STAT GR5261

FINAL PROJECT

Analysis of Volatility Modelling in VaR Estimation

Group 9 Members:
Yunxiao Zhao (yz3380)
Zhichao Liu (zl2686)
Junling Zheng (jz2985)
Yun Zhang (yz3384)
Yixuan Wang (yw3163)
Junzhi Sheng (js5300)

Instructor: Prof. Zhiliang Ying

May 5th, 2019

Analysis of Volatility Modelling in VaR Estimation

May, 2019

Yunxiao Zhao, Zhichao Liu, Junling Zheng, Yun Zhang, Yixuan Wang, Junshi Sheng
Course Instructor: Prof. Zhiliang Ying

***Abstract*—We evaluated different approaches and family of GARCH models in forecasting daily Value-at-Risk (VaR) of several types of selected financial data including single stocks, market indices and foreign exchange rates, using a variety of distributional assumptions and parameters. We find, first, the volatility models provide a better result than simple parametric approaches; The increasing sample size (training size) will improve the accuracy of the estimation, whereas forecast length in a small range seems indifferent; Also, the choice of conditional mean has almost no effect on stock return forecast. Finally, different volatility models with optimal VaR forecast are suggested for each type of financial data.**

I. Introduction

The financial world has always been risky, and understanding models used in risk management has now become more important than ever. There are many different types of risk. In this report we try to quantify the estimation of market risk, which is due to changes in market prices and returns.

The most well-known market risk measure is Value-at-Risk (VaR), which refers to a portfolio's worst outcome that is expected to occur over a predetermined period and at a given confidence level, denoted by T and $1-\alpha$, respectively. If L is the loss over the holding period T, then VaR(α) is the αth upper quantile of L. For any loss distribution, continuous or not, VaR is defined as:

$$VaR(\alpha) = inf(x : P(L > x) \leq \alpha)$$

To estimate VaR, we model selected financial data in a time series perspective. Transformation of the data is proposed as daily log return, a stylized fact $X_t = 100 \times ln(\frac{P_t}{P_{t-1}})$, where P_t is the closing price on trading day t. Logarithmic returns are useful in statistics and mathematical finance. One of the advantages is that log returns are symmetric, while ordinary returns are not. As time moves, log return with equal magnitude but opposite signs will eventually cancel out, resulting in a martingale process, which will help simplify the models.

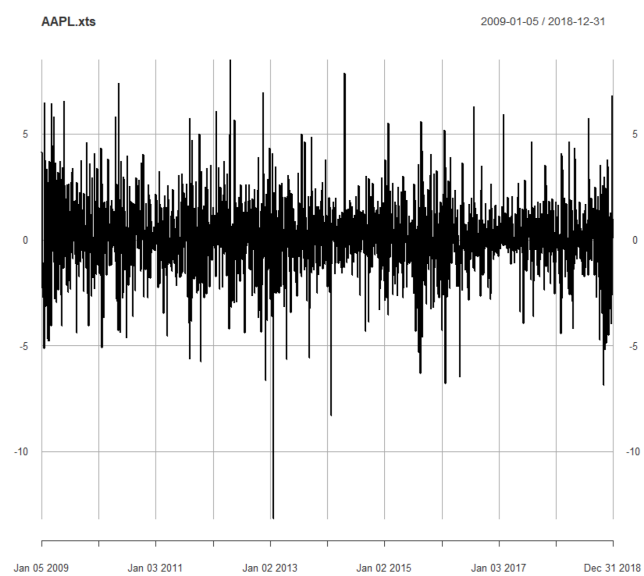

Figure 1.1: Log return time series plot of AAPL from Jan 05, 2009 to Dec 31, 2018

The purpose of this report is twofold. First to evaluate different volatility models and their forecast results on VaR estimation under risk management framework, including change of parameters such as sample size, forecast length, assumptions on innovations and order of conditional mean. Rolling forecast, which is a general back test approach in modeling time series data, is combined with VaR violation rate to assess forecast results. The results are judged in two criteria: accuracy and consistency. Second, based on comparisons among different models, optimal models will be suggested for selected financial data.

II. Volatility Models

It is often found that financial market data exhibits volatility clustering, where time series show periods of high volatility and periods of low volatility. In fact, with economic and financial data, time-varying volatility is more common than constant volatility, and accurate modeling of time-varying volatility is of great importance in financial risk management. When estimating financial VaR, the most commonly used simple parametric models, which, however, is not enough to capture the dependence between return variance since it only measures the conditional expectation of a process given the past. Thus, alternative methods, like GARCH models, should be considered if we want to make more accurate predictions.

A. ARCH

For modeling changing volatility aforementioned, (Engle, 1982) introduced the ARCH(p) process X_t as a solution of the equations

$$Z_t = \sqrt{h_t}e_t, \quad e_t \sim IID(0,1)$$

where h_t is a (positive) function of $\{Z_s, s < t\}$, defined by

$$h_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i Z_{t-i}^2,$$

with $\alpha_0 > 0$ and $\alpha_i \geq 0$, $i = 1, \cdots, p$. The name ARCH signifies autoregressive conditional heteroscedasticity. h_t is the conditional variance of Z_t given $\{Z_s, s < t\}$.

B. GARCH

The GARCH(p, q) process (Bollerslev, 1986) is a generalization of the ARCH(p) process in which the variance equation is replaced by

$$Z_t = \sqrt{h_t}e_t, \quad e_t \sim IID(0,1)$$

$$h_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i Z_{t-i}^2 + \sum_{i=1}^{q} \beta_i h_{t-i},$$

where $\alpha_0 > 0$ and $\alpha_i \geq 0, \beta_i \geq 0$, for each i. Comparing with ARCH models, GARCH models permit a wider range of behavior, in particular, more persistent volatility since the β parameter part adding a slowly decaying trend to the variance equation.

C. IGARCH

IGARCH (Integrated GARCH) models are unit-root GARCH models. An IGARCH(1, 1) model can be written as

$$Z_t = \sqrt{h_t}e_t, \quad e_t \sim IID(0,1)$$

$$h_t = \alpha_0 + \beta_1 Z_{t-1}^2 + (1-\beta_1)h_{t-1},$$

where $\alpha_0 > 0$ and $0 < \beta_1 < 1$. The major difference between GARCH and IGARCH models is that the unconditional variance of h_t, for IGARCH models, is not defined. From a theoretical point of view, the IGARCH phenomenon might be caused by occasional level shifts in volatility.

D. Innovations

For modeling purposes, it is usually assumed in addition that either

$$e_t \sim N(0,1)$$

$$\text{or } \sqrt{\tfrac{\nu}{\nu-2}}e_t \sim t_\nu, \nu > 2$$

where t_ν denotes scaled Student's t - distribution with ν degrees of freedom in GARCH family models. In the analysis of empirical financial data such as percentage daily stock returns, it is usually found that better fits to the data are obtained by using the heavier-tailed Student's t - distribution for the distribution of Z_t given $\{Z_s, s < t\}$. To verify the claim with selected data, models with both innovation assumptions are tested and conclusions are provided afterwards.

III. Evaluating Different Models

A. Rolling Forecast

To evaluate the performance of models, we introduce rolling forecast, which is a technique that allows us to predict by adding and dropping approach that creates test periods on a rolling basis. With rolling forecast, we can establish a set of periods after which to update the forecast. In practice, we separate each selected data into two portions, sample (training) and test data set. The sample data is used to fit the model and the test data is used to evaluate model forecast. Different sample data size and test data size (forecast length) are adjusted during the model assessment process in order to achieve optimal forecast results.

B. VaR Violation Rate

We choose violation test to check whether the model produces satisfying result. The violation rate, which is the proportion of real loss greater than predicted VaR, is calculated after each period's model is fitted. As VaR represents the maximum possible loss of a financial asset with a specific significance level α, the violation rate can be interpreted as probability of a true loss greater than expected maximum loss, which is generally accepted in risk management framework.

C. Assessment Criteria

The accuracy and consistency of the forecast result are considered in model assessment. For the accuracy part, the outcome should be as close as the true significance level of VaR. Besides, the variation of violation rate should be limited. As for consistency, a relevant stable violation rate sequence lies on single side of the true significance level should be preferred. If the majority of the outcome is above or below the threshold, the model tends to have overall a consistent underestimated or overestimated prediction. In this case, it would be easy to adjust forecast result by adding or dropping some numbers in practical usage.

IV. Data and Results

We randomly select 10-year financial data from 3 major categories, which are stocks, market indices and exchange rates. All data are transformed into daily log return as described before. For each type of financial data, we perform same parameter test and evaluation over different models. The results will be discussed in the following part. Table 4.1 provides a summary statistic.

	AAPL	AMZN	NFLX	MSFT	SNE
Mean	0.09935	0.13196	0.04586	0.06396	0.031436
Median	0.08965	0.09125	0.16456	0.04799	0
Maximum	8.50223	23.7402	35.22296	10.0008	16.2913
Minimum	-13.1885	-13.5325	-42.9179	-12.4578	-15.5406
SD	1.675831	2.138322	3.315306	1.562336	2.162768
	CAC40	DAX30	HSI	N225	SPY
Mean	0.06524	0.1756	0.1373	-0.0177	0.03935
Median	0.12338	-0.08	0.1365	0.1681	0.05819
Maximum	179.176	783.3996	273.9419	778.4473	6.83664
Minimum	-98.2249	-782.004	-269.634	-778.406	-6.89584
SD	13.97089	62.69725	16.36616	60.97005	1.047811
	USD/CAD	USD/CHF	USD/JPY	EUR/USD	TNX
Mean	0.004202	-0.0031	0.007156	-0.00762	0.0106
Median	0.00783	0	0	0	-0.1708
Maximum	2.85252	9.23883	3.463915	3.733324	207.9442
Minimum	-2.1882	-17.1445	-2.77221	-2.65288	-165.823
SD	0.559723	0.724934	0.619503	0.598097	15.7861

Table 4.1: Preliminary data statistic

A. Fixed vs Conditional Volatility Models

Before evaluating model parameters, we use the stock data to compare results between fixed volatility parametric models and simple volatility models. As shown in Table 4.2, the forecast results from fixed volatility models are relatively inaccurate and highly unstable for both confidence levels.

95%VaR	AAPL	AMZN	MSFT	SNE	NFLX
Normal	4.93%	3.85%	4.30%	4.16%	3.76%
t	2.35%	2.08%	2.17%	2.31%	2.08%
GARCH	6.45%	5.34%	5.56%	5.08%	4.94%
99%VaR	AAPL	AMZN	MSFT	SNE	NFLX
Normal	1.86%	1.58%	1.76%	1.90%	1.76%
t	0.41%	0.45%	0.50%	0.41%	0.36%
GARCH	1.46%	1.32%	1.28%	1.37%	1.50%

Table 4.2: Violation rates of fixed and conditional volatility models

B. Sample Size

In measuring effect of sample size, we try different size as 50, 100, 250, 500, and 1000 trading days. The result in Table 4.3 shows that as the rolling sample size increases, the violation rate becomes closer to the specified significance level, for all types of financial data.

Sample Size	95%VaR	99%VaR
50	7.32%	2.41%
100	6.32%	1.71%
250	5.82%	1.45%
500	5.53%	1.26%
1000	5.27%	1.05%

Table 4.3: Averaged violation rates for different sample size

C. Forecast Length

As for forecast length, the choices measured are 5, 10, 20, 60 and 120 trading days. The outputs show that chosen length of forecast set within a month does not have a significant effect on overall performance of models. We believe the reason behind is that the returns for most of our data do not have much unusual fluctuations during 2009 to 2019, therefore small additional length of forecast set will still be correctly predicted by the persistent variance nature of GARCH models.

Forecast Length	95%VaR	99%VaR
5	5.75%	1.42%
10	5.82%	1.45%
20	5.82%	1.45%
60	6.12%	1.56%
120	6.02%	1.61%

Table 4.4: Averaged violation rates for different forecast length

D. Assumptions on Innovations

The assumption on innovations is an important part of volatility models. When build GARCH models, we assume either normal or t innovation. As in previous, the averaged violation rate is used as a metric to evaluate the effect of innovation choices. Table 4.5 shows the result.

Normal		t	
95%VaR	99%VaR	95%VaR	99%VaR
5.47%	2.19%	5.90%	1.50%

Table 4.5: Averaged violation rates with different innovation assumptions

For 95% confidence level, the normal innovation model seems have slightly better outcomes than t innovation ones since violation rates are close to 5%. However, when it turns to 99% confidence level, models with t innovation have a much accurate result on average. The reason leads to this, we believe, should be the heavy tail characteristic of daily stock returns which is discussed above.

E. Conditional Mean

The general GARCH model contains an AMRA part which is used to measure conditional mean. To evaluate the effect of conditional mean, we experiment different ARMA parameters in producing model forecast given GARCH parameters fixed. For stocks data, the results show that ARMA part in models does not make significant difference in forecast result. One explanation for this phenomenon could be that using log transformation introduces property of martingale. Under efficient market hypothesis, the conditional expectation of log return would be uncorrelated and have a zero mean.

95%VaR					
Model	AAPL	AMZN	MSFT	SNE	NFLX
Zero Mean	6.45%	5.34%	5.56%	5.08%	4.94%
AR(1)	6.28%	5.40%	5.81%	5.37%	5.02%
AR(2)	6.12%	5.48%	5.87%	5.30%	5.04%
ARMA(1,1)	6.27%	5.58%	6.09%	5.28%	5.07%
99%VaR					
Model	AAPL	AMZN	MSFT	SNE	NFLX
Zero Mean	1.46%	1.32%	1.28%	1.37%	1.50%
AR(1)	1.39%	1.37%	1.35%	1.32%	1.33%
AR(2)	1.35%	1.37%	1.37%	1.32%	1.35%
ARMA(1,1)	1.42%	1.39%	1.40%	1.33%	1.32%

Table 4.6: Violation rates for stocks with different conditional means

F. Model Selection

With the comparison of different models, we try to discover an optimal model for selected financial data based on criteria mentioned above. However, the result suggests that there is no global optimal model for all data. According to Table 4.7 in Appendix A, we find that for all the stocks, IGARCH model (1,1) produces stable forecast violation rate closest to the threshold. As a consequence, IGARCH(1,1) model is chosen as the best model for stocks. Next, for the indices, ARCH(1,0) model has a better performance since the violation test result is more accurate. When considering exchange rates, GARCH(1,1) is the optimal model because it provides a consistent output sequence, despite all models generates relevantly accurate forecast results.

V. Conclusion

Following the extensive and detailed model assessment and comparison, briefly presented in the preceding sections, a number of conclusions are proposed, aiming to summarize our results in analysis of models.

For selected financial data described in this report, we have strong indications that the GARCH models in VaR estimation are superior than simple parametric models with constant volatility. It is obvious to see that GARCH models achieve more accurate and consistent forecast in both confidence levels, as we expected before. The size of rolling sample used in estimation turns out to be rather important. The forecast accuracy improves with an increasing sample size. Meanwhile, length of forecast is indifferent to results within a small range, which means the short time persistence in variance of our model works well for all data. Moreover, using Normal innovations yields slightly better results under 95% confidence level. However, it is never the case for 99% confidence level. Because of the heavy-tail nature in financial data, Student's t innovation are more appropriate than the Normal in estimating tail values under VaR framework, which is clearly verified by more accurate forecasts we observe. Furthermore, the mean process specification in our models seems to play no important role. The conditional mean does not add any significance to the VaR forecast other than complexity in the estimation procedure. Finally, there is no global choice of best model. Optimal models depend on type of data and indeed can vary. For practical purpose, we suggest one model for each type of financial data based on criteria introduced above.

References

[1] Robert F. Engle, "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," Econometrica, vol.50, issue 4, pp.987-1007, Jul. 1982.

[2] Tim Bollerslev, "Generalized autoregressive conditional heteroskedasticity," Journal of Econometrics, vol.31, issue 3, pp.307-327, Apr. 1986.

[3] David Ruppert, David S. Matteson, Statistics and Data Analysis for Financial Engineering(Second Edition), Cornell University, Springer, 2015, pp.307-452, 553-579.

[4] Peter J. Brockwell, Richard A. Davis, Introduction to Time Series and Forecasting(Second Edition), Colorado State University, Springer, 2002, pp.331-368.

[5] Ruey S. Tsay, Analysis of Financial Time Series(Third Edition), University of Chicago, Wiley, 2010, pp.1-174.

APPENDIX A
TABLES & SEVERAL RESULTS

	AAPL		AMZN		MSFT		SNE		NFLX	
	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
ARCH	6.14%	1.55%	5.52%	1.19%	6.00%	1.37%	4.94%	1.37%	5.25%	1.41%
GARCH(1,1)	6.31%	1.41%	5.65%	1.50%	6.05%	1.32%	5.39%	1.28%	4.90%	1.41%
GARCH(2,1)	6.31%	1.46%	5.61%	1.55%	6.23%	1.46%	5.34%	1.28%	5.03%	1.50%
GARCH(1,2)	6.36%	1.32%	5.65%	1.32%	6.09%	1.37%	5.43%	1.37%	5.08%	1.28%
GARCH(2,2)	6.23%	1.37%	5.47%	1.41%	6.09%	1.46%	5.30%	1.37%	5.08%	1.28%
IGARCH(1,1)	5.78%	1.24%	5.17%	1.19%	5.56%	1.10%	4.72%	1.15%	4.64%	1.10%
	CAC40		DAX30		HSI		N225		S&P500	
	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
ARCH	6.07%	1.69%	9.18%	2.93%	6.31%	1.58%	7.30%	2.16%	7.42%	2.08%
GARCH(1,1)	7.38%	2.08%	9.13%	2.75%	6.94%	1.94%	6.94%	2.52%	7.59%	2.30%
GARCH(2,1)	7.46%	2.17%	9.53%	2.84%	7.03%	2.12%	6.62%	2.39%	7.73%	2.34%
GARCH(1,2)	7.33%	2.17%	9.18%	2.93%	7.07%	1.80%	7.17%	2.57%	7.68%	2.43%
GARCH(2,2)	7.46%	2.21%	9.40%	2.84%	7.07%	2.03%	6.94%	2.52%	7.86%	2.34%
IGARCH(1,1)	7.25%	1.87%	8.83%	2.53%	6.94%	1.89%	6.89%	2.48%	7.28%	1.90%
	USD_CAD		USD_CHF		USD_JPY		EUR_USD			
	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR		
ARCH	4.52%	0.88%	4.73%	1.17%	5.49%	0.88%	5.01%	1.23%		
GARCH(1,1)	4.36%	0.96%	5.32%	1.42%	5.23%	1.09%	5.64%	1.15%		
GARCH(2,1)	4.44%	1.01%	5.32%	1.34%	5.19%	1.13%	5.64%	1.10%		
GARCH(1,2)	4.48%	1.01%	5.40%	1.34%	5.53%	1.05%	5.81%	1.19%		
GARCH(2,2)	4.36%	1.09%	5.32%	1.30%	5.44%	0.96%	5.77%	1.06%		
IGARCH(1,1)	4.43%	0.99%	5.22%	1.31%	5.38%	1.02%	5.57%	1.15%		

Table 4.7: Comparison among different models for each type of financial data

Stocks & 10yr Treasury Bond

Summary Statistic

Summary Statistic	AAPL	AMZN	NFLX	MSFT	SNE	TNX
Mean	0.09935	0.13196	0.04586	0.06396	0.03143	0.0106
Median	0.08965	0.09125	0.16456	0.04799	0	-0.1708
Maximum	8.50223	23.7402	35.22296	10.0008	16.2913	207.9442
Minimum	-13.18847	-13.53254	-42.91789	-12.45784	-15.54063	-165.8228
SD	1.675831	2.138322	3.315306	1.562336	2.162768	15.7861
Skewness	-0.1418577	0.843166	-0.2716893	-0.1723071	0.1455264	0.5547326
Kurtosis	3.77387	11.96672	23.65228	7.579667	4.960382	43.17545

ARMA(1,1)+GARCH(1,1)

	forecast=10	AAPL		AMZN		MSFT		SNE		NFLX		10yTsy	
	Rolling size	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
Normal	50	6.77%	3.12%	6.94%	3.53%	7.30%	2.60%	6.21%	2.88%	6.73%	3.16%	8.15%	3.73%
	100	5.96%	2.36%	5.80%	2.24%	5.71%	2.07%	5.34%	2.07%	4.55%	2.40%	6.63%	2.82%
	250	5.74%	2.12%	4.59%	1.90%	5.30%	1.94%	4.94%	2.08%	4.42%	2.21%	6.67%	2.21%
	500	5.91%	1.94%	4.67%	1.89%	4.67%	1.79%	4.47%	1.89%	3.82%	2.03%	6.45%	1.99%
	1000	4.62%	1.52%	4.42%	1.45%	4.22%	1.72%	3.17%	1.12%	3.10%	1.19%	6.40%	2.51%
t	50	6.86%	2.03%	7.26%	2.56%	7.59%	2.23%	6.17%	2.23%	7.30%	2.23%	8.72%	3.20%
	100	6.50%	1.74%	6.25%	1.61%	6.34%	1.66%	6.00%	1.41%	5.63%	1.37%	7.20%	2.44%
	250	6.31%	1.41%	5.65%	1.50%	6.05%	1.32%	5.39%	1.28%	4.90%	1.41%	6.62%	1.77%
	500	6.30%	1.44%	5.61%	1.34%	5.26%	1.19%	4.71%	0.99%	5.26%	1.44%	6.05%	1.14%
	1000	5.68%	1.12%	5.08%	0.99%	5.02%	1.06%	4.16%	0.59%	4.75%	0.99%	6.93%	1.52%
	rolling=250	AAPL		AMZN		MSFT		SNE		NFLX		10yTsy	
	Forecasting size	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
Normal	5	5.83%	2.03%	4.68%	1.99%	5.03%	1.94%	4.94%	1.99%	4.33%	2.03%	6.62%	2.12%
	10	5.74%	2.12%	4.59%	1.90%	5.30%	1.94%	4.94%	2.08%	4.42%	2.21%	6.67%	2.21%
	20	5.74%	2.03%	5.03%	1.99%	5.43%	1.90%	5.08%	1.99%	4.46%	2.25%	6.45%	2.25%
	60	5.70%	2.25%	5.25%	1.94%	5.47%	2.12%	5.34%	1.85%	4.55%	2.16%	6.49%	2.43%
	120	6.18%	2.43%	4.99%	2.12%	6.05%	2.43%	4.86%	1.81%	6.62%	3.97%	7.20%	3.22%
t	5	6.18%	1.46%	5.56%	1.46%	5.92%	1.28%	5.39%	1.24%	4.86%	1.37%	6.58%	1.72%
	10	6.31%	1.41%	5.65%	1.50%	6.05%	1.32%	5.39%	1.28%	4.90%	1.41%	6.62%	1.77%
	20	6.40%	1.46%	5.39%	1.37%	6.09%	1.28%	5.39%	1.28%	5.03%	1.55%	6.62%	1.77%
	60	6.36%	1.50%	5.87%	1.59%	6.62%	1.37%	5.30%	1.28%	5.52%	1.63%	7.02%	1.99%
	120	6.40%	1.41%	5.56%	1.59%	6.58%	1.50%	4.81%	1.10%	5.12%	1.28%	7.64%	2.78%

10 years, rolling.size=250, forecast.size=10

		AAPL		AMZN		MSFT		SNE		NFLX		10yTsy	
	Model	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
	pure normal	4.93%	1.86%	3.85%	1.58%	4.30%	1.76%	4.16%	1.90%	3.76%	1.76%	5.29%	2.90%
	pure t	2.61%	0.49%	2.43%	0.44%	1.81%	0.62%	2.21%	0.58%	1.63%	0.35%	3.01%	0.84%
	arma(1,1)+garch(1,1), normal error	5.74%	2.12%	4.59%	1.90%	5.30%	1.94%	4.94%	2.08%	4.42%	2.21%	6.67%	2.21%
	arma(1,1)+garch(1,1), t error	6.31%	1.41%	5.65%	1.50%	6.05%	1.32%	5.39%	1.28%	4.90%	1.41%	6.62%	1.77%
	arma(1,1)+garch(1,0)	6.14%	1.55%	5.52%	1.19%	6.00%	1.37%	4.94%	1.37%	5.25%	1.15%	5.17%	1.33%
	arma(1,1)+garch(2,1)	6.31%	1.46%	5.61%	1.55%	6.23%	1.46%	5.34%	1.28%	5.03%	1.50%	5.53%	1.64%
	arma(1,1)+garch(1,2)	6.36%	1.32%	5.65%	1.32%	6.09%	1.37%	5.43%	1.37%	5.08%	1.28%	5.35%	1.72%
	arma(1,1)+garch(2,2)	6.23%	1.37%	5.47%	1.41%	6.09%	1.46%	5.30%	1.37%	5.08%	1.28%	5.48%	1.68%
	arma(1,1)+igarch(1,1)	5.78%	1.24%	5.17%	1.19%	5.56%	1.10%	4.72%	1.15%	4.64%	1.10%	6.49%	1.81%
	arma(1,1)+egarch(1,1)	6.31%	2.12%	N/A	N/A	6.53%	2.08%	5.96%	1.81%	5.87%	2.08%	N/A	N/A
use t for all following	arma(0,0)+garch(1,1)	6.45%	1.46%	5.34%	1.32%	5.56%	1.28%	5.08%	1.37%	4.94%	1.50%	6.27%	1.72%
	arma(0,1)+garch(1,1)	6.31%	1.41%	5.43%	1.37%	5.74%	1.37%	5.43%	1.28%	4.86%	1.50%	6.53%	1.72%
	arma(1,0)+garch(1,1)	6.31%	1.37%	5.47%	1.32%	5.70%	1.37%	5.47%	1.28%	4.86%	1.46%	6.53%	1.72%
	arma(1,1)+garch(1,1)	6.31%	1.41%	5.65%	1.50%	6.05%	1.32%	5.39%	1.28%	4.90%	1.41%	6.62%	1.77%
	arma(2,0)+garch(1,1)	6.09%	1.32%	5.56%	1.37%	5.78%	1.37%	5.34%	1.28%	4.86%	1.50%	6.58%	1.68%
test ar1	arma(1,0)+garch(1,0) (ar1+arch)	6.05%	1.37%	5.39%	1.19%	5.74%	1.28%	5.25%	1.32%	5.03%	1.19%	5.13%	1.37%
	arma(1,0)+garch(1,1)	6.31%	1.37%	5.47%	1.32%	5.70%	1.37%	5.47%	1.28%	4.86%	1.46%	5.40%	1.64%
	arma(1,0)+garch(2,1)	6.31%	1.46%	5.34%	1.50%	5.87%	1.37%	5.43%	1.32%	5.03%	1.50%	5.40%	1.59%
	arma(1,0)+garch(1,2)	6.40%	1.32%	5.39%	1.32%	5.74%	1.37%	5.39%	1.32%	5.08%	1.28%	5.40%	1.64%
	arma(1,0)+garch(2,2)	6.31%	1.41%	5.43%	1.50%	6%	1.37%	5.30%	1.37%	5.12%	1.24%	5.62%	1.55%
	average	6.28%	1.39%	5.40%	1.37%	5.81%	1.35%	5.37%	1.32%	5.02%	1.33%	5.39%	1.56%
test ar2	arma(2,0)+garch(1,0) (ar1+arch)	5.87%	1.41%	5.56%	1.24%	5.83%	1.37%	5.08%	1.28%	4.99%	1.24%	5.09%	1.37%
	arma(2,0)+garch(1,1)	6.09%	1.32%	5.56%	1.37%	5.78%	1.37%	5.34%	1.28%	4.86%	1.50%	5.44%	1.68%
	arma(2,0)+garch(2,1)	6.23%	1.46%	5.39%	1.46%	5.92%	1.37%	5.30%	1.28%	5.08%	1.46%	5.40%	1.59%
	arma(2,0)+garch(1,2)	6.23%	1.28%	5.56%	1.32%	5.83%	1.32%	5.43%	1.37%	5.08%	1.32%	5.40%	1.64%
	arma(2,0)+garch(2,2)	6.18%	1.28%	5.34%	1.46%	6.00%	1.41%	5.34%	1.37%	5.21%	1.24%	5.44%	1.55%
	average	6.12%	1.35%	5.48%	1.37%	5.87%	1.37%	5.30%	1.32%	5.04%	1.35%	5.35%	1.57%
test arma(1,1)	arma(1,1)+garch(1,0) (ar1+arch)	6.14%	1.55%	5.52%	1.19%	6.00%	1.37%	4.94%	1.37%	5.25%	1.15%	5.17%	1.33%
also given arma(1,1), test different garch	arma(1,1)+garch(1,1)	6.31%	1.41%	5.65%	1.50%	6.05%	1.32%	5.39%	1.28%	4.90%	1.41%	6.62%	1.77%
	arma(1,1)+garch(2,1)	6.31%	1.46%	5.61%	1.55%	6.23%	1.46%	5.34%	1.28%	5.03%	1.50%	5.53%	1.64%
	arma(1,1)+garch(1,2)	6.36%	1.32%	5.65%	1.32%	6.09%	1.37%	5.43%	1.37%	5.08%	1.28%	5.35%	1.72%
	arma(1,1)+garch(2,2)	6.23%	1.37%	5.47%	1.41%	6.09%	1.46%	5.30%	1.37%	5.08%	1.28%	5.48%	1.68%
	average	6.27%	1.42%	5.58%	1.39%	6.09%	1.40%	5.28%	1.33%	5.07%	1.32%	5.63%	1.63%

Indices

Summary Statistic

Summary Statistics	CAC40	DAX30	HSI	N225	SPY
Mean	0.06524	0.1756	0.1373	-0.0177	0.03935
Median	0.12338	-0.08	0.1365	0.1681	0.05819
Maximum	179.17595	783.3996	273.9419	778.4473	6.83664
Minimum	-98.22487	-782.0038	-269.6335	-778.4057	-6.89584
SD	13.97089	62.69725	16.36616	60.97005	1.047811
Skewness	0.8683233	-0.5502493	0.2191497	-0.1323181	-0.328313
Kurtosis	26.96359	75.72581	114.0948	69.92024	5.003327

ARMA(1,1)+GARCH(1,1)

	forecast=10	CAC40		DAX30		HSI		N225		S&P500	
	Rolling size	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
Normal	50	9.70%	4.83%	7.76%	4.38%	8.97%	4.75%	6.90%	3.84%	9.70%	4.50%
	100	7.17%	2.85%	7.22%	3.49%	7.43%	3.21%	5.95%	3.25%	7.41%	3.52%
	250	6.51%	2.65%	5.86%	2.80%	5.54%	2.39%	5.05%	2.79%	6.87%	2.91%
	500	5.74%	2.00%	4.91%	2.55%	5.13%	2.34%	4.93%	2.74%	6.45%	2.88%
t	50	10.02%	3.63%	10.33%	4.10%	9.79%	3.76%	10.67%	4.75%	9.98%	4.02%
	100	8.19%	2.00%	8.70%	2.83%	8.40%	2.62%	9.33%	3.33%	8.03%	2.82%
	250	7.38%	2.08%	9.13%	2.75%	6.94%	1.94%	6.94%	2.52%	7.59%	2.30%
	500	6.72%	1.41%	9.67%	3.29%	7.36%	1.83%	7.06%	2.39%	7.20%	1.99%
	rolling=250	CAC40		DAX30		HSI		N225		S&P500	
	forecast size	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
Normal	5	6.46%	2.73%	5.73%	2.84%	5.90%	2.57%	5.05%	2.84%	6.75%	2.78%
	10	6.51%	2.65%	5.86%	2.80%	5.54%	2.39%	5.05%	2.79%	6.87%	2.91%
	20	6.46%	2.65%	5.94%	2.84%	5.36%	2.52%	4.87%	2.66%	6.80%	3.00%
	60	6.38%	2.52%	5.51%	2.75%	5.05%	2.21%	4.91%	2.48%	6.98%	3.00%
t	5	7.55%	2.13%	8.78%	2.88%	7.07%	1.85%	6.76%	2.57%	7.55%	2.25%
	10	7.38%	2.08%	9.13%	2.75%	6.94%	1.94%	6.94%	2.52%	7.59%	2.30%
	20	7.25%	1.78%	9.05%	2.62%	6.89%	1.76%	7.21%	2.48%	7.64%	2.25%
	60	7.20%	1.61%	10.14%	2.97%	6.67%	1.85%	7.08%	2.84%	8.08%	2.38%

10 years, rolling.size=250, forecast.size=10

		CAC40		DAX30		HSI		N225		S&P500	
	Model	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
	pure normal	3.85%	1.90%	4.07%	2.31%	4.07%	2.22%	2.99%	1.86%	5.29%	2.49%
	pure t	2.26%	0.59%	2.67%	1.04%	2.67%	0.95%	2.08%	0.81%	0.31%	0.50%
	arma(1,1)+garch(1,1), normal error	6.51%	2.65%	5.86%	2.80%	5.54%	2.39%	5.05%	2.79%	6.87%	2.91%
	arma(1,1)+garch(1,1), t error	7.38%	2.08%	9.13%	2.75%	6.94%	1.94%	6.94%	2.52%	7.59%	2.30%
test arma(1,1)	arma(1,1)+garch(1,0) (ar1+arch)	6.07%	1.69%	9.18%	2.93%	6.31%	1.58%	7.30%	2.16%	7.42%	2.08%
	arma(1,1)+igarch(1,1)	7.25%	1.87%	8.83%	2.53%	6.94%	1.89%	6.89%	2.48%	7.28%	1.90%
	arma(1,1)+egarch(1,1)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	7.06%	2.38%
use t for all following	arma(0,0)+garch(1,1)	7.11%	1.74%	8.61%	2.67%	6.76%	1.89%	6.80%	2.52%	6.84%	2.16%
	arma(0,1)+garch(1,1)	7.29%	1.82%	8.65%	2.58%	6.76%	1.94%	6.89%	2.39%	7.28%	2.16%
	arma(1,0)+garch(1,1)	7.25%	1.87%	8.65%	2.53%	6.76%	1.94%	6.94%	2.43%	7.28%	2.16%
	arma(1,1)+garch(1,1)	7.38%	2.08%	9.13%	2.75%	6.94%	1.94%	6.94%	2.52%	7.59%	2.30%
	arma(2,0)+garch(1,1)	7.25%	1.95%	8.52%	2.67%	7.16%	1.98%	6.99%	2.52%	7.24%	2.03%
test ar1	arma(1,0)+garch(1,0) (ar1+arch)	5.81%	1.48%	8.26%	2.40%	5.95%	1.58%	6.49%	2.07%	6.45%	1.81%
	arma(1,0)+garch(1,1)	7.25%	1.87%	8.65%	2.53%	6.76%	1.94%	6.94%	2.43%	7.28%	2.16%
	arma(1,0)+garch(2,1)	7.07%	2.00%	9.00%	2.62%	6.80%	2.16%	6.76%	2.48%	7.28%	2.21%
	arma(1,0)+garch(1,2)	7.11%	1.95%	8.61%	2.58%	6.85%	1.80%	6.94%	2.52%	7.28%	2.12%
	arma(1,0)+garch(2,2)	6.98%	1.91%	8.87%	2.62%	6.71%	1.98%	6.80%	2.34%	7.42%	2.21%
	average	7.05%	1.93%	8.74%	2.60%	6.61%	1.89%	6.79%	2.37%	7.14%	2.10%
test ar2	arma(2,0)+garch(1,0) (ar1+arch)	5.68%	1.65%	8.26%	2.53%	6.04%	1.62%	6.49%	1.89%	6.62%	1.90%
	arma(2,0)+garch(1,1)	7.25%	1.95%	8.52%	2.67%	7.16%	1.98%	6.99%	2.52%	7.24%	2.03%
	arma(2,0)+garch(2,1)	7.07%	1.91%	8.74%	2.67%	7.16%	2.21%	6.80%	2.43%	7.37%	2.12%
	arma(2,0)+garch(1,2)	7.07%	2.00%	8.65%	2.75%	7.07%	1.85%	6.99%	2.48%	7.37%	2.08%
	arma(2,0)+garch(2,2)	7.11%	1.95%	8.78%	2.71%	7.21%	2.07%	6.89%	2.43%	7.55%	2.08%
	average	6.84%	1.89%	8.59%	2.67%	6.93%	1.95%	6.83%	2.35%	7.23%	2.04%
test arma(1,1)	arma(1,1)+garch(1,0) (ar1+arch)	6.07%	1.69%	9.18%	2.93%	6.31%	1.58%	7.30%	2.16%	7.42%	2.08%
also given arma(1,1), test different garch	arma(1,1)+garch(1,1)	7.38%	2.08%	9.13%	2.75%	6.94%	1.94%	6.94%	2.52%	7.59%	2.30%
	arma(1,1)+garch(2,1)	7.46%	2.17%	9.53%	2.84%	7.03%	2.12%	6.62%	2.39%	7.73%	2.34%
	arma(1,1)+garch(1,2)	7.33%	2.17%	9.18%	2.93%	7.07%	1.80%	7.17%	2.57%	7.68%	2.43%
	arma(1,1)+garch(2,2)	7.46%	2.21%	9.40%	2.84%	7.07%	2.03%	6.94%	2.52%	7.86%	2.34%
	average	7.14%	2.06%	9.28%	2.86%	6.88%	1.89%	6.99%	2.43%	7.66%	2.30%

Foreign Exchanges

Summary Statistic

Summary Statistics	USD_CAD	USD_CHF	USD_JPY	EUR_USD
Mean	0.004202	-0.003095	0.007156	-0.007618
Median	0.00783	0	0	0
Maximum	2.85252	9.23883	3.463915	3.733324
Minimum	-2.188198	-17.144541	-2.77221	-2.652879
SD	0.5597232	0.7249343	0.6195027	0.598097
Skewness	0.161073	-4.229645	-0.05465961	0.02103314
Kurtosis	1.679328	129.2069	3.610319	1.833073

ARMA(1,1)+GARCH(1,1)

	forecast=10	USD_CAD		USD_CHF		USD_JPY		EUR_USD	
	Rolling size	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
Normal	50	6.80%	2.32%	7.15%	2.55%	7.15%	3.17%	6.73%	2.07%
	100	5.67%	1.30%	5.63%	1.77%	5.67%	2.09%	6.30%	1.95%
	250	4.48%	1.30%	5.19%	1.93%	4.65%	1.68%	5.30%	1.70%
	500	4.07%	0.98%	5.19%	1.82%	5.10%	1.64%	4.70%	1.33%
t	50	7.11%	2.05%	7.19%	2.24%	7.23%	2.05%	7.12%	1.60%
	100	5.56%	1.02%	5.79%	1.54%	6.03%	1.54%	6.58%	1.48%
	250	4.36%	0.96%	5.32%	1.42%	5.23%	1.09%	5.64%	1.15%
	500	4.16%	0.84%	4.82%	1.08%	5.47%	1.03%	4.98%	0.95%
	rolling=250	USD_CAD		USD_CHF		USD_JPY		EUR_USD	
	forecast size	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
Normal	5	4.44%	1.26%	5.19%	1.84%	4.77%	1.68%	5.43%	1.78%
	10	4.48%	1.30%	5.19%	1.93%	4.65%	1.68%	5.30%	1.70%
	20	4.27%	1.26%	5.44%	1.97%	4.56%	1.68%	5.13%	1.78%
	60	4.10%	1.30%	5.57%	2.01%	4.27%	1.59%	5.01%	1.65%
t	5	4.44%	0.92%	5.40%	1.38%	5.23%	1.05%	5.73%	1.10%
	10	4.36%	0.96%	5.32%	1.42%	5.23%	1.09%	5.64%	1.15%
	20	4.27%	1.01%	5.32%	1.38%	5.32%	1.13%	5.60%	1.27%
	60	4.10%	1.05%	5.53%	1.38%	5.36%	1.17%	5.43%	1.27%

10 years, rolling.size=250, forecast.size=10

		USD_CAD		USD_CHF		USD_JPY		EUR_USD	
	Model	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR	95%VaR	99%VaR
	pure normal	4.16%	1.54%	3.71%	1.36%	4.62%	1.27%	4.89%	1.54%
	pure t	1.90%	0.05%	1.86%	0.18%	2.04%	0.27%	2.22%	0.23%
	arma(1,1)+garch(1,1), normal error	4.48%	1.30%	5.19%	1.93%	4.65%	1.68%	5.30%	1.70%
	arma(1,1)+garch(1,1), t error	4.36%	0.96%	5.32%	1.42%	5.23%	1.09%	5.64%	1.15%
	arma(1,1)+garch(1,0) (ar1+arch)	4.52%	0.88%	4.73%	1.17%	5.49%	0.88%	5.01%	1.23%
	arma(1,1)+igarch(1,1)	4.36%	0.71%	5.44%	1.13%	5.15%	0.88%	5.26%	9.80%
	arma(1,1)+egarch(1,1)	5.44%	1.68%	6.74%	2.09%	5.78%	1.59%	5.73%	1.61%
use t for all following	arma(0,0)+garch(1,1)	4.48%	1.09%	5.53%	1.30%	5.03%	1.17%	5.39%	1.15%
	arma(0,1)+garch(1,1)	4.15%	1.05%	5.40%	1.30%	5.28%	1.09%	5.52%	1.02%
	arma(1,0)+garch(1,1)	4.15%	1.05%	5.44%	1.30%	5.19%	1.09%	5.43%	1.02%
	arma(1,1)+garch(1,1)	4.36%	0.96%	5.32%	1.42%	5.23%	1.09%	5.64%	1.15%
	arma(2,0)+garch(1,1)	4.40%	1.05%	5.57%	1.21%	5.15%	1.26%	5.43%	1.15%
test ar1	arma(1,0)+garch(1,0) (ar1+arch)	4.44%	0.75%	4.52%	1.30%	5.19%	0.84%	5.05%	1.27%
	arma(1,0)+garch(1,1)	4.15%	1.05%	5.44%	1.30%	5.19%	1.09%	5.43%	1.02%
	arma(1,0)+garch(2,1)	4.27%	1.13%	5.53%	1.38%	5.11%	1.13%	5.47%	1.10%
	arma(1,0)+garch(1,2)	4.31%	1.05%	5.44%	1.26%	5.40%	1.17%	5.64%	1.02%
	arma(1,0)+garch(2,2)	4.36%	1.13%	5.40%	1.30%	5.28%	1.09%	5.64%	1.10%
	average	4.31%	1.02%	5.27%	1.31%	5.23%	1.06%	5.45%	1.10%
test ar2	arma(2,0)+garch(1,0) (ar1+arch)	4.44%	0.80%	4.52%	1.17%	5.32%	0.92%	4.96%	1.23%
	arma(2,0)+garch(1,1)	4.40%	1.05%	5.57%	1.21%	5.15%	1.26%	5.43%	1.15%
	arma(2,0)+garch(2,1)	4.40%	1.13%	5.53%	1.17%	5.15%	1.21%	5.43%	1.23%
	arma(2,0)+garch(1,2)	4.48%	1.05%	5.57%	1.17%	5.36%	1.17%	5.60%	1.15%
	arma(2,0)+garch(2,2)	4.61%	1.13%	5.53%	1.13%	5.49%	1.21%	5.60%	1.19%
	average	4.47%	1.03%	5.34%	1.17%	5.29%	1.15%	5.40%	1.19%
test arma(1,1)	arma(1,1)+garch(1,0) (ar1+arch)	4.52%	0.88%	4.73%	1.17%	5.49%	0.88%	5.01%	1.23%
also given arma(1,1), test different garch	arma(1,1)+garch(1,1)	4.36%	0.96%	5.32%	1.42%	5.23%	1.09%	5.64%	1.15%
	arma(1,1)+garch(2,1)	4.44%	1.01%	5.32%	1.34%	5.19%	1.13%	5.64%	1.10%
	arma(1,1)+garch(1,2)	4.48%	1.01%	5.40%	1.34%	5.53%	1.05%	5.81%	1.19%
	arma(1,1)+garch(2,2)	4.36%	1.09%	5.32%	1.30%	5.44%	0.96%	5.77%	1.06%
	average	4.43%	0.99%	5.22%	1.31%	5.38%	1.02%	5.57%	1.15%

APPENDIX B
CODES

```
# packages
library(MASS)
library(rugarch)
library(fGarch)
require(xts)
require(scales)

# intermediate functions
# rolling forecast using simple normal
VaR_normal <- function(dat,train,test,alpha){
  m <- mean(dat[1:train,1])
  v <- var(dat[1:train,1])
  VaR <- v * qnorm(alpha) - m
  Violation <- mean(dat[(train+1):(train+test),1] < -VaR)
  return(c(VaR,Violation))
}

# rolling forecast using simple t
VaR_t <- function(dat,train,test,alpha){
  k <- floor((length(dat)-train)/test)
  VaRs <- c()
  Violations <- c()
  for (i in 1:k){
    dat_sub<-dat[(1+test*(i-1)):(train+test*(i-1))]
    fit<-fitdist(distribution = 'std',x=dat_sub) # estimate t parameters
    m <- fit$pars[1]
    v <- fit$pars[2]
    df <- fit$pars[3]
    VaRs[i] <- v*qt(alpha,df=df) - m
    Violations[i] <- mean(dat[(train+1+test*(i-1)):(train+test*i)] < -VaRs[i])
  }
  VaR <- mean(VaRs)
  Violation <- mean(Violations)

  return(c(VaR,Violation))
}

# Rolling VaR estimation function(ARMA+GARCH)
Rolling_VaR <- function(data, ar=1, ma=1, alpha=1, beta=1, dist='std',
                        model='sGARCH', rolling, refit){
  #n <- length(data)
  model <- ugarchspec(variance.model=list(model=model, garchOrder=c(alpha, beta)),
                      mean.model=list(armaOrder=c(ar, ma)), distribution.model=dist)
  modelroll <- ugarchroll(model, data=data, n.ahead=1, n.start=rolling, refit.every=refit,
                          refit.window=c("moving"), calculate.VaR=TRUE,
                          VaR.alpha=c(0.01, 0.05), keep.coef=TRUE)
                          # rolling forecast, use 'rolling' days forecaste 'refit' days VaR
  VaR95 <- modelroll@forecast$VaR[,2]
  VaR99 <- modelroll@forecast$VaR[,1]
  real <- modelroll@forecast$VaR[,3]
  rate95 <- mean(real<VaR95)
  rate99 <- mean(real<VaR99)
  shortfall95 <- mean(real[real<VaR95])
  shortfall99 <- mean(real[real<VaR99])
```

```
  result <-list(VaR95=VaR95, VaR99=VaR99, rate95=rate95, rate99=rate99,
                shortfall95=shortfall95, shortfall99=shortfall99)
  return(result)
}

# final function
Rolling <- function(datalist, ar=1, ma=1, alpha=1, beta=1, rolling, refit){
  n <- length(datalist)
  RollVaRn <- matrix(0, nrow=n, ncol=2)
  stocknames <- names(datalist)
  rownames(RollVaRn) <- stocknames
  colnames(RollVaRn) <- c("95VaR␣Violation%", "99VaR␣Violation%")
  RollVaRt <- matrix(0, nrow=n, ncol=2)
  rownames(RollVaRt) <- stocknames
  colnames(RollVaRt) <- c("95VaR␣Violation%", "99VaR␣Violation%")

  # normal assumption
  for (i in 1:n){
    result <- Rolling_VaR(datalist[[i]], ar=ar, ma=ma, alpha=alpha, beta=beta,
                          dist='norm', rolling=rolling, refit=refit)
    RollVaRn[i,1] <- result$rate95
    RollVaRn[i,2] <- result$rate99
  }
  # t assumption
  for (i in 1:n){
    result <- Rolling_VaR(datalist[[i]], ar=ar, ma=ma, alpha=alpha, beta=beta,
                          rolling=rolling, refit=refit)
    RollVaRt[i,1] <- result$rate95
    RollVaRt[i,2] <- result$rate99
  }

  RollVaRn[] <- percent(RollVaRn)
  RollVaRt[] <- percent(RollVaRt)
  return(list(RollVaRn=RollVaRn, RollVaRt=RollVaRt))
} # Rolling estimate VaR

ie_Rolling <- function(datalist, ar=1, ma=1, alpha=1, beta=1, rolling, refit){
  n <- length(datalist)
  RollVaRi <- matrix(0, nrow=n, ncol=2)
  stocknames <- names(datalist)
  rownames(RollVaRi) <- stocknames
  colnames(RollVaRi) <- c("95VaR␣Violation%", "99VaR␣Violation%")
  RollVaRe <- matrix(0, nrow=n, ncol=2)
  rownames(RollVaRe) <- stocknames
  colnames(RollVaRe) <- c("95VaR␣Violation%", "99VaR␣Violation%")

  # IGARCH assumption
  for (i in 1:n){
    result <- Rolling_VaR(datalist[[i]], ar=ar, ma=ma, alpha=alpha, beta=beta,
                          model='iGARCH', rolling=rolling, refit=refit)
    RollVaRi[i,1] <- result$rate95
    RollVaRi[i,2] <- result$rate99
  }
  # EGARCH assumption
  for (i in 1:n){
    result <- Rolling_VaR(datalist[[i]], ar=ar, ma=ma, alpha=alpha, beta=beta,
                          model='eGARCH', rolling=rolling, refit=refit)
```

```
    RollVaRe[i,1] <- result$rate95
    RollVaRe[i,2] <- result$rate99
  }

  RollVaRi[] <- percent(RollVaRi)
  RollVaRe[] <- percent(RollVaRe)
  return(list(RollVaRi=RollVaRi, RollVaRe=RollVaRe))
} # Rolling estimate VaR using EGARCH and IGARCH


# 100*log return
lg_return_100<-function(data){
  data<-as.vector(as.numeric(data))
  n<-length(data)
  logreturn<-100*log(data[-1]/data[-n])
  return(logreturn)
}

# data preparation
price <- read.csv("stocks.csv")
bond<-read.csv("TNX_bond.csv")
Date <- as.Date(price$Date[-1], "%m/%d/%Y") # save the date for tseries
bDate <-as.Date(bond$Date[-1], "%m/%d/%Y")
n<-nrow(price)

# transform price data into 100*log return Time series
return <- price[-1,]/price[-n,]
return <- 100*log(return[,-1])
breturn<-lg_return_100(as.numeric(bond$price))

AAPL.xts <- na.omit(xts(x = return$AAPL.Close, order.by = Date))
AMZN.xts <- na.omit(xts(x = return$AMZN.Close, order.by = Date))
NFLX.xts <- na.omit(xts(x = return$NFLX.Close, order.by = Date))
MSFT.xts <- na.omit(xts(x = return$MSFT.Close, order.by = Date))
SNE.xts <- na.omit(xts(x = return$SNE.Close, order.by = Date))
bond.xts<-na.omit(xts(x=breturn,order.by=bDate))

# input data formula as a list of tseries
Stocks <- list('AAPL'=AAPL.xts, 'AMZN'=AMZN.xts, 'MSFT'=MSFT.xts, 'SNE'=SNE.xts,
               'NFLX'=NFLX.xts, 'Bond'=bond.xts)

# sample experiments
# fixed refit test rolling
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=1000, refit=10)
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=500, refit=10)
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=250, refit=10)
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=100, refit=10)
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=50, refit=10)

# fixed rolling test refit
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=250, refit=5)
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=250, refit=10)
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=250, refit=20)
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=250, refit=60)
Rolling(datalist=Stocks, ar=1, ma=1, alpha=1, beta=1, rolling=250, refit=120)
```