5-2013

Volatility of Volatility Structural Parameter Estimation and Subsequent Cross-Sectional Returns

Tyson Van Alfen
Utah State University

VOLATILITY OF VOLATILITY STRUCTURAL PARAMETER ESTIMATION AND SUBSEQUENT CROSS-SECTIONAL RETURNS

by

Tyson Van Alfen

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Financial Economics

Approved:

Tyler Brough Major Professor	Benjamin Blau Committee Member

Ryan Whitby
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2013

ABSTRACT

Volatility of Volatility Structural Parameter Estimation
and Subsequent Cross-Sectional Returns

by

Tyson Van Alfen, Master of Science
Utah State University, 2013

Major Professor: Tyler Brough
Department: Finance and Economics

Finance theory suggests that there is a direct positive relationship between a stock's return and that same stock's risk. Similarly, a common variable used as an attempt to quantify that risk is volatility. However, volatility almost certainly falls short of accounting for all the relevant risks that investors face. I hypothesize in this paper that the added volatility of volatility measure may help in the explanation of a stock's subsequent returns. I estimate volatility of volatility (vol of vol) by imposing a structural model on the data and then subsequently estimating the vol of vol parameter. My results show that this structural parameter estimate is unable to explain any of the subsequent (or current) stock returns, and thus fails to provide any evidence to support my hypothesis. I subsequently use a more simple estimate for the vol of vol and find that it is almost perfectly correlated with plain vanilla volatility which does instead have a significant relationship with returns.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION AND LITERATURE REVIEW

A. *Volatility and Stock Returns*

The study of how stock returns are related to volatility is nothing new: Merton estimated the relationship between the market risk premium and volatility (1980); French et al. tried to isolate the ex ante measure of variance and estimate its relationship to stock returns (1987); and many others have branched off in similar veins. Additionally, there has been an explosion of models that have sought to explain the evolution of stock prices in an environment of stochastic volatility. Perhaps the most noteworthy stochastic volatility model is the Heston model (1993). It describes the stock price movement as a stochastic process that also depends on the stochastic volatility process:

$$dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t^S \tag{1.1a}$$

$$dv_t = \kappa(\theta - v_t)dt + \xi\sqrt{v_t} dW_t^v \tag{1.1b}$$

where μ is the rate of return of the asset, κ is the rate of mean reversion, θ is the long-run mean variance, the dW's are Weiner processes, and ξ is the volatility of volatility.

These stochastic volatility (SV) models seem to be based on what we constantly observe. There are market periods of relative calm; and there are market periods of high volatility because of high levels of uncertainty. However, a couple of questions arise at this point. First, can we accurately and reliably estimate these structural parameters? And second, do these parameters help us to explain or predict the current or future stock price movements? This paper attempts to answer both of those questions.

B. Structural Parameters

According to Reiss and Wolak (2007), whether or not there exists substantial economic theory on a particular topic should aid the econometrician in deciding what type of econometric model to estimate. If economic theory is scarse on a particular topic, then the econometrician would be best served by estimating descriptive econometric models. However, when economic theory is present in abundance, the econometrician's best choice may be to estimate a structural econometric model.

Given that theoretical models of stochastic volatility are relatively abundant, I chose to estimate a structural parameter. The model that I chose to use comes from Rambharat and Brockwell and is given here:

$$S_{t+1} = S_t \cdot \exp[(r - \frac{\sigma_{t+1}^2}{2})\Delta + \sigma_{t+1}\sqrt{\Delta}[\sqrt{1-\rho^2}Z_{1,t+1} + \rho Z_{2,t+1}]] \tag{1.2a}$$

$$\sigma_{t+1} = e^{Y_{t+1}} \tag{1.2b}$$

$$Y_{t+1} = \beta^* + e^{-\alpha\Delta}(Y_t - \beta^*) + \gamma\sqrt{(\frac{1-e^{-2\alpha\Delta}}{2\alpha})}Z_{2,t+1} \tag{1.2c}$$

In this discrete time approximation model, r is the risk-free rate, σ represents the volatility, ρ is the correlation coefficient between the two random processes (Z_1 and Z_2), α is the volatility mean reversion rate, β is the volatility mean reversion level, Δ is the time step, and γ is the primary parameter of interest, the volatility of volatility.

DATA AND METHODS

The primary issue at hand is estimating the relationship between the vol of vol and stock returns. This is attempted, after estimating the parameter, by using regression analysis. In attempting to isolate the effect of vol of vol on returns, several other variables are added to the regression model. First of all, I added the raw volatility variable in order to understand the difference between these two measures. In addition to volatility, I add market capitalization, returns lagged by one month, and a measure for illiquidity.

I add market cap to the regression equation because of the past financial theory (most notably from Fama and French 1992) that suggests that since smaller firms are more risky investors demand more of a return as compensation. Smaller firms may be more risky for investors for several reasons: less analyst coverage for the investor to reference, smaller firms may on average be younger firms as well and therefore more likely to default, etc. There may also exist an inverse relationship between market cap and volatility for the reasons mentioned above. So in order to keep the error term uncorrelated with the independent variables, market cap was included in the regression analysis.

I have also added the lagged May returns to the regression equation to attempt to control for the potential momentum effect. Work by Jegadeesh and Titman (1993) suggests that lagged returns are indicative of what prices will do in the short run. So in order to control for this possible effect I also included this variable in the regression.

I added the Amihud measure of illiquidity for precisely the reasons Amihud suggests in his paper (2002). Investors obviously face more risk when buying an illiquid stock. The added risk is in the added difficulty of selling that stock back during turbulent market conditions. This is the source of the illiquidity premium that Amihud talks about. Also, illiquidity may be positively correlated with volatility; and in order to avoid violating a Gauss-Markov assumption I include it in the model.

A. *Stock Data*

The stock data comes from The Center for Research in Security Prices which I accessed via Wharton Research Data Services. From there I pulled the daily closing stock prices, daily returns, daily volume, and the shares outstanding from June 2002 to June 2007. I realize that this is a time period of above average market growth and because of this it is a biased sample. However, the next five to ten years may arguably be similar in nature depending on the strength of the continuing recovery from the recession.

Since this is an analysis on the cross-section of returns, the data up until May 2007 is primarily used to estimate the volatility parameters. The other parameters use the May and June 2007 data.

Due to the nature of calculating the vol of vol, I excluded all stock observations from my sample that came into existence after June 2002, and I also excluded all observations that had gone out of existence prior to June 2007. This probably causes my sample to suffer from survivorship bias, but it was necessary since a rather large price vector was required to get a somewhat reliable measures of the vol of vol.

B. *Parameters of Interest*

The plain vanilla volatility parameter in this cross-sectional analysis was estimated by taking the standard deviation of the daily stock returns in the sample up through May 2007.

The first estimate of the vol of vol comes from Rambharat and Brockwell's SV model. But this is discussed in more detail in the following section.

The second, and more simple, vol of vol estimate was calculated by first estimating the monthly volatility by taking the standard deviation of daily stock returns.

Then the standard deviation of all the monthly standard deviations of stock returns was calculated, and returned a single vol of vol estimate for each stock at the time of cross sectional analysis. This measure will be referred to as the *Sd of Sd* estimate from now on since it is the standard deviation of standard deviations of stock returns.

The market cap for each firm was also calculated from the CRSP data by multiplying the May 2007 price by the number of shares outstanding in May and then scaled so that it is measured in thousands of dollars.

The May 2007 return for each stock is calculated by summing up the daily returns during that month (as reported by CRSP). The June 2007 return is calculated in a similar manner.

A measure of illiquidity is included for the regression analysis. In this paper I use the Amihud measure of illiquidity which is calculated by dividing the absolute value of the May 2007 returns for each stock by the May 2007 total volume for each stock. This measure is then scaled up to make the interpretation easier.

The summary statistics for each of these measure is given below in table 1. The May and June returns are given as percentages.

Table 1: Summary Statistics

Variable	N	Mean	Median	Std Dev	Min	Max
Gamma	4793	121.14	1.9022	1171.73	0.0331	24205
Sd of Sd	4793	0.0123	0.0091	0.0133	0.0007	0.5213
Volatility	4793	0.0279	0.0235	0.0187	0.0035	0.5775
Mkt Cap	4793	4,150,545	457,778	17,439,923	283	468,519,056
May Ret	4793	2.59	1.91	11.47	-133.28	210.03
June Ret	4793	-0.65	-1.16	9.37	-62.36	88.28
Illiquidity	4793	2.0066	0.1154	11.321	2.29e-7	322.13

C. Gamma Parameter

I estimated the vol of vol by using these stock price vectors from the CRSP data and I applied the Bayesian estimation approach as laid out by Rambharat and

Brockwell in their paper (2010). This vol of vol estimate is referred to from now on as the *Gamma* estimate since it is the γ parameter that I am estimating in the Rambharat and Brockwell model.

The first step in this process is to pass the (normalized) prior parameter distributions and the stock price vector to the Markov Chain Monte Carlo algorithm[1]. The distributions of the parameters are referred to as θ, while the transformed distributions of the parameters that have been transfored to have a multi-variate Gaussian distribution are referred to as $\tilde{\theta}$. From within this algorithm (Algorithm 7) it references two other algorithms: the Kitagawa Log-Likelihood Approximation algorithm (Algorithm 6) and the Sequential Monte Carlo Estimation algorithm (Algorithm 1). After these transformed data are passed to Algorithm 7, Algorithm 6 is used in conjunction wit Algorithm 1 to compute a log-posterior value.

Posterior log-likelihoods are estimated by simulating m particles from the "distribution of the first-order autoregressive process" (Rambharat 2010). These simulations are then used to determine the appropriate weights using the conditional densities. A new sample is then drawn using probabilities proportional to the weights. After doing this process m times, a log-likelihood value is returned to Algorithm 7, which in turn uses a uniform distribution decision rule to either update or not update the posterior distribution. After that has occured 50,000 times, the inverse transformation is applied to $\tilde{\theta}$ to arrive at our estimate for the θ. However, for this analysis, I am only interested in one value of θ and that is the *Gamma* estimate.

In their paper, Rambharat and Brockwell used 500 particles and 50,000 MCMC iterations. One difficulty that was immediately present in their work was their use of R to estimate the posterior densities. Though R has some very convenient features,

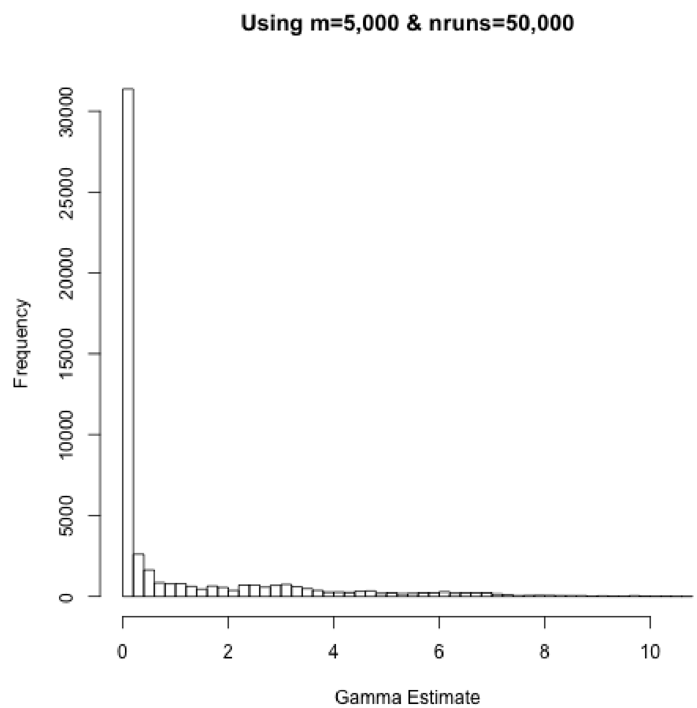
[1] I will be consistent with the algorithm referencing used by Rambharat and Brockwell. The Markov Chain Monte Carlo algorithm is Algorithm 7, the Kitagawa Log-Likelihood Approximation algorithm is Algorithm 6, and the Sequential Monte Carlo Estimation algorithm is Algorithm 1. See their paper for formal algorithm definitions.

it falls short in the efficiency race. This would prove to be a significant hurdle since I wanted to estimate vol of vol for every stock in the market. For this reason, I translated their R code into OX code. OX proved to be a valuable hybrid programming language. It has many handy features especially for matrix manipulation, and it also sacrifices little efficiency for these tools. Because of this I was able to use 5,000 particles and 50,000 MCMC iterations without losing too much in computational time. I also experimented with using 500,000 MCMC iterations expecting a more reliable vol of vol estimate. However, this increase in iterations did not result in a more accurate estimate, so I returned to using 50,000 iterations to save on computational time. It took approximately 16 hours total to run this program to obtain a vol of vol estimate for every stock in my sample at the end of May 2007.

In order to test the accuracy of the estimates I was getting, I generated a random stock price vector simulating 10 years of daily stock prices using their exact SV model, and assigning arbitrary but realistic parameters. I then used the aforementioned Bayesian parameter estimation technique to solve for the γ parameter that I had assigned[2]. Unfortunately, I quickly discovered that though the estimated parameter had an unbiased distribution, it was wildly inconsistent. I could estimate the parameter and get a value close to the correct value of 1 and then immediately estimate it again and get a value closer to 10. A histogram of the gamma estimates can be seen in figure 1. Also, oddly enough, this parameter estimate's variance was not decreased by increasing the number of simulations. This brings into question the reliability of the Rambharat and Brockwell method for finding these estimates, and we should keep this in mind when interpreting the estimates in the *Results* section.

[2]The arbitrary *Gamma* parameter that I had chosen in this instance was equal to 1.

Figure 1: Gamma Estimate Histogram



D. Methods

In order to determine the effect of volatility and vol of vol on stock returns, two different (but similar) methods are employed. The first is regression analysis, and the second is a simple ranking of the stocks by their volatility measures and then analyzing their corresponding returns.

In the regression analysis, June's returns are initially regressed on each variable individually. Subsequent regressions are then run with varying combinations of volatility measures, but all controlling for the non-volatility measures.

In order to ascertain whether the volatility measures affect the current returns, subsequent returns, both, or neither, stocks are ranked into deciles based on their volatility measures. Both the May and the June returns are then given for each decile.

RESULTS

A. *Regression Results*

Estimated coefficients and t-statistics (in parentheses) for the regressions are given in table 2. As you can see, not only was the R^2 extremely low in all cases, but the only estimates that were statistically significant were the estimated coefficients for raw *Volatility* and the *Sd of Sd*. This is rather surprising given the results that were found in the other papers previously cited. Investors during this time frame are not compensated for holding stocks that are less liquid. Nor are they compensated for investing in smaller firms. Also, lagged May returns are completely unable to explain June's returns. These results are not what I expected to see at all given the finance literature that is commonly cited in regards to these topics.

It is possible that this is a unique cross-sectional sample that does represent the general market conditions. And this would give justification to further the analysis by extending this into a time series project. However, that is beyond the scope of this paper.

Interestingly, the estimated relationship between next month's returns and the structural parameter gamma, the primary variable in question, is statistically zero. This could suggest a couple of things. First, that it is possible that the gamma estimates themselves are bad enough to be considered unusable[3]. Or it is also possible that we have good estimates for the vol of vol in this gamma parameter, but that investors are not compensated for this added risk of greater variability in volatility. This would not necessarily be a bad conclusion to reach if I had more confidence in the *Gamma* estimates.

Even when we look at the magnitude of the $\hat{\beta}_{volatility}$ in the different regression equations, we see economically meaningful values. In regression (3) we see that an

[3]This is especially likely given the difficulty in estimation that was mentioned earlier.

increase in standard deviation of 1% rewards investors, on average, with an additional 0.5% in June return[4]. This result is also seen in regressions (9) and (10).

Another interesting observation in these regressions is the interaction between *Volatility* and the *Sd of Sd* estimates. Each of these measures is approximately equal in economic magnitude when included in regressions separately. However, in regression (11), when they are both included in the model, their estimated betas deviate wildly from their previous values and their statistical significance decreases. This is probably due to the highly multicollinear relationship between the two variables. In fact, this may also be evidence that these two measures are essentially approximating the same risk. This is fleshed out in more detail in the section on multicollinearity.

B. *Heteroscedasticity*

In order to test for heteroscedasticity in the regression models, I employed the White test which tests the null hypothesis of homoscedasticity. In every regression in table 2 I was able to reject that null hypothesis. The only exception to this was in regression (6). In that regression I was able to reject the null of homoscedasticity at the 10% level but not at the 5% level. In each of the other regressions, I was able to reject the null at the 5% level or less.

In order to correct for this heteroscedasticity, the regressions were rerun using White standard errors, but the results from that analysis are not included in this paper since the estimated beta coefficients were not noticeably different. The statistical significance of the estimates was also not changed in a meaningful way.

C. *Multicollinearity*

The Pearson correlation coefficients and the variance inflation factors were used to determine the amount of multicollinearity in the regression model. Table 3 reports

[4]Or about 6% in annualized terms.

Table 2: Regression Results

Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Gamma	5.7e-7						5.7e-7			4.2e-7	
	(0.49)						(0.49)			(0.37)	
Sd of Sd		.5845						.5784			-.7559
		(5.79)						(5.70)			(-2.54)
Volatility			.5108						.5083	0.508	1.018
			(7.08)						(6.99)	(6.98)	(4.77)
Mkt Cap				-9e-8			-9.3e-8	-6e-8	-3e-8	-3e-8	-1e-8
				(-1.25)			(-1.20)	(-0.77)	(-0.39)	(-0.39)	(-0.14)
May Ret					.0024		0.003	.0067	.0074	0.0074	.007
					(0.20)		(0.25)	(0.57)	(0.63)	(0.63)	(0.60)
Illiquidity						0.0002	0.0002	.0001	.0001	0.0001	.0002
						(1.35)	(1.30)	(1.17)	(1.20)	(1.20)	(1.27)
Intercept	-.006	-.0137	-.0207	-.006	-.007	-.007	-0.007	-.014	-.021	-0.021	-.026
	(-4.80)	(-7.45)	(-8.56)	(-4.36)	(-4.70)	(-4.94)	(-4.50)	(-7.15)	(-8.31)	(-8.31)	(-8.12)
Adj. R^2	.000	.0067	.0102	.0001	.000	.0002	0.000	.0066	.01	.0098	.0111

Table 3: Pearson Correlation Coefficients

	Gamma	Sd of Sd	Volatility	May Ret	Jun Ret	Mkt Cap	Amihud Ill.
Gamma	1.00	0.0124	0.0176	0.0098	0.0071	-0.0061	-0.0074
		(0.3915)	(0.2221)	(0.4978)	(0.6243)	(0.6730)	(0.6105)
Sd of Sd		1.00	0.9403	-0.0569	0.0833	-0.0788	0.0264
			(<.0001)	(<.0001)	(<.0001)	(<.0001)	(0.0676)
Volatility			1.00	-0.0569	0.1018	-0.1185	0.0198
				(<.0001)	(<.0001)	(<.0001)	(0.1701)
May Ret				1.00	0.0030	0.0359	-0.0085
					(0.8379)	(0.0130)	(0.5542)
Jun Ret					1.00	-0.0181	0.0195
						(0.2114)	(0.1783)
Mkt Cap						1.00	-0.0402
							(0.0054)
Amihud Ill.							1.00

Table 4: Variance Inflation Factors

Parameter	Regression 10	Regression 11
Gamma	1.00	
Sd of Sd		8.72
Volatility	1.02	8.78
Mkt Cap	1.02	1.02
May Ret	1.00	1.00
Amihud Ill.	1.00	1.00

the correlation coefficients matrix and table 4 reports the variance inflation factors for regressions 10 and 11.

As is evident from table 3 the only source of severe multicollinearity comes from the inclusion of both *Sd of Sd* and *Volatility*. With a Pearson correlation coefficient of 0.94 it is safe to say that they are pretty much measuring the same thing. For this reason, regressions (8) or (9) should be preferred to regression equation (11).

Similar conclusions can be drawn from looking at the variance inflation factors: there exists very little collinearity in all cases except for the relationship between volatility and Sd of Sd.

Another interesting thing to note is the negative and significant relationship between volatility and May returns, and the positive and significant relationship between volatility and June returns. I will delve deeper into a possible explanation in a subsequent section.

One final thing that needs to be mentioned has to do with the correlation between *Gamma* and *Sd of Sd*. Notice that there is essentially zero correlation between the two variables. This again could be hinting towards the idea that these variables are not measuring what they were intended to measure[5].

D. *Volatility Rankings*

To further illustrate the relationship between the two vol of vol measures and returns, included below are tables 5 and 6 that show the May and June stock returns sorted by *Gamma* rank and also sorted by *Sd of Sd* rank (with higher rank meaning a larger vol of vol). Figures 2 and 3 plot the data in table 5. Similarly, figures 4 and 5 plot the data in table 6.

Table 5: Mean Returns (in percent) by Gamma Rank

Rank	May	June
1	2.27	-1.16
2	2.18	-0.92
3	2.69	0.06
4	2.64	-0.57
5	3.16	-0.72
6	2.50	-0.98
7	2.52	-0.44
8	2.51	-0.38
9	2.86	-0.36
10	2.58	-0.99

It is interesting to note that while there appears to be no obvious trend to the relationship between *Gamma* and returns (in either May or June), this is not the case for the relationship between *Sd of Sd* (or similarly *Volatility*) and returns (in both

[5]This should be specifically applied to the *Gamma* estimates; but it can also be argued that the *Sd of Sd* parameter is unreliable as well.

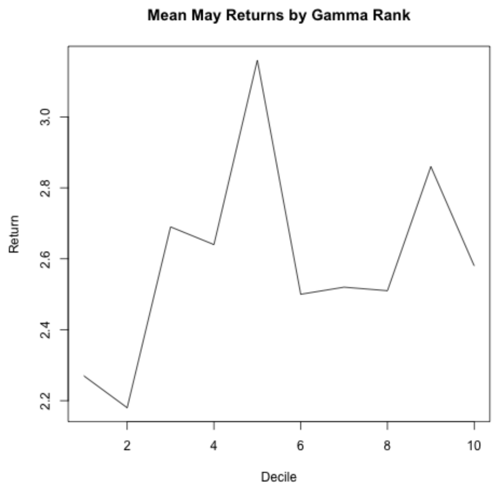


Figure 2: May Gamma

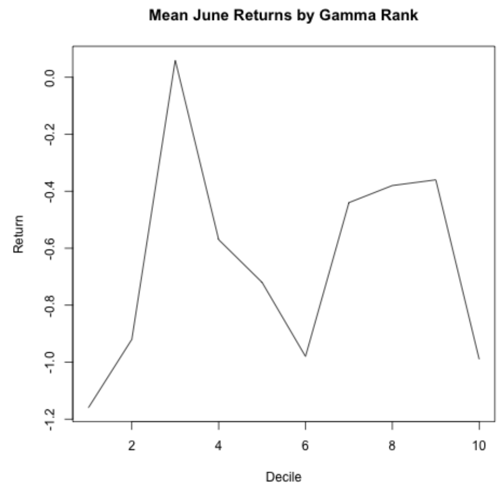


Figure 3: June Gamma

May and June).

In figure 4 we see what appears to be a quadratic relationship between *Sd of Sd* and returns. During the month of May — which happened to be a month with above average market returns — stocks with above average vol of vol and below average vol of vol underperformed the middle deciles by about 300 basis points. However, during the month of June — which happened to be a month with below average market returns — we seem to see a more linear effect. Stocks with the lowest amount of vol of vol underperformed stocks with the highest vol of vol, again by about 300 basis points.

These results are interesting beecause they are counterintuitive. Assuming figures 4 and 5 are an accurate sample of the market as a whole, figure 5 would suggest that the investor buy the highest decile of vol of vol and short the lowest decile when anticipating a market downturn. Similarly, figure 4 suggests that the investor should buy the middle deciles of vol of vol and avoid the highest and lowest deciles when anticipating a bull market.

Table 6: Mean Returns (in percent) by Sd of Sd Rank

Rank	May	June
1	0.46	-3.26
2	2.78	-1.23
3	3.40	-1.43
4	3.49	-0.94
5	3.86	-0.52
6	4.21	-0.39
7	2.29	-0.17
8	2.58	0.23
9	2.35	0.71
10	0.49	0.53

Figure 4: May Sd of Sd

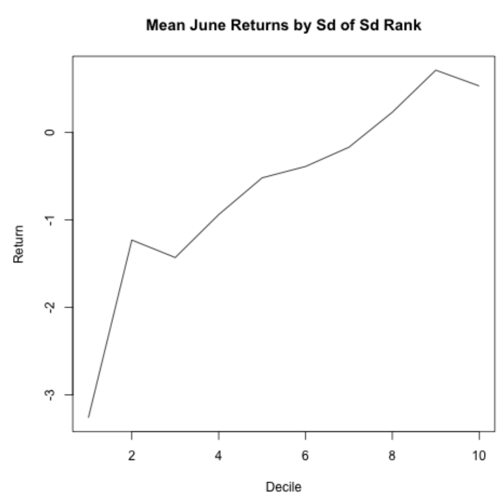


Figure 5: June Sd of Sd

CONCLUSIONS

So after this analysis, I can say with relative confidence that I was unable to estimate the γ parameter from the Rambharat and Brockwell SV model with any degree of accuracy. However, there may be something unique about this model that makes it especially difficult to estimate. It it possible that other models may be easier and more reliable — such as the Heston model.

Also, surprisingly, in June 2007 we can say that investors were not compensated for buying illiquidity, nor were they compensated for buying stocks of smaller companies. In addition, during that time period, no momentum effect was observed. These results are rather interesting given the pre-existing literature that suggests that these effects do in fact exist.

The validity of the second measure of vol of vol that I used was also put into a dubious light after this analysis — particularly because of the rather high correlation coefficient between *Sd of Sd* and *Volatility*.

Also, though the positive and significant regression estimate on the *Volatility* variable is consistent with risk/return theory. It does run counter to the results obtained by Ang et al. (2006). However, it is very possible that this is simply due to the fact that they used idiosyncratic volatility and I used a measure of raw volatility. Further research is required in this area.

In conclusion, though the final results were not what I was expecting to see, it opens up doors for future research. I may use other methods of estimation with other models in order to get a better measure of the vol of vol. In addition, I can expand this analysis to look at other time periods and other stock samples that suffer from less bias. The conclusion is far from reached.

Appendix A - Ox Code

```
#include <oxstd.h>
#include <oxfloat.h>  // so I can use pi and other constants
#include <oxdraw.h>  // useful for creating graphics
#include <oxprob.h>  // to use the ranbinomial() function

equityfilter(retvec, thetavec, delta, sqrtdelta, m);
equitymcmc(datavec, delta, m, nruns, ndim, vcovarr, priorarr, ivalues);
sample(particles, n, wts);

main()
{
//       println
    ("-----------------------------------------------------------------");
        /*
          DEFINITIONS OF VARIABLES
          retvec => ln-returns of the equity prices
          thetavec => (mu, rhotilde, alphatilde, beta, gammatilde) --
              see section 5.1 of paper
                        Notice that mu is the statistical (real-world)
                            drift of the price process.
                        Risk neutral requires that mu = risk-free rate
                        Also notice that the parameters are on the
                            transformed (tilde) scale.
          delta => time step (i.e. 1/252 for daily steps)
          m = number of SMC particles

          datavec => actual share prices on the original scale
          nruns => number of MCMC iterations
          ndim => number of parameters
          vcovarr => scaling matrix for our Gaussian proposal
              distribution
          priorarr => parameters of prior distribution
          ivalues =>  initial values for parameters (taken to be prior
              mean)
        */
```

```
        decl alist, datavec, datavect, delta, m, nruns, ndim, vcovarr,
            priorarr, ivalues, seed;
        alist = arglist(); // this is so that I can pass a file to the
            program from the command line
        println("alist: ", alist);

        datavect = loadmat(alist[1]); // load the matrix from the file
            into datavect
        datavec = datavect[][1];
//      println("datavec: ", datavec[0:10]);

        // Setting up the initial parameters
        delta = 1/252;
        m = 5000; // 500
        nruns = 50000; // 5000
        ndim = 5;
        vcovarr = <0.001, 0, 0, 0, 0; // mu, rho, alpha, beta, gamma
                        0, 0.001, 0, 0, 0;
                        0, 0, 0.005, 0, 0;
                        0, 0, 0, 0.0025, 0;
                        0, 0, 0, 0, 0.001>;
        priorarr = <.01, .01, .01, .01, .01; // mu, rho, alpha, beta,
            gamma
                              1, 1, 1, 1, 1>;
        ivalues = <.01, .01, .01, .01, .01>; // mu, rho, alpha, beta,
            gamma

        // Setting the seed for the RNG
        decl tim, time1, time2, time3, time4, time5, time6;
        tim = time();
//      println(tim);
        time1 = tim[0];
        time2 = tim[1];
        time3 = tim[3];
        time4 = tim[4];
        time5 = tim[6];
```

```
        time6 = tim[7];
        seed = time3 * time4 * time5 * time6;
//      seed = 1;  // use this if I want to use the same random numbers
        ranseed(seed);

        decl final;
        final = equitymcmc(datavec, delta, m, nruns, ndim, vcovarr,
            priorarr, ivalues);

//      println("  mean mu  |  mean rho  |  mean alpha  |  mean beta  |
    mean gamma");
//      println(meanc(final));
//      println("Standard Deviations:");
//      println(sqrt(varc(final)));

        decl mu, rho, alpha, beta, gamma, name, name1, tempfile;
        mu = meanc(final[][0]);
        rho = meanc(final[][1]);
        alpha = meanc(final[][2]);
        beta = meanc(final[][3]);
        gamma = meanc(final[][4]);

        // This code is for outputting the value of gamma to a .txt file
             named after the permno.
        name = string(alist[1]);
        name1 = name[0:12] + "txtfiles/" + name[21:25];
        name1 = name1 + ".txt";
        println("name: ", name1);
        savemat(name1, gamma, 1);

        // This code is for outputting a vector of estimates for each
            parameter.
//      savemat("muvec2.txt", mu, 1);
//      savemat("rhovec2.txt", rho, 1);
//      savemat("alphavec2.txt", alpha, 1);
//      savemat("betavec2.txt", beta, 1);
//      savemat("gammavec2.txt", gamma, 1);
```

```
}

// This is a function to draw from a sample with differing weights
sample(particles, n, wts)
{
        decl nsides, newsample, pns, prob, i, j, mass, draw;

        nsides = columns(wts);
        newsample = zeros(1, n);
        pns = zeros(1, n);

        for(i=0; i<n; i++)
        {
                mass = 1;
                for(j=0; j<nsides; j++)
                {
                        prob = wts[j] / mass;
                        draw = ranbinomial(1, 1, 1, prob);
                        if(draw > 0)
                        {
                                newsample[i] = j;
                                break;
                        }else{
                                mass = mass - wts[j];
                        }
                }
                pns[i] = particles[newsample[i]];
        }
        return pns;
}

// This is a function to compute the log-likelihood of stochastic
    volatility models under the GOU volatility process
equityfilter(retvec, thetavec, delta, sqrtdelta, m)
{
        // declare all necessary variables
        decl outmat, nsteps, halfpi, muval, rhoval, alphaval, betaval,
```

```
  gammaval, ifiltermean, ifiltersd, oldfilter,
        filtermean, filtersd, rhoconst, likevec, filterdraws,
            newfilter, predvol, sigma, wtmean, wtsd, wtvec,
        temp, bootsmc, lnlikevals, lnlikeout;

// Generic and Parameter Inputs
nsteps = columns(retvec);  // filter iterations => nsteps - 1
halfpi = M_PI / 2;
muval = thetavec[0];  // GBM (statistical) drift
rhoval = atan(thetavec[1]) / halfpi;  // Correlation
alphaval = exp(thetavec[2]);  // ln-vol mean reversion rate
betaval = thetavec[3];  // ln-vol mean reversion level
gammaval = exp(thetavec[4]);  // ln-vol volatility

// Initial particles come from stationary distribution
ifiltermean = betaval;
ifiltersd = gammaval / sqrt(2 * alphaval);
oldfilter = rann(1, m);
oldfilter = oldfilter * ifiltersd + ifiltermean;

// Filter sd to be used below
filtersd = gammaval * sqrt((1- exp(-2 * alphaval * delta)) / (2
    * alphaval));

// Constant involving rho
rhoconst = sqrt(1 - rhoval * rhoval);

// Log-likelihood results
likevec = zeros(1, nsteps-1);

// Filter Loop
decl i, j;
for(i=1; i<nsteps; i++)
{
        // Posterior and Predictive Calculations
        filtermean = betaval + exp(-alphaval * delta) * (
            oldfilter - betaval);
```

```
filterdraws = rann(1, m);
newfilter = filtermean + filtersd * filterdraws;

// Predictive Distribution
predvol = exp(newfilter);  // p(sigma(n+1) | D(n), theta
    ) estimate
sigma = predvol;  // See filter draws above

// Weights: uses Euler discretization for both mean and
    sd
wtmean = (muval - (sigma .* sigma / 2)) * delta + sigma
    * rhoval * sqrtdelta .* filterdraws;
wtsd = sigma * rhoconst * sqrtdelta;

// Note: a very small value for wtsd may cause the
    likelihood function to spike above 1
// Hence the log-likelihood could be greater than 1

// Original solution
temp = (retvec[i] - wtmean) ./ wtsd;  // retvec[i] needs
     to be evaluated at every wtmean & wtsd (i.e. "./")
wtvec = densn(temp);
println("wtvec: ", wtvec);
// Resample particles
bootsmc = sample(oldfilter, m, wtvec);
println("bootsmc");

// Shifted Log Weights solution in R code

// Traditional SMC result
oldfilter = bootsmc;

// KDE step: helps to "replenish" particles
// Included in R code

// For the log-likelihood computation
likevec[i-1] = meanr(wtvec);
```

```
        }
        lnlikevals = log(likevec);
        lnlikeout = sumr(lnlikevals);
}

// This is a function to estimate the posetrior distribution of the
    model parameters
equitymcmc(datavec, delta, m, nruns, ndim, vcovarr, priorarr, ivalues)
{
        decl lndatavec, returnsvec, halfpi, sqrtdelta, posterior, outmat
            , mumean, musd, rhomean, rhosd,
                alphamean, alphasd, betamean, betasd, gammamean,
                    gammasd, currentvec, indepmean, indepsd,
                varvec, currentlikevalue, priorcurrentmu,
                    priorcurrentrho, priorcurrentalpha,
                    priorcurrentbeta,
                priorcurrentgamma, currentpostvalue, temp, i, samplevec
                    , samplelikevalue, priorsamplemu,
                priorsamplerho, priorsamplealpha, priorsamplebeta,
                    priorsamplegamma, samplepostvalue, lnpostdiff,
                logunif;

        // Log and Returns Data Transformation
        lndatavec = log(datavec);
        returnsvec = diff0(lndatavec, 1);
        returnsvec = returnsvec[1:];

        // Generic setup
        halfpi = M_PI / 2;
        sqrtdelta = sqrt(delta);
        posterior = zeros(nruns+1, ndim);
        outmat = zeros(nruns+1, ndim+3);

        // GBM drift parameter: Gaussian prior
        mumean = priorarr[0][0];
        musd = priorarr[1][0];
        // Transformed correlation parameter: Gaussian prior
```

```
rhomean = priorarr[0][1];
rhosd = priorarr[1][1];
// Transformed ln-OU mean reversion rate parameter: Gaussian
    prior
alphamean = priorarr[0][2];
alphasd = priorarr[1][2];
// Ln-OU mean reversion level parameter: Gaussian prior
betamean = priorarr[0][3];
betasd = priorarr[1][3];
// Transformed ln-OU volatility: Gaussian prior
gammamean = priorarr[0][4];
gammasd = priorarr[1][4];

// To be used when using independent sampler
currentvec = ivalues;
indepmean = currentvec;
varvec = diagonal(vcovarr);  // variance vector
indepsd = sqrt(varvec);

// Initial run of particle filter to get ln-posterior for
    initial current values
currentlikevalue = equityfilter(returnsvec, currentvec, delta,
    sqrtdelta, m);  // calling the equity filter function
priorcurrentmu = log(densn((currentvec[0] - indepmean[0]) /
    indepsd[0]));  // notice: normalizing, then 'dnorm', then
    log
priorcurrentrho = log(densn((currentvec[1] - indepmean[1]) /
    indepsd[1]));
priorcurrentalpha = log(densn((currentvec[2] - indepmean[2]) /
    indepsd[2]));
priorcurrentbeta = log(densn((currentvec[3] - indepmean[3]) /
    indepsd[3]));
priorcurrentgamma = log(densn((currentvec[4] - indepmean[4]) /
    indepsd[4]));
currentpostvalue = currentlikevalue + priorcurrentmu +
    priorcurrentrho + priorcurrentalpha + priorcurrentbeta +
    priorcurrentgamma;
```

```
// Fill in postmat initialization
posterior[0][] = currentvec;
// Fill in outmat initialization
outmat[0][0:ndim-1] = posterior[0][];
outmat[0][ndim] = 1;  // A/R decision: we "accept" initial point
outmat[0][ndim+1] = currentpostvalue;
outmat[0][ndim+2] = 0;  // iteration

for(i=1; i<nruns+1; i++)
{
        // Sample candidate parameter value: no covariance
           structure is included.
        samplevec = rann(1, ndim);
        samplevec = samplevec .* indepsd + indepmean;

        // Evaluate the log posterior
        samplelikevalue = equityfilter(returnsvec, samplevec,
           delta, sqrtdelta, m);
        priorsamplemu = log(densn((samplevec[0] - indepmean[0])
           / indepsd[0]));
        priorsamplerho = log(densn((samplevec[1] - indepmean[1])
            / indepsd[1]));
        priorsamplealpha = log(densn((samplevec[2] - indepmean
           [2]) / indepsd[2]));
        priorsamplebeta = log(densn((samplevec[3] - indepmean
           [3]) / indepsd[3]));
        priorsamplegamma = log(densn((samplevec[4] - indepmean
           [4]) / indepsd[4]));
        samplepostvalue = samplelikevalue + priorsamplemu +
           priorsamplerho + priorsamplealpha + priorsamplebeta
           + priorsamplegamma;

        // Grab the previous iteration's log-posterior value of
           the current parameter vector.
        currentpostvalue = outmat[i-1][ndim+1];
```

```
// Evaluate the difference
lnpostdiff = samplepostvalue - currentpostvalue;
logunif = log(ranu(1,1));

// Usual acceptance condition
if(logunif <= lnpostdiff)
{
        posterior[i][0] = samplevec[0];
        posterior[i][1] = atan(samplevec[1]) / halfpi;
        posterior[i][2] = exp(samplevec[2]);
        posterior[i][3] = samplevec[3];
        posterior[i][4] = exp(samplevec[4]);

        outmat[i][0:ndim-1] = posterior[i][];
        outmat[i][ndim] = 1;
        outmat[i][ndim+1] = samplepostvalue;
        outmat[i][ndim+2] = i;

        // MCMC update for acceptance
        currentvec = samplevec;
        indepmean = currentvec;
}
// Usual rejection condition
else{
        posterior[i][0] = currentvec[0];
        posterior[i][1] = atan(currentvec[1]) / halfpi;
        posterior[i][2] = exp(currentvec[2]);
        posterior[i][3] = currentvec[3];
        posterior[i][4] = exp(currentvec[4]);

        outmat[i][0:ndim-1] = posterior[i][];
        outmat[i][ndim] = 0;
        outmat[i][ndim+1] = currentpostvalue;
        outmat[i][ndim+2] = i;

        // MCMC update for rejection
        indepmean = currentvec;
```

```
                    }
//                  println(outmat[i][]);
            }

            // Remove the first row that is due to initialization
            posterior = posterior[1:][];
            outmat = outmat[1:][];

            return posterior;
}
```

REFERENCES

Amihud, Y., 2002. Illiquidity and Stock Returns: Cross-Section and Time-Series Effects. Journal of Financial Markets, Vol 5, Issue 1, 31-56.

Ang, A., Hodrick, R., Xing, Y., Zhang, X., 2006. The Cross-Section of Volatility and Expected Returns. The Journal of Finance, Vol 61, Issue 1, 259-299.

Fama, E., French, K., 1992. The Cross-Section of Expected Stock Returns. The Journal of Finance, Vol 47, Issue 2, 427-465.

French, K., Schwert, G., Stambaugh, R., 1987. Expected Stock Returns and Volatility. Journal of Financial Economics 19, 3-29.

Heston, S., 1993. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. The Review of Financial Studies, Vol 6, No 2, 327-343.

Jegadeesh, N., Titman, S., 1993. Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency. The Journal of Finance, Vol 48, Issue 1, 65-91.

Merton, R., 1980. On Estimating the Expected Return on the Market: An Exploratory Investigation. Journal of Financial Economics 8, 323-361.

Rambharat, B., Brockwell, A., 2010. Sequential Monte Carlo Pricing of American-Style Options Under Stochastic Volatility Models. The Annals of Applied Statistics, Vol 4, No. 1, 222-265.

Reiss, P., Wolak, F., 2007. Structural Econometric Modeling: Rationales and Examples from Industrial Organization. Handbook of Econometrics, Vol 6A, Chapter 64, 4281-4284.