An Analysis of Volatility

Jerry Qinghui Yu   John Peiyang Li

Oct 2018

Contents

1 Introduction

1.1 What is VIX

VIX, or CBOE Volatility Index, as defined by CBOE [2018] is a measure of implied volatility of 30-day S&P500 options (SPX) created by the Chicago Board Options Exchange (CBOE). The CBOE is the largest options exchange in the world, selling through both electronic and open outcry channels. VIX measures the volatility as priced in the S&P500 index options by the market. Near term (23 day) and Next term (>30 day) at-the-money and out-of-the-money options are weighed by time to maturity and moneyness to interpolate a 30-day price of volatility. VIX is also known as the fear index, based on its correlation with market panics in the past. Normally when market panics or crashes, VIX spikes to reflect a new-found short-term volatility. A related index is VXO, or the "original" VIX, where the underlying index is S&P100, or OEX, and only at-the-money option is considered.

With the VIX futures and options launched by the CBOE in 2004 and 2006 respectively, investors have an instrument to hedge against market volatility. Investors could include VIX options in their portfolio to shield it against an unexpected market downturn. It has since become the most successful security the CBOE has launched. Because of its success, the CBOE launched a few related indices like 9-day volatility index VXST, 3-month volatility index VXV, 6-month volatility index VXMT, Nasdaq-100 30-day volatility VXN, DJIA volatility VXD, Russell 2000 volatility index RVX.

Implied volatility is the future volatility under no arbitrage principle as priced in the option. Formulas such as Black-Scholes take account of such volatility when pricing options, and similarly, we can back off the volatility from the option price. A related, but distinct concept is that of a historical volatility, which is calculated as the standard error of S&P500 in a specific time period of the past (such as past 30 days). It is an accurate ex-post measure of volatility but is useless in hedging unforeseen market events.

1.2 Evolution of VIX

Figure 1 presents an evolution of VIX and VXO from 2004 to present. There are a few spikes along the way, on the days that S&P500 index fell sharply or unexpectedly. The largest spike is around 2008 financial crisis where the stocks fell continuously. The spike around 2011 when the US's credit rating was downgraded. A spike around summer 2015 is caused by Greek's sovereign bond crisis. Then there was a spike in 2018 due to a large correction and worries about Fed overtightening.

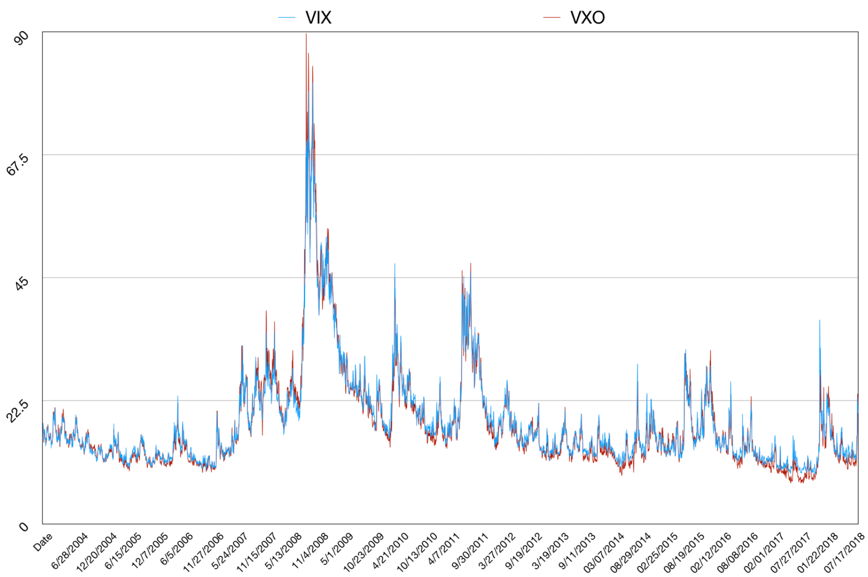

Figure 1: Evolution of VIX and VXO

Although the evolution of VIX and VXO are largely similar, there are pronounced differences around the 2008 financial crisis, as shown in figure 2, where diff = VIX − VXO. The anomaly around 2008 is in small part due to the larger diversification of S&P500 index as opposed to the S&P100. The larger part is due to VIX capturing out-of-the-money puts investors bought when the market is in free fall.

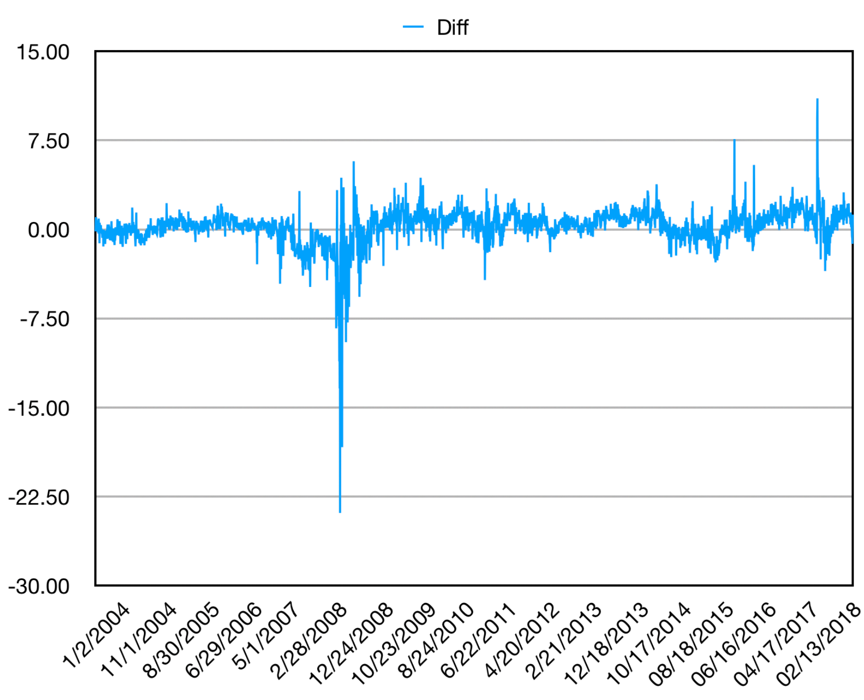

Figure 2: Evolution of the Difference

1.3 Historical Data

In this section I present statistical summary and fitness gauges with regarding to historical VIX and VXO data, Table 1 presents the first four moments

	Mean	Variance	Skewness	Kurtosis
VIX	18.45766	78.86871	2.663491	12.51215
VXO	18.18334	89.70345	2.766474	13.64604

Table 1: Moments for VIX and VXO

From the previous section we could see a clear correlation between VIX and VXO, but one might wonder how they are distributed. From the moments, it is clear that they do not fit a Gaussian distribution, which has Skewness 0 and Kurtosis 3 regardless of mean and variance. Table 1 strongly refutes that VIX and VXO are Gaussian.

Regardless, we fit a Gaussian and a Lognormal distribution to both VIX and VXO. The results are presented below. Figure 3 and 5 shows a best-fit Gaussian and Lognormal distributions to VIX and VXO respectively. From figure 4 and figure 6, we could clearly see that both VIX and VXO are not Gaussian, with Q-Q plots deviating far from the 45 degree line. On the other hand, Lognormal distribution fits the data quite well, as shown by both the best fit figures and Q-Q plots. The only exception is right tail data, which is fatter than our theoretical distribution.

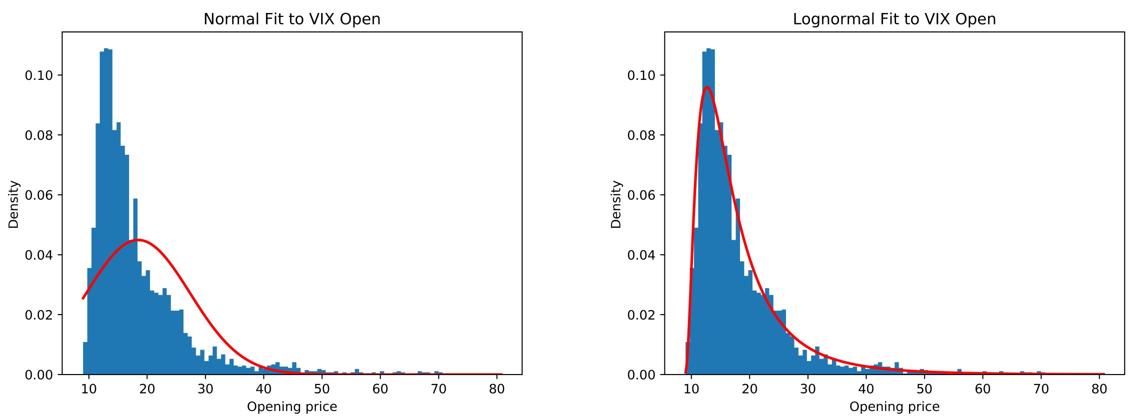

Figure 3: Best fit for VIX

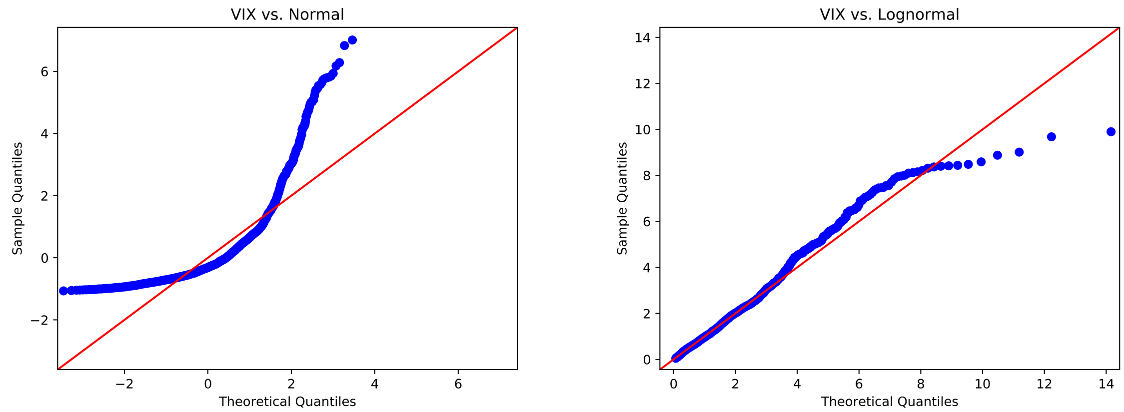

Figure 4: QQ plot for VIX

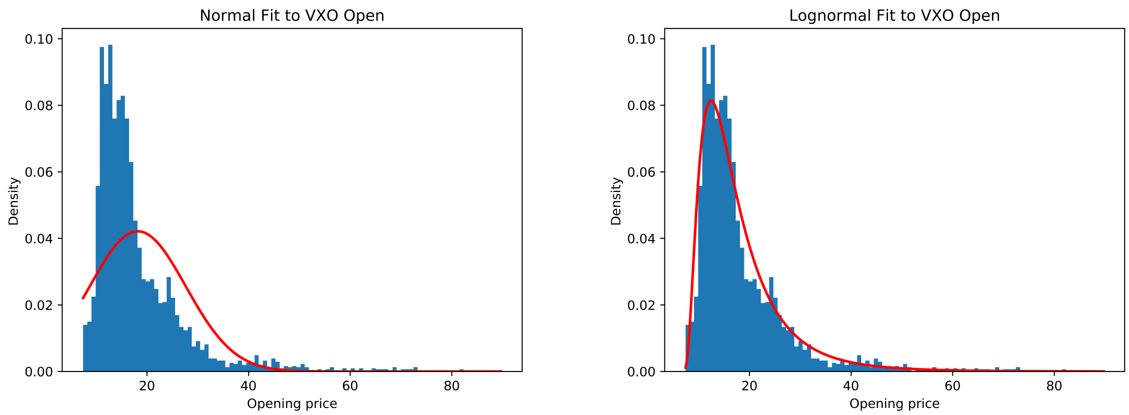

Figure 5: Best fit for VXO

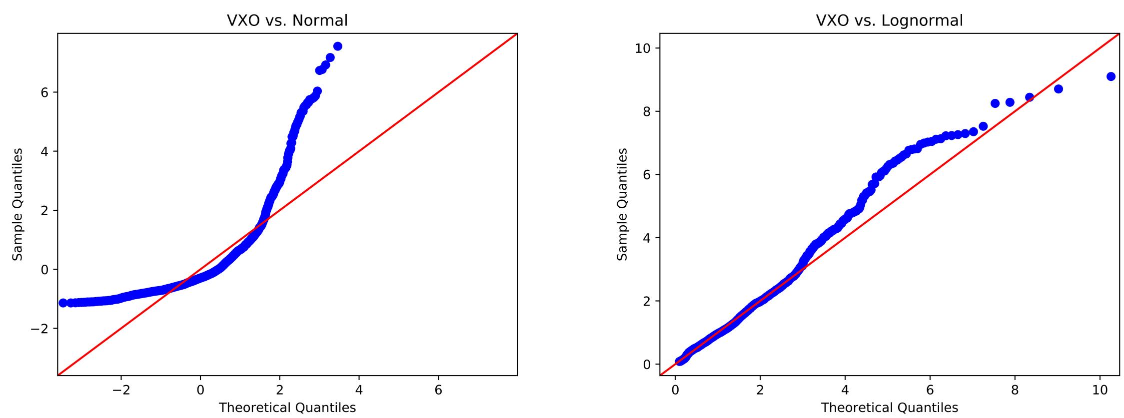

Figure 6: QQ plot for VXO

1.4 Rolling Window

In this section we present the rolling mean, variance, VaR at 25% and VaR at 75% for both VIX and VXO. See figure 7 and figure 8. We compute the 2-year rolling period by holding the number of days in each period constant, and we assume each month is 21 days, such that each 2-year period is $24 \times 21 = 504$ days. The value on x-axis represent the starting date of the time period.

As shown in the figures, the realized volatility of VIX over a 2-year window is incredibly varied as well. We observe that during the 2008 (2006 on the x-axis) financial crisis, where VIX unprecedentedly shot up, the realized variance of VIX went up, and the gap between 25%, 50% and 75% widened accordingly. We also see that outside the periods of great uncertainty, the variance is generally low.

Since participants in the capital markets trade VIX, it is only reasonable that they want to hedge against volatility of VIX. To provide a more concrete example, VIX ETNs require rolling VIX future contracts, and since they are in contango - meaning the spot price decreases the nearer to maturity - that the price required to roll the futures are significant. Thus, all VIX ETNs exhibit an inevitable downward slope. This creates an opportunity for institutions to short VIX futures to ETN providers and to reap the risk premium. Because of this, many big institutional investors are constantly shorting VIX futures. To provide hedge, they want to long the volatility of VIX, such that with a small risk premium, a large loss from their short VIX futures positions could be avoided. Knowing this, the CBOE publishes a VVIX index that tracks the implied volatility of VIX. Parties could also trade volatility swaps on VIX.

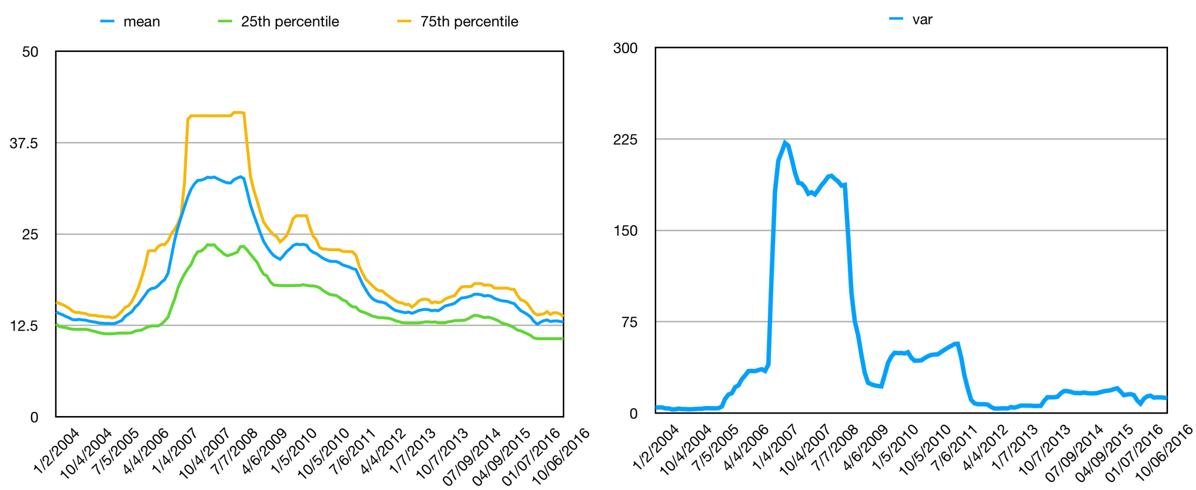

Figure 7: VIX 2-Year Rolling Stats

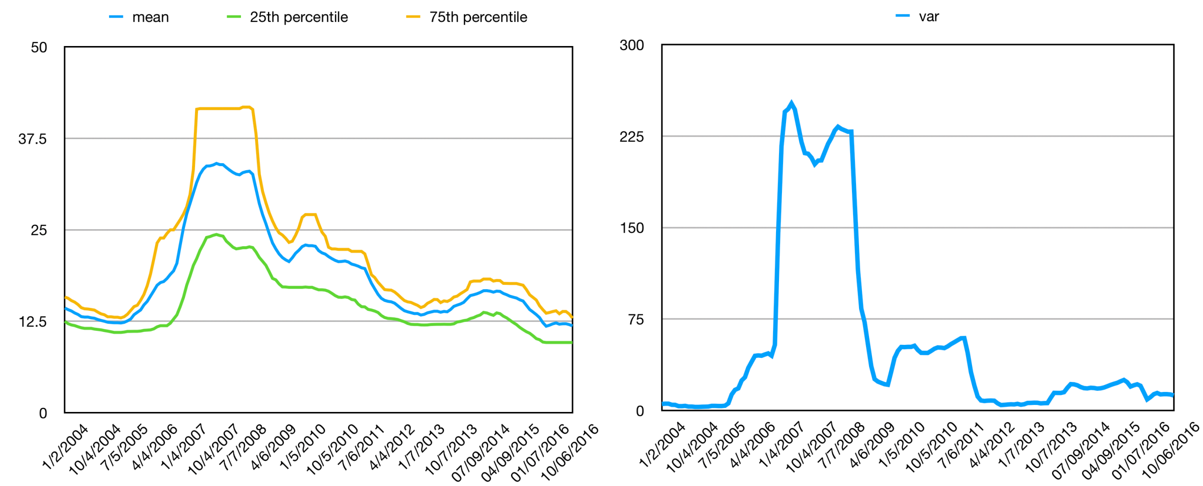

Figure 8: VXO 2-Year Rolling Stats

1.5 Comparison with VXO

As mentioned previously, there was a change of methodology in 2003 when calculating VIX. Previously the calculation is based on at-the-money S&P100 (OEX) options, then it changed to both at-the-money and out-of-the-money S&P500 (SPX) options. This change has 2 underlying reasons as stated by Whaley [2008].

1. At inception of VIX in 1993, OEX is the most actively traded options on the market, accounting for 75% of index option volume, six times that of SPX, however, as investors embraced diversification, and SPX futures and options proliferated, it has overtaken OEX in great numbers. In fact, by 2008, SPX volume is more than 12 times that of OEX.

2. Moneyness in calculation changed because when VIX is first introduced, only near-the-money options are frequently traded, however, that quickly changed as portfolio insurers bought large volumes of SPX puts to guard against losses.

These two reasons prompted the CBOE to change the methodology provide investors with an up-to-date index that reflects the investment landscapes and needs of the day.

2 Random Walk and VIX

Fama [1995] states that stock prices followed a random walk pattern where price moves are independent. As a result, any effort to predict the future movement based on the past, assuming the theory is true, would not be effective. In addition, fundamental analysts, which concerns themselves only with intrinsic value of stocks, are no better than random selection unless they have private information or insights regarding existing information that the market does not capture. In this section we lay out the mathematical terms of the theory and apply it to the VIX data of previous chapters.

2.1 Mathematics of Random Walk

Denote the price of a security at time t by p_t and let ϵ_t an independently distributed random variable that represents the price adjustments at time i when new information are revealed. Then security's price follows a random walk given initial price p_0 if $p_t = p_0 + \sum_{i=0}^{t} \epsilon_i$. We do not impose identical distribution on ϵ_i since the information revealed on i is not necessarily identical across time. It is clear that

$$\mathbb{E}(p_{t+1} \mid p_t) = p_t + \mathbb{E}(\epsilon_{t+1}) = \mathbb{E}(p_{t+1} \mid p_t, p_{t-1}, p_{t-2} \dots)$$

Meaning that the current price is as useful in predicting the future as the entire history of prices. In other words, given we already have the price today, past price information is not useful in predicting the future in any way. Thus, random walk hypothesis is a direct rebuttal to technical analysis, which assumes past patterns repeat in the future.

2.2 Autocorrelation of VIX

Define the time series $y_t = \log(VIX_t)$, we run the autocorrelation function (ACF) on the series with lag 260 to denote a financial year. If there is a seasonal pattern, it should be seen in ACF. The result is presented in figure 9. Clearly, the autocorrelation function is a decaying function, however, the decay is very slow, to the extend that the autocorrelation at lag 260 is 0.403.

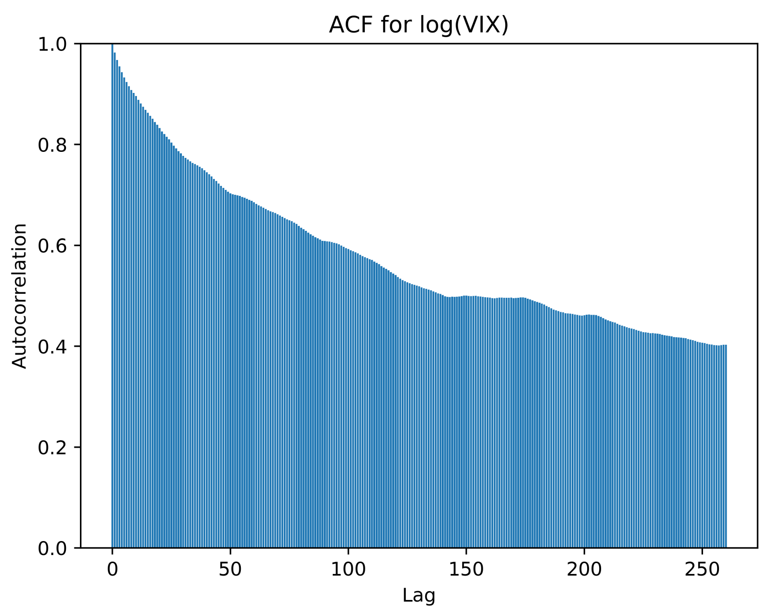

Figure 9: Autocorrelation for y_t

One might wonder, given the high autocorrelation, if a unit root is present or that the time series is not trend stationary. We apply Augmented Dickey-Fuller test for unit root and obtain a p-value of 0.0071, well below $\alpha = 0.05$, which means the null hypothesis that a unit root is present must be rejected. We also apply a Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test to complement the unit root test, and we obtain a p-value of 0.01, also below $\alpha = 0.05$. However, because H_0 for KPSS test is that no unit root exists, the result of KPSS test directly contradicts with ADF test. What gives? It turns out that there are specific examples where ADF test do not provide a convincing answer to the stationarity problem, such as $x_t = (-1)^t$. As a result, we conclude that it is not clear whether or not y_t is stationary just by looking at test results.

2.3 Fitting an ARMA Model

Regardless, we fit an ARMA model to y_t. Using a small scale model search based on the Akaike information criterion (AIC), we obtain an optimal set of order $(2, 1)$. The autoregressive coefficients are shown in table 2. Both coefficients are very significant with p-value extremely close to 0, and $\phi_1 + \phi_2 = 1.759 - 0.761 = 0.998$, extremely close to a unit root.

	Coefficient	p-value
ϕ_1	1.759	6.64e-179
ϕ_2	-0.761	9.47600231e-254

Table 2: Autoregressive coefficients for ARMA(2,1)

The residuals from our ARMA(2,1) model is plotted in figure 10. It seems that the residual is indeed a white noise process as an ARMA model would imply. But when we run Ljung–Box test for autocorrelation of residuals, we find that it is not the case. The p-values for Ljung-Box q-statistic at all lags are 0, which suggests that the residuals are serially dependent, a likely case of model misspecification.

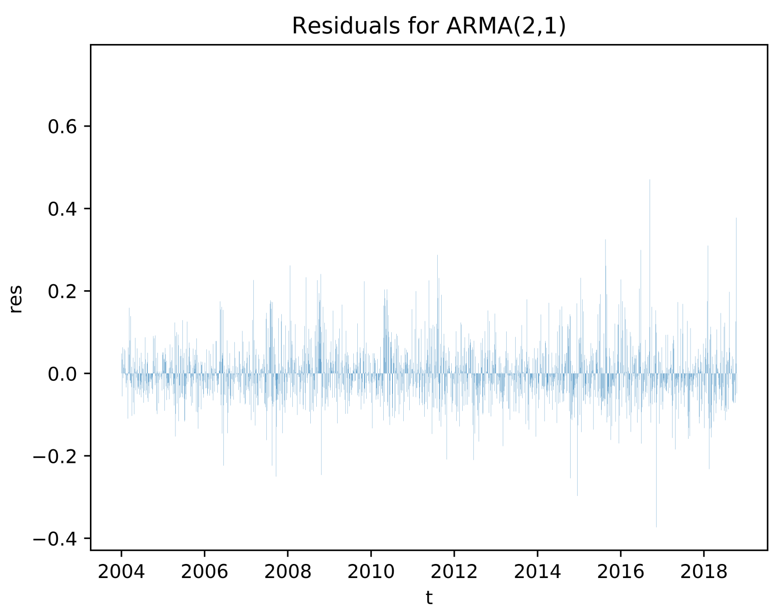

Figure 10: Residuals from an ARMA(2,1) for y_t

2.4 Differencing the Time Series

Now we integrate y_t by order 1, i.e. define a new time series $\Delta y_t = y_t - y_{y-1}$, which represents the log return of VIX. We perform the same analysis from the previous subsection on this integrated data set. The ACF is shown in figure 11. We can see that autocorrelation quickly drops to 0 compared to the ACF for y_t.

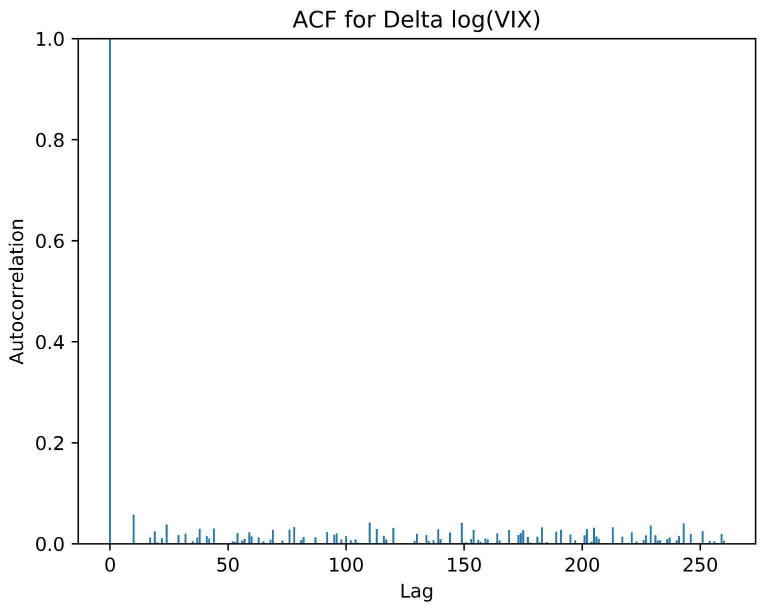

Figure 11: Autocorrelation for Δy_t

Unit root tests also confirm that Δy_t is trend stationary, ADF test gives a p-value of 0, rejecting the existence of a unit root, while KPSS test returns a p-value of > 0.1, also confirming stationarity. The coefficients from table 3 also confirms the stationarity tests, since $0.818 + 0.0186 = 0.8366 < 1$.

	Coefficient	p-value
ϕ_1	0.818	9.91680787e-01
ϕ_2	0.0186	1.16485807e-69

Table 3: Autoregressive coeffecients for ARIMA(2,1,1)

Again, Ljung–Box test rejects the null hypothesis with p-value = 0 and concludes serial dependence exists in residuals. This is likely due to model misspecification in that the linear structure of ARMA is unsuited for VIX data.

2.5 Applying EMH to VIX Returns

If the weak form of efficient market hypothesis holds, the price of a security, in our case VIX, would resemble a random walk up to a drift. In mathematical terms,

$$\begin{aligned} \log p_t &= \mu + \log p_{t-1} + \epsilon_t \\ \log p_t - \log p_{t-1} &= \mu + \epsilon_t \\ \Delta p_t &= \mu + \epsilon_t \end{aligned}$$

To test efficient market hypothesis on our VIX returns from last section, we define a regression test of VIX return on its lagged return.

$$\Delta y_t = \beta_0 + \beta_1 \Delta y_{t-1} + \beta_2 \Delta y_{t-1}^2 + \epsilon_t \tag{1}$$

If the random walk hypothesis holds, we should identify a coefficient of 0 on lagged returns and risk premium, i.e. $\beta_1 = \beta_2 = 0$. The regression results of equation 1 is presented in table 5 below, notice the both β_1, β_2 are non-zero, and that p-value for both β_1 and β_2 are very significant. Clearly there is a statistically significant, albeit near zero, short-term predictability associated with VIX. However, the more salient point here is that risk premium on VIX is in fact large and negative, meaning that the volatility is negatively correlated with short-term return. This is reasonable, considering VIX derives its utility by acting as an insurance plan to guard against volatility in a stock portfolio, and if VIX is volatile, then its insurance value is somewhat negated. We reject either of them being 0, a direct contradiction to the random walk hypothesis which dictates that past information has no value in predicting future returns.

	Coefficient	p-value
β_0	0.00096234	4.32595449e-01
β_1	-0.07106933	2.36555536e-05
β_2	-0.18976056	3.28175691e-02

Table 4: Testing Random Walk Hypothesis on VIX

2.6 Long-Term Predictability

In the previous section, we observe that in the short-term we can predict VIX to a small extent, but what if we stretch the time window to weeks, months or even years? Let $0 < b < e$, $b, e \in \mathbb{N}$ be beginning and end date, define $\Delta y_{b,e} = \Delta y_b + \Delta y_{b+1} \cdots + \Delta y_{e-1} + \Delta y_e$, which is the log-return between b and e. Let h be the prediction window length in days, then $\Delta y_{t-h+1,t}$ is the log-return of past h periods, and $\Delta y_{t+1,t+h}$ is the log-return of h periods into the future. We formulate

$$\Delta y_{t+1,t+h} = \beta_h \Delta y_{t-h+1,t} + w_{t,h}$$

Where w is a white noise. The β_h term, as a function of h, describes relationship between past return and future return, while R^2 signifies predictability. Result of the regression is presented in figure 12. We note that VIX return has a strongly cyclical pattern over period of roughly 260 days, or a year. Since the β_h is negative for all periods, VIX also exhibits strong mean reversion, especially for prediction window of 2 or 4 years, per the cyclical pattern. When it comes to R^2, it is clear that it has a upward trend with strong uncertainty. We believe that, perhaps counterintuitively, that VIX return is more predictable over the long run because of its mean reversion property. This is a result of the cyclical nature of the real economy, where credit cycles stresses the stock market during certain periods while accommodates it during others. Our assertion is supported by Fama and French [1989], who notes that the return is dependent on business conditions through the risk premium.

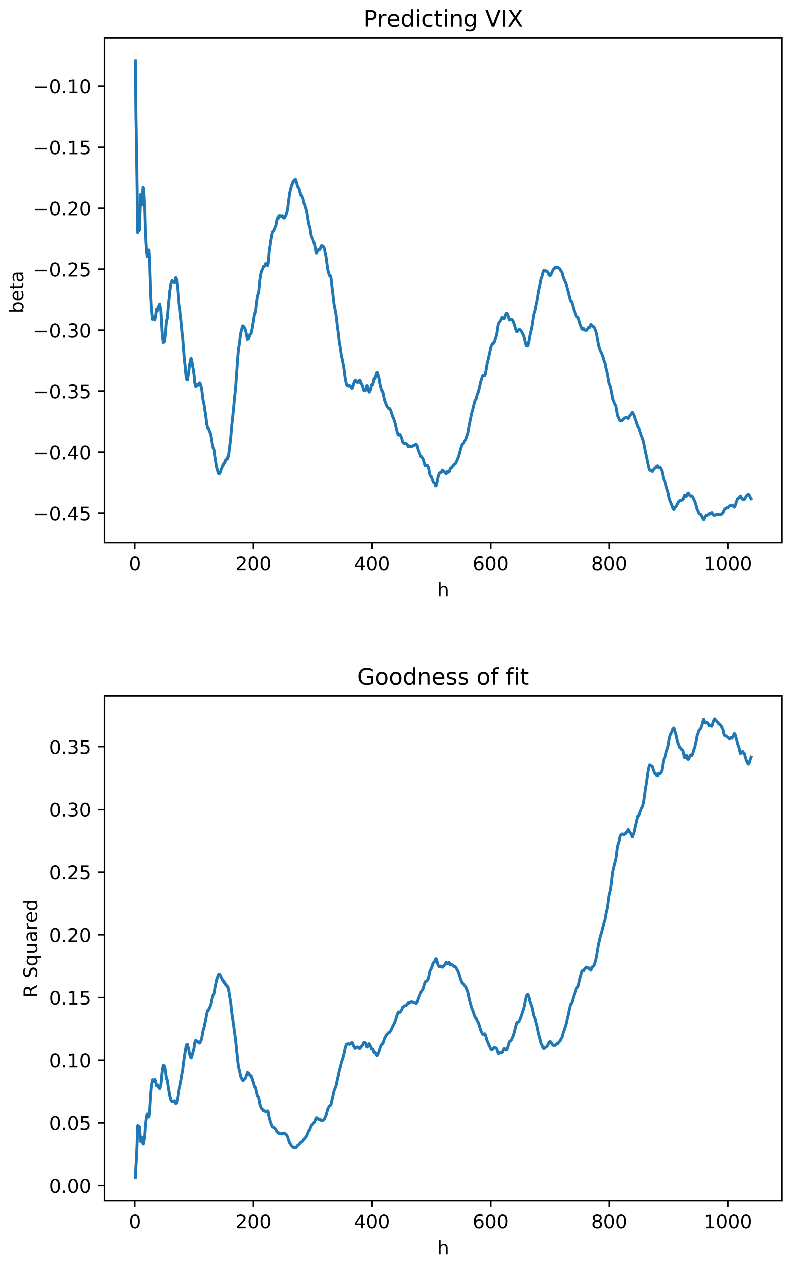

Figure 12: Effect of past VIX on future returns

3 Realized Volatility

3.1 Measures of Realized Volatility

In this section we construct two measures of realized volatility on SPX, and contrasts it with implied volatility VIX. They are defined as follows,

$$
\begin{aligned}
r_t &= \frac{SPX_t - SPX_{t-1}}{SPX_{t-1}} \\
v_{t,1} &= r_t^2 \\
v_{t,2} &= \frac{1}{25} \sum_{i=1}^{25} v_{t-i,1}
\end{aligned}
$$

The measure $v_{t,1}$, or squared daily return, is a very short term volatility measure while $v_{t,2}$ is a smoothed series by taking the 25-day moving average of $v_{t,1}$.

3.2 SPX Data

We take daily SPX closing price from CBOE and calculate r_t based on formula above, 13 plots the series r_t. The mean daily return is about 0.0343%, very close to 0. However, it does not mean that we should simply round it off and claim it is 0. Indeed, if we annualize, it is about 9%, very far from zero return. To judge whether EMH applies to SPX or not, we plot the ACF for r_t in figure 14, and it is clear from the sharp drop-off that autocorrelation isn't significant. To strengthen the claim, we ran ADF and KPSS tests, which gave p-value of 0 and > 0.1 respectively, both confirming stationarity. It is possible and from the numerical evidence, even likely, that EMH applies for SPX.

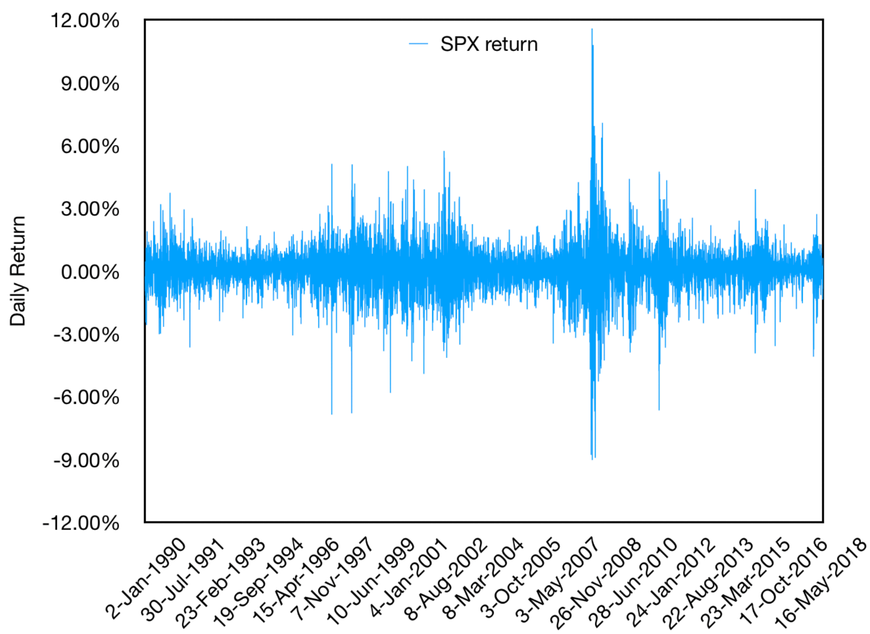

Figure 13: SPX Daily Returns

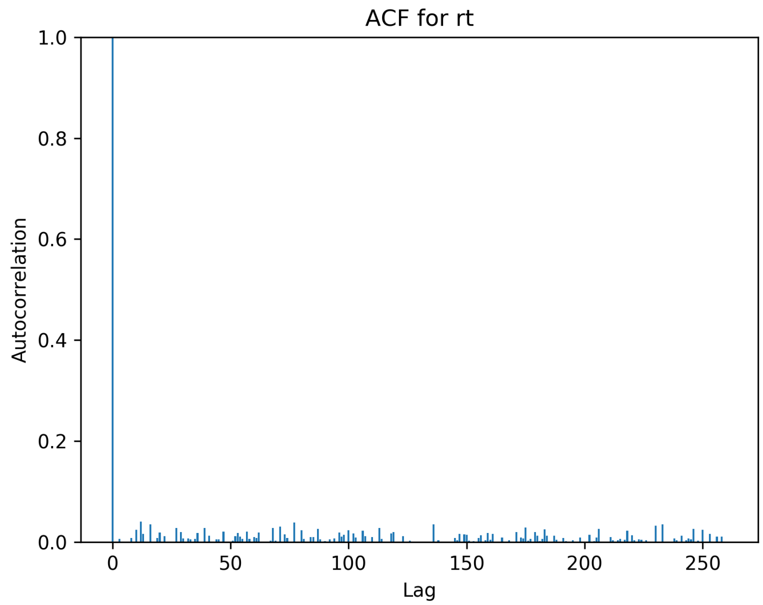

Figure 14: Autocorrelation for SPX Return

3.3 Volatility Data

The evolution of VIX, $v_{t,1}$ and $v_{t,2}$ are shown in figure 15. From the figure it is clear that up to a multiplicative factor, VIX anticipates volatility, as measured by square returns, rather well. Overall the peaks and troughs of $v_{t,1}$, $v_{t,2}$ and VIX track very closely. All measures peaked near Asian financial crisis of 1997, dot-com bubble of 2000, terrorist threat of 2001, financial crisis of 2008 and debt crisis of 2010. The daily volatility measure of $v_{t,1}$ varies quite a bit more, while the implied and smoothed measures of VIX and $v_{t,2}$ see dampened movement. From the shapes alone, we note that VIX and $v_{t,2}$ are very similar. To analyze it further, we computed the mean and variance-covariance matrix, and they are as follows

	Mean	Variance
VIX	19.3	61.61
$v_{t,1}$	1.23	16.56
$v_{t,2}$	1.23	4.71

Table 5: Mean and Variance for Volatility Measures

$$\Sigma = \begin{bmatrix} \sigma^2_{VIX,VIX} & \sigma^2_{VIX,v_{t,1}} & \sigma^2_{VIX,v_{t,2}} \\ \sigma^2_{v_{t,1},VIX} & \sigma^2_{v_{t,1},v_{t,1}} & \sigma^2_{v_{t,1},v_{t,2}} \\ \sigma^2_{VIX,v_{t,2}} & \sigma^2_{v_{t,1},v_{t,2}} & \sigma^2_{v_{t,2},v_{t,2}} \end{bmatrix} = \begin{bmatrix} 61.61 & 14.78 & 13.34 \\ 14.78 & 16.56 & 3.90 \\ 13.34 & 3.90 & 4.71 \end{bmatrix}$$

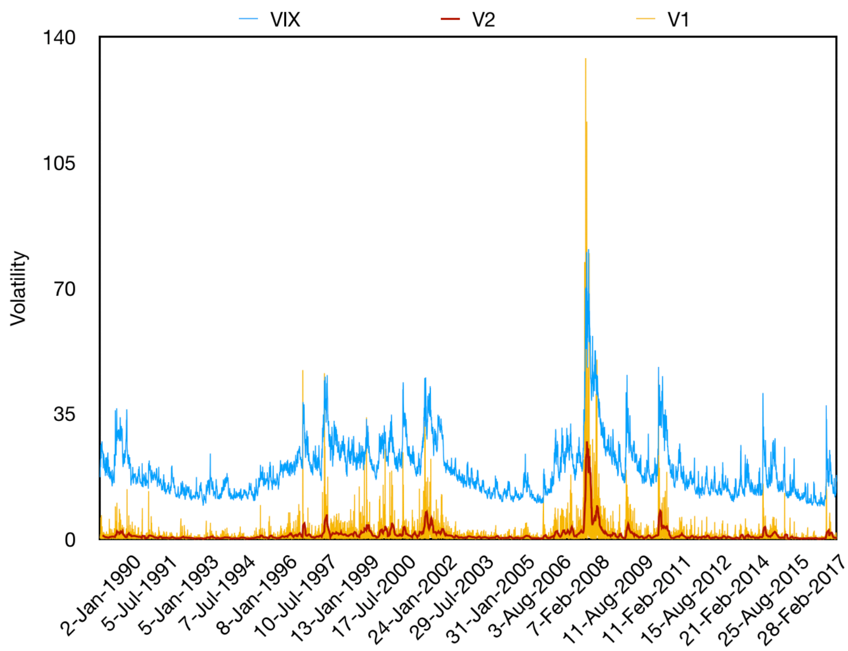

Figure 15: Evolution of Realized Volatility

Using spectral decomposition, we write $\Sigma = Q\Lambda Q^{-1}$, where Q is the matrix whose columns are eigenvectors of Σ, and Λ is a diagonal matrix where the values on the diagonal are the eigenvalues whose order is the same as the order of their corresponding eigenvectors in Q. We obtain a decomposition as follows

$$Q = \begin{bmatrix} -0.94 & -0.28 & -0.20 \\ -0.28 & 0.96 & -0.05 \\ -0.21 & -0.01 & 0.98 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 69.02 & 0 & 0 \\ 0 & 12.14 & 0 \\ 0 & 0 & 1.71 \end{bmatrix}$$

Clearly, the smallest eigenvalue is 1.71, different from a risk-free portfolio of 0. As a result, we could not have found an arbitrage portfolio such that by buying and selling it and a safe asset we obtain limitless risk-free return.

3.4 Vector Autoregression

We estimate a vector autoregression (VAR) model on the time series

$$\boldsymbol{Y_t} = \begin{bmatrix} VIX_t \\ v_{t,1} \\ v_{t,2} \end{bmatrix}$$

$$\boldsymbol{Y_t} = \boldsymbol{c} + A\boldsymbol{Y_{t-1}} + \boldsymbol{\epsilon}$$

The result is as follows

$$\boldsymbol{c} = \begin{bmatrix} 0.605 \\ -2.06 \\ -0.01 \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} 0.963 & 0.144 & 0.0003 \\ -0.0155 & -0.0185 & 0.043 \\ 0.0983 & 0.427 & 0.959 \end{bmatrix}$$

Granger causality test shows that $v_{t,2}$ and $v_{t,1}$ granger causes the other two, while VIX granger causes $v_{t,1}$. We apply the same eigendecomposition procedure on the variance-covariance matrix of the residual from the VAR model, and we obtain the following results

$$\Sigma_\epsilon = \begin{bmatrix} 2.313 & 0.677 & -0.002 \\ 0.677 & 12.915 & 0.045 \\ -0.002 & 0.045 & 0.02 \end{bmatrix} Q_\epsilon = \begin{bmatrix} 0.063 & 0.998 & 0.002 \\ 0.998 & -0.063 & -0.004 \\ 0.003 & -0.002 & 1 \end{bmatrix}, \ \Lambda_\epsilon = \begin{bmatrix} 12.96 & 0 & 0 \\ 0 & 2.27 & 0 \\ 0 & 0 & 0.019 \end{bmatrix}$$

The smallest eigenvalue is $\lambda_3 = 0.019$, substantially closer to 0 than we previously observed. Does it mean we have an arbitrage opportunity this time? To answer this we must note the difference between the settings by which we conduct the eigendecomposition. In the last section, the decomposition is on the volatility measures we constructed. Assuming we could trade on each of the measures, if we have an eigenvalue of 0, we are certain that there is a portfolio by which we could arbitrage against a safe asset. Here it is different as we are dealing with the residual of a VAR model and its variance-covariance matrix. A eigenvalue of 0 means that the residuals do not vary with a portfolio with allocation given by the eigenvector corresponding to the eigenvalue 0. This is in contrast with the previous notion where the level of volatilities themselves did not vary. As a result, we could either subsume the residual term into the constant term or eliminate it outright, given that residuals have zero mean. It would entail that we have removed the stochastic part and we are left with a deterministic linear trend, meaning we could predict the future levels of volatilities with certainty. However, it would still not constitute an arbitrage opportunity, as we may not make money if the price of the goods are the same, so lastly we would require the forward price of the portfolio of volatilities to diverge from our surefire prediction of the future price. In practice our eigenvalue is still different from 0, meaning that there are still risks involved and it would not constitute an arbitrage opportunity.

4 Causality and Conclusion

What do we gain from analyzing implied and realized volatilities? Do one of them predict the other, given the VAR analysis in the previous section? To answer this question, we breakdown the causality chain in VAR model in figure 16. It is clear that if we do not restrict the distribution of error term to be structural, i.e. each shock or innovation being independent of each other, determining the causality would be difficult because of the ”instantaneous causality” between the error terms.

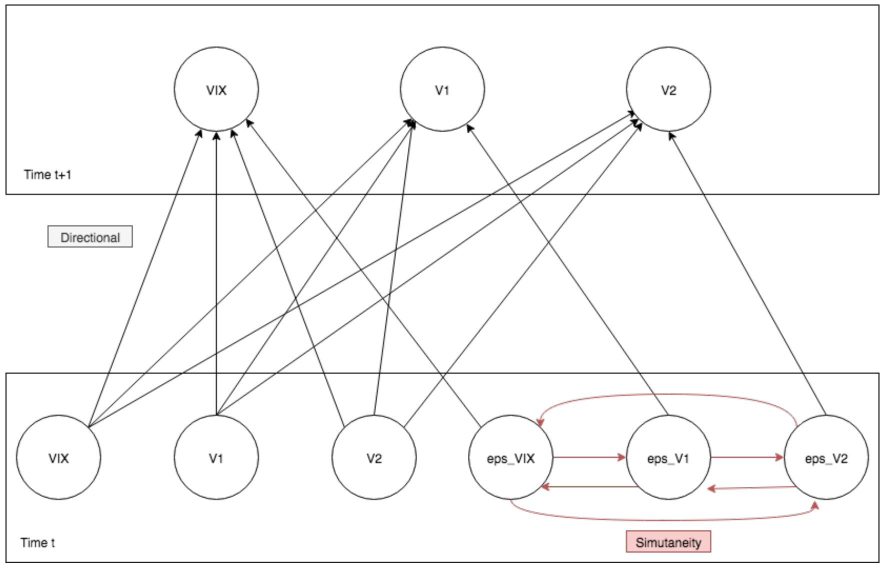

Figure 16: Chain of Influence in a VAR Model

At the end, we would like to comment on some of the differences between implied and realized volatility. Implied volatility is a measure of future volatility priced in by the market through derivatives. Realized volatility is a mathematical construct on past stock returns. Both measures are backward facing as they are dependent historical data, but Christensen and Prabhala [1998] has found that implied volatility contains more information in predicting the future compared to historical volatility. This is as we should expect, that if the option market is efficient, the price of options used to calculate the implied volatility should contain all the public information such that implied volatility reflects the best predictor of what's going to happen.

References

CBOE. Vix white paper. 2018.

B. J. Christensen and N. R. Prabhala. The relation between implied and realized volatility1. *Journal of financial economics*, 50(2):125–150, 1998.

E. F. Fama. Random walks in stock market prices. *Financial Analysts Journal*, 51(1):75–80, 1995. doi: 10.2469/faj.v51.n1.1861. URL `https://doi.org/10.2469/faj.v51.n1.1861`.

E. F. Fama and K. R. French. Business conditions and expected returns on stocks and bonds. *Journal of Financial Economics*, 25(1):23 – 49, 1989. ISSN 0304-405X. doi: https://doi.org/10.1016/0304-405X(89)90095-0. URL `http://www.sciencedirect.com/science/article/pii/0304405X89900950`.

R. E. Whaley. Understanding vix. 2008.

5 Appendix

util.py

```
import csv
import math
import numpy as np

def importvix(path):
    vix = []
    dates = []
    with open(path, 'r') as csvfile:
        reader = csv.reader(csvfile)
        for row in reader:
            dates.append(row[0])
            vix.append(row[1])
    return dates[1:], np.array([float(a) for a in vix[1:]])
```

v.py

```
import numpy as np
import csv
import scipy.stats
from util import importvix

def rolling_window_stats(infile, outfile):
    dates, vixopen = importvix(infile)
    t = 0
    with open(outfile, 'w') as f:
        writer = csv.writer(f)
        writer.writerow(['date', 'mean', 'var', '25th␣percentile', '75th␣percentile'])
        while (t+23) * 21 < len(dates):
            window = np.array(vixopen[t * 21 : (t+23) * 21])
            row = [dates[t*21], scipy.stats.tmean(window), scipy.stats.tvar(window), \
                np.percentile(window, 25), np.percentile(window, 75)]
            writer.writerow(row)
            t += 1

if __name__ == '__main__':
```

```
    rolling_window_stats('./vxocurrent.csv', 'vxostats.csv')
```

lognormal.py

```
from util import importvix
import matplotlib.pyplot as plt
import numpy as np
import csv
import math
import scipy.stats as stats

def export_fig(dist, infile, outfile, title):
    # VIX
    dates, vixopen = importvix(infile)
    count, bins, ignored = plt.hist(vixopen, 100, density=True, align='mid')
    x = np.linspace(min(bins), max(bins), 10000)
    param = dist.fit(vixopen)
    pdf = dist.pdf(x, *param[:-2], loc=param[-2], scale=param[-1])
    plt.plot(x, pdf, linewidth=2, color='r')
    plt.title(title)
    plt.xlabel("Opening price")
    plt.ylabel("Density")
    plt.savefig(outfile, dpi=750)
    plt.clf()

def main():
    export_fig(stats.norm, './vixcurrent.csv', 'vixnormal.png', \
        'Normal Fit to VIX Open')
    export_fig(stats.lognorm, './vixcurrent.csv', 'vixlognormal.png', \
        'Lognormal Fit to VIX Open')
    export_fig(stats.norm, './vxocurrent.csv', 'vxonormal.png', \
        'Normal Fit to VXO Open')
    export_fig(stats.lognorm, './vxocurrent.csv', 'vxolognormal.png', \
        'Lognormal Fit to VXO Open')

if __name__ == '__main__':
    main()
```

qqplot.py

```
from util import importvix
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as stats
import statsmodels.api as sm

dates, vixopen = importvix('./vixcurrent.csv')
ppnorm = sm.ProbPlot(vixopen, stats.norm, fit=True)
pplognorm = sm.ProbPlot(vixopen, stats.lognorm, fit=True)
fig1 = ppnorm.qqplot(line='45')
fig1.gca().set_title('VIX vs. Normal')
plt.savefig('vixqqnorm.png', dpi=750)

fig2 = pplognorm.qqplot(line='45')
fig2.gca().set_title('VIX vs. Lognormal')
plt.savefig('vixqqlognorm.png', dpi=750)
```

ar.py

```
import util
import math
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as stats
import statsmodels.tsa.stattools
import statsmodels.tsa.arima_model

def plot_acf(ts, nlags, outfile):
    autocorr = statsmodels.tsa.stattools.acf(ts, unbiased=True, nlags=nlags)
    x = range(0, nlags+1)
    plt.bar(x, autocorr)
    plt.ylim((0, 1))
    plt.title('ACF for log(VIX)')
    plt.xlabel("Lag")
    plt.ylabel("Autocorrelation")
    plt.savefig(outfile, dpi=750)
    plt.clf()

def test_stationarity(y):
    adf, pvalue_adf, nlags_adf, nobs_adf, critical_adf, \
        icbest_adf = statsmodels.tsa.stattools.adfuller(y, regression='ct')
    kpss, pvalue_kpss, nlags_kpss, critical_kpss = \
        statsmodels.tsa.stattools.kpss(y, regression='ct')
    print('Augmented Dickey Fuller test result: {0}, p-value is {1}'.format \
        (adf, pvalue_adf))
    print('KPSS test result: {0}, p-value is {1}'.format(kpss, pvalue_kpss))

if __name__ == "__main__":
    dates, vix = util.importvix('./vixcurrent.csv')
    y = np.log(vix)
    plot_acf(y, 260, 'logvixacf.png')
    test_stationarity(y)

    min_aic = float("inf")
    arma_model = statsmodels.tsa.arima_model.ARMA(y, (2, 1))
    arma_res = arma_model.fit()

    print(arma_res.arparams)
    print(arma_res.pvalues)

    plt.bar(dates, arma_res.resid)
    plt.title('Residuals for ARMA(2,1)')
    plt.xlabel("t")
    plt.ylabel("res")
    plt.savefig('armares.png', dpi=750)
    plt.clf()

    autocorr = statsmodels.tsa.stattools.acf(arma_res.resid, \
        unbiased=True, nlags=260)
    q_stat, p_value_q = statsmodels.tsa.stattools.q_stat \
        (autocorr, nobs=len(y))
    print(len(q_stat))
    print('Ljung-Box test result: {0}, p-value is {1}'.format \
        (q_stat, p_value_q))

    plot_acf(arma_res.resid, 260, 'resacf.png')
```

arima.py

```
import util
import math
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as stats
import statsmodels.tsa.stattools
import statsmodels.tsa.arima_model

def plot_acf(ts, nlags, outfile):
    autocorr = statsmodels.tsa.stattools.acf(ts, unbiased=True, nlags=nlags)
    x = range(0, nlags+1)
    plt.bar(x, autocorr)
    plt.ylim((0, 1))
    plt.title('ACF for Delta log(VIX)')
    plt.xlabel("Lag")
    plt.ylabel("Autocorrelation")
    plt.savefig(outfile, dpi=750)
    plt.clf()

def test_stationarity(y):
    adf, pvalue_adf, nlags_adf, nobs_adf, critical_adf, \
        icbest_adf = statsmodels.tsa.stattools.adfuller(y, regression='ct')
    kpss, pvalue_kpss, nlags_kpss, critical_kpss = \
        statsmodels.tsa.stattools.kpss(y, regression='ct')
    print('Augmented Dickey Fuller test result: {0}, p-value is {1}'.format \
        (adf, pvalue_adf))
    print('KPSS test result: {0}, p-value is {1}'.format(kpss, pvalue_kpss))

if __name__ == "__main__":
    dates, vix = util.importvix('../vixcurrent.csv')
    y = np.log(vix)
    deltay = y[1:] - y[:-1]
    plot_acf(deltay, 260, '../figures/deltayacf.png')
    test_stationarity(deltay)

    arma_model = statsmodels.tsa.arima_model.ARMA(deltay, (2, 1))
    arma_res = arma_model.fit()
    print(arma_res.arparams)
    print(arma_res.pvalues)

    plt.bar(dates[1:], arma_res.resid)
    plt.title('Residuals for ARIMA(2,1,1)')
    plt.xlabel("t")
    plt.ylabel("res")
    plt.savefig('../figures/arimares.png', dpi=750)
    plt.clf()

    autocorr = statsmodels.tsa.stattools.acf(arma_res.resid, \
        unbiased=True, nlags=260)
    q_stat, p_value_q = statsmodels.tsa.stattools.q_stat \
        (autocorr, nobs=len(y))
    print(len(q_stat))
    print('Ljung-Box test result: {0}, p-value is {1}'.format \
        (q_stat, p_value_q))

    plot_acf(arma_res.resid, 260, '../figures/deltaresacf.png')
```

emh.py

```
import util
import math
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm

if __name__ == "__main__":
    dates, vix = util.importvix('../vixcurrent.csv')
    y = np.log(vix)
    deltay = y[1:] - y[:-1]

    Y = deltay[1:]
    X = np.append(np.expand_dims(deltay[:-1], axis=1), \
        np.expand_dims(np.power(deltay[:-1], 2), axis=1), axis=1)
    X = sm.add_constant(X)
    model = sm.OLS(Y,X)
    results = model.fit()
    print(results.params)
    print(results.pvalues)
    print(results.f_pvalue)
```

klagreg.py

```
import util
import math
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm

if __name__ == "__main__":
    dates, vix = util.importvix('../vixcurrent.csv')
    y = np.log(vix)
    deltay = y[1:] - y[:-1]

    k = list(range(1, 1040))
    beta = []
    rsquare = []
    for lag in k:
        y = []
        x = []
        start = lag
        end = len(deltay) - lag
        for t in range(start, end):
            past = deltay[t-lag : t]
            future = deltay[t : t + lag]
            y.append(np.sum(future))
            x.append(np.sum(past))
        model = sm.OLS(y,x,hasconst=False)
        results = model.fit()
        beta.append(results.params[0])
        rsquare.append(results.rsquared)
    plt.plot(k, beta)
    plt.title('Predicting VIX')
    plt.xlabel("h")
    plt.ylabel("beta")
    plt.savefig('../figures/klagbeta.png', dpi=750)
    plt.clf()
```

```
    plt.plot(k, rsquare)
    plt.title('Goodness␣of␣fit')
    plt.xlabel("h")
    plt.ylabel("R␣Squared")
    plt.savefig('../figures/klagrsquared.png', dpi=750)
    plt.clf()
```

spxret.py

```
import csv
import math
import datetime
import numpy as np
import statsmodels.tsa.stattools
import matplotlib.pyplot as plt

def import_data(path):
    data = []
    dates = []
    with open(path, 'r') as csvfile:
        reader = csv.reader(csvfile)
        for row in reader:
            tokens = row[0].split('/')
            if len(tokens) > 1:
                dates.append(datetime.date(int(tokens[2]), int(tokens[0]), int(tokens[1])))
            else:
                dates.append(row[0])
            data.append(row[1])
    return dates[1:], np.array([float(a) for a in data[1:]])

if __name__ == "__main__":
    dates, ret = import_data('../spxret.csv')
    adf, pvalue_adf, nlags_adf, nobs_adf, critical_adf, \
        icbest_adf = statsmodels.tsa.stattools.adfuller(ret, regression='ct')
    kpss, pvalue_kpss, nlags_kpss, critical_kpss = \
        statsmodels.tsa.stattools.kpss(ret, regression='ct')
    print('Augmented␣Dickey␣Fuller␣test␣result:␣{0},␣p-value␣is␣{1}'.format \
        (adf, pvalue_adf))
    print('KPSS␣test␣result:␣{0},␣p-value␣is␣{1}'.format(kpss, pvalue_kpss))
    autocorr = statsmodels.tsa.stattools.acf(ret, unbiased=True, nlags=260)
    x = range(0, 260+1)
    plt.bar(x, autocorr)
    plt.ylim((0, 1))
    plt.title('ACF␣for␣rt')
    plt.xlabel("Lag")
    plt.ylabel("Autocorrelation")
    plt.savefig('../figures/spxretacf.png', dpi=300)
    plt.clf()
    import pdb; pdb.set_trace()
```

realizedvol.py

```
import pandas as pd
import numpy as np
from statsmodels.tsa.api import VAR

if __name__ == "__main__":
    data = pd.read_csv('../realizedvol.csv').drop('Date', 1).values
    mean = np.mean(data, axis=0)
```

```
vol = np.cov(data, rowvar=False)
evalue, evector = np.linalg.eig(vol)
model = VAR(data)
results = model.fit(1)
residual = results.resid
res_vol = np.cov(residual ,rowvar=False)
res_evalue, res_evector = np.linalg.eig(res_vol)
```