==d(03071997)
replace Holiday = 1 if date ==d(02071998)
replace Holiday = 1 if date ==d(02071999)
replace Holiday = 1 if date ==d(03072000)
replace Holiday = 1 if date ==d(03072001)
replace Holiday = 1 if date ==d(03072002)
replace Holiday = 1 if date ==d(03072003)
replace Holiday = 1 if date ==d(02072004)
replace Holiday = 1 if date ==d(01072005)
replace Holiday = 1 if date ==d(03072006)
replace Holiday = 1 if date ==d(03072007)
replace Holiday = 1 if date ==d(03072008)
replace Holiday = 1 if date ==d(02072009)
replace Holiday = 1 if date ==d(02072010)

*****************************************************************
*** Holliday dummy Labor Day                                  ****
*****************************************************************

replace Holiday = 1 if date ==d(30081996)
replace Holiday = 1 if date ==d(29081997)
replace Holiday = 1 if date ==d(04091998)
replace Holiday = 1 if date ==d(03091999)
replace Holiday = 1 if date ==d(01092000)
replace Holiday = 1 if date ==d(31082001)
replace Holiday = 1 if date ==d(30082002)
replace Holiday = 1 if date ==d(29082003)
replace Holiday = 1 if date ==d(03092004)
replace Holiday = 1 if date ==d(02092005)
replace Holiday = 1 if date ==d(01092006)
replace Holiday = 1 if date ==d(31082007)
replace Holiday = 1 if date ==d(29082008)
replace Holiday = 1 if date ==d(04092009)
replace Holiday = 1 if date ==d(03092010)
```

```
*****************************************************************
*** Holliday dummy Thanksgiving
      ****
*****************************************************************

replace Holiday = 1 if date ==d(27111996)
replace Holiday = 1 if date ==d(26111997)
replace Holiday = 1 if date ==d(25111998)
replace Holiday = 1 if date ==d(24111999)
replace Holiday = 1 if date ==d(22112000)
replace Holiday = 1 if date ==d(21112001)
replace Holiday = 1 if date ==d(27112002)
replace Holiday = 1 if date ==d(26112003)
replace Holiday = 1 if date ==d(24112004)
replace Holiday = 1 if date ==d(23112005)
replace Holiday = 1 if date ==d(22112006)
replace Holiday = 1 if date ==d(21112007)
replace Holiday = 1 if date ==d(26112008)
replace Holiday = 1 if date ==d(25112009)
replace Holiday = 1 if date ==d(24112010)


*****************************************************************
*** Holliday dummy Christmas
      ****
*****************************************************************

replace Holiday = 1 if date ==d(24121996)
replace Holiday = 1 if date ==d(24121997)
replace Holiday = 1 if date ==d(24121998)
replace Holiday = 1 if date ==d(23121999)
replace Holiday = 1 if date ==d(22122000)
replace Holiday = 1 if date ==d(24122001)
replace Holiday = 1 if date ==d(24122002)
replace Holiday = 1 if date ==d(24122003)
replace Holiday = 1 if date ==d(23122004)
replace Holiday = 1 if date ==d(23122005)
replace Holiday = 1 if date ==d(22122006)
replace Holiday = 1 if date ==d(21122007)
replace Holiday = 1 if date ==d(24122008)
replace Holiday = 1 if date ==d(24122009)
*****************************************************************
*** dummy Tripple Witch                                      ****
*****************************************************************

replace triplewitch = 1 if date == d(15031996)
replace triplewitch = 1 if date == d(21061996)
replace triplewitch = 1 if date == d(20091996)
replace triplewitch = 1 if date == d(20121996)

replace triplewitch = 1 if date == d(21031997)
replace triplewitch = 1 if date == d(20061997)
replace triplewitch = 1 if date == d(19091997)
replace triplewitch = 1 if date == d(19121997)
```

```
replace triplewitch = 1 if date == d(20031998)
replace triplewitch = 1 if date == d(19061998)
replace triplewitch = 1 if date == d(18091998)
replace triplewitch = 1 if date == d(18121998)

replace triplewitch = 1 if date == d(19031999)
replace triplewitch = 1 if date == d(18061999)
replace triplewitch = 1 if date == d(17091999)
replace triplewitch = 1 if date == d(17121999)

replace triplewitch = 1 if date == d(17032000)
replace triplewitch = 1 if date == d(16062000)
replace triplewitch = 1 if date == d(15092000)
replace triplewitch = 1 if date == d(15122000)

replace triplewitch = 1 if date == d(16032001)
replace triplewitch = 1 if date == d(15062001)
replace triplewitch = 1 if date == d(21092001)
replace triplewitch = 1 if date == d(21122001)

replace triplewitch = 1 if date == d(15032002)
replace triplewitch = 1 if date == d(21062002)
replace triplewitch = 1 if date == d(20092002)
replace triplewitch = 1 if date == d(20122002)

replace triplewitch = 1 if date == d(21032003)
replace triplewitch = 1 if date == d(20062003)
replace triplewitch = 1 if date == d(19092003)
replace triplewitch = 1 if date == d(19122003)

replace triplewitch = 1 if date == d(19032004)
replace triplewitch = 1 if date == d(18062004)
replace triplewitch = 1 if date == d(17092004)
replace triplewitch = 1 if date == d(17122004)

replace triplewitch = 1 if date == d(18032005)
replace triplewitch = 1 if date == d(17062005)
replace triplewitch = 1 if date == d(16092005)
replace triplewitch = 1 if date == d(16122005)

replace triplewitch = 1 if date == d(17032006)
replace triplewitch = 1 if date == d(16062006)
replace triplewitch = 1 if date == d(15092006)
replace triplewitch = 1 if date == d(15122006)

replace triplewitch = 1 if date == d(16032007)
replace triplewitch = 1 if date == d(15062007)
replace triplewitch = 1 if date == d(21092007)
replace triplewitch = 1 if date == d(21122007)

replace triplewitch = 1 if date == d(21032008)
replace triplewitch = 1 if date == d(20062008)
replace triplewitch = 1 if date == d(19092008)
replace triplewitch = 1 if date == d(19122008)

replace triplewitch = 1 if date == d(20032009)
replace triplewitch = 1 if date == d(19062009)
```

```
replace triplewitch = 1 if date == d(18092009)
replace triplewitch = 1 if date == d(18122009)

replace triplewitch = 1 if date == d(19032010)
replace triplewitch = 1 if date == d(18062010)
replace triplewitch = 1 if date == d(17092010)
replace triplewitch = 1 if date == d(17122010)



*************************************************************
*** Holliday dummy Monday Effect                                  ****
*************************************************************
forvalues i = 13156(7)18564 {
      replace Monday = 1 if date == `i'
      }
```

Appendix 11K: Regression

```
local dummies " halloween Jan Jan2 TOM Holiday Monday triplewitch"

set more off


foreach d in `dummies' {
      xtreg   delta_excess_return `d', fe vce(cluster date)nonest
}

preserve

drop if ncp_flag ==2

foreach d in `dummies' {
      xtreg   delta_excess_return `d', fe vce(cluster date)nonest
}

restore
preserve

drop if ncp_flag ==1

foreach d in `dummies' {
      xtreg   delta_excess_return `d', fe vce(cluster date)nonest
}
```

Appendix 11L: Interaction Dummies

```
set more off

rename Jan january
rename Jan2 january2
rename TOM turnofthemonth
rename Holiday holiday
rename Monday monday

generate hal = 0
```

```
replace hal = 1 if halloween == 1 &  january == 0 & turnofthemonth == 0 &
holiday == 0 & monday == 0 & triplewitch == 0

generate hol = 0
replace hol = 1 if halloween == 0 &  january == 0 & turnofthemonth == 0 &
holiday == 1 & monday == 0 & triplewitch == 0

generate mon = 0
replace mon = 1 if halloween == 0 &  january == 0 & turnofthemonth == 0 &
holiday == 0 & monday == 1 & triplewitch == 0

generate tom = 0
replace tom = 1 if halloween == 0 &  january == 0 & turnofthemonth == 1 &
holiday == 0 & monday == 0 & triplewitch == 0

generate triple = 0
replace triple = 1 if halloween == 0 &  january == 0 & turnofthemonth ==
0 & holiday == 0 & monday == 0 & triplewitch == 1

generate halhol = 0
replace halhol   = 1 if halloween == 1 & holiday == 1 & monday == 0 &
january == 0 & turnofthemonth == 0 & triplewitch == 0

generate halmon = 0
replace halmon   = 1 if halloween == 1 & holiday == 0 & monday == 1 &
january == 0 & turnofthemonth == 0 & triplewitch == 0

generate haljan = 0
replace haljan   = 1 if halloween == 1 & holiday == 0 & monday == 0 &
january == 1 & turnofthemonth == 0 & triplewitch == 0

generate haltom = 0
replace haltom   = 1 if halloween == 1 & holiday == 0 & monday == 0 &
january == 0 & turnofthemonth == 1 & triplewitch == 0

generate haltriple = 0
replace haltriple = 1 if halloween == 1 & holiday == 0 & monday == 0 &
january == 0 & turnofthemonth == 0 & triplewitch == 1

generate holtom = 0
replace holtom   = 1 if halloween == 0 & holiday == 1 & monday == 0 &
january == 0 & turnofthemonth == 1 & triplewitch == 0


generate montom = 0
replace montom = 1 if halloween == 0 & holiday == 0 & monday == 1 &
january == 0 & turnofthemonth == 1 & triplewitch == 0

compress
generate halholmon = 0
replace halholmon      = 1 if halloween == 1 & holiday == 1 & monday == 1
& january == 0 & turnofthemonth == 0 & triplewitch == 0

generate halholjan = 0
replace halholjan = 1 if halloween == 1 & holiday == 1 & monday == 0 &
january == 1 & turnofthemonth == 0 & triplewitch == 0
```

```
generate halholtom = 0
replace halholtom      = 1 if halloween == 1 & holiday == 1 & monday == 0
& january == 0 & turnofthemonth == 1 & triplewitch == 0

generate halholtriple = 0
replace halholtriple = 1 if halloween == 1 & holiday == 1 & monday == 0 &
january == 0 & turnofthemonth == 0 & triplewitch == 1

generate halmonjan = 0
replace halmonjan = 1 if halloween == 1 & holiday == 0 & monday == 1 &
january == 1 & turnofthemonth == 0 & triplewitch == 0

generate halmontom = 0
replace halmontom = 1 if halloween == 1 & holiday == 0 & monday == 1 &
january == 0 & turnofthemonth == 1 & triplewitch == 0


generate haljantom = 0
replace haljantom = 1 if halloween == 1 & holiday == 0 & monday == 0 &
january == 1 & turnofthemonth == 1 & triplewitch == 0

compress

generate halholmontom = 0
replace halholmontom = 1 if halloween == 1 & holiday == 1 & monday == 1 &
january == 0 & turnofthemonth == 1 & triplewitch == 0

generate halmonjantom = 0
replace halmonjantom = 1 if halloween == 1 & holiday == 0 & monday == 1 &
january == 1 & turnofthemonth == 1 & triplewitch == 0

compress
```

Appendix 11N: VW Regression

```
local dummies " halloween Jan Jan2 TOM Holiday Monday triplewitch"

set more off

generate vw = open_interest *price

foreach d in `dummies' {
      areg delta_excess_return `d' [weight = vw],absorb(optioned)
vce(cluster date)
}

preserve

drop if ncp_flag ==2

foreach d in `dummies' {
      areg delta_excess_return `d' [weight = vw],absorb(optioned)
vce(cluster date) }

restore
preserve
```

```
drop if ncp_flag ==1

foreach d in `dummies' {
      areg delta_excess_return `d' [weight = vw],absorb(optioned)
vce(cluster date)
}
```

Appendix 11N: S&P 100

```
clear

use "E:\Data New\Dummy\1996dummy.dta", clear

      generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="01381710" |   cusip =="02553710" |   cusip =="02581610" |   cusip
=="05430310" |   cusip =="05722410" |   cusip =="07181310" |   cusip
=="92343V10" |   cusip =="09702310" |   cusip =="11012210" |   cusip
=="19121610" |   cusip =="19416210" |   cusip =="25468710" |   cusip
=="26054310" |   cusip =="26353410" |   cusip =="31428X10" |   cusip
=="34537086" |   cusip =="36955010" |   cusip =="36960410" |   cusip
=="40621610" |   cusip =="42307410" |   cusip =="42823610" |   cusip
=="45920010" |   cusip =="47816010" |   cusip =="58013510" |   cusip
=="58933Y10" |   cusip =="88579Y10" |   cusip =="65584410" |   cusip
=="71344810" |   cusip =="75511150" |   cusip =="80685710" |   cusip
=="84258710" |   cusip =="88250810" |   cusip =="91301710" |   cusip
=="96945710" |   cusip =="13985910" |   cusip =="32054810" |   cusip
=="39144210" |   cusip =="25384910" |   cusip =="55267310" |   cusip
=="06605010" |   cusip =="31945A10" |   cusip =="17399Z93" |   cusip
=="03190510" |   cusip =="03189710" |   cusip =="03095410" |   cusip
=="41387510" |   cusip =="60705910" |   cusip =="43850610" |   cusip
=="71694110" |   cusip =="04882510" |   cusip =="15852510" |   cusip
=="56123210" |   cusip =="90491110" |   cusip =="08750910"
      generate tag2 = 0
            replace tag2  = 1 if cusip =="73109510" |     cusip
=="11704310" |   cusip =="15699X92" |   cusip =="34341210" |   cusip
=="43761410" |   cusip =="45950610" |   cusip =="81235010" |   cusip
=="67459910" |   cusip =="87913110" |   cusip =="75127730" |   cusip
=="65656850" |   cusip =="71713U10" |   cusip =="81238710" |   cusip
=="89233593" |   cusip =="24736170" |   cusip =="30231G10" |   cusip
=="93114210" |   cusip =="96216610" |   cusip =="98412110" |   cusip
=="00195750" |   cusip =="90921430" |   cusip =="75043810" |   cusip
=="67622P10" |   cusip =="09179710" |   cusip =="27746110" |   cusip
=="20536310" |   cusip =="63764010" |   cusip =="53271610" |   cusip
=="01741R10" |   cusip =="77390310" |   cusip =="37045V10" |   cusip
=="02687478" |   cusip =="12550910" |   cusip =="46014610" |   cusip
=="59099Z00" |   cusip =="12189T10" |   cusip =="29364G10" |   cusip
=="41399X93" |   cusip =="28335P00" |   cusip =="02635110" |   cusip
=="45814010" |   cusip =="57777810" |   cusip =="17119610" |   cusip
=="32019510" |   cusip =="67076810" |   cusip =="41651510" |   cusip
=="40412C10" |   cusip =="68389X10" |   cusip =="17275R10"

      keep if tag1 ==1 | tag2 == 1

drop tag1 tag2
save "E:\Data New\S&P100\1996S&P100.dta", replace
```

```
***
use "E:\Data New\Dummy\1997dummy.dta", clear

      generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="39144210" |   cusip =="25384910" |   cusip =="55267310" |   cusip
=="06605010" |   cusip =="31945A10" |   cusip =="17399Z93" |   cusip
=="03190510" |   cusip =="03189710" |   cusip =="03095410" |   cusip
=="41387510" |   cusip =="60705910" |   cusip =="43850610" |   cusip
=="71694110" |   cusip =="04882510" |   cusip =="15852510" |   cusip
=="56123210" |   cusip =="90491110"

      generate tag2 = 0
            replace tag2 = 1 if cusip =="087509105" |     cusip
=="731095105" |  cusip =="08750910" |   cusip =="73109510" |   cusip
=="11704310" |   cusip =="15699X92" |   cusip =="34341210" |   cusip
=="43761410" |   cusip =="45950610" |   cusip =="81235010" |   cusip
=="67459910" |   cusip =="87913110" |   cusip =="75127730" |   cusip
=="65656850" |   cusip =="71713U10" |   cusip =="81238710" |   cusip
=="89233593" |   cusip =="24736170" |   cusip =="30231G10" |   cusip
=="93114210" |   cusip =="96216610" |   cusip =="98412110" |   cusip
=="00195750" |   cusip =="90921430" |   cusip =="75043810" |   cusip
=="67622P10" |   cusip =="09179710" |   cusip =="27746110" |   cusip
=="20536310" |   cusip =="63764010" |   cusip =="53271610" |   cusip
=="01741R10" |   cusip =="77390310" |   cusip =="37045V10" |   cusip
=="02687478" |   cusip =="12550910" |   cusip =="46014610" |   cusip
=="59099Z00" |   cusip =="12189T10" |   cusip =="29364G10" |   cusip
=="41399X93" |   cusip =="28335P00" |   cusip =="02635110" |   cusip
=="45814010" |   cusip =="57777810" |   cusip =="17119610" |   cusip
=="67076810" |   cusip =="41651510" |   cusip =="40412C10" |   cusip
=="68389X10" |   cusip =="17275R10" |   cusip =="06050510"

      generate tag3 = 0
            replace tag3 = 1 if cusip =="59491810"

      keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3
save "E:\Data New\S&P100\1997S&P100.dta", replace


***
use "E:\Data New\Dummy\1998dummy.dta", clear

generate tag1 = 0
```

```
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |    cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |    cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |    cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |    cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |    cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |    cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |    cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |    cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |    cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |    cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |    cusip =="91301710" |   cusip =="96945710" |   cusip
=="25384910" |    cusip =="55267310" |   cusip =="06605010" |   cusip
=="31945A10" |    cusip =="17399Z93" |   cusip =="03190510" |   cusip
=="03189710" |    cusip =="03095410" |   cusip =="41387510" |   cusip
=="60705910" |    cusip =="43850610" |   cusip =="71694110" |   cusip
=="04882510" |    cusip =="15852510" |   cusip =="56123210" |   cusip
=="90491110" |    cusip =="08750910"

      generate tag2 = 0
            replace tag2 = 1 if cusip == "73109510" |     cusip
=="73109510" |    cusip == "73109510" |  cusip =="73109510" |   cusip
=="11704310" |    cusip =="15699X92" |   cusip =="34341210" |   cusip
=="43761410" |    cusip =="45950610" |   cusip =="81235010" |   cusip
=="67459910" |    cusip =="87913110" |   cusip =="75127730" |   cusip
=="65656850" |    cusip =="71713U10" |   cusip =="81238710" |   cusip
=="89233593" |    cusip =="24736170" |   cusip =="30231G10" |   cusip
=="93114210" |    cusip =="96216610" |   cusip =="98412110" |   cusip
=="00195750" |    cusip =="90921430" |   cusip =="75043810" |   cusip
=="67622P10" |    cusip =="09179710" |   cusip =="27746110" |   cusip
=="20536310" |    cusip =="63764010" |   cusip =="53271610" |   cusip
=="01741R10" |    cusip =="77390310" |   cusip =="37045V10" |   cusip
=="02687478" |    cusip =="12550910" |   cusip =="46014610" |   cusip
=="59099Z00" |    cusip =="12189T10" |   cusip =="29364G10" |   cusip
=="41399X93" |    cusip =="28335P00" |   cusip =="02635110" |   cusip
=="45814010" |    cusip =="57777810" |   cusip =="17119610" |   cusip
=="41651510" |    cusip =="40412C10" |   cusip =="68389X10" |   cusip
=="17275R10" |    cusip =="06050510" |   cusip =="59491810" |   cusip
=="74271810"
      generate tag3 = 0
                  replace tag3 = 1 if cusip =="06423A10" |      cusip ==
"06423A10" |     cusip == "06423A10" |  cusip =="06423A10" |   cusip
=="13442910" |    cusip =="90297310" |   cusip =="17296710" |   cusip
=="12490K10"

      keep if tag1 ==1 | tag2 == 1 | tag3 == 1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\1998S&P100.dta", replace


***
use  "E:\Data New\Dummy\1999dummy.dta", clear

                  generate tag1 = 0
```

```
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="03189710" |   cusip =="03095410" |   cusip =="41387510" |   cusip
=="60705910" |   cusip =="43850610" |   cusip =="71694110" |   cusip
=="04882510" |   cusip =="15852510" |   cusip =="56123210" |   cusip
=="90491110" |   cusip =="08750910" |   cusip =="73109510" |   cusip
=="11704310" |   cusip =="15699X92" |   cusip =="34341210" |   cusip
=="43761410" |   cusip =="45950610"
      generate tag2 = 0
            replace tag2 = 1 if cusip =="81235010" |      cusip
=="67459910" |   cusip =="87913110" |   cusip =="75127730" |   cusip
=="65656850" |   cusip =="71713U10" |   cusip =="81238710" |   cusip
=="89233593" |   cusip =="24736170" |   cusip =="30231G10" |   cusip
=="93114210" |   cusip =="96216610" |   cusip =="98412110" |   cusip
=="00195750" |   cusip =="90921430" |   cusip =="75043810" |   cusip
=="67622P10" |   cusip =="09179710" |   cusip =="27746110" |   cusip
=="20536310" |   cusip =="63764010" |   cusip =="53271610" |   cusip
=="01741R10" |   cusip =="77390310" |   cusip =="37045V10" |   cusip
=="02687478" |   cusip =="12550910" |   cusip =="46014610" |   cusip
=="59099Z00" |   cusip =="12189T10" |   cusip =="29364G10" |   cusip
=="41399X93" |   cusip =="28335P00" |   cusip =="02635110" |   cusip
=="45814010" |   cusip =="57777810" |   cusip =="41651510" |   cusip
=="40412C10" |   cusip =="68389X10" |   cusip =="17275R10" |   cusip
=="06050510" |   cusip =="59491810" |   cusip =="74271810" |   cusip
=="06423A10" |   cusip =="13442910" |   cusip =="90297310" |   cusip
=="17296710" |   cusip =="12490K10" |   cusip =="94974610" |   cusip
=="54946310"
      generate tag3 = 0
            replace tag3 = 1 if cusip =="43707610" |      cusip
=="03116210" |   cusip =="80311110" |   cusip =="43851610"


      keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\1999S&P100.dta", replace


***
use "E:\Data New\Dummy\2000dummy.dta", clear


                     generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
```

```
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="71694110" |   cusip =="04882510" |   cusip =="15852510" |   cusip
=="56123210" |   cusip =="90491110" |   cusip =="08750910" |   cusip
=="73109510" |   cusip =="11704310" |   cusip =="15699X92" |   cusip
=="34341210" |   cusip =="43761410" |   cusip =="45950610" |   cusip
=="81235010" |   cusip =="67459910" |   cusip =="87913110" |   cusip
=="75127730" |   cusip =="65656850"

      generate tag2 = 0
            replace tag2 = 1 if cusip =="71713U10" |      cusip
=="71713U10" |   cusip =="81238710" |   cusip =="89233593" |   cusip
=="24736170" |   cusip =="30231G10" |   cusip =="93114210" |   cusip
=="96216610" |   cusip =="98412110" |   cusip =="00195750" |   cusip
=="90921430" |   cusip =="75043810" |   cusip =="67622P10" |   cusip
=="09179710" |   cusip =="27746110" |   cusip =="20536310" |   cusip
=="63764010" |   cusip =="53271610" |   cusip =="01741R10" |   cusip
=="77390310" |   cusip =="37045V10" |   cusip =="02687478" |   cusip
=="12550910" |   cusip =="46014610" |   cusip =="59099Z00" |   cusip
=="12189T10" |   cusip =="29364G10" |   cusip =="41399X93" |   cusip
=="28335P00" |   cusip =="02635110" |   cusip =="45814010" |   cusip
=="57777810" |   cusip =="41651510" |   cusip =="40412C10" |   cusip
=="68389X10" |   cusip =="17275R10" |   cusip =="06050510" |   cusip
=="59491810" |   cusip =="74271810" |   cusip =="06423A10" |   cusip
=="13442910" |   cusip =="90297310" |   cusip =="17296710" |   cusip
=="12490K10" |   cusip =="94974610" |   cusip =="54946310" |   cusip
=="43707610" |   cusip =="03116210" |   cusip =="80311110" |   cusip
=="43851610" |   cusip =="61744644"

      generate tag3 = 0
            replace tag3 = 1 if cusip =="26864810" |      cusip
=="26864810" |   cusip =="12485720" |   cusip =="88731730" |   cusip
=="H8912810" |   cusip =="30161N10" |   cusip =="29356110" |   cusip
=="37576610" |   cusip =="46625H10" |   cusip =="71708110" |   cusip
=="G3921A17" |   cusip =="65332V10" |   cusip =="58469910" |   cusip
=="12502P10" |   cusip =="52490810" |   cusip =="00130H10"


      keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\2000S&P100.dta", replace


***
use "E:\Data New\Dummy\2001dummy.dta", clear

      generate tag1 = 0
```

```
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="75127730" |   cusip =="65656850" |   cusip =="71713U10" |   cusip
=="81238710" |   cusip =="89233593" |   cusip =="24736170" |   cusip
=="30231G10" |   cusip =="93114210" |   cusip =="96216610" |   cusip
=="98412110" |   cusip =="00195750" |   cusip =="90921430" |   cusip
=="75043810" |   cusip =="67622P10" |   cusip =="09179710" |   cusip
=="27746110" |   cusip =="20536310"
      generate tag2 = 0
            replace tag2 = 1 if cusip =="63764010" |      cusip
=="53271610" |   cusip =="01741R10" |   cusip =="77390310" |   cusip
=="37045V10" |   cusip =="02687478" |   cusip =="12550910" |   cusip
=="46014610" |   cusip =="59099Z00" |   cusip =="12189T10" |   cusip
=="29364G10" |   cusip =="41399X93" |   cusip =="28335P00" |   cusip
=="02635110" |   cusip =="45814010" |   cusip =="57777810" |   cusip
=="41651510" |   cusip =="40412C10" |   cusip =="68389X10" |   cusip
=="17275R10" |   cusip =="06050510" |   cusip =="59491810" |   cusip
=="74271810" |   cusip =="06423A10" |   cusip =="13442910" |   cusip
=="90297310" |   cusip =="17296710" |   cusip =="94974610" |   cusip
=="54946310" |   cusip =="43707610" |   cusip =="03116210" |   cusip
=="80311110" |   cusip =="43851610" |   cusip =="61744644" |   cusip
=="26864810" |   cusip =="12485720" |   cusip =="88731730" |   cusip
=="H8912810" |   cusip =="30161N10" |   cusip =="29356110" |   cusip
=="37576610" |   cusip =="46625H10" |   cusip =="71708110" |   cusip
=="G3921A17" |   cusip =="65332V10" |   cusip =="58469910" |   cusip
=="12502P10" |   cusip =="52490810" |   cusip =="00130H10" |   cusip
=="28336L10"

      generate tag3 = 0
            replace tag3 = 1 if cusip =="90297330" |      cusip
=="00206R10" |   cusip =="02209S10" |   cusip =="03522910" |   cusip
=="58505510"


      keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\2001S&P100.dta", replace


***
use "E:\Data New\Dummy\2002dummy.dta", clear

      generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
```

```
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="65656850" |   cusip =="71713U10" |   cusip =="81238710" |   cusip
=="89233593" |   cusip =="24736170" |   cusip =="30231G10" |   cusip
=="93114210" |   cusip =="96216610" |   cusip =="98412110" |   cusip
=="00195750" |   cusip =="90921430" |   cusip =="75043810" |   cusip
=="67622P10" |   cusip =="09179710" |   cusip =="27746110" |   cusip
=="20536310" |   cusip =="63764010"

      generate tag2 = 0
            replace tag2 = 1 if cusip =="53271610" |      cusip
=="01741R10" |   cusip =="77390310" |   cusip =="37045V10" |   cusip
=="02687478" |   cusip =="12550910" |   cusip =="46014610" |   cusip
=="59099Z00" |   cusip =="12189T10" |   cusip =="29364G10" |   cusip
=="41399X93" |   cusip =="45814010" |   cusip =="57777810" |   cusip
=="41651510" |   cusip =="40412C10" |   cusip =="68389X10" |   cusip
=="17275R10" |   cusip =="06050510" |   cusip =="59491810" |   cusip
=="74271810" |   cusip =="06423A10" |   cusip =="13442910" |   cusip
=="17296710" |   cusip =="94974610" |   cusip =="54946310" |   cusip
=="43707610" |   cusip =="03116210" |   cusip =="80311110" |   cusip
=="43851610" |   cusip =="61744644" |   cusip =="26864810" |   cusip
=="12485720" |   cusip =="88731730" |   cusip =="H8912810" |   cusip
=="30161N10" |   cusip =="37576610" |   cusip =="46625H10" |   cusip
=="71708110" |   cusip =="65332V10" |   cusip =="58469910" |   cusip
=="12502P10" |   cusip =="52490810" |   cusip =="00130H10" |   cusip
=="28336L10" |   cusip =="90297330" |   cusip =="00206R10" |   cusip
=="02209S10" |   cusip =="03522910" |   cusip =="58505510" |   cusip
=="38141G10"


      keep if tag1 ==1 | tag2 == 1

drop tag1 tag2

save "E:\Data New\S&P100\2002S&P100.dta", replace


***
use "E:\Data New\Dummy\2003dummy.dta", clear
      generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
```

```
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="71713U10" |   cusip =="81238710" |   cusip =="89233593" |   cusip
=="24736170" |   cusip =="30231G10" |   cusip =="93114210" |   cusip
=="96216610" |   cusip =="98412110" |   cusip =="00195750" |   cusip
=="90921430" |   cusip =="75043810" |   cusip =="67622P10" |   cusip
=="09179710" |   cusip =="27746110" |   cusip =="20536310" |   cusip
=="63764010" |   cusip =="53271610"

      generate tag2 = 0
            replace tag2 = 1 if cusip =="01741R10" |      cusip
=="77390310" |   cusip =="37045V10" |   cusip =="02687478" |   cusip
=="12550910" |   cusip =="46014610" |   cusip =="59099Z00" |   cusip
=="12189T10" |   cusip =="29364G10" |   cusip =="41399X93" |   cusip
=="45814010" |   cusip =="57777810" |   cusip =="41651510" |   cusip
=="40412C10" |   cusip =="68389X10" |   cusip =="17275R10" |   cusip
=="06050510" |   cusip =="59491810" |   cusip =="74271810" |   cusip
=="06423A10" |   cusip =="13442910" |   cusip =="17296710" |   cusip
=="94974610" |   cusip =="54946310" |   cusip =="43707610" |   cusip
=="03116210" |   cusip =="80311110" |   cusip =="43851610" |   cusip
=="61744644" |   cusip =="26864810" |   cusip =="12485720" |   cusip
=="88731730" |   cusip =="H8912810" |   cusip =="30161N10" |   cusip
=="37576610" |   cusip =="46625H10" |   cusip =="71708110" |   cusip
=="65332V10" |   cusip =="58469910" |   cusip =="12502P10" |   cusip
=="52490810" |   cusip =="00130H10" |   cusip =="28336L10" |   cusip
=="90297330" |   cusip =="00206R10" |   cusip =="02209S10" |   cusip
=="03522910" |   cusip =="58505510" |   cusip =="38141G10" |   cusip
=="02000210"

      keep if tag1 ==1 | tag2 == 1

drop tag1 tag2

save "E:\Data New\S&P100\2003S&P100.dta", replace


***
use "E:\Data New\Dummy\2004dummy.dta", clear

      generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="81238710" |   cusip =="89233593" |   cusip =="24736170" |   cusip
=="30231G10" |   cusip =="93114210" |   cusip =="96216610" |   cusip
=="98412110" |   cusip =="00195750" |   cusip =="90921430" |   cusip
=="75043810" |   cusip =="67622P10" |   cusip =="09179710" |   cusip
```

```
=="27746110" |   cusip =="20536310" |   cusip =="63764010" |   cusip
=="53271610" |   cusip =="01741R10"
      generate tag2 = 0
            replace tag2 = 1 if cusip =="77390310" |      cusip
=="37045V10" |   cusip =="02687478" |   cusip =="12550910" |   cusip
=="46014610" |   cusip =="59099Z00" |   cusip =="12189T10" |   cusip
=="29364G10" |   cusip =="41399X93" |   cusip =="45814010" |   cusip
=="57777810" |   cusip =="41651510" |   cusip =="40412C10" |   cusip
=="68389X10" |   cusip =="17275R10" |   cusip =="06050510" |   cusip
=="59491810" |   cusip =="74271810" |   cusip =="06423A10" |   cusip
=="13442910" |   cusip =="17296710" |   cusip =="94974610" |   cusip
=="54946310" |   cusip =="43707610" |   cusip =="03116210" |   cusip
=="80311110" |   cusip =="43851610" |   cusip =="61744644" |   cusip
=="26864810" |   cusip =="12485720" |   cusip =="88731730" |   cusip
=="H8912810" |   cusip =="30161N10" |   cusip =="37576610" |   cusip
=="46625H10" |   cusip =="71708110" |   cusip =="65332V10" |   cusip
=="58469910" |   cusip =="12502P10" |   cusip =="52490810" |   cusip
=="00130H10" |   cusip =="28336L10" |   cusip =="90297330" |   cusip
=="00206R10" |   cusip =="02209S10" |   cusip =="03522910" |   cusip
=="58505510" |   cusip =="38141G10" |   cusip =="02000210" |   cusip
=="24702R10"

      keep if tag1 ==1 | tag2 == 1
drop tag1 tag2

save "E:\Data New\S&P100\2004S&P100.dta", replace


***
use "E:\Data New\Dummy\2005dummy.dta", clear

      generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="81238710" |   cusip =="89233593" |   cusip =="24736170" |   cusip
=="30231G10" |   cusip =="93114210" |   cusip =="96216610" |   cusip
=="98412110" |   cusip =="00195750" |   cusip =="90921430" |   cusip
=="75043810" |   cusip =="67622P10" |   cusip =="09179710" |   cusip
=="27746110" |   cusip =="20536310" |   cusip =="63764010" |   cusip
=="53271610" |   cusip =="01741R10"
      generate tag2 = 0
            replace tag2 = 1 if cusip =="77390310" |      cusip
=="37045V10" |   cusip =="02687478" |   cusip =="12550910" |   cusip
=="46014610" |   cusip =="59099Z00" |   cusip =="12189T10" |   cusip
=="29364G10" |   cusip =="41399X93" |   cusip =="45814010" |   cusip
=="57777810" |   cusip =="41651510" |   cusip =="40412C10" |   cusip
=="68389X10" |   cusip =="17275R10" |   cusip =="06050510" |   cusip
```

```
=="59491810" |   cusip =="74271810" |   cusip =="13442910" |   cusip
=="17296710" |   cusip =="94974610" |   cusip =="54946310" |   cusip
=="43707610" |   cusip =="03116210" |   cusip =="80311110" |   cusip
=="43851610" |   cusip =="61744644" |   cusip =="26864810" |   cusip
=="12485720" |   cusip =="88731730" |   cusip =="H8912810" |   cusip
=="30161N10" |   cusip =="37576610" |   cusip =="46625H10" |   cusip
=="71708110" |   cusip =="65332V10" |   cusip =="58469910" |   cusip
=="12502P10" |   cusip =="52490810" |   cusip =="00130H10" |   cusip
=="28336L10" |   cusip =="90297330" |   cusip =="00206R10" |   cusip
=="02209S10" |   cusip =="03522910" |   cusip =="58505510" |   cusip
=="38141G10" |   cusip =="02000210" |   cusip =="24702R10" |   cusip
=="20030N10"
      generate tag3 = 0
            replace tag3 = 1 if cusip =="87612E10" |      cusip
=="85206110" |   cusip =="14912310" |   cusip =="30231G10" |   cusip
=="93114210" |   cusip =="96216610" |   cusip =="98412110" |   cusip
=="00282410" |   cusip =="16676410" |   cusip =="91131210"


      keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\2005S&P100.dta", replace



***
use "E:\Data New\Dummy\2006dummy.dta", clear

generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="90921430" |   cusip =="75043810" |   cusip =="67622P10" |   cusip
=="09179710" |   cusip =="27746110" |   cusip =="20536310" |   cusip
=="63764010" |   cusip =="53271610" |   cusip =="01741R10" |   cusip
=="77390310" |   cusip =="37045V10" |   cusip =="02687478" |   cusip
=="12550910" |   cusip =="46014610" |   cusip =="59099Z00" |   cusip
=="12189T10" |   cusip =="29364G10"

      generate tag2 = 0
            replace tag2 = 1 if cusip =="41399X93" |      cusip
=="45814010" |   cusip =="41651510" |   cusip =="40412C10" |   cusip
=="68389X10" |   cusip =="17275R10" |   cusip =="06050510" |   cusip
=="59491810" |   cusip =="74271810" |   cusip =="13442910" |   cusip
=="17296710" |   cusip =="94974610" |   cusip =="54946310" |   cusip
=="43707610" |   cusip =="03116210" |   cusip =="80311110" |   cusip
```

```
=="43851610" |   cusip =="61744644" |   cusip =="26864810" |   cusip
=="12485720" |   cusip =="88731730" |   cusip =="H8912810" |   cusip
=="30161N10" |   cusip =="46625H10" |   cusip =="71708110" |   cusip
=="58469910" |   cusip =="12502P10" |   cusip =="52490810" |   cusip
=="00130H10" |   cusip =="28336L10" |   cusip =="90297330" |   cusip
=="00206R10" |   cusip =="02209S10" |   cusip =="03522910" |   cusip
=="58505510" |   cusip =="38141G10" |   cusip =="02000210" |   cusip
=="24702R10" |   cusip =="20030N10" |   cusip =="87612E10" |   cusip
=="85206110" |   cusip =="14912310" |   cusip =="30231G10" |   cusip
=="93114210" |   cusip =="96216610" |   cusip =="98412110" |   cusip
=="00282410" |   cusip =="16676410" |   cusip =="91131210" |   cusip
=="92990310"

      generate tag3 = 0
            replace tag3 = 1 if cusip =="7591EP10" |      cusip
=="38259P50" |   cusip =="20825C10" |   cusip =="14040H10"


      keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\2006S&P100.dta", replace


***
use "E:\Data New\Dummy\2007dummy.dta",clear


                      generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="09179710" |   cusip =="27746110" |   cusip =="20536310" |   cusip
=="63764010" |   cusip =="53271610" |   cusip =="01741R10" |   cusip
=="77390310" |   cusip =="37045V10" |   cusip =="02687478" |   cusip
=="12550910" |   cusip =="46014610" |   cusip =="59099Z00" |   cusip
=="12189T10" |   cusip =="29364G10" |   cusip =="41399X93" |   cusip
=="45814010" |   cusip =="41651510"

      generate tag2 = 0
            replace tag2 = 1 if cusip =="68389X10" |      cusip
=="17275R10" |   cusip =="06050510" |   cusip =="59491810" |   cusip
=="74271810" |   cusip =="13442910" |   cusip =="17296710" |   cusip
=="94974610" |   cusip =="43707610" |   cusip =="03116210" |   cusip
=="80311110" |   cusip =="43851610" |   cusip =="61744644" |   cusip
=="26864810" |   cusip =="12485720" |   cusip =="88731730" |   cusip
=="H8912810" |   cusip =="30161N10" |   cusip =="46625H10" |   cusip
```

```
=="71708110" |   cusip =="58469910" |   cusip =="12502P10" |   cusip
=="52490810" |   cusip =="00130H10" |   cusip =="28336L10" |   cusip
=="90297330" |   cusip =="00206R10" |   cusip =="02209S10" |   cusip
=="03522910" |   cusip =="58505510" |   cusip =="38141G10" |   cusip
=="02000210" |   cusip =="24702R10" |   cusip =="20030N10" |   cusip
=="87612E10" |   cusip =="85206110" |   cusip =="14912310" |   cusip
=="30231G10" |   cusip =="93114210" |   cusip =="96216610" |   cusip
=="98412110" |   cusip =="00282410" |   cusip =="16676410" |   cusip
=="91131210" |   cusip =="92990310" |   cusip =="7591EP10" |   cusip
=="38259P50" |   cusip =="20825C10" |   cusip =="14040H10" |   cusip
=="12665010"

      generate tag3 = 0
            replace tag3 = 1 if cusip =="50075N10" |      cusip
=="03783310" |   cusip =="06405810" |   cusip =="G2554F11" |   cusip
=="62949110"


      keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\2007S&P100.dta", replace


***
use  "E:\Data New\Dummy\2008dummy.dta", clear


                        generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="01741R10" |   cusip =="77390310" |   cusip =="37045V10" |   cusip
=="02687478" |   cusip =="12550910" |   cusip =="46014610" |   cusip
=="59099Z00" |   cusip =="12189T10" |   cusip =="29364G10" |   cusip
=="41399X93" |   cusip =="45814010" |   cusip =="41651510" |   cusip
=="68389X10" |   cusip =="17275R10" |   cusip =="06050510" |   cusip
=="59491810" |   cusip =="74271810"
      generate tag2 = 0
            replace tag2 = 1 if cusip =="13442910" |      cusip
=="17296710" |   cusip =="94974610" |   cusip =="43707610" |   cusip
=="03116210" |   cusip =="80311110" |   cusip =="43851610" |   cusip
=="61744644" |   cusip =="26864810" |   cusip =="12485720" |   cusip
=="88731730" |   cusip =="H8912810" |   cusip =="30161N10" |   cusip
=="46625H10" |   cusip =="71708110" |   cusip =="12502P10" |   cusip
=="52490810" |   cusip =="00130H10" |   cusip =="28336L10" |   cusip
=="90297330" |   cusip =="00206R10" |   cusip =="02209S10" |   cusip
```

```
=="03522910" |   cusip =="58505510" |   cusip =="38141G10" |   cusip
=="02000210" |   cusip =="24702R10" |   cusip =="20030N10" |   cusip
=="87612E10" |   cusip =="85206110" |   cusip =="14912310" |   cusip
=="30231G10" |   cusip =="93114210" |   cusip =="96216610" |   cusip
=="98412110" |   cusip =="00282410" |   cusip =="16676410" |   cusip
=="91131210" |   cusip =="92990310" |   cusip =="7591EP10" |   cusip
=="38259P50" |   cusip =="20825C10" |   cusip =="14040H10" |   cusip
=="12665010" |   cusip =="50075N10" |   cusip =="03783310" |   cusip
=="06405810" |   cusip =="G2554F11" |   cusip =="62949110" |   cusip
=="91324P10"
                generate tag3 = 0
           replace tag3 = 1 if cusip =="71817210" |      cusip
=="63707110" |   cusip =="57636Q10" |   cusip =="74752510" |   cusip
=="67459910" |   cusip =="98302410" |   cusip =="53983010" |   cusip
=="54866110" |   cusip =="65410610" |   cusip =="93142210" |   cusip
=="25179M10" |   cusip =="37555810" |   cusip =="80660510"


     keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\2008S&P100.dta", replace



***
use "E:\Data New\Dummy\2009dummy.dta", clear

     generate tag1 = 0
           replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="59099Z00" |   cusip =="12189T10" |   cusip =="29364G10" |   cusip
=="45814010" |   cusip =="68389X10" |   cusip =="17275R10" |   cusip
=="06050510" |   cusip =="59491810" |   cusip =="74271810" |   cusip
=="13442910" |   cusip =="17296710" |   cusip =="94974610" |   cusip
=="43707610" |   cusip =="03116210" |   cusip =="80311110" |   cusip
=="43851610" |   cusip =="61744644"

     generate tag2 = 0
           replace tag2 = 1 if cusip =="26864810" |      cusip
=="88731730" |   cusip =="H8912810" |   cusip =="30161N10" |   cusip
=="46625H10" |   cusip =="71708110" |   cusip =="90297330" |   cusip
=="00206R10" |   cusip =="02209S10" |   cusip =="58505510" |   cusip
=="38141G10" |   cusip =="02000210" |   cusip =="24702R10" |   cusip
=="20030N10" |   cusip =="87612E10" |   cusip =="85206110" |   cusip
=="14912310" |   cusip =="30231G10" |   cusip =="93114210" |   cusip
```

```
=="96216610" |   cusip =="98412110" |   cusip =="00282410" |   cusip
=="16676410" |   cusip =="91131210" |   cusip =="92990310" |   cusip
=="7591EP10" |   cusip =="38259P50" |   cusip =="20825C10" |   cusip
=="14040H10" |   cusip =="12665010" |   cusip =="50075N10" |   cusip
=="03783310" |   cusip =="06405810" |   cusip =="G2554F11" |   cusip
=="62949110" |   cusip =="91324P10" |   cusip =="71817210" |   cusip
=="63707110" |   cusip =="57636Q10" |   cusip =="74752510" |   cusip
=="67459910" |   cusip =="98302410" |   cusip =="53983010" |   cusip
=="54866110" |   cusip =="65410610" |   cusip =="93142210" |   cusip
=="25179M10" |   cusip =="37555810" |   cusip =="80660510" |   cusip
=="22160K10"

      generate tag3 = 0
            replace tag3 = 1 if cusip =="02313510" |      cusip
=="61166W10" |   cusip =="59156R10" |   cusip =="65248E10" |   cusip
=="35671D85"


      keep if tag1 ==1 | tag2 == 1 | tag3 ==1

drop tag1 tag2 tag3

save "E:\Data New\S&P100\2009S&P100.dta", replace


***
use "E:\Data New\Dummy\2010dummy.dta", clear

generate tag1 = 0
            replace tag1 = 1 if cusip =="01381710" |      cusip
=="02553710" |   cusip =="02581610" |   cusip =="05430310" |   cusip
=="05722410" |   cusip =="07181310" |   cusip =="92343V10" |   cusip
=="09702310" |   cusip =="11012210" |   cusip =="19121610" |   cusip
=="19416210" |   cusip =="25468710" |   cusip =="26054310" |   cusip
=="26353410" |   cusip =="31428X10" |   cusip =="34537086" |   cusip
=="36955010" |   cusip =="36960410" |   cusip =="40621610" |   cusip
=="42307410" |   cusip =="42823610" |   cusip =="45920010" |   cusip
=="47816010" |   cusip =="58013510" |   cusip =="58933Y10" |   cusip
=="88579Y10" |   cusip =="65584410" |   cusip =="71344810" |   cusip
=="75511150" |   cusip =="80685710" |   cusip =="84258710" |   cusip
=="88250810" |   cusip =="91301710" |   cusip =="96945710" |   cusip
=="12189T10" |   cusip =="29364G10" |   cusip =="45814010" |   cusip
=="68389X10" |   cusip =="17275R10" |   cusip =="06050510" |   cusip
=="59491810" |   cusip =="74271810" |   cusip =="13442910" |   cusip
=="17296710" |   cusip =="94974610" |   cusip =="43707610" |   cusip
=="03116210" |   cusip =="80311110" |   cusip =="43851610" |   cusip
=="61744644" |   cusip =="26864810"

      generate tag2 = 0
            replace tag2 = 1 if cusip =="88731730" |      cusip
=="30161N10" |   cusip =="46625H10" |   cusip =="71708110" |   cusip
=="90297330" |   cusip =="00206R10" |   cusip =="02209S10" |   cusip
=="58505510" |   cusip =="38141G10" |   cusip =="02000210" |   cusip
=="24702R10" |   cusip =="20030N10" |   cusip =="87612E10" |   cusip
=="85206110" |   cusip =="14912310" |   cusip =="30231G10" |   cusip
=="93114210" |   cusip =="96216610" |   cusip =="98412110" |   cusip
=="00282410" |   cusip =="16676410" |   cusip =="91131210" |   cusip
```

```
=="7591EP10" |   cusip =="38259P50" |   cusip =="20825C10" |   cusip
=="14040H10" |   cusip =="12665010" |   cusip =="50075N10" |   cusip
=="03783310" |   cusip =="06405810" |   cusip =="62949110" |   cusip
=="91324P10" |   cusip =="71817210" |   cusip =="63707110" |   cusip
=="57636Q10" |   cusip =="74752510" |   cusip =="67459910" |   cusip
=="53983010" |   cusip =="54866110" |   cusip =="65410610" |   cusip
=="93142210" |   cusip =="25179M10" |   cusip =="37555810" |   cusip
=="22160K10" |   cusip =="02313510" |   cusip =="61166W10" |   cusip
=="59156R10" |   cusip =="65248E10" |   cusip =="35671D85" |   cusip
=="08467070"

      keep if tag1 ==1 | tag2 == 1

drop tag1 tag2

save "E:\Data New\S&P100\2010S&P100.dta", replace


****cusip   06405810
```

Appendix 11O: Delta Bucket Regression

```
set more off
local dummies " halloween Jan Jan2 TOM Holiday Monday triplewitch"

preserve


drop if delta < 0 & ncp_flag == 1 | delta > .2 & ncp_flag == 1
drop if delta > 0 & ncp_flag == 2 | delta < -.2 & ncp_flag == 2


foreach d in `dummies' {
xtreg delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve

drop if delta < .2 & ncp_flag == 1 | delta > .4 & ncp_flag == 1
drop if delta > -.2 & ncp_flag == 2 | delta < -.4 & ncp_flag == 2


foreach d in `dummies' {
xtreg delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve

drop if delta < .4 & ncp_flag == 1 | delta > .6 & ncp_flag == 1
drop if delta > -.4 & ncp_flag == 2 | delta < -.6 & ncp_flag == 2

foreach d in `dummies' {
xtreg delta_excess_return `d', fe vce(cluster date)nonest
}
```

```
restore
preserve

drop if delta < .6 & ncp_flag == 1 | delta > .8 & ncp_flag == 1
drop if delta > -.6 & ncp_flag == 2 | delta < -.8 & ncp_flag == 2

foreach d in `dummies' {
xtreg delta_excess_return `d', fe vce(cluster date)nonest
 }

restore
preserve

drop if delta < .8 & ncp_flag == 1 | delta > 1 & ncp_flag == 1
drop if delta > -.8 & ncp_flag == 2 | delta < -1 & ncp_flag == 2

foreach d in `dummies' {
xtreg delta_excess_return `d', fe vce(cluster date)nonest
 }
```

Appendix 11P: Maturity Regression

```
set more off
local dummies " halloween Jan Jan2 TOM Holiday Monday triplewitch"

preserve

keep if daten <=10


foreach d in `dummies' {
xtreg excess_return `d', fe vce(cluster date)nonest
}

restore
preserve

drop if daten <= 10 | daten > 31


foreach d in `dummies' {
xtreg excess_return `d', fe vce(cluster date)nonest
}


restore

keep if daten >= 32

foreach d in `dummies' {
xtreg excess_return `d', fe vce(cluster date)nonest
}

use "E:\Data New\Af\Nieuwe intrest berekening\Mean\van 1 naar 10.dta",
clear
```

```
preserve

keep if daten <=10


foreach d in `dummies' {
xtreg  delta_excess_returns `d', fe vce(cluster date)nonest
}

restore
preserve

drop if daten <= 10 | daten > 31


foreach d in `dummies' {
xtreg  delta_excess_returns `d', fe vce(cluster date)nonest
}


restore

keep if daten >= 32

foreach d in `dummies' {
xtreg  delta_excess_returns `d', fe vce(cluster date)nonest
}
```

Appendix 11Q: Per Annum

```
local dummies " halloween Jan Jan2 TOM Holiday Monday triplewitch"

set more off

preserve

keep if date >= d(01011996) & date <= d(31121996)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}

restore
preserve

keep if date >= d(01011997) & date <= d(31121997)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}

restore
preserve

keep if date >= d(01011998) & date <= d(31121998)

foreach d in `dummies' {
```

```
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}

restore
preserve

keep if date >= d(01011999) & date <= d(31121999)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve

keep if date >= d(01012000) & date <= d(31122000)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve


keep if date >= d(01012001) & date <= d(31122001)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
      }
restore
preserve

      preserve

keep if date >= d(01012002) & date <= d(31122002)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve

      keep if date >= d(01012003) & date <= d(31122003)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve
      keep if date >= d(01012004) & date <= d(31122004)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve

      keep if date >= d(01012005) & date <= d(31122005)
```

```
foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve


      keep if date >= d(01012006) & date <= d(31122006)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve

      keep if date >= d(01012007) & date <= d(31122007)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve

      keep if date >= d(01012008) & date <= d(31122008)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve

      keep if date >= d(01012009) & date <= d(31122009)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
restore
preserve


keep if date >= d(01012010) & date <= d(31122010)

foreach d in `dummies' {
      xtreg  delta_excess_return `d', fe vce(cluster date)nonest
}
```

Appendix 11R: Straddle 1996 – 2010

```
local year "1996"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"
```

```
      use `year', clear
            append using `yt'
            drop T
            compress
            duplicates drop optionid date, force

cd "E:\Data New\Straddle\"
cd `year'

      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in 13196     13224 13259 13287 13322 13350 13378 13413 13441
      13469 13504 13532 {
            use `year', clear
            keep if exdate == `i'
            save `i'

                  foreach j in 50155022  5029  5036  5046  5048  5049  5058
      5061  5067  5079  5083  5086  5087  5089  5097  5115  5133  5142  5158
      5169  5176  5180  5191  5212  5218  5240  5252  5258  5273  5278  5297
      5309  5317  5319  5351  5361  5385  5399  5402  5413  5428  5459  5496
      5510  5528  5549  5563  5564  5569  5579  5587  5591  5603  5628  5644
      5646  5670  5684  5692  5693  5699  5712  5724  5764  5782  5788  5804
      5811  5817  5828  5837  5854  5882  5909  5915  5934  5944  5959  5963
      5969  5970  5973  5983  6008  6012  6021  6029  6099  6250  6572  6677
      6740  7333  7631  7737  8042  8155  8274  8285  8672  8747  9265
      100917      100930      100972      101121      101164      101199
      101201      101204      101227      101233      101273      101275
      101305      101322      101325      101368      101375      101382
      101384      101387      101397      101508      101549      101558
      101578      101594      101610      101639      101669      101697
      101795      101800      101828      101849      101920      101930
      101966      101988      102015      102021      102041      102042
      102043      102067      102082      102117      102163      102210
      102244      102265      102267      102296      102339      102349
      102381      102386      102408      102409      102525      102555
      102561      102571      102583      102588      102660      102796
      102833      102845      102927      102936      102968      103015
      103037      103042      103049      103106      103125      103157
      103202      103282      103296      103302      103313      103355
      103370      103401      103402      103404      103434      103466
      103498      103546      103574      103654      103682      103736
      103749      103772      103861      103879      103912      103920
      103932      103936      103937      103969      103979      104049
      104117      104118      104124      104144      104153      104172
      104286      104338      104355      104361      104508      104533
      104550      104560      104626      104628      104633      104635
      104664      104847      104878      104891      104893      104894
      104939      104958      104967      105003      105076      105119
      105120      105168      105169      105174      105206      105223
      105244      105327      105338 105340    105356     105379
      105384      105452      105512      105536      105550      105557
      105573      105581      105588      105669      105675      105688
      105696      105700      105730      105759      105776      105785
```

```
105833     105955     105976     105985     106045     106119
106203     106247     106276     106281     106295     106329
106553     106566     106567     106629     106638     106665
106674     106689     106713     106744     106776     106967
106969     106975     107006     107010     107015     107044
107045     107050     107078     107079     107107     107244
107255     107280     107297     107309     107317     107318
107321     107335     107398     107407     107430     107525
107544     107585     107605     107616     107691     107693
107704     107916     107951     107952     107975     108117
108120     108130     108156     108160     108161     108185
108186     108196     108231     108265     108279     108385
108406     108463     108505     108609     108617     108642
108645     108649     108670     108716     108768     108849
108888     108893     108910     108948     108960     108962
108965     108969     109036     109040     109047     109085
109118     109143     109224     109285     109297     109347
109447     109463     109497     109528     109557     109652
109678     109692     109715     109749     109752     109775
109823     109848     109866     109923     109954     109956
109970     109998     110002     110086     110117     110126
110152     110262     110337     110357     110433     110460
110576     110611     110627     110634     110649     110716
110752     110766     110810     110860     110914     110926
110945     110950     110967     110972     110979     111008
111020     111051     111054     111086     111208     111256
111283     111296     111298     111337     111381     111394
111404     111459     111469     111501     111504     111537
111652     111668     111850     111860     111861     111953
111957     112011     112015     112022     112037     112042
112136     112142     112169     112185     112286  {
```

```
use `i', clear
keep if secid ==`j'
generate min = (abs(delta) -0.5)^2
sort date ncp_flag min
drop if strike_price ~=strike_price[1]
```

```
drop if ncp_flag ==ncp_flag[_n-1]
generate T = (exdate - date)/365
generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1
```

```
                save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\1996\filelist.txt"

      file open myfile using "E:\Data New\Straddle\1996\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
save master_data, replace




local year "1997"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in 13567     13595 13623 13651 13686 13714 13742 13777 13805
      13840 13868 13896 13931 {
            use `year', clear
            keep if exdate == `i'
            save `i'

                  foreach j in 50155022  5029  5036  5046  5048  5049  5058
      5061  5067  5083  5086  5087  5089  5097  5115  5133  5142  5169  5176
      5191  5212  5240  5252  5273  5278  5297  5309  5317  5319  5385  5399
      5402  5413  5418  5428  5435  5459  5496  5510  5520  5528  5549  5564
      5569  5579  5591  5603  5628  5644  5646  5670  5692  5693  5699  5712
      5724  5764  5782  5788  5792  5804  5811  5817  5828  5837  5854  5882
      5909  5915  5934  5944  5959  5963  5969  5973  5983  6008  6012  6021
```

```
6029  6099  6572  6677  6740  7333  7631  7737  8042  8155  8274  8285
8672  8747  9265  100917      100930      100972      101062
101121      101164      101199      101201      101204      101227
101233      101273      101275      101305      101322      101325
101368      101375      101382      101384      101387      101397
101508      101533      101549      101558      101578      101580
101594      101610      101639      101669      101697      101795
101800      101806      101828      101849      101863      101920
101930      101966      101988      102015      102021      102041
102042      102043      102067      102082      102088      102117
102119      102163      102210      102244      102265      102267
102296      102339      102349      102386      102408      102409
102525      102561      102571      102583      102588      102660
102733      102796      102833      102845      102927      102936
102968      103015      103029      103037      103042      103049
103106      103125      103157      103202      103282      103296
103302      103313      103355      103370      103401      103402
103404      103434      103466      103477      103498      103546
103574      103654      103682      103736      103749      103772
103861      103879      103912      103920      103932      103936
103937      103969      103979      104049      104117      104118
104124      104144      104153      104172      104286      104338
104355      104361      104411      104508      104533      104550
104560      104626      104628      104633      104635      104664
104847      104878      104891      104893      104894      104939
104958      104967      105003      105076      105119      105120
105168      105169      105174      105206      105223      105244
105327 105338    105340      105356      105379      105384
105452      105512      105550      105557      105573      105581
105588      105615      105669      105675      105688      105696
105700      105730      105759      105776      105785      105833
105846      105955      105976      105985      106045      106119
106203      106247      106276      106281      106295      106329
106553      106566      106567      106595      106629      106638
106665      106674      106689      106713      106744      106776
106967      106969      106975      107006      107010      107015
107036      107044      107045      107050      107078      107079
107107      107244      107255      107280      107297      107309
107317      107318      107321      107335      107398      107407
107430      107525      107544      107585      107605      107616
107691      107693      107704      107916      107951      107952
107975      108117      108120      108130      108156      108160
108161      108185      108186      108196      108231      108265
108279      108385      108429      108463      108505      108573
108609      108617      108642      108645      108649      108670
108716      108756      108768      108849      108888      108893
108910      108948      108960      108962      108965      108969
109036      109047      109085      109118      109143      109224
109245      109274      109285      109297      109347      109396
109447      109463      109497      109528      109557      109652
109678      109692      109715      109724      109749      109752
109775      109823      109848      109866      109923      109954
109956      109965      109970      109998      110002      110086
110117      110126      110152      110262      110337      110357
110433      110460      110490      110576      110611      110627
110634      110649      110707      110716      110752      110766
110810      110860      110914      110926      110945      110950
```

```
      110967      110972      110979      110991      111008      111020
      111051      111054      111086      111206      111208      111256
      111283      111296      111298      111337      111381      111394
      111404      111459      111469      111501      111504      111537
      111652      111668      111850      111860      111861      111884
      111953      111957      112011      112015      112022      112037
      112042      112136      112142      112158      112169      112286  {


                        use `i', clear
                        keep if secid ==`j'
                        generate min = (abs(delta) -0.5)^2
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\1997\filelist.txt"

      file open myfile using "E:\Data New\Straddle\1997\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace
```

```
local year "1998"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in 13896     13931 13959 13987 14015 14050 14078 14113 14141
      14169 14204 14232 14260 14295  {
            use `year', clear
            keep if exdate == `i'
            save `i'

                  foreach j in 50155036  5046  5048  5049  5061  5067  5083
      5086  5087  5089  5097  5133  5142  5169  5176  5191  5212  5252  5273
      5278  5297  5309  5317  5319  5385  5399  5402  5413  5418  5435  5459
      5496  5510  5520  5528  5549  5569  5579  5602  5603  5628  5644  5646
      5693  5712  5724  5746  5748  5764  5782  5788  5792  5804  5811  5828
      5837  5854  5882  5915  5934  5944  5959  5963  5973  5983  6008  6012
      6021  6029  6099  6214  6572  6677  7333  8042  8155  8239  8285  8672
      8680  8747  100900      100917      100930      100972      101062
      101121      101164      101199      101201      101204      101227
      101233      101273      101275      101305      101322      101325
      101368      101375      101382      101384      101387      101397
      101508      101533      101549      101558      101578      101580
      101594      101610      101639      101669      101697      101725
      101795      101800      101806      101828      101849      101863
      101903      101920      101930      101966      101988      102015
      102021      102041      102042      102043      102061      102067
      102082      102088      102117      102163      102210      102244
      102265      102267      102296      102339      102349      102386
      102408      102409      102525      102561      102571      102583
      102588      102660      102702      102733      102752      102796
      102833      102845      102927      102936      102968      103015
      103029      103037      103042      103049      103106      103125
      103126      103157      103202      103282      103296      103302
      103313      103355      103363      103370      103401      103402
      103404      103434      103466      103477      103498      103546
      103574      103654      103676      103682      103736      103749
      103772      103861      103879      103912      103920      103932
      103936      103937      103969      103979      104049      104117
      104118      104124      104144      104153      104172      104286
```

```
104338      104355      104361      104411      104508      104533
104550      104560      104626      104628      104633      104635
104664      104847      104878      104891      104893      104939
104958      104967      104992      105003      105076      105119
105120      105138      105168      105169      105174      105206
105223      105244      105327      105338      105340      105351
105356 105379    105384      105452      105512      105550
105557      105573      105581      105588      105615      105669
105675      105688      105696      105700      105730      105759
105776      105785      105833      105846      105955      105976
105985      106045      106119      106203      106276      106281
106295      106329      106553      106566      106567      106595
106629      106638      106665      106674      106689      106713
106723      106744      106776      106893      106967      106969
106975      107006      107010      107015      107036      107044
107045      107050      107078      107079      107107      107203
107244      107248      107255      107280      107297      107309
107317      107318      107321      107335      107398      107407
107430      107438      107525      107544      107585      107605
107616      107691      107693      107704      107884      107916
107951      107952      107975      108117      108120      108130
108146      108156      108160      108161      108185      108186
108196      108226      108231      108265      108279      108385
108429      108463      108505      108573      108609      108617
108642      108645      108649      108670      108716      108756
108768      108804      108849      108877      108887      108888
108893      108910      108948      108960      108962      108965
108969      109036      109047      109085      109118      109143
109224      109245      109274      109285      109297      109347
109396      109447      109463      109497      109528      109543
109557      109652      109678      109692      109715      109724
109749      109752      109775      109823      109848      109854
109866      109923      109954      109956      109965      109970
109998      110000      110002      110051      110086      110117
110152      110262      110282      110337      110357      110432
110433      110434      110460      110464      110490      110594
110611      110627      110634      110649      110707      110716
110752      110766      110810      110860      110914      110926
110945      110950      110967      110972      110979      110991
111008      111020      111051      111054      111086      111206
111208      111256      111283      111296      111298      111337
111359      111381      111394      111395      111404      111459
111469      111501      111504      111537      111652      111668
111850      111860      111861      111884      111893      111953
111957      112011      112015      112022      112037      112042
112136      112142      112158      112169      112286{
```

```
use `i', clear
keep if secid ==`j'
generate min = (abs(delta) -0.5)^2
sort date ncp_flag min
drop if strike_price ~=strike_price[1]
```

```
                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\1998\filelist.txt"

      file open myfile using "E:\Data New\Straddle\1998\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "1999"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
```

```
            duplicates drop optionid date, force
                  save `year', replace


foreach i in 14260     14295 14323 14351 14386 14414 14442 14477 14505
      14533 14568 14596 14631 14659  {
            use `year', clear
            keep if exdate == `i'
            save `i'

                  foreach j in 50365046  5048  5061  5067  5083  5087  5133
      5142  5169  5176  5212  5278  5317  5385  5399  5418  5435  5459  5496
      5510  5528  5549  5602  5628  5646  5693  5695  5724  5746  5748  5764
      5782  5792  5804  5828  5837  5854  5915  5934  5944  5963  5983  6008
      6029  6099  6214  6572  6677  8042  8239  8285  8672  8680  100871
      100892      100900      100917      100930      100972      101053
      101062      101121      101164      101199      101201      101204
      101227      101233      101263      101273      101275      101305
      101322      101325      101328      101368      101375      101382
      101384      101387      101397      101508      101523      101533
      101535      101549      101558      101578      101580      101594
      101610      101639      101669      101697      101725      101795
      101800      101806      101828      101849      101863      101903
      101920      101930      101966      101988      102015      102021
      102041      102042      102043      102061      102067      102068
      102082      102088      102113      102117      102163      102210
      102244      102265      102267      102296      102339      102349
      102386      102408      102409      102525      102536      102561
      102571      102583      102588      102660      102702      102733
      102752      102796      102833      102845      102885      102936
      102968      103015      103029      103037      103042      103049
      103065      103106      103125      103126      103157      103202
      103282      103296      103302      103307      103312      103313
      103355      103363      103370      103401      103402      103404
      103434      103466      103477      103498      103546      103574
      103654      103676      103682      103736      103749      103751
      103772      103861      103879      103912      103920      103932
      103936      103937      103969      103979      104049      104117
      104118      104124      104153      104172      104206      104286
      104338      104355      104361      104411      104508      104533
      104550      104560      104626      104628      104633      104635
      104664      104847      104866      104878      104891      104893
      104939      104958      104967      104992      105003      105076
      105119      105120      105138      105168      105169      105174
      105206      105223      105244      105327      105338      105340
      105351      105356      105379      105384      105452      105512
      105550      105573      105581      105588      105615      105669
      105675      105688      105696 105700    105730     105759
      105776      105785      105833      105846      105955      105976
      105985      106045      106119      106203      106276      106281
      106295      106329      106553      106566      106567      106595
      106625      106629      106638      106665      106674      106689
      106713      106723      106744      106776      106891      106893
      106917      106967      106969      106975      107006      107010
      107015      107036      107044      107045      107050      107078
      107079      107107      107203      107244      107248      107255
      107280      107297      107309      107317      107318      107321
```

```
107325      107335      107398      107407      107430      107438
107525      107544      107585      107605      107616      107657
107691      107693      107704      107884      107916      107951
107952      107975      108055      108117      108120      108130
108146      108156      108160      108161      108185      108186
108196      108226      108231      108265      108279      108385
108401      108419      108429      108463      108505      108573
108609      108617      108642      108645      108649      108670
108710      108716      108756      108768      108804      108849
108877      108887      108888      108893      108910      108948
108960      108962      108965      108969      109024      109036
109047      109085      109118      109143      109181      109224
109245      109274      109285      109297      109347      109348
109396      109447      109463      109497      109528      109543
109557      109652      109678      109692      109715      109724
109749      109752      109775      109848      109854      109866
109923      109954      109956      109965      109970      109998
110000      110002      110051      110086      110117      110152
110262      110282      110337      110356      110357      110432
110433      110434      110460      110464      110490      110594
110611      110627      110634      110649      110707      110716
110752      110766      110810      110860      110914      110926
110945      110950      110952      110967      110972      110979
110991      111008      111020      111051      111054      111086
111093      111206      111208      111256      111283      111296
111298      111337      111359      111381      111394      111395
111404      111459      111469      111501      111504      111537
111652      111668      111822      111850      111860      111861
111884      111893      111907      111951      111953      111957
112011      112015      112037      112042      112136      112142
112158      112161      112169      112180      112286      112333{
```

```
use `i', clear
keep if secid ==`j'
generate min = (abs(delta) -0.5)^2
sort date ncp_flag min
drop if strike_price ~=strike_price[1]
```

```
drop if ncp_flag ==ncp_flag[_n-1]
generate T = (exdate - date)/365
generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
```

```
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\1999\filelist.txt"

      file open myfile using "E:\Data New\Straddle\1999\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2000"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in 14631     14659 14687 14722 14750 14778 14813 14841 14869
      14904 14932 14960 14995 15023    {
            use `year', clear
            keep if exdate == `i'
            save `i'

                  foreach j in 50675142  5169  5176  5212  5317  5385  5418
      5496  5510  5695  5748  5782  5828  5837  5934  5983  6008  6029  8672
```

```
8680  9412  100871      100892      100900      100917      100930
100972      101053      101062      101121      101149      101164
101199      101201      101204      101226      101227      101233
101263      101273      101275      101293      101305      101311
101322      101325      101328      101368      101375      101382
101384      101387      101397      101426      101508      101523
101533      101535      101549      101558      101578      101580
101594      101610      101639      101669      101697      101725
101795      101800      101806      101820      101828      101849
101863      101903      101920      101930      101966      101988
102015      102021      102042      102043      102061      102067
102068      102082      102088      102113      102117      102157
102163      102210      102244      102265      102267      102296
102339      102349      102362      102364      102386      102408
102409      102525      102529      102536      102561      102571
102583      102588      102660      102702      102733      102752
102796      102833      102845      102885      102924      102936
102968      102999      103015      103029      103037      103042
103049      103065      103106      103125      103126      103157
103202      103282      103296      103302      103307      103312
103313      103338      103355      103363      103370      103393
103401      103402      103404      103434      103466      103477
103498      103546      103574      103654      103676      103682
103736      103749      103751      103772      103802      103861
103879      103912      103920      103932      103936      103937
103969      103979      103981      104019      104049      104059
104117      104118      104124      104153      104172      104206
104286      104338      104355      104361      104411      104508
104533      104560      104626      104628      104633      104635
104664      104847      104866      104878      104891      104893
104939      104946      104958      104967      104992      105003
105119      105120      105138      105168      105169      105174
105206      105223      105244      105327      105338      105340
105351      105356      105379      105384      105452      105512
105558      105581      105588      105615      105669      105688
105696      105700      105730      105759      105776      105785
105833      105846      105955      105976      105985      106045
106119      106203      106276      106281      106295      106329
106367 106505    106553      106566      106567      106595
106625      106629      106638      106665      106674      106677
106689      106697      106713      106723      106744      106776
106891      106893      106917      106967      106969      106975
106982      107006      107010      107015      107036      107044
107045      107050      107078      107079      107107      107203
107244      107248      107255      107280      107289      107297
107309      107317      107318      107321      107325      107335
107379      107398      107407      107430      107434      107438
107484      107525      107544      107585      107605      107616
107657      107686      107693      107704      107846      107884
107885      107916      107951      107952      107975      108055
108117      108120      108130      108146      108156      108160
108161      108166      108185      108186      108196      108226
108231      108265      108266      108279      108385      108401
108419      108429      108463      108505      108573      108609
108617      108642      108645      108649      108670      108710
108716      108719      108756      108768      108804      108849
108877      108887      108888      108893      108910      108948
```

```
      108960      108962      108965      108969      109024      109036
      109047      109085      109118      109131      109143      109181
      109224      109245      109274      109285      109297      109331
      109347      109348      109396      109447      109463      109497
      109528      109543      109557      109652      109667      109678
      109692      109715      109724      109749      109752      109775
      109841      109848      109854      109866      109906      109921
      109923      109954      109956      109965      109970      109998
      110000      110002      110051      110086      110117      110143
      110152      110262      110282      110337      110356      110357
      110432      110433      110434      110460      110464      110472
      110490      110531      110578      110594      110611      110627
      110634      110649      110685      110707      110716      110752
      110766      110810      110860      110914      110926      110945
      110952      110967      110972      110979      110991      111008
      111020      111045      111051      111054      111086      111093
      111206      111208      111256      111283      111296      111298
      111337      111359      111381      111394      111395      111404
      111459      111469      111501      111504      111537      111663
      111668      111822      111850      111860      111861      111884
      111893      111907      111951      111953      111957      112011
      112015      112037      112042      112136      112142      112158
      112161      112169      112180      112333      112487      112507 {
```

```
                        use `i', clear
                        keep if secid ==`j'
                        generate min = (abs(delta) -0.5)^2
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}
```

```
! dir *.dta /a-d /b >"E:\Data New\Straddle\2000\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2000\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace



local year "2001"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in 14995     15023 15051 15086 15114 15142 15177 15205 15240
      15268 15296 15331 15359   {
            use `year', clear
            keep if exdate == `i'
            save `i'

                  foreach j in 100871    100892      100900      100917
      100930      100933      100972      101053      101062      101121
      101149      101164      101199      101201      101204      101226
      101227      101233      101263      101273      101275      101293
      101305      101311      101322      101325      101328      101368
      101375      101382      101384      101387      101397      101426
      101475      101508      101523      101533      101535      101549
      101558      101578      101580      101594      101610      101612
      101639      101697      101795      101800      101806      101820
      101828      101849      101863      101903      101920      101930
```

101966 101988 102015 102021 102042 102043
102061 102067 102068 102082 102088 102113
102157 102163 102210 102244 102265 102267
102296 102349 102362 102364 102386 102408
102409 102525 102529 102536 102561 102571
102583 102588 102660 102702 102733 102752
102796 102833 102845 102885 102924 102936
102968 102999 103015 103022 103029 103037
103042 103046 103049 103054 103065 103106
103125 103126 103157 103202 103282 103296
103302 103307 103312 103313 103325 103338
103355 103363 103370 103393 103401 103402
103404 103434 103466 103477 103574 103654
103676 103682 103736 103749 103751 103772
103802 103861 103879 103912 103920 103932
103936 103937 103969 103979 104019 104049
104059 104117 104118 104124 104153 104172
104206 104286 104338 104355 104361 104411
104420 104425 104508 104533 104550 104560
104593 104626 104628 104633 104635 104664
104847 104866 104870 104878 104891 104939
104946 104958 104992 105003 105076 105119
105120 105138 105168 105169 105174 105206
105216 105223 105244 105327 105338 105340
105356 105384 105452 105512 105535 105558
105581 105588 105615 105617 105669 105688
105696 105700 105730 105759 105776 105785
105833 105846 105955 105985 106010 106045
106119 106203 106276 106281 106293 106295
106329 106367 106505 106521 106553 106566
106567 106569 106595 106629 106638 106665
106674 106677 106689 106697 106713 106723
106744 106776 106891 106893 106917 106967
106969 106975 106982 107006 107010 107015
107036 107044 107045 107050 107078 107079
107107 107203 107244 107248 107255 107280
107289 107297 107309 107317 107318 107321
107325 107335 107379 107398 107407 107430
107434 107438 107484 107525 107544 107605
107616 107657 107686 107704 107846 107885
107916 107951 107952 107975 108055 108117
108120 108130 108146 108156 108160 108161
108166 108175 108185 108186 108196 108226
108231 108265 108266 108279 108321 108385
108401 108419 108429 108505 108609 108617
108639 108642 108645 108649 108670 108710
108716 108719 108756 108768 108804 108849
108877 108887 108890 108893 108910 108948
108960 108962 108965 108969 109024 109036
109047 109118 109131 109143 109181 109224
109245 109274 109285 109297 109331 109347
109348 109396 109447 109463 109497 109528
109543 109557 109667 109678 109692 109715
109724 109752 109775 109841 109848 109854
109866 109906 109921 109923 109954 109956
109965 109970 110000 110002 110051 110117
110143 110152 110262 110282 110337 110356

```
      110357      110433      110434      110460      110464      110472
      110490      110531      110578      110594      110611      110627
      110634      110649      110685      110707      110716      110740
      110752      110766      110771      110810      110860      110914
      110926      110945      110952      110967      110972      110979
      110991      111008      111045      111051      111054      111086
      111093      111206      111208      111256      111283      111296
      111298      111337      111359      111381      111394      111395
      111404      111459      111469      111498      111501      111504
      111537      111663      111668      111822      111850      111860
      111861      111884      111893      111907      111951      111953
      111957      112011      112015      112037      112042      112136
      112142      112158      112161      112169      112180      112233
      112333      112339      112487      112507      112888      113069
      115279{
```

```
                        use `i', clear
                        keep if secid ==`j'
                        generate min = (abs(delta) -0.5)^2
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2001\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2001\filelist.txt",
read

file read myfile line
```

```
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2002"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in  15359    15387 15415 15450 15478 15513 15541 15569 15604
      15632 15660 15695 15723   {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100871    100892     100900     100917
      100930      100933      100972      101062      101121      101149
      101164      101199      101201      101204      101226      101233
      101263      101273      101275      101293      101311      101322
      101325      101328      101368      101375      101384      101387
      101397      101426      101441      101475      101508      101523
      101533      101535      101549      101558      101578      101580
      101590      101594      101610      101612      101639      101697
      101795      101800      101806      101820      101828      101849
      101863      101886      101903      101920      101930      101966
      101988      102015      102021      102042      102043      102061
      102067      102068      102082      102088      102113      102157
      102163      102210      102244      102265      102267      102296
      102349      102362      102379      102386      102408      102409
      102525      102536      102561      102571      102583      102660
      102702      102733      102752      102796      102833      102845
```

102885 102924 102936 102968 102999 103015
103022 103029 103037 103042 103046 103049
103054 103065 103106 103125 103126 103157
103198 103202 103282 103296 103302 103307
103312 103313 103325 103338 103355 103363
103370 103393 103401 103402 103404 103434
103466 103477 103498 103574 103654 103676
103682 103736 103749 103751 103772 103802
103861 103879 103912 103920 103932 103936
103937 103969 103979 104019 104049 104059
104117 104118 104124 104138 104153 104172
104206 104229 104286 104338 104361 104411
104420 104425 104508 104533 104550 104560
104593 104626 104628 104633 104635 104664
104845 104847 104866 104870 104878 104891
104939 104946 104958 104992 105003 105119
105120 105138 105168 105169 105174 105206
105216 105223 105244 105327 105329 105338
105340 105356 105384 105452 105512 105535
105558 105581 105588 105615 105617 105669
105688 105696 105700 105730 105759 105785
105833 105846 105955 105985 106010 106045
106119 106203 106276 106281 106293 106295
106329 106367 106505 106521 106553 106566
106567 106569 106595 106629 106638 106665
106674 106677 106689 106697 106713 106723
106744 106776 106891 106893 106917 106967
106969 106975 106982 107006 107010 107015
107044 107045 107050 107078 107079 107107
107203 107244 107247 107248 107255 107280
107289 107297 107309 107317 107318 107321
107325 107335 107379 107398 107407 107430
107434 107438 107484 107525 107544 107605
107616 107657 107676 107686 107704 107846
107885 107916 107951 107975 108055 108117
108120 108130 108146 108156 108160 108161
108166 108175 108185 108186 108196 108213
108226 108231 108265 108266 108279 108321
108385 108401 108429 108505 108609 108617
108639 108642 108645 108649 108670 108710
108716 108719 108756 108768 108804 108849
108877 108887 108890 108893 108910 108948
108960 108962 108965 108969 109024 109036
109047 109073 109131 109143 109181 109224
109245 109274 109285 109297 109331 109348
109371 109396 109447 109485 109497 109528
109543 109557 109621 109667 109678 109692
109715 109724 109752 109775 109841 109848
109854 109866 109869 109906 109921 109923
109954 109956 109965 109970 110000 110002
110051 110117 110143 110152 110188 110262
110282 110337 110356 110357 110433 110434
110460 110464 110472 110490 110531 110578
110611 110627 110634 110649 110685 110707
110716 110740 110752 110766 110771 110810
110860 110914 110926 110945 110952 110972
110979 110991 111008 111045 111086 111206

```
      111208      111256      111283      111296      111298      111337
      111359      111381      111394      111395      111404      111434
      111459      111469      111498      111501      111504      111537
      111663      111668      111822      111860      111861      111884
      111893      111902      111907      111951      111953      111957
      112011      112015      112037      112042      112136      112142
      112158      112161      112169      112180      112233      112254
      112333      112339      112487      112507      112888      113069
      113096      113119      113437      113859      115279{
```

```
                        use `i', clear
                        keep if secid ==`j'
                        generate min = (abs(delta) -0.5)^2
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2002\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2002\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
```

```
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2003"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in  15723    15758 15786 15814 15842 15877 15905 15933 15968
      15996 16031 16059 16087 16122    {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100871    100892     100900     100917
      100930      100933      100972      101062      101121      101149
      101164      101199      101204      101226      101233      101263
      101273      101275      101293      101311      101322      101325
      101328      101368      101375      101384      101387      101397
      101426      101441      101475      101508      101523      101533
      101535      101549      101558      101578      101580      101583
      101590      101594      101610      101612      101639      101697
      101795      101800      101802      101806      101820      101828
      101849      101863      101886      101903      101920      101930
      101966      101988      102015      102042      102043      102061
      102067      102068      102082      102088      102113      102157
      102163      102210      102244      102265      102267      102296
      102349      102362      102379      102386      102408      102409
      102525      102536      102561      102571      102583      102660
      102702      102733      102752      102796      102833      102845
      102885      102924      102936      102968      102999      103015
      103022      103029      103037      103042      103046      103049
      103054      103065      103106      103125      103126      103157
      103198      103202      103296      103302      103307      103312
      103313      103325      103355      103363      103370      103393
      103401      103402      103404      103434      103466      103477
```

103498 103574 103654 103676 103682 103736
103749 103751 103772 103802 103861 103879
103912 103920 103932 103936 103937 103969
103979 104019 104049 104059 104117 104118
104124 104138 104153 104172 104206 104229
104286 104338 104361 104411 104420 104425
104508 104522 104533 104560 104593 104626
104628 104633 104634 104635 104664 104845
104847 104866 104870 104878 104891 104939
104946 104958 104992 105003 105119 105120
105138 105168 105169 105174 105206 105216
105223 105244 105327 105329 105338 105340
105356 105384 105452 105512 105535 105558
105581 105588 105615 105617 105669 105688
105696 105700 105730 105759 105785 105833
105846 105894 105955 105985 106119 106203
106276 106281 106293 106295 106329 106367
106505 106521 106553 106566 106567 106569
106595 106629 106638 106665 106674 106677
106689 106697 106713 106723 106744 106776
106891 106893 106917 106967 106969 106975
106982 107006 107010 107015 107044 107045
107050 107078 107079 107107 107203 107244
107247 107248 107255 107280 107289 107297
107309 107315 107317 107318 107321 107325
107379 107398 107407 107430 107434 107438
107484 107525 107544 107605 107616 107657
107676 107686 107704 107846 107885 107916
107951 107975 108055 108117 108120 108130
108146 108160 108161 108166 108175 108185
108186 108213 108226 108231 108265 108266
108279 108321 108385 108401 108429 108505
108609 108617 108639 108642 108645 108649
108670 108710 108716 108756 108768 108804
108849 108877 108887 108890 108893 108910
108948 108960 108962 108965 108969 109024
109036 109073 109131 109143 109181 109224
109245 109248 109274 109285 109297 109331
109348 109371 109396 109447 109485 109497
109528 109543 109557 109621 109667 109678
109692 109715 109752 109775 109841 109848
109854 109866 109869 109906 109923 109954
109956 109965 109970 110000 110002 110051
110117 110143 110152 110188 110262 110282
110337 110356 110357 110433 110434 110460
110464 110472 110490 110531 110578 110611
110627 110634 110649 110684 110685 110707
110716 110740 110752 110771 110810 110860
110914 110926 110945 110952 110972 110979
110991 111008 111045 111086 111206 111208
111256 111283 111296 111298 111337 111359
111394 111395 111404 111434 111459 111469
111498 111501 111504 111537 111663 111668
111822 111860 111861 111884 111893 111902
111907 111951 111953 111957 112011 112015
112042 112136 112142 112158 112161 112169
112180 112233 112254 112333 112339 112487

```
      112507      112888      113096      113119      113437      113859
      115279      117627 {
```

```
                        use `i', clear
                        keep if secid ==`j'
                        generate min = (abs(delta) -0.5)^2
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2003\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2003\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace
```

```
local year "2004"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in  16087    16122 16150 16178 16213 16241 16269 16304 16332
      16360 16395 16423 16458 16486    {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100871    100892     100900     100930
      100933      100972      101062      101121      101137      101149
      101164      101199      101204      101226      101227      101233
      101263      101273      101275      101293      101311      101322
      101325      101328      101368      101375      101384      101387
      101397      101426      101441      101475      101508      101523
      101533      101535      101549      101558      101578      101580
      101583      101590      101594      101610      101612      101639
      101641      101697      101795      101800      101802      101806
      101820      101828      101849      101863      101886      101903
      101920      101930      101966      101988      102015      102042
      102043      102061      102067      102068      102082      102088
      102113      102163      102210      102244      102265      102267
      102296      102349      102362      102379      102386      102408
      102409      102525      102536      102561      102571      102583
      102660      102702      102733      102741      102752      102796
      102833      102845      102885      102924      102936      102968
      102999      103015      103022      103029      103037      103042
      103046      103049      103054      103065      103106      103109
      103125      103126      103157      103198      103202      103283
      103296      103302      103307      103312      103313      103325
      103355      103363      103370      103393      103401      103402
      103404      103434      103466      103477      103498      103533
      103574      103654      103676      103682      103736      103749
      103751      103772      103802      103861      103879      103912
      103920      103932      103936      103937      103969      103979
      104019      104049      104059      104081      104117      104118
      104124      104138      104153      104172      104206      104229
      104286      104338      104361      104411      104420      104425
```

```
      104508      104522      104533      104560      104593      104626
      104628      104633      104634      104635      104664      104845
      104847      104866      104870      104878      104879      104891
      104939      104946      104958      104992      105003      105119
      105120      105138      105168      105169      105174      105206
      105216      105223      105243      105244      105327      105329
      105338      105340      105356      105384      105452      105512
      105535      105558      105581      105588      105617      105669
      105688      105696      105700      105730      105759      105785
      105833      105846      105894      105955      105985      106119
      106203      106276      106281      106293      106295      106329
      106367      106505      106521      106553      106566      106567
      106569      106595      106629      106638      106665      106674
106677      106689      106697      106713      106723      106744
      106776      106780      106790      106891      106893      106917
      106967      106969      106975      106982      107006      107010
      107015      107044      107045      107050      107074      107078
      107079      107107      107203      107244      107247      107248
      107255      107280      107289      107297      107309      107315
      107318      107321      107325      107379      107398      107407
      107430      107434      107438      107484      107525      107544
      107605      107616      107657      107676      107686      107704
      107823      107846      107885      107916      107951      107975
      108055      108117      108120      108130      108146      108160
      108161      108166      108175      108185      108186      108213
      108226      108231      108265      108266      108279      108321
      108385      108401      108429      108505      108609      108617
      108639      108642      108645      108649      108670      108710
      108716      108756      108768      108804      108849      108877
      108887      108890      108893      108910      108948      108962
      108965      108969      109024      109036      109073      109131
      109143      109181      109224      109245      109248      109274
      109285      109297      109331      109348      109371      109396
      109447      109497      109528      109543      109557      109621
      109667      109678      109692      109715      109752      109775
      109841      109848      109854      109866      109869      109906
      109923      109954      109956      109965      109970      110000
      110002      110051      110117      110143      110152      110188
      110262      110282      110337      110356      110357      110433
      110434      110460      110464      110472      110490      110531
      110578      110611      110627      110634      110649      110684
      110685      110707      110716      110740      110752      110771
      110810      110860      110914      110926      110945      110952
      110972      110979      110991      111008      111045      111086
      111206      111208      111256      111283      111296      111298
      111337      111359      111394      111395      111404      111434
      111459      111469      111498      111501      111504      111537
      111560      111663      111668      111822      111860      111861
      111884      111893      111902      111907      111951      111953
      111957      112011      112015      112042      112136      112142
      112158      112161      112169      112180      112233      112254
      112333      112339      112487      112507      112888      113096
      113119      113437      113859      115279      117627      121383
      122337  {
```

```
                        use `i', clear
                        keep if secid ==`j'
                        generate min = (abs(delta) -0.5)^2
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2004\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2004\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2005"
local yt = `year' + 1
```

```
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in  16458    16486 16514 16542 16577 16605 16633 16668 16696
      16731 16759 16787 16822 16850  {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100871    100892     100900     100930
      100972      101062      101121      101137      101149      101164
      101199      101204      101226      101227      101233      101263
      101273      101275      101293      101310      101311      101322
      101325      101328      101368      101375      101387      101397
      101426      101441      101475      101508      101523      101533
      101535      101549      101558      101578      101580      101583
      101590      101594      101610      101612      101639      101641
      101697      101795      101800      101802      101806      101820
      101828      101849      101863      101886      101903      101920
      101930      101966      102015      102042      102043      102061
      102067      102068      102082      102088      102113      102163
      102210      102244      102265      102267      102296      102349
      102362      102386      102408      102409      102525      102536
      102561      102571      102583      102660      102702      102733
      102741      102752      102796      102833      102845      102885
      102936      102968      102999      103015      103022      103029
      103037      103042      103046      103049      103054      103065
      103106      103109      103125      103126      103157      103198
      103202      103283      103296      103302      103307      103312
      103313      103355      103363      103368      103370      103393
      103401      103402      103404      103434      103466      103477
      103485      103533      103574      103651      103654      103676
      103682      103736      103749      103751      103802      103861
      103879      103912      103920      103932      103936      103937
      103969      103979      104019      104049      104059      104081
      104117      104118      104124      104138      104153      104172
      104206      104229      104286      104338      104361      104411
      104420      104425      104508      104522      104533      104560
      104593      104626      104628      104633      104634      104635
      104664      104845      104847      104866      104870      104878
      104879      104939      104946      104958      104992      105003
      105119      105120      105138      105168      105169      105174
      105206      105216      105223      105243      105244      105327
```

```
      105329      105338      105340      105356      105384      105452
      105512      105558      105581      105588      105617      105669
      105688      105696      105700      105730      105759      105785
      105833      105846      105894      105955      105985      106119
      106203      106276      106281      106293      106295      106329
      106367      106505      106521      106553      106566      106567
      106569      106595      106629      106638      106665      106674
      106677      106689      106697      106713      106723      106744
106776      106780      106790      106891      106893      106899
      106917      106967      106969      106975      106982      107006
      107010      107015      107044      107045      107050      107074
      107078      107079      107107      107203      107244      107247
      107248      107255      107280      107289      107297      107309
      107315      107318      107321      107325      107379      107398
      107407      107430      107434      107438      107484      107525
      107544      107605      107616      107657      107676      107686
      107704      107812      107823      107846      107885      107916
      107939      107951      107975      108055      108117      108120
      108130      108146      108160      108161      108166      108175
      108185      108186      108213      108226      108231      108265
      108266      108279      108321      108385      108401      108429
      108505      108609      108617      108639      108642      108645
      108649      108670      108710      108716      108756      108768
      108793      108804      108849      108877      108890      108893
      108910      108948      108962      108965      108969      109024
      109036      109073      109131      109143      109181      109224
      109245      109248      109274      109285      109286      109297
      109331      109348      109371      109396      109447      109497
      109528      109543      109557      109621      109667      109678
      109692      109715      109752      109775      109841      109848
      109854      109866      109869      109906      109923      109954
      109956      109965      109970      109988      110000      110002
      110051      110117      110143      110152      110188      110262
      110282      110337      110357      110433      110460      110464
      110472      110488      110490      110531      110578      110611
      110627      110634      110649      110684      110685      110707
      110716      110740      110752      110771      110810      110860
      110914      110926      110945      110952      110972      110979
      110991      111045      111086      111206      111208      111283
      111285      111296      111298      111337      111359      111394
      111404      111434      111459      111469      111498      111501
      111504      111537      111560      111663      111668      111814
      111822      111860      111861      111884      111893      111902
      111907      111921      111953      111957      112011      112015
      112042      112142      112158      112161      112169      112180
      112233      112254      112333      112339      112487      112507
      112888      113096      113119      113437      113859      115279
      117073      117627      121383      121535      122337      124943{
```

```
                        use `i', clear
                        keep if secid ==`j'
                        generate min = (abs(delta) -0.5)^2
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2005\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2005\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2006"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
```

```
            append using `yt'
            drop T

cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in  16822    16850 16878 16913 16941 16969 17004 17032 17060
      17095 17123 17151 17186 17214 {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100871    100892      100900      100972
      101062      101121      101137      101149      101164      101199
      101204      101226      101227      101233      101263      101273
      101275      101293      101310      101311      101322      101325
      101328      101368      101375      101387      101397      101426
      101441      101475      101508      101523      101533      101535
      101549      101558      101578      101580      101583      101590
      101594      101610      101612      101639      101641      101697
      101795      101800      101802      101806      101820      101828
      101849      101863      101886      101903      101920      101930
      101966      102015      102025      102042      102043      102061
      102067      102068      102082      102088      102113      102163
      102210      102244      102265      102267      102294      102296
      102349      102362      102386      102408      102409      102525
      102536      102561      102571      102583      102660      102702
      102733      102741      102752      102796      102822      102833
      102845      102885      102936      102963      102968      102999
      103015      103022      103037      103042      103046      103049
      103054      103065      103106      103109      103125      103126
      103138      103157      103198      103202      103208      103283
      103296      103302      103307      103312      103313      103354
      103355      103363      103368      103370      103393      103401
      103402      103404      103434      103466      103477      103485
      103533      103574      103651      103654      103676      103682
      103736      103749      103802      103861      103879      103912
      103920      103932      103936      103937      103969      103979
      104019      104049      104059      104081      104117      104118
      104124      104138      104153      104172      104206      104229
      104286      104338      104361      104411      104420      104425
      104508      104522      104533      104560      104593      104626
      104628      104633      104634      104635      104664      104845
      104847      104866      104870      104878      104879      104939
      104946      104958      104992      105003      105119      105120
      105138      105168      105169      105174      105206      105216
      105243      105327      105329      105338      105340      105356
      105452      105512      105558      105562      105581      105588
      105617      105669      105688      105696      105700      105730
      105759      105785      105833      105846      105894      105955
      105985      106119      106203      106276      106281      106293
```

```
      106295      106329      106367      106505      106521      106553
      106566      106567      106569      106581      106595      106629
      106638      106665      106674      106677      106689      106690
106697      106713      106723      106744      106776      106780
      106790      106858      106891      106892      106893      106899
      106917      106967      106969      106975      106982      107006
      107010      107015      107044      107045      107050      107074
      107078      107107      107203      107244      107247      107248
      107255      107280      107289      107309      107315      107318
      107321      107325      107379      107398      107407      107430
      107434      107438      107484      107525      107544      107605
      107616      107657      107676      107686      107704      107812
      107823      107846      107885      107916      107939      107951
      107975      108055      108117      108120      108130      108160
      108161      108166      108175      108185      108186      108213
      108226      108231      108265      108266      108279      108321
      108385      108401      108429      108505      108609      108617
      108639      108642      108645      108649      108670      108710
      108716      108756      108768      108793      108804      108849
      108877      108890      108893      108910      108948      108962
      108965      108969      109024      109036      109073      109143
      109181      109224      109245      109248      109285      109286
      109297      109331      109348      109371      109373      109396
      109447      109497      109528      109543      109557      109621
      109667      109678      109692      109715      109752      109775
      109841      109848 109854     109866      109869      109901
      109906      109923      109954      109956      109965      109970
      109988      110000      110051      110117      110143      110152
      110188      110248      110262      110282      110337      110357
      110433      110460      110464      110472      110488      110490
      110531      110578      110590      110611      110627      110634
      110649      110684      110685      110707      110716      110740
      110752      110771      110810      110860      110914      110926
      110945      110952      110955      110972      110979      110991
      111045      111086      111206      111208      111283      111285
      111296      111298      111310      111337      111359      111394
      111404      111434      111459      111469      111498      111504
      111537      111560      111666      111668      111814      111822
      111860      111861      111884      111893      111902      111907
      111921      111953      111957      112011      112015      112026
      112042      112142      112158      112161      112169      112180
      112233      112254      112333      112339      112487      112507
      112717      112888      113096      113119      113437      113859
      115179      115279      116532      117073      117627      119891
      121383      121535      121568      121812      122337      123082
      124943      125232      125237      126835      127317      127593
      128182 {
```

```
use `i', clear
keep if secid ==`j'
```

```
                        generate min = (abs(delta) -0.5)^2
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                                drop if ncp_flag ==ncp_flag[_n-1]
                                generate T = (exdate - date)/365
                                generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                                generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                                generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                                generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                                generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                    save `i'`j', replace
                }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2006\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2006\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2007"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T
```

```
cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in   17186   17214 17242 17277 17305 17333 17368 17396 17431
      17459 17487 17522 17550 17578   {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100871    100892     100900     100972
      100974      101062      101121      101137      101149      101164
      101176      101204      101226      101227      101233      101263
      101273      101275      101293      101310      101311      101322
      101325      101328      101354      101368      101375      101387
      101397      101426      101441      101444      101475      101508
      101533      101535      101558      101578      101580      101583
      101590      101594      101610      101639      101641      101697
      101795      101800      101802      101806      101812      101820
      101828      101849      101863      101886      101903      101920
      101930      101966      102015      102025      102042      102043
      102061      102067      102068      102082      102088      102113
      102163      102210      102244      102265      102267      102294
      102296      102349      102362      102386      102408      102524
      102525      102536      102561      102571      102583      102660
      102702      102733      102741      102752      102796      102822
      102845      102885      102936      102963      102968      103015
      103022      103029      103037      103042      103046      103049
      103054      103065      103106      103109      103125      103126
      103138      103157      103198      103202      103208      103283
      103296      103302      103307      103312      103313      103354
      103355      103363      103368      103370      103393      103401
      103404      103434      103466      103477      103485      103533
      103574      103651      103654      103676      103682      103736
      103749      103800      103802      103861      103879      103912
      103920      103932      103936      103937      103969      103979
      104019      104049      104056      104059      104081      104117
      104118      104124      104138      104153      104172      104206
      104229      104286      104361      104411      104420      104425
      104508      104517      104522      104533      104560      104593
      104626      104628      104633      104634      104635      104664
      104845      104847      104866      104870      104878      104939
      104946      104958      104992      105003      105119      105120
      105168      105169      105174      105206      105216      105243
      105329      105338      105340      105356      105512      105558
      105562      105581      105588      105617      105669      105688
      105696      105700      105730      105759      105785      105805
      105824      105833      105846      105894      105955      105985
      106119      106203      106276      106281      106293      106295
      106329      106367      106505      106521      106529      106566
      106567      106569      106581      106595      106629      106638
```

```
      106674      106677      106689      106690      106697      106723
106744      106776      106780      106790      106858      106891
      106892      106893      106899      106905      106917      106967
      106969      106975      106982      107006      107010      107015
      107045      107074      107078      107095      107107      107200
      107203      107244      107247      107248      107255      107280
      107289      107315      107318      107321      107325      107379
      107398      107407      107430      107438      107484      107525
      107532      107544      107605      107616      107657      107676
      107686      107704      107812      107823      107846      107885
      107916      107939      107951      108055      108117      108120
      108130      108160      108161      108166      108173      108175
      108185      108186      108226      108231      108265      108266
      108279      108321      108385      108401      108429      108505
      108609      108617      108639      108642      108645      108649
      108670      108710      108716      108756      108768      108793
      108804      108849      108877      108890      108893      108910
      108948      108962      108965      108969      109024      109036
      109073      109088      109120      109143      109148      109181
      109224      109245      109248      109285      109286      109297
      109331      109348      109371      109373      109396      109447
      109476      109497      109543      109557      109621      109667
      109678      109692      109715      109752      109775      109841
      109848      109854      109866      109869      109901      109906
      109923      109954      109956      109965      109988      110000
      110051      110117      110152 110188    110248      110262
      110282      110337      110357      110433      110460      110464
      110472      110488      110490      110531      110578      110590
      110611      110627      110634      110649      110684      110685
      110707      110716      110740      110752      110771      110810
      110860      110914      110926      110945      110952      110955
      110962      110972      110979      110991      111045      111062
      111086      111206      111208      111283      111285      111296
      111298      111310      111337      111359      111394      111404
      111434      111459      111469      111498      111504      111537
      111560      111628      111666      111668      111814      111822
      111838      111860      111861      111884      111893      111902
      111907      111921      111953      111957      112011      112015
      112026      112042      112142      112158      112161      112169
      112180      112233      112254      112333      112339      112487
      112507      112717      112888      113096      113119      113437
      113859      113993      115179      115247      115279      116532
      117073      117627      119891      120322      121383      121535
      121568      121812      122337      123082      124209      124943
      125175      125232      125237      125812      126835      127317
      127593      128604      129709      129711      129716      134207 {
```

```
use `i', clear
keep if secid ==`j'
generate min = (abs(delta) -0.5)^2
```

```
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2007\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2007\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2008"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T
```

```
cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in  17550    17578 17613 17641 17669 17704 17732 17760 17795
      17823 17858 17886 17914    {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100892    100900     100903     100972
      100974      101062      101121      101137      101149      101164
      101176      101204      101226      101227      101233      101263
      101273      101293      101310      101311      101322      101325
      101328      101354      101368      101375      101387      101397
      101441      101444      101475      101508      101519      101530
      101533      101535      101558      101578      101580      101583
      101590      101594      101610      101639      101697      101795
      101800      101802      101806      101812      101828      101849
      101863      101886      101903      101920      101930      101966
      102015      102025      102043      102061      102067      102068
      102088      102113      102210      102244      102265      102267
      102294      102296      102349      102362      102379      102386
      102408      102524      102525      102536      102561      102571
      102583      102595      102660      102702      102733      102752
      102796      102822      102845      102885      102886      102936
      102963      102968      103015      103022      103029      103037
      103042      103046      103049      103054      103065      103106
      103109      103125      103126      103138      103157      103198
      103202      103208      103296      103302      103307      103313
      103354      103355      103363      103368      103370      103393
      103399      103401      103404      103434      103466      103477
      103485      103533      103574      103651      103654      103676
      103682      103715      103736      103749      103784      103800
      103802      103861      103879      103912      103920      103932
      103936      103969      103979      103981      104019      104049
      104056      104059      104081      104117      104118      104124
      104138      104153      104172      104206      104229      104286
      104361      104411      104416      104425      104508      104517
      104522      104533      104560      104593      104618      104626
      104628      104633      104634      104635      104664      104845
      104847      104866      104870      104878      104916      104918
      104939      104946      104958      104992      105003      105119
      105120      105168      105169      105174      105206      105216
      105243      105329      105338      105340      105356      105512
      105558      105562      105573      105581      105588      105613
      105669      105688      105696      105700      105759      105785
      105805      105824      105833      105846      105894      105955
      105985      106119      106203      106276      106281      106293
      106295      106329      106367      106369      106387      106505
      106521      106529      106566      106567      106569      106581
      106595      106629      106638      106674      106689      106690
```

```
106697      106723      106744      106776      106780      106790
      106858      106891      106892      106893      106899      106905
      106917      106967      106969      106975      106982      107006
      107010      107015      107045      107074      107078      107095
      107107      107200      107244      107247      107248      107255
      107280      107315      107318      107321      107325      107398
      107407      107430      107438      107484      107525      107532
      107544      107605      107616      107657      107676      107686
      107704      107812      107823      107885      107916      107939
      107951      108006      108055      108117      108120      108130
      108160      108161      108166      108173      108175      108185
      108186      108226      108231      108265      108266      108279
      108321      108385      108401      108429      108505      108609
      108617      108642      108645      108649      108670      108710
      108716      108768      108793      108804      108849      108871
      108890      108893      108910      108948      108965      108969
      109024      109029      109036      109073      109088      109120
      109143      109148      109181      109224      109245      109248
      109285      109286      109297      109331      109348      109371
      109373      109396      109447      109476      109497      109543
      109557      109586      109621      109667      109678      109692
      109715      109752      109775      109848      109854      109866
      109869      109901      109923      109954      109956      109965
      109988      110000      110051      110117      110152      110188
      110248      110262      110337      110357      110366      110433
      110460      110464      110472      110488 110490    110510
      110531      110578      110590      110611      110627      110634
      110649      110684      110707      110716      110740      110752
      110771      110810      110914      110945      110952      110955
      110962      110972      110979      110991      111045      111062
      111086      111107      111206      111283      111285      111296
      111298      111310      111337      111359      111394      111404
      111434      111459      111469      111504      111537      111560
      111628      111666      111668      111814      111822      111838
      111860      111861      111884      111893      111902      111907
      111921      111953      111957      112011      112015      112026
      112042      112087      112142      112158      112161      112169
      112180      112233      112254      112333      112339      112487
      112507      112717      112888      113096      113119      113437
      113859      113928      113993      115179      115247      115279
      115429      115885      116416      116532      117073      117627
      119891      120322      121383      121535      121568      121592
      121812      122337      123082      124209      124758      124943
      125175      125232      125237      125812      126835      126938
      127317      127593      128604      129709      129711      129716
      134207      135403      136216      137155  {
```

```
use `i', clear
keep if secid ==`j'
generate min = (abs(delta) -0.5)^2
```

```
                        sort date ncp_flag min
                        drop if strike_price ~=strike_price[1]


                              drop if ncp_flag ==ncp_flag[_n-1]
                              generate T = (exdate - date)/365
                              generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2008\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2008\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2009"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
            append using `yt'
            drop T
```

```
cd "E:\Data New\Straddle\"
cd `year'

      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in  17914    17949 17977 18005 18033 18068 18096 18131 18159
      18187 18222 18250 18278 18313  {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100892    100900     100903     100972
      100974      101062      101121      101137      101149      101164
      101167      101176      101204      101226      101227      101233
      101273      101293      101310      101322      101325      101328
      101354      101368      101375      101387      101397      101444
      101475      101508      101519      101530      101533      101535
      101578      101580      101583      101590      101594      101610
      101639      101795      101800      101802      101806      101812
      101828      101849      101863      101886      101903      101920
      101930      101966      102015      102043      102067      102068
      102088      102113      102210      102244      102265      102294
      102296      102349      102362      102379      102408      102524
      102525      102536      102561      102571      102583      102595
      102660      102702      102733      102752      102796      102822
      102845      102885      102886      102936      102963      102968
      103015      103022      103029      103042      103046      103049
      103054      103065      103098      103106      103109      103125
      103126      103138      103157      103198      103202      103296
      103302      103307      103313      103354      103355      103363
      103368      103370      103393      103399      103401      103404
      103434      103466      103485      103533      103574      103651
      103654      103676      103682      103715      103736      103749
      103770      103784      103800      103802      103803      103821
      103879      103912      103920      103932      103936      103969
      103979      103981      104019      104049      104056      104059
      104081      104117      104118      104124      104138      104153
      104172      104206      104229      104286      104361      104411
      104416      104425      104508      104517      104522      104533
      104548      104550      104560      104593      104618      104633
      104634      104635      104664      104845      104866      104870
      104878      104916      104918      104939      104946      104958
      104992      105003      105119      105120      105168      105169
      105174      105206      105216      105243      105329      105338
      105340      105356      105512      105558      105562      105573
      105581      105588      105613      105633      105669      105696
      105700      105759      105785      105800      105805      105824
      105833      105846      105894      105955      105985      106119
      106203      106276      106281      106293      106295      106329
      106367      106369      106387      106407      106505      106521
      106529      106566      106567      106569      106581      106595
      106629      106638      106674      106689      106690      106697
      106723      106744      106776      106780      106790      106858
```

```
106891      106892      106899      106905      106917      106967
      106969      106975      106982      107010      107015      107045
      107074      107078      107095      107200      107244      107247
      107248      107255      107280      107315      107318      107321
      107325      107398      107407      107430      107438      107484
      107525      107532      107544      107605      107616      107657
      107676      107686      107704      107812      107823      107885
      107939      107951      108006      108055      108117      108120
      108130      108160      108161      108166      108173      108175
      108185      108186      108220      108226      108231      108265
      108266      108279      108321      108385      108401      108429
      108505      108573      108609      108642      108645      108649
      108670      108710      108716      108768      108793      108804
      108849      108871      108890      108893      108910      108948
      108965      108969      109024      109029      109036      109073
      109088      109120      109143      109148      109181      109182
      109224      109245      109248      109285      109286      109297
      109331      109348      109356      109371      109373      109396
      109447      109476      109497      109513      109543      109557
      109586      109621      109667      109678      109692      109699
      109706      109715      109752      109775      109854      109866
      109869      109901      109923      109942      109954      109956
      109965      110000      110051      110117      110152      110188
      110248      110262      110337      110357      110366      110433
      110460      110464      110472      110488      110490 110510
      110531      110578      110590      110611      110627      110634
      110649      110684      110716      110740      110752      110771
      110810      110914      110945      110952      110962      110972
      110979      110991      111045      111062      111086      111107
      111206      111283      111285      111296      111298      111337
      111359      111394      111434      111459      111469      111504
      111537      111560      111628      111647      111666      111668
      111814      111822      111838      111860      111861      111893
      111902      111907      111921      111953      111987      112011
      112015      112026      112042      112087      112158      112161
      112169      112180      112233      112339      112487      112507
      112717      112888      113096      113119      113437      113859
      113928      113993      115179      115247      115259      115279
      115429      115885      116416      116532      117073      117627
      119891      120322      121383      121535      121568      121592
      121764      121812      122337      123082      124209      124758
      124943      125175      125232      125237      125812      126835
      126938      127317      127593      127813      128220      128604
      129054      129709      129711      129716      134207      135403
      135419      136216      137155      138637      139386      139977  {
```

```
use `i', clear
keep if secid ==`j'
generate min = (abs(delta) -0.5)^2
sort date ncp_flag min
```

```
                        drop if strike_price ~=strike_price[1]


                                drop if ncp_flag ==ncp_flag[_n-1]
                                generate T = (exdate - date)/365
                                generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
                                generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                                generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                                generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                                generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                    save `i'`j', replace
                }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2009\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2009\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace




local year "2010"
local yt = `year' + 1
set more off

cd "E:\Data New\Dummy\"

      use `year', clear
                  drop T

cd "E:\Data New\Straddle\"
cd `year'
```

```
      drop if missing(beta)
      compress
            duplicates drop optionid date, force
                  save `year', replace


foreach i in  18278    18313 18341 18369 18404 18432 18460 18495 18523
      18551 18586 18614 {
            use `year', clear
            keep if exdate == `i'
            save `i'

            foreach j in 100892    100900      100903      100972
      100974      101062      101121      101137      101149      101164
      101167      101176      101204      101226      101227      101233
      101273      101293      101310      101322      101325      101328
      101368      101375      101397      101444      101475      101508
      101519      101530      101533      101535      101578      101580
      101583      101590      101594      101610      101639      101795
      101800      101802      101806      101812      101828      101849
      101863      101886      101903      101920      101930      101966
      102015      102043      102067      102068      102088      102109
      102113      102210      102244      102265      102294      102296
      102349      102362      102379      102408      102524      102525
      102536      102561      102571      102583      102587      102595
      102660      102702      102733      102752      102796      102822
      102885      102886      102893      102936      102963      102968
      103015      103029      103038      103042      103046      103049
      103054      103065      103098      103106      103109      103125
      103126      103138      103157      103198      103202      103296
      103302      103307      103313      103354      103355      103363
      103368      103370      103399      103404      103434      103466
      103485      103533      103574      103651      103654      103676
      103682      103715      103736      103749      103770      103784
      103802      103803      103821      103879      103912      103920
      103932      103936      103969      103979      103981      104049
      104059      104081      104117      104118      104124      104138
      104153      104172      104206      104229      104286      104361
      104411      104416      104425      104508      104517      104522
      104533      104548      104550      104560      104593      104618
      104633      104634      104635      104644      104664      104845
      104866      104870      104878      104916      104918      104939
      104946      104958      104992      105003      105119      105120
      105168      105169      105174      105206      105216      105243
      105329      105338      105340      105356      105512      105558
      105562      105573      105581      105588      105613      105633
      105669      105675      105696      105700      105759      105785
      105800      105805      105824      105833      105846      105894
      105955      105985      106119      106203      106276      106281
      106293      106295      106329      106367      106369      106387
      106407      106505      106521      106529      106566      106567
      106581      106595      106638      106674      106689      106690
      106697      106723      106744      106776      106780      106790
      106858      106891      106892      106899      106905      106917
106967      106969      106975      106982      107010      107015
      107045      107074      107095      107244      107247      107248
```

```
107255      107280      107315      107318      107321      107325
107398      107407      107430      107438      107484      107525
107532      107544      107605      107616      107657      107676
107686      107704      107812      107823      107885      107939
107951      108006      108055      108117      108120      108121
108130      108160      108161      108166      108173      108185
108186      108220      108226      108231      108265      108266
108279      108321      108385      108401      108429      108463
108505      108573      108609      108642      108645      108649
108670      108710      108716      108768      108793      108804
108849      108871      108890      108893      108910      108948
108965      108969      109024      109029      109036      109073
109088      109120      109143      109148      109181      109182
109224      109245      109248      109285      109286      109297
109331      109348      109356      109371      109373      109396
109447      109476      109497      109513      109543      109557
109586      109621      109667      109678      109699      109706
109715      109752      109775      109854      109866      109869
109901      109923      109942      109956      109965      110000
110051      110117      110152      110188      110248      110262
110337      110357      110366      110433      110460      110464
110472      110488      110490      110510      110531      110578
110590      110611      110627      110634      110649      110684
110716      110740      110752 110771    110810      110914
110945      110952      110962      110972      110979      110991
111045      111062      111086      111107      111206      111283
111285      111296      111337      111359      111394      111434
111459      111469      111504      111512      111537      111560
111628      111647      111666      111668      111814      111822
111838      111860      111861      111893      111902      111907
111953      111987      112011      112015      112026      112042
112087      112158      112161      112169      112180      112233
112254      112339      112487      112507      112717      112888
113096      113119      113437      113859      113928      113993
115179      115247      115259      115279      115422      115429
115885      116416      116532      117073      117627      119846
119891      120322      121383      121535      121568      121592
121764      121812      122337      123082      124166      124209
124758      124943      125175      125232      125237      125812
126938      127317      127593      127813      128220      128604
129054      129716      134207      135403      135419      136216
137155      138637      139386      143086   {
```

```
use `i', clear
keep if secid ==`j'
generate min = (abs(delta) -0.5)^2
sort date ncp_flag min
drop if strike_price ~=strike_price[1]
```

```
drop if ncp_flag ==ncp_flag[_n-1]
generate T = (exdate - date)/365
generate D1 =
(((ln(prc/(strike_price/1000))+((rate/1000)+
impl_volatility^2)*T)/(impl_volatility*(T^0.5))))*(T^0.5)
```

```
                              generate Bcall
=(prc/price)*normalden(D1)*beta if ncp_flag==1
                              generate Bput =(prc/price)*(normalden(D1)-
1)*beta if ncp_flag==2
                              generate straddleR = (-Bput[_n+1]/(Bcall-
Bput[_n+1]))*return + (-Bput[_n+1]/(Bcall-Bput[_n+1]))*return[_n+1] if
ncp_flag ==1
                              generate ExcessstraddleR = (-
Bput[_n+1]/(Bcall-Bput[_n+1]))* excess_return + (-Bput[_n+1]/(Bcall-
Bput[_n+1]))* excess_return[_n+1] if ncp_flag ==1


                  save `i'`j', replace
            }
}

! dir *.dta /a-d /b >"E:\Data New\Straddle\2010\filelist.txt"

      file open myfile using "E:\Data New\Straddle\2010\filelist.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if missing(straddleR)
save master_data, replace
```

Appendix 11S: Skewness Asset

```
local year "1996"

set more off

cd "E:\Data New\Gefilterde Data\"

use `year'filter, clear

cd "E:\Data New\Straddle\1996\"
compress
save `year', replace

foreach i in 13168     13196 13224 13259 13287 13322 13350 13378 13413
      13441 13469 13504 13532  {
            use `year', clear
                  keep if exdate == `i'
                        save `i', replace
```

```
            foreach j in 50155022  5029  5036  5046  5048  5049  5058
5061  5067  5079  5083  5086  5087  5089  5097  5115  5133  5142  5158
5169  5176  5180  5191  5212  5218  5240  5252  5258  5273  5278  5297
5309  5317  5319  5351  5361  5385  5399  5402  5413  5428  5459  5496
5510  5528  5549  5563  5564  5569  5579  5587  5591  5603  5628  5644
5646  5670  5684  5692  5693  5699  5712  5724  5764  5782  5788  5804
5811  5817  5828  5837  5854  5882  5909  5915  5934  5944  5959  5963
5969  5970  5973  5983  6008  6012  6021  6029  6099  6250  6572  6677
6740  7333  7631  7737  8042  8155  8274  8285  8672  8747  9265
100917      100930      100972      101121      101164      101199
101201      101204      101227      101233      101273      101275
101305      101322      101325      101368      101375      101382
101384      101387      101397      101508      101549      101558
101578      101594      101610      101639      101669      101697
101795      101800      101828      101849      101920      101930
101966      101988      102015      102021      102041      102042
102043      102067      102082      102117      102163      102210
102244      102265      102267      102296      102339      102349
102381      102386      102408      102409      102525      102555
102561      102571      102583      102588      102660      102796
102833      102845      102927      102936      102968      103015
103037      103042      103049      103106      103125      103157
103202      103282      103296      103302      103313      103355
103370      103401      103402      103404      103434      103466
103498      103546      103574      103654      103682      103736
103749      103772      103861      103879      103912      103920
103932      103936      103937      103969      103979      104049
104117      104118      104124      104144      104153      104172
104286      104338      104355      104361      104508      104533
104550      104560      104626      104628      104633      104635
104664      104847      104878      104891      104893      104894
104939      104958      104967      105003      105076      105119
105120      105168      105169      105174      105206      105223
105244      105327      105338 105340    105356     105379
105384      105452      105512      105536      105550      105557
105573      105581      105588      105669      105675      105688
105696      105700      105730      105759      105776      105785
105833      105955      105976      105985      106045      106119
106203      106247      106276      106281      106295      106329
106553      106566      106567      106629      106638      106665
106674      106689      106713      106744      106776      106967
106969      106975      107006      107010      107015      107044
107045      107050      107078      107079      107107      107244
107255      107280      107297      107309      107317      107318
107321      107335      107398      107407      107430      107525
107544      107585      107605      107616      107691      107693
107704      107916      107951      107952      107975      108117
108120      108130      108156      108160      108161      108185
108186      108196      108231      108265      108279      108385
108406      108463      108505      108609      108617      108642
108645      108649      108670      108716      108768      108849
108888      108893      108910      108948      108960      108962
108965      108969      109036      109040      109047      109085
109118      109143      109224      109285      109297      109347
109447      109463      109497      109528      109557      109652
109678      109692      109715      109749      109752      109775
109823      109848      109866      109923      109954      109956
```

```
      109970      109998      110002      110086      110117      110126
      110152      110262      110337      110357      110433      110460
      110576      110611      110627      110634      110649      110716
      110752      110766      110810      110860      110914      110926
      110945      110950      110967      110972      110979      111008
      111020      111051      111054      111086      111208      111256
      111283      111296      111298      111337      111381      111394
      111404      111459      111469      111501      111504      111537
      111652      111668      111850      111860      111861      111953
      111957      112011      112015      112022      112037      112042
      112136      112142      112169      112185      112286  {


                        use `i', clear
                        keep if secid ==`j'
                        generate min = (abs(delta) -0.1)^2
                        sort ncp_flag min
                        drop if strike_price ~=strike_price[1] & ncp_flag
== 1

                        drop min
                        generate min = (abs(delta) - 0.1)^2
                        gsort -ncp_flag min
                        drop if strike_price ~=strike_price[1] & ncp_flag
== 2

                        sort date ncp_flag
                        drop if ncp_flag ==ncp_flag[_n-1]

                        capture duplicates drop

                        generate Pskew = abs(price + ((-
vega/vega[_n+1])*price[_n+1]) - ((delta + delta[_n+1]*(-
vega/vega[_n+1]))*prc)) if ncp_flag ==1


                        drop if missing(Pskew)
                        sort date
                        generate Rskew = (Pskew - Pskew[_n-1])/ Pskew[_n-
1]
                        generate T = date - date[_n-1]
                        generate ExcessSkew = Rskew - (rate/1000)*T

                save `i'`j', replace


            }
}


! dir *.dta /a-d /b >"E:\Data New\Straddle\1996\filelist1.txt"

      file open myfile using "E:\Data New\Straddle\1996\filelist1.txt",
read

file read myfile line
```

```
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
save master_data, replace

local year "1997"

local nextyear = `year' + 1
set more off

cd "E:\Data New\Gefilterde Data\"

use `year'filter, clear
      append using `nextyear'filter

cd "E:\Data New\Straddle\1997\"
compress
save `year', replace

foreach i in 13532     13567 13595 13623 13651 13686 13714 13742 13777
      13805 13840 13868 13896 13931  {
            use `year', clear
                  keep if exdate == `i'
                        save `i', replace

                  foreach j in 50155022  5029  5036  5046  5048  5049  5058
      5061  5067  5083  5086  5087  5089  5097  5115  5133  5142  5169  5176
      5191  5212  5240  5252  5273  5278  5297  5309  5317  5319  5385  5399
      5402  5413  5418  5428  5435  5459  5496  5510  5520  5528  5549  5564
      5569  5579  5591  5603  5628  5644  5646  5670  5692  5693  5699  5712
      5724  5764  5782  5788  5792  5804  5811  5817  5828  5837  5854  5882
      5909  5915  5934  5944  5959  5963  5969  5973  5983  6008  6012  6021
      6029  6099  6572  6677  6740  7333  7631  7737  8042  8155  8274  8285
      8672  8747  9265  100917      100930      100972      101062
      101121      101164      101199      101201      101204      101227
      101233      101273      101275      101305      101322      101325
      101368      101375      101382      101384      101387      101397
      101508      101533      101549      101558      101578      101580
      101594      101610      101639      101669      101697      101795
      101800      101806      101828      101849      101863      101920
      101930      101966      101988      102015      102021      102041
      102042      102043      102067      102082      102088      102117
      102119      102163      102210      102244      102265      102267
      102296      102339      102349      102386      102408      102409
      102525      102561      102571      102583      102588      102660
      102733      102796      102833      102845      102927      102936
      102968      103015      103029      103037      103042      103049
      103106      103125      103157      103202      103282      103296
      103302      103313      103355      103370      103401      103402
      103404      103434      103466      103477      103498      103546
      103574      103654      103682      103736      103749      103772
```

```
      103861      103879      103912      103920      103932      103936
      103937      103969      103979      104049      104117      104118
      104124      104144      104153      104172      104286      104338
      104355      104361      104411      104508      104533      104550
      104560      104626      104628      104633      104635      104664
      104847      104878      104891      104893      104894      104939
      104958      104967      105003      105076      105119      105120
      105168      105169      105174      105206      105223      105244
      105327 105338    105340      105356      105379      105384
      105452      105512      105550      105557      105573      105581
      105588      105615      105669      105675      105688      105696
      105700      105730      105759      105776      105785      105833
      105846      105955      105976      105985      106045      106119
      106203      106247      106276      106281      106295      106329
      106553      106566      106567      106595      106629      106638
      106665      106674      106689      106713      106744      106776
      106967      106969      106975      107006      107010      107015
      107036      107044      107045      107050      107078      107079
      107107      107244      107255      107280      107297      107309
      107317      107318      107321      107335      107398      107407
      107430      107525      107544      107585      107605      107616
      107691      107693      107704      107916      107951      107952
      107975      108117      108120      108130      108156      108160
      108161      108185      108186      108196      108231      108265
      108279      108385      108429      108463      108505      108573
      108609      108617      108642      108645      108649      108670
      108716      108756      108768      108849      108888      108893
      108910      108948      108960      108962      108965      108969
      109036      109047      109085      109118      109143      109224
      109245      109274      109285      109297      109347      109396
      109447      109463      109497      109528      109557      109652
      109678      109692      109715      109724      109749      109752
      109775      109823      109848      109866      109923      109954
      109956      109965      109970      109998      110002      110086
      110117      110126      110152      110262      110337      110357
      110433      110460      110490      110576      110611      110627
      110634      110649      110707      110716      110752      110766
      110810      110860      110914      110926      110945      110950
      110967      110972      110979      110991      111008      111020
      111051      111054      111086      111206      111208      111256
      111283      111296      111298      111337      111381      111394
      111404      111459      111469      111501      111504      111537
      111652      111668      111850      111860      111861      111884
      111953      111957      112011      112015      112022      112037
      112042      112136      112142      112158      112169      112286 {
```

```
                    use `i', clear
                    keep if secid ==`j'
                    generate min = (abs(delta) -0.1)^2
                    sort ncp_flag min
                    drop if strike_price ~=strike_price[1] & ncp_flag
== 1

                    drop min
```

```
                        generate min = (abs(delta) - 0.1)^2
                        gsort -ncp_flag min
                        drop if strike_price ~=strike_price[1] & ncp_flag
== 2

                        sort date ncp_flag
                        drop if ncp_flag ==ncp_flag[_n-1]

                        capture duplicates drop

                        generate Pskew = abs(price + ((-
vega/vega[_n+1])*price[_n+1]) - ((delta + delta[_n+1]*(-
vega/vega[_n+1]))*prc)) if ncp_flag ==1


                        drop if missing(Pskew)
                        sort date
                        generate Rskew = (Pskew - Pskew[_n-1])/ Pskew[_n-
1]
                        generate T = date - date[_n-1]
                        generate ExcessSkew = Rskew - (rate/1000)*T

                save `i'`j', replace


            }
}



! dir *.dta /a-d /b >"E:\Data New\Straddle\1997\filelist1.txt"

      file open myfile using "E:\Data New\Straddle\1997\filelist1.txt",
read

file read myfile line
use `line'
save master_data, replace

file read myfile line
while r(eof)==0 { /* while you're not at the end of the file */
      append using `line'
      file read myfile line
}
file close myfile
drop if  missing( Rskew)
save master_data, replace
```

Appendix 11T:S&P500 Index Options

```
*** Index options

cd "E:\Data New\S&P500 index\"

local a "S&P500bewerkt"
```

```
generate price = (best_bid + best_offer)/2
            egen ncp_flag = group( cp_flag)

      sort date exdate strike_price ncp_flag optionid
                  save `a', replace

      rename date DATE
      rename price PRICEb

      ** Generate Option Prices -t
      foreach i in 1 2 3 4 5 {
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag optionid
                        merge date exdate strike_price ncp_flag optionid
using `a', keep(price)
      drop  _merge
      rename date date`i'
      rename price price`i'
      compress
}
      foreach i in 1 2 3  {
            replace date`i' =. if missing(price`i')
      }

      foreach i in 2 3   {
            replace date1 = date`i' if missing(date1)
      }

foreach i in 2 3   {
      replace price1 = price`i' if missing(price1)
      }

      rename DATE date
      rename PRICE price
      rename price1 price_t
      rename date1 date_t
      drop date2
      drop date3
      drop date4
      drop date5

      drop price2
      drop price3
      drop price4
      drop price5

      drop if missing(secid)

      *** added SP500 return and price
      sort date
            merge date using "E:\Data New\S&P500 index\SP500stock.dta"
                  drop if missing(secid)
                        drop _merge


*** price -t
```

```
      rename date DATE
      rename spindx SPINDX

      foreach i in 1 2 3 4 {
            generate date = DATE - `i'
                  sort date
                        merge date using "E:\Data New\S&P500
index\SP500stock.dta"

      drop if missing(secid)
      drop date
      drop  _merge

      rename spindx spindx`i'
      }

      foreach i in 2 3 4  {
      replace spindx1 = spindx`i' if missing(spindx1)
      }


      rename DATE date
      rename spindx1 spindx_t
      rename SPINDX spindx
      drop spindx2
      drop spindx3
      drop spindx4

save `a', replace

*** Filling in Missing IV
cd "E:\Data New\S&P500 index\
local a "S&P500bewerkt"


use `a', clear

preserve
      drop if ncp_flag == 2
            sort date exdate strike_price
                  save `a'call, replace

restore
preserve
      drop if ncp_flag == 1
            sort date exdate strike_price
                  save `a'put, replace

restore
      sort date exdate strike_price
            save `a', replace


use `a', clear
      rename impl_volatility IV

***IV step 1 for Calls
```

```
      sort date exdate strike_price
            merge date exdate strike_price using `a'put,
keep(impl_volatility)
                  replace  IV = impl_volatility if missing(IV) & ncp_flag
== 1
                        drop impl_volatility
                        drop  _merge

***IV step 1 for Puts
      sort  date exdate strike_price
            merge  date exdate strike_price using `a'call,
keep(impl_volatility)
                  replace  IV = impl_volatility if missing(IV) & ncp_flag
== 2
                        drop impl_volatility
                        drop  _merge

*** Rename impl_volatility IV, Sort and Save File
      rename IV impl_volatility
                  sort date exdate strike_price ncp_flag
                        save `a'IV, replace


*** Rename date to DATE and IV to impl_volatility
      rename date DATE
      rename impl_volatility IV


***IV Step 2 for Calls and Puts, goes back 10 calendar days
      foreach i in 1 2 4 5 6 7 8 9 10 {
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag
                        merge  date exdate strike_price ncp_flag using
`a'IV, keep(impl_volatility)

      drop if missing(secid)

      replace  IV =  impl_volatility if missing(IV)
      drop impl_volatility
      drop date
      drop  _merge
      }

rename DATE date
rename IV impl_volatility

save `a', replace

cd "E:\Data New\S&P500 index\
local a "S&P500bewerkt"

use `a', clear



local b "filter"
```

```
*** Create variable Spread, Spread of Midpoint, Bid Price in Percent of
Stock, Offer price of Stock Price
      generate Spread = best_offer - best_bid
      generate SpreadofMidPoint = (best_offer - best_bid)/(( best_bid+
best_offer)/2)
      generate BidPriceofStock =  best_bid / spindx
      generate OfferPriceofStock = best_offer/ spindx


*** Save and Sort
                  sort date exdate strike_price ncp_flag
                        compress
                              save `b', replace

*** Rename date DATE and best_bid BestBid
      rename date DATE
      rename best_bid BestBid


*** Day t-2 Bid Price is less than $0.50, drop
      foreach i in 2 3 4 5{
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag
                        merge  date exdate strike_price ncp_flag using
`b', keep(best_bid)

      drop if missing(secid)
      drop date
      drop  _merge

      recode best_bid .= 999
      rename best_bid best_bid`i'
}

      drop if best_bid2 <=.50
      drop if best_bid3 <=.50 & best_bid2 ==999
      drop if best_bid4 <=.50 & best_bid2 ==999 & best_bid3 ==999
      drop if best_bid5 <=.50 & best_bid2 ==999 & best_bid3 ==999 &
best_bid4 ==999

      drop best_bid2
      drop best_bid3
      drop best_bid4
      drop best_bid5

*** Day t-2 Bid Price is 0.1% or less of the price of the underlying
stock, drop
      foreach i in 2 3 4 5{
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag
                        merge date exdate strike_price ncp_flag using `b',
keep(BidPriceofStock)

      drop if missing(secid)
      drop date
      drop  _merge
```

```
      recode BidPriceofStock .= 999
      rename BidPriceofStock BidPriceofStock`i'
}
      drop if BidPriceofStock2 <=.001
      drop if BidPriceofStock3 <=.001 & BidPriceofStock2 ==999
      drop if BidPriceofStock4 <=.001 & BidPriceofStock2 ==999 &
BidPriceofStock3 ==999
      drop if BidPriceofStock5 <=.001 & BidPriceofStock2 ==999 &
BidPriceofStock3 ==999 & BidPriceofStock5 ==999


      drop BidPriceofStock2
      drop BidPriceofStock3
      drop BidPriceofStock4
      drop BidPriceofStock5

*** Day t-2 Spread is more than 25% of the midpoint, drop
      foreach i in 2 3 4 5{
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag
                        merge date exdate strike_price ncp_flag using `b',
keep(SpreadofMidPoint)

      drop if missing(secid)
      drop date
      drop  _merge

      recode SpreadofMidPoint .= -999
      rename SpreadofMidPoint SpreadofMidPoint`i'
}

      drop if SpreadofMidPoint2 >=.25
      drop if SpreadofMidPoint3 >=.25 & SpreadofMidPoint2 ==-999
      drop if SpreadofMidPoint4 >=.25 & SpreadofMidPoint2 ==-999 &
SpreadofMidPoint3 ==-999
      drop if SpreadofMidPoint5 >=.25 & SpreadofMidPoint2 ==-999 &
SpreadofMidPoint3 ==-999 & SpreadofMidPoint4 ==-999

      drop SpreadofMidPoint2
      drop SpreadofMidPoint3
      drop SpreadofMidPoint4
      drop SpreadofMidPoint5

***Day t-1 Offer price is less than Bid price, drop
      foreach i in 1 2 3 4{
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag
                        merge date exdate strike_price ncp_flag using `b',
keep(Spread)

      drop if missing(secid)
      drop date
      drop  _merge

      recode Spread .= 999
      rename Spread Spread`i'
}
```

```
      drop if Spread1 <0
      drop if Spread2 <0 & Spread1 ==999
      drop if Spread3 <0 & Spread1 ==999 & Spread2 ==999
      drop if Spread4 <0 & Spread1 ==999 & Spread2 ==999 & Spread3 ==999

      drop Spread1
      drop Spread2
      drop Spread3
      drop Spread4

*** Day t Offer price is less than Bid price, drop
      generate Spread =  best_offer - BestBid
            drop if Spread <0
                  drop Spread

*** Day t-1 Offer Price is twice the Stock Price
      foreach i in 1 2 3 4{
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag
                        merge date exdate strike_price ncp_flag using `b',
keep(OfferPriceofStock)

      drop if missing(secid)
      drop date
      drop  _merge

      recode OfferPriceofStock .= -999
      rename OfferPriceofStock OfferPriceofStock`i'

}
      drop if OfferPriceofStock1 >2
      drop if OfferPriceofStock2 >2 & OfferPriceofStock1 ==-999
      drop if OfferPriceofStock3 >2 & OfferPriceofStock1 ==-999 &
OfferPriceofStock2 ==-999
      drop if OfferPriceofStock3 >2 & OfferPriceofStock1 ==-999 &
OfferPriceofStock2 ==-999 & OfferPriceofStock3 ==-999


      drop OfferPriceofStock1
      drop OfferPriceofStock2
      drop OfferPriceofStock3
      drop OfferPriceofStock4

*** Dat t Offer Price twice the Stock Price
      generate OfferPriceofStock = best_offer /  spindx
            drop if OfferPriceofStock >2
                  drop OfferPriceofStock

*** Day t-1 Spread is more than $5
      foreach i in 1 2 3 4{
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag
                        merge date exdate strike_price ncp_flag using `b',
keep(Spread)

      drop if missing(secid)
```

```
      drop date
      drop  _merge

      recode Spread .= -999
      rename Spread Spread`i'
}
      drop if Spread1 >=5
      drop if Spread2 >=5 & Spread1 ==-999
      drop if Spread3 >=5 & Spread1 ==-999 & Spread2 ==-999
      drop if Spread3 >=5 & Spread1 ==-999 & Spread2 ==-999 & Spread3 ==-
999

      drop Spread1
      drop Spread2
      drop Spread3
      drop Spread4

*** Day t Spread is more than $5
      generate Spread =  best_offer - BestBid
            drop if Spread >=5
                  drop Spread

*** Day t-1 Spread 200% of midpoint
      foreach i in 1 2 3 4{
            generate date = DATE - `i'
                  sort date exdate strike_price ncp_flag
                        merge date exdate strike_price ncp_flag using `b',
keep(SpreadofMidPoint)

      drop if missing(secid)
      drop date
      drop  _merge

      recode SpreadofMidPoint .= -999
      rename SpreadofMidPoint SpreadofMidPoint`i'
}

      drop if SpreadofMidPoint1 >=2
      drop if SpreadofMidPoint2 >=2 & SpreadofMidPoint1 ==-999
      drop if SpreadofMidPoint3 >=2 & SpreadofMidPoint1 ==-999 &
SpreadofMidPoint2 ==-999
      drop if SpreadofMidPoint3 >=2 & SpreadofMidPoint1 ==-999 &
SpreadofMidPoint2 ==-999 & SpreadofMidPoint3 ==-999

      drop SpreadofMidPoint1
      drop SpreadofMidPoint2
      drop SpreadofMidPoint3
      drop SpreadofMidPoint4

*** Day t Spread 200% of midpoint
      generate SpreadofMidPoint = (BestBid - best_offer)/((best_offer +
BestBid )/2)
            drop if SpreadofMidPoint >=2
                  drop SpreadofMidPoint

      rename DATE date
```

```
*** add zeroyield
sort date
                merge date using "E:\Data New\zeroyield", keep(rate)
                drop if missing(secid)
                replace rate = ((((rate/100)+1)^(1/365))-1) *100



*** Generate Greeks, replace them and drop the variable
      generate P   = (exdate - date)/365
      generate DeltaCall1 = normal((ln(
spindx/(strike_price/1000))+((rate)+
impl_volatility^2)*P)/(impl_volatility*(P^0.5))) if  ncp_flag == 1
      generate DeltaPut = -1 +
normal((ln(spindx/(strike_price/1000))+(rate+
impl_volatility^2)*P)/(impl_volatility*(P^0.5))) if  ncp_flag == 2

      replace  delta = DeltaCall if missing(delta) & ncp_flag == 1
      replace  delta = DeltaPut if missing(delta) & ncp_flag == 2

      drop DeltaCall
      drop DeltaPut

      generate Vega =
spindx*normalden(((ln(spindx/(strike_price/1000))+((rate)+
impl_volatility^2)*P)/(impl_volatility*(P^0.5))))*(P^0.5)

      replace vega = Vega if missing(vega)

      drop Vega
      drop P

      ***generate return
      generate return = (price - price_t)/price_t


*** Excess Returns
      generate T =  date - date_t
      generate excess_return = (price - price_t)/price_t -T*(rate/100)


***Delta hedgde excess returns
      generate delta_return = (price - price_t)/price_t -((delta*
spindx_t)/price_t)*(( spindx -  spindx_t)/ spindx_t)
      generate delta_excess_return = ((price - price_t)/price_t -
(T*rate))-(((delta* spindx_t)/price_t)*((spindx -  spindx_t)/ spindx_t) -
T*(rate/100))


      drop if missing(secid)
      duplicates drop strike_price ncp_flag date exdate, force
      compress
```