Application of Volatility Models on Stock Return Prediction

Applied Project

Quantitative Analysis

By

Tuo Xiong

CID: 02169869

2021-2022 Academic Year

Introduction and Client Specification

Our client is a global asset management organization. Nowadays most of asset managers tend to focus on investigating specific firms and build a large portfolio to diversify the firm-specific volatility effectively. The non-diversified risk, or market risk is less frequently considered in the process of investment. One main target of this project is linking the market volatility and pricing models together, in order to better capture the features of moving market price. Volatility is a significant term which we can find in nearly any financial models. Popular models such as the Vasicek and CIR models in fixed-income area, the Black Scholes model in option pricing area, the log-normal stock price model and the Fama-French model are all based on a volatility term which may consist of a specific value (σ) and Brownian Motion process. In the basic form of these models, the volatility value is always assumed as a constant, but more and more evidence from practice shows that such assumption may be not sufficient and far from reality. Hence in this report, we are interested in how non-constant volatility works in these models, how non-constant volatility is correlated to itself with different time intervals, or whether forecasting asset volatility will improve the performance of these models. We are going to exhibit a process of estimating the daily volatility of a specific underlying asset by several commonly used volatility models, and then applying such estimated volatility on the log-normal stock price distribution model. We expect to see that the log stock price or return movements are supposed to be explained by the volatility in a certain degree.

CONTENTS

1. Data Selection

In this applied project, we select the daily close price of the S&P 500 index as the main asset utilized for quantitative analysis, and the data period includes 252 working days in the one year before July 2022. For simplicity and easiness for reading, we are going to use number 1 to 252 to represent time in the following plots. One of the great advantages of using daily S&P 500 index as data is its stability. Commonly, S&P 500 index is considered as an essential benchmark of overall U.S. market, and some investors even consider it as the market portfolio which has a nearly perfect correlation with the market movement. The companies in the list of S&P 500 are generally large corporations, and stable market sectors are weighted more heavily (Plaehn, 2022). The stability adds less randomness of specific companies in our close price data and more consistent with our log-normal price model without the assumption of jump process. Under this context, in a way we can regard the volatility of S&P 500 Index daily price as the market volatility.

2. Data Preprocessing

2.1 Overall Log Close Price

First, we can have an overall look at the whole log price plot during the chosen one-year period, which is shown in Figure 1 below.

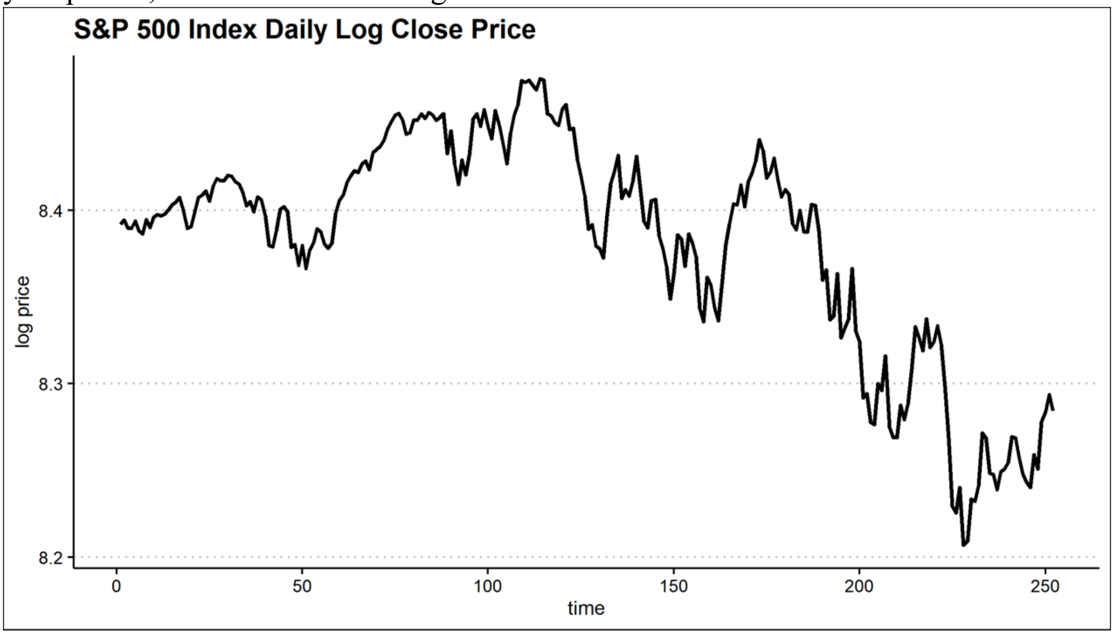


Figure 1: Moving Log Close Price of S&P 500 Index

The mean log close price in this year is roughly 8.38 and the standard deviation is about

0.064. We have almost 94% of samples being apart from the mean within 2 standard deviations, and no extreme price jump was found in this group of data.

2.2 Overall Return

Similarly, from the return plot in Figure 2, we can see the absolute values of all daily returns are within 5%, which is much lower than its historical average return and may be considered as stable. The daily return in Figure 2 is obtained straightly from the ratio of two successive close prices. The other method is to obtain return from the log ratio, which is exactly the continuously compounded daily return, but both methods get nearly the same outcomes due to the small absolute value of return.

$$\textit{Proportional Return:}\ \ R_t = \frac{S_t}{S_{t-1}} - 1$$

$$\textit{Log Return:}\ \ R_t = log\left(\frac{S_t}{S_{t-1}}\right)$$

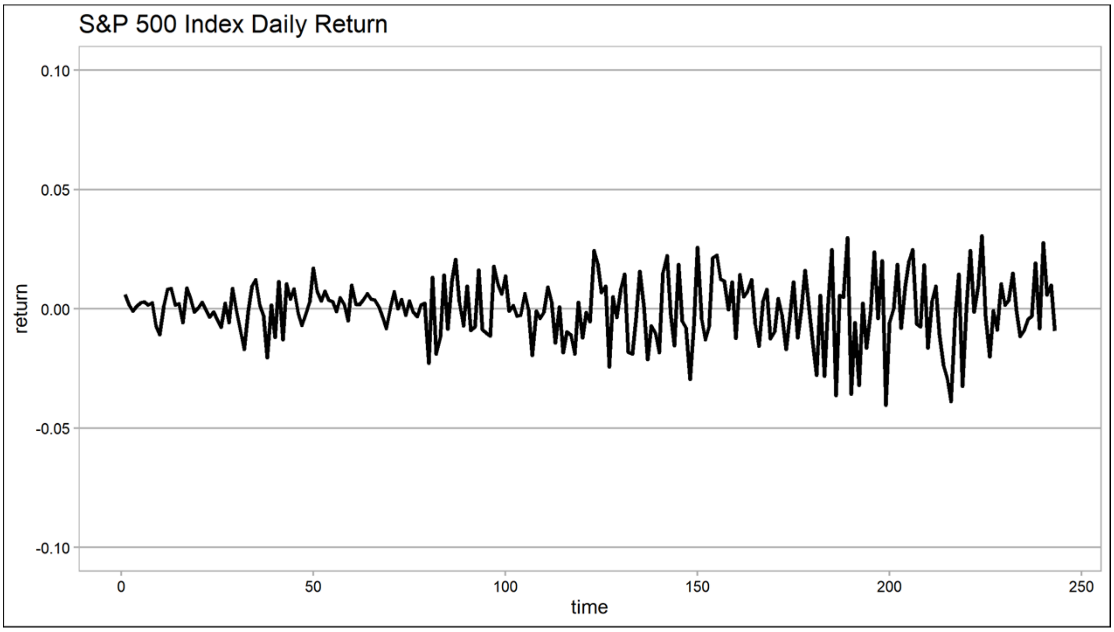


Figure 2: Daily Return of S&P 500 Index

2.3 Auto Correlation Analysis

Some investors may think the historical return of an asset may help forecast its return in near future. This does work especially when good news is published successively, and strategies such as looking at moving average and momentum are based on historical return data. While abnormal return happens less or neutral news are majority, this may not be that effective. At least in practice, the auto correlation of daily return seems to be weak and stochastic. In the following Figure 3, we have a plot of S&P 500 index daily return auto correlation with lag orders from 1 to 30. The result has no surprise that the signs of correlation across x-axis are dramatically floated, and the absolute values are all very small. This means that we can obtain little information on future return from the return today, at least for S&P 500 index. By contrast, when we square the return, the result reverses totally. As exhibited by Figure 4, now the auto correlation values of squared return are consistently positive across x-axis and focus on 0.2 to 0.3 roughly. In addition, we observe that the auto correlation values with small lag orders tend to be higher and denser than those with large lag orders. Such observations suggest that there exists a strong clustering of volatility, which means large price or return changes are likely to be subsequently followed by substantial changes, and vice versa.

The following one is the plot of volatility auto correlation in Figure 5. Here we need to set a time window to determine the daily volatility of S&P 500 index, and make sure each date is associated with a unique volatility value. We preset the default length of time window as 10 days. It means that we use the sample standard deviation of the 10-day-before close price data, as we are not able to realize in advance the information in the future, to obtain the volatility of a specific date. Then Figure 5 has comparable results to that of squared return volatility, but compared to Figure 4, volatility auto correlation has stronger and higher value of positive correlation. Furthermore, we observe that the correlation drops down gradually to 0.4 approximately as the lag order grows. We expect that as lag order increases above 30, the correlation may still remain at a level of 0.3 to 0.4.

The characteristics of volatility clustering shown above actually suggest that the auto-regressive conditional heteroskedasticity model (ARCH) is supposed to be an excellent tool to complete the estimation of return volatility (Kurniawan, 2015). Specifically, we choose the most popular GARCH(1,1) model in this project, which we will soon explain and explore more in the next pages. A certain degree of predictability of the return volatility rather than the asset return itself is the major reason that we are interested in applying the estimated volatility on the stock price model to indirectly forecast the return.

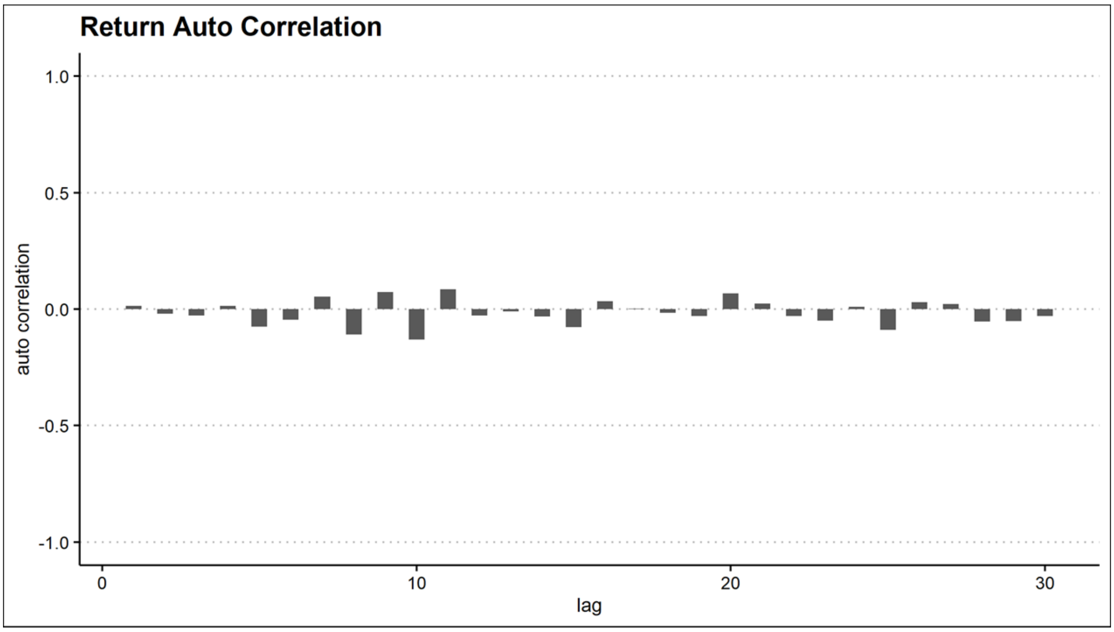


Figure 3: Daily Return Auto Correlation

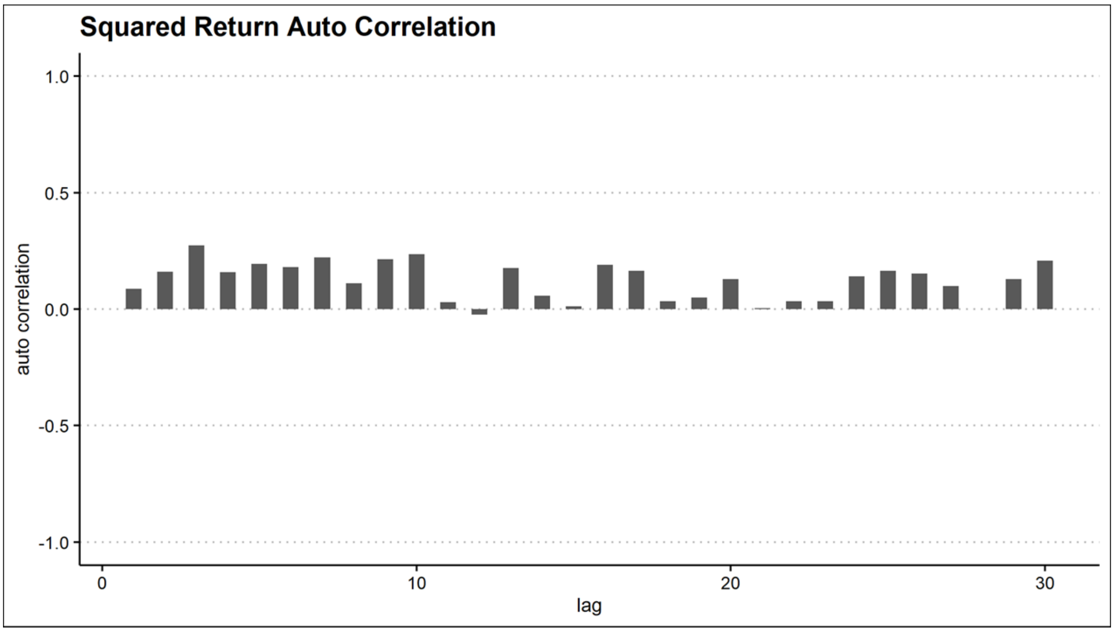


Figure 4: Squared Daily Return Auto Correlation

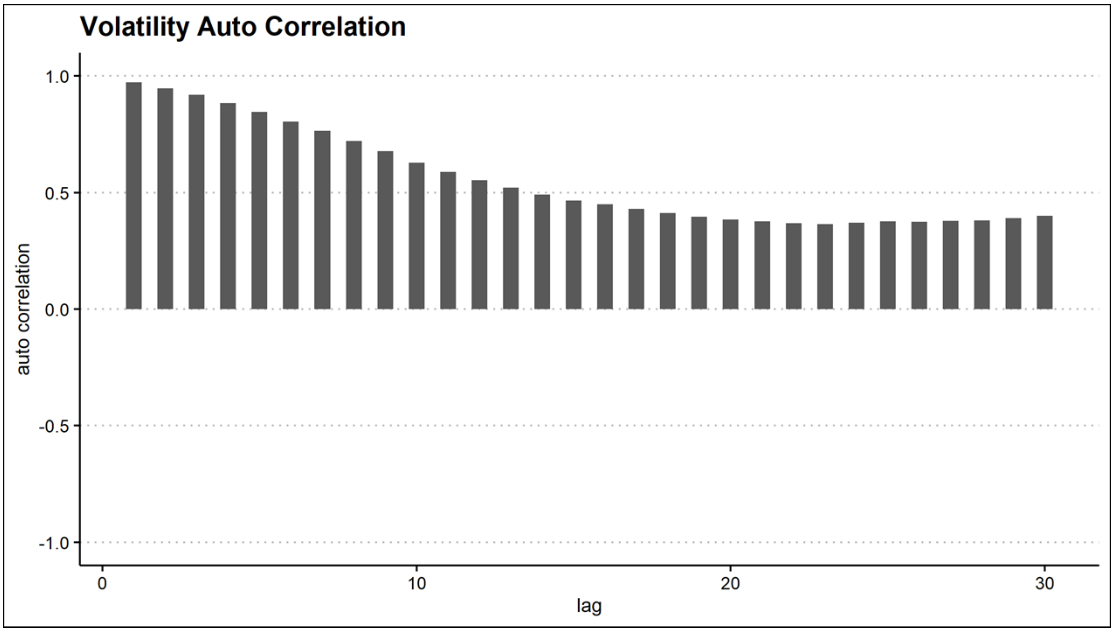


Figure 5: Historical Volatility Auto Correlation

2.4 Data Splitting

Finally, before stepping into the next stage, we have our last preprocessing of data, which is to split the whole 252 samples of working days into two sets, the test and train set. This split will help significantly in the following application and verification of models. We here preset the latest 50 working days, from May 11^{th} to July 22^{nd}, as the test group and other dates constitute the train group. To be more specific, we are supposing now that we are at the start of May 11^{th}, 2022 and have no ideas on the future information of S&P 500 index after this date. The only information we are assumed to acquire is from May 11^{th}, 2022 and the days before.

3. Volatility Models and Application

Basically, in this project, we design four sorts of methods to obtain daily volatility, in order to compare the estimated volatility by volatility models and the ordinarily defined volatility. The volatility models we will apply, and cover are two of these. We will briefly introduce all these four methods, from simple to complex, which respectively are constant volatility, close to close historical volatility, estimated volatility by EWMA and GARCH(1,1) models.

3.1 Constant Volatility

Constant volatility is the most straight one and can be easily obtained from calculate the volatility of daily return during a given time window. Here we continue our assumption of preset time window hyperparameter, which is 10 days consistently with the previous part of volatility auto correlation analysis (2.3, pp. 5-8). Then the up-to-date constant volatility we can obtain is based on the information of May 11^{th} and 9 days before, which is about 0.0237. Such constant volatility satisfies the assumption by the most common log-normal price model and Black Scholes model, but the constant assumption of the latter has been empirically proven and argued to be not sufficient in practice (Sataputera, 2003). Based on this, we infer the volatility assumed in log-normal price model is not supposed to be constant as well.

3.2 Historical Volatility (Close to Close Historical Volatility)

Close to close historical volatility, more known as historical volatility, is the most popular method to calculate the moving daily volatility. Most of daily volatility data from public financial website or organization is based on this method. Our volatility samples utilized in the volatility auto correlation analysis are exactly calculated by this method. The formula of historical volatility is explained by the equation below. In addition, daily volatility multiplied by the square root of number of working days in a year gets annualized volatility, but in this project, the non-annualized one is more suitable for our log-normal asset price model (Bennett and Gil, 2012). We can obtain and record the daily volatility for each date in both the test and train dataset based on the past 10-day information. This method breaks the restriction of constant volatility and captures the volatile feature of the underlying asset more precisely.

$$\sigma_t^2 = \frac{1}{n-1}\sum_{i=0}^{n-1}(R_{t-i}-\overline{R_i})$$

3.3 EWMA Volatility Model

EWMA model, more often called RiskMetrics model, is equivalent to Exponentially Weighted Moving Average model. Compared to the previous two methods, the EWMA volatility takes both historical return and volatility into account. The EWMA method has a formula as shown below for forecasting the daily volatility one day ahead. It includes one unknown parameter λ constricted within 0 to 1, which represents the proportion of information of the volatility yesterday we want to keep and at the meantime determines the proportion of information we want from the squared return yesterday. By plugging in this equation recursively, we can get a K-day ahead forecast of volatility once we know the information of the return in the past K days and the volatility K days before (Christoffersen, 2011). A large parameter gives more weight to older volatility. According to the guidance from a handbook by NIST/SEMATECH, the value of λ is usually set between 0.7 and 0.8. Therefore, here we set a median of 0.75 (NIST/SEMATECH, 2003). In EWMA model method, we still use a time window of 10 days.

$$\sigma_t^2 = \lambda\sigma_{t-1}^2 + (1-\lambda)R_{t-1}^2$$
$$0 < \lambda < 1$$

3.4 GARCH(1,1) Volatility Model

GARCH(1,1) model, is a statistical model abbreviated from Generalized Auto Regressive Condition Heteroskedasticity with order (1,1). The GARCH method in this project is in a way inspired by the volatility auto correlation analysis in the previous pages. GARCH has three unknown parameters ω, α and β as shown in the equation below. It features mean reversion towards a long run mean and is no longer constricted by adding the weights of historical squared return and variance up to 1. Compared to the EWMA model which is its simplified version with ω = 0, α = 1 - λ and β = λ, the GARCH(1,1) model can explain the effects of past squared return and volatility better. Then instead of presetting parameters like how we implement in the EWMA method, we want to estimate these three parameters depending on our train dataset. To estimate the paraments, we have three possible ways. The first one is the Maximum Likelihood Estimation (MLE) under the assumption that the daily return is equal to the volatility at the same date multiplied by a standard normally distributed random factor (Christoffersen, 2011). The second one is the Ordinary Least Square (OLS) estimator, where we simply regard the first parameter ω as intercept and the other two as effect coefficients. The last one way is just aligning the weight of historical variance β = λ = 0.75 with that in the EWMA method, and then repeating the first or second way with two unknown parameters ω and α. In this project, we choose the MLE suggested by literature and obtain ω = 0.00000524, α = 0.1408 and β = 0.8269 approximately. Still we use consistent time window of 10 days, estimating the daily volatility by plugging into the GARCH(1,1) equation recursively.

$$\sigma_t^2 = \omega + \alpha R_{t-1}^2 + \beta\sigma_{t-1}^2$$

Long Run Mean Variance: $\sigma^2 = \frac{\omega}{1-\alpha-\beta}$

$$\alpha + \beta < 1$$

After obtaining the volatility by the four methods above, we visualize the moving daily volatility in the test dataset as exhibited by the attached Figure 6 to help readers compare their difference more intuitively. In the graph, we can clearly see that the moving paths of the EWMA and GARCH daily volatility are similar, but the GARCH line seems to be slightly lower than the EWMA line. This assists to explain the final output of this project in some ways.

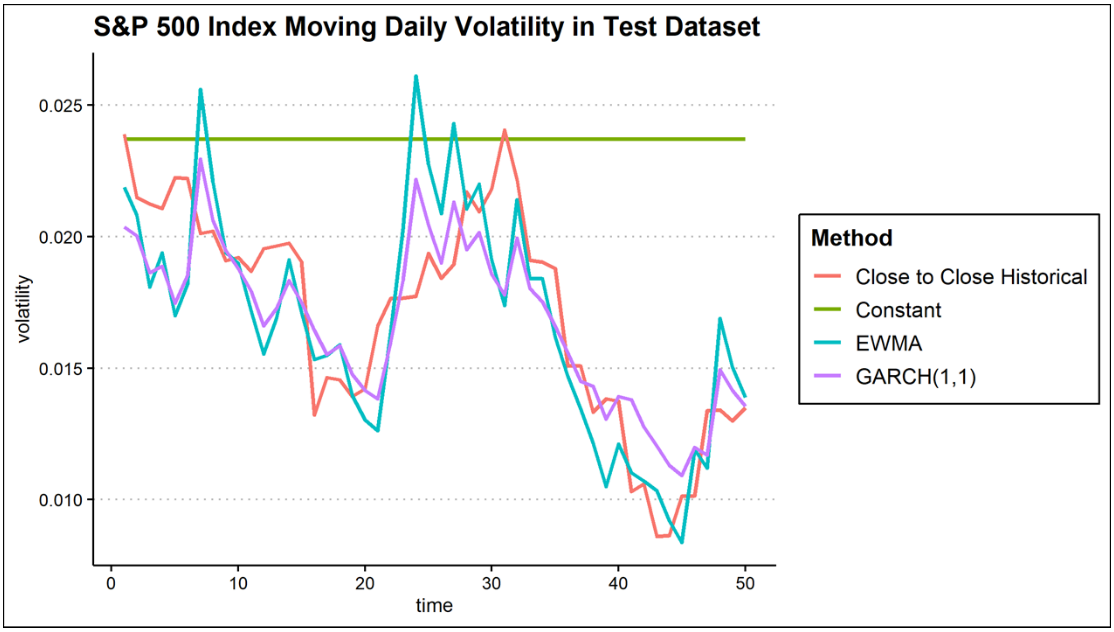


Figure 6: 10-Day Estimated Volatility Curves

4. Log-Normal Price Model and Estimation

4.1 Log-Normal Price Model

In this part, we are going to combine the volatility models we cover above and the log-normal price model to predict the daily return and log close price of S&P 500 index in the test dataset. The log-normal price model is the basis of pricing many derivatives which describes the feature of moving asset price as log-normally distributed. The model includes parameters μ and σ, which represent the trend of price moving and volatility respectively. For simplicity, we assume μ to be constant. By applying Ito's Lemma, we obtain the stochastic differential equation of log close price. The continuous version of this model exhibited below can be also transformed into a discrete-time version, which is more suitable for our dataset (Sharpe). Since we have the volatility estimated in the last part, the only parameter we need to estimate in this model is μ. Based on the MLE and OLS method, we can easily learn μ from the train dataset. The OLS estimator is just the arithmetic mean of the proportional daily return, and both the MLE and OLS provide similar outcomes of about -0.04%.to -0.05%. Generally, we select the estimated μ by the MLE method.

SDE:

$$dS = \mu S dt \ + \sigma S dW_t$$

$$dlog(S) = \left(\mu - \frac{1}{2}\sigma^2\right)dt + \sigma dW_t$$

Discrete-Time Version:

$$R_t = \frac{S_t - S_{t-1}}{S_{t-1}} = \mu + \sigma Z_{t-1}$$

$$\log(S_t) = \log(S_{t-1}) + \mu - \frac{1}{2}\sigma^2 + \sigma Z_{t-1}$$

$$Z_t \sim N(0,1) \ \ \mathit{for\ all\ t}$$

4.2 Prediction

Prediction by model is simple. We just utilize the estimated mean return μ and different estimated volatility 10 days before to create a random noise factor and predict the future log price and return recursively. To avoid the effects of unexpected extreme distribution of the noise, we set a seed to control random distribution and make sure the result evaluations by different methods are perfectly correlated and comparable. This makes the lines of moving predicted price have a similar pattern and their paths overlap, as shown in Figure 7. We observe that compared to the line by constant volatility, lines by the other three methods press closer to the real log close price line in grey.

$$R_t = \frac{S_t - S_{t-1}}{S_{t-1}} = \mu + \sigma_{t-1} Z_{t-1}$$

$$\log(S_t) = \log(S_{t-1}) + \mu - \frac{1}{2}{\sigma_{t-1}}^2 + \sigma_{t-1} Z_{t-1}$$

$$Z_t \sim N(0,1) \ \ \mathit{for\ all\ t}$$

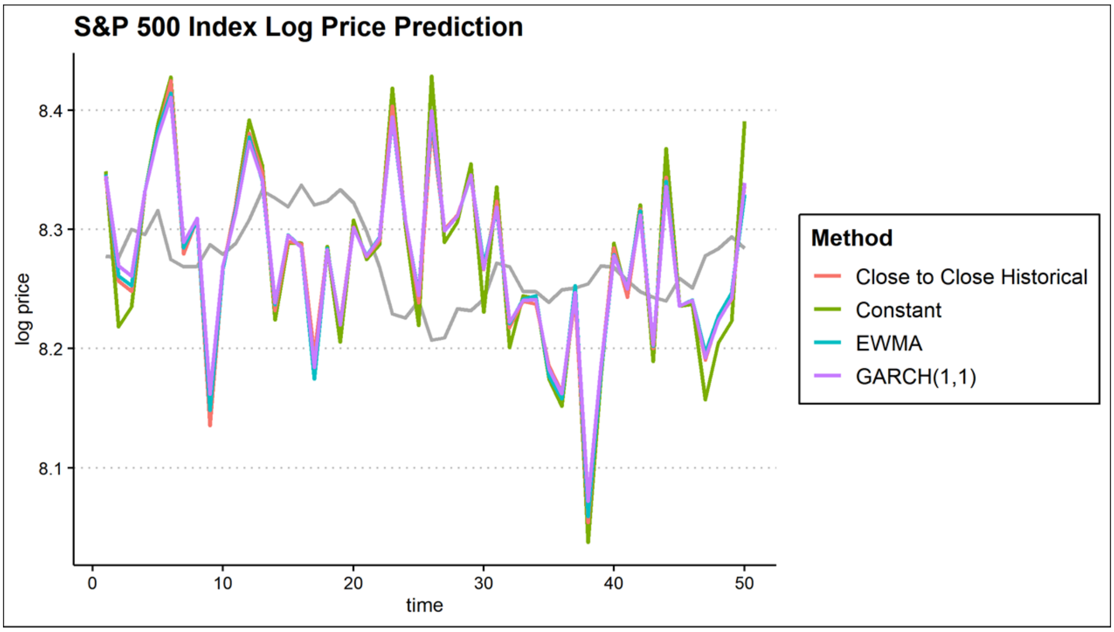


Figure 7: 10-Day Log Close Price Prediction Curves

4.3 Evaluation

After obtaining the predicted log price and daily return of S&P 500 Index, we get an opportunity to evaluate their relative performance based on certain evaluation metrics. Root Mean Square Error (RMSE) is one of the most popular evaluation metrics measuring the average magnitude of the error, which has a formula as shown below with x representing the variable we predict. Compared to Mean Square Error (MSE) and Mean Absolute Error (MAE) methods, RMSE owns the same unit as the x, and further puts more weight on large errors. These properties make RMSE more useful when extremely large errors are undesirable, and at the meantime, provide an intuitional comparison to its corresponding variable (JJ, 2016).

$$RMSE = \sqrt{\frac{1}{m}\sum_{j=1}^{m=50} \left(\hat{x}_j - x_j\right)^2}$$

From Table 1, we can see the RMSE of predicted S&P 500 Index log daily close price by constant volatility is the only one above 0.088 and larger than the other three. The historical, EWMA and GARCH results all reduce about 13% compared to the constant one. The results by these three sorts of daily volatility are close to each other, but as expected, the EWMA and GARCH results perform slightly better than the historical one with GARCH the best. Furthermore, the RMSE results of predicted S&P 500 Index daily return by the four methods follow the same pattern. The result by constant volatility reaches nearly to 3% and those by the other three improve about 18%. GARCH result still performs the best with approximate 2.36% and is followed by EWMA and historical results in sequence.

	Constant	Historical	EWMA	GARCH
Return RMSE	0.02955	0.02412	0.02379	0.02357

Table 1: 10-Day Daily Return Prediction RMSE

	Constant	Historical	EWMA	GARCH
Log Price RMSE	0.08873	0.07909	0.07806	0.07635

Table 2: 10-Day Log Close Price Prediction RMSE

5. Extension and Conclusion

In addition to the preset time window of 10 days, we can alter it to be wider or narrower. At the end of this project, we try a short window of 3 days and a long window of 20 days. The result with the 3-day time window, unsurprisingly follows the same pattern as that of the 10-day one, but in this set, the RMSE of both predicted return and log price by constant volatility method is closer to the other three. We expect to observe this because short time window provides more and put more weight on updated information of current volatility so that enables the model to capture the feature of volatility relatively more accurately. At the meantime, under this set, the predicted log price curve path is more alike to the real one.

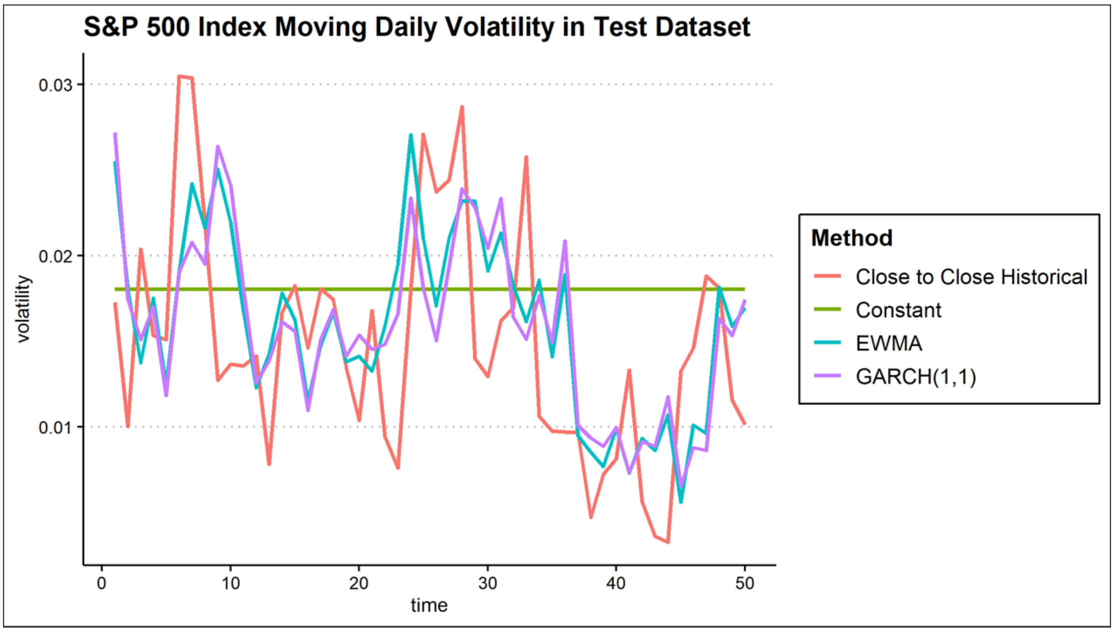


Figure 8: 3-Day Estimated Volatility Curves

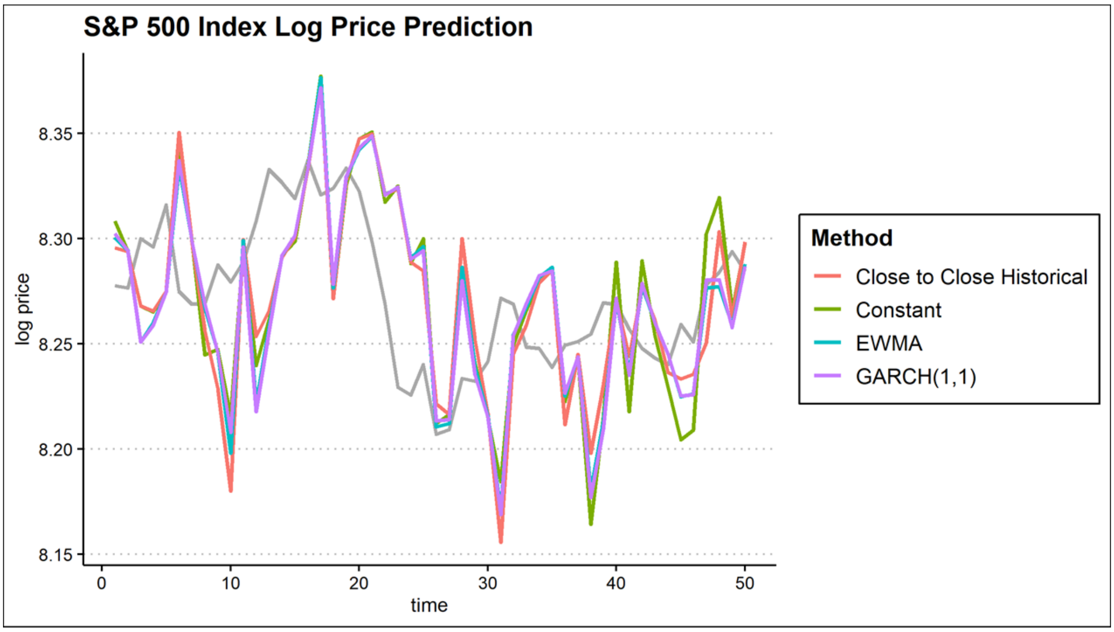


Figure 9: 3-Day Log Close Price Prediction Curves

	Constant	Historical	EWMA	GARCH
Return RMSE	0.02441	0.02276	0.02243	0.02198

Table 3: 3-Day Daily Return Prediction RMSE

	Constant	Historical	EWMA	GARCH
Log Price RMSE	0.04393	0.0437	0.04354	0.0435

Table 4: 3-Day Log Close Price Prediction RMSE

For the long time window of 20 days, as we expected as well, the RMSE of predicted return follows a similar pattern, but now the result of EWMA performs slightly better than GARCH. Surprisingly, under this set, the historical method performs the best in the RMSE of predicted log price, with GARCH and EWMA the second and third. This reveals the prediction can be unstable as the length of time window increases.

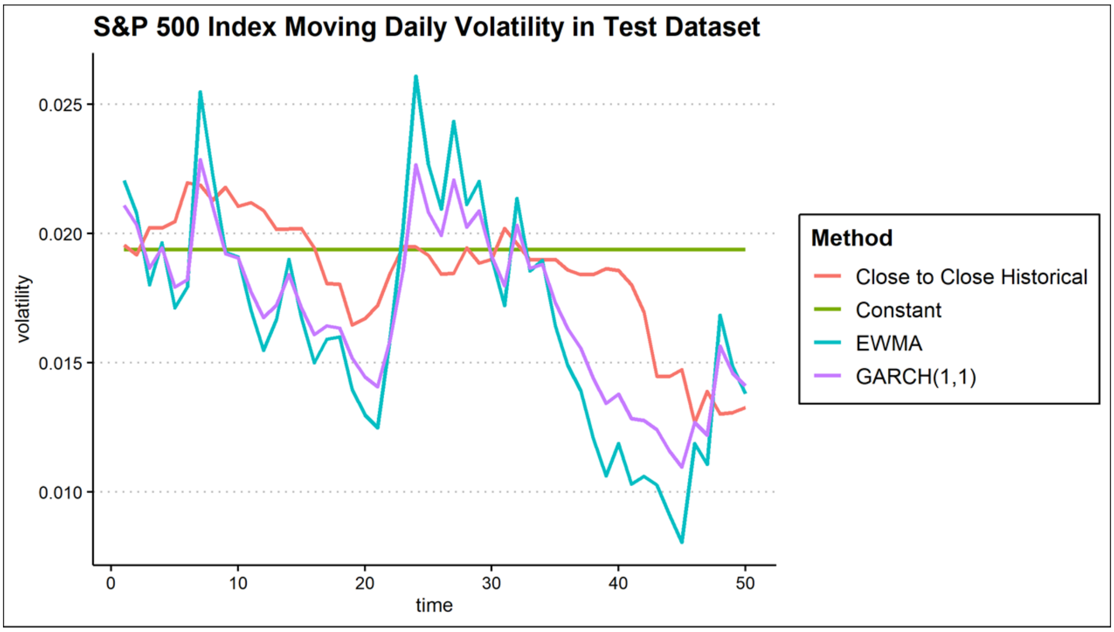


Figure 10: 20-Day Estimated Volatility Curves

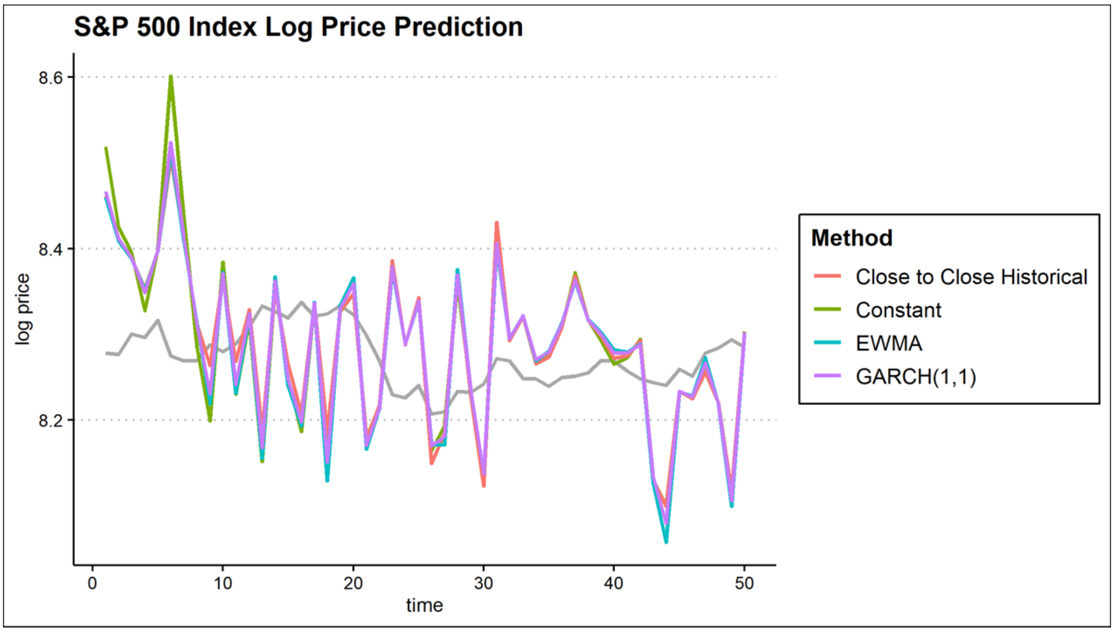


Figure 11: 20-Day Log Close Price Prediction Curves

	Constant	Historical	EWMA	GARCH
Return RMSE	0.02556	0.02497	0.02378	0.02389

Table 5: 20-Day Daily Return Prediction RMSE

	Constant	Historical	EWMA	GARCH
Log Price RMSE	0.10576	0.09434	0.10049	0.09808

Table 6: 20-Day Log Close Price Prediction RMSE

In conclusion, application of the EWMA and GARCH(1,1) volatility models has positive effects on the S&P 500 Index price and return prediction, even though as the time window widens, the close to close historical volatility method also has good performance as these two methods. Overall, the short-time ahead forecast of price and return is more suitable and long-time ahead forecast will be unstable. Currently, the simple log-normal price model is not sufficient to capture all features of the price moving. There is no great improvement after application of the EWMA and GARCH models on the log-normal price model, but we have reasons to believe that the combination of volatility models and more advanced price prediction model will be more effective.

References

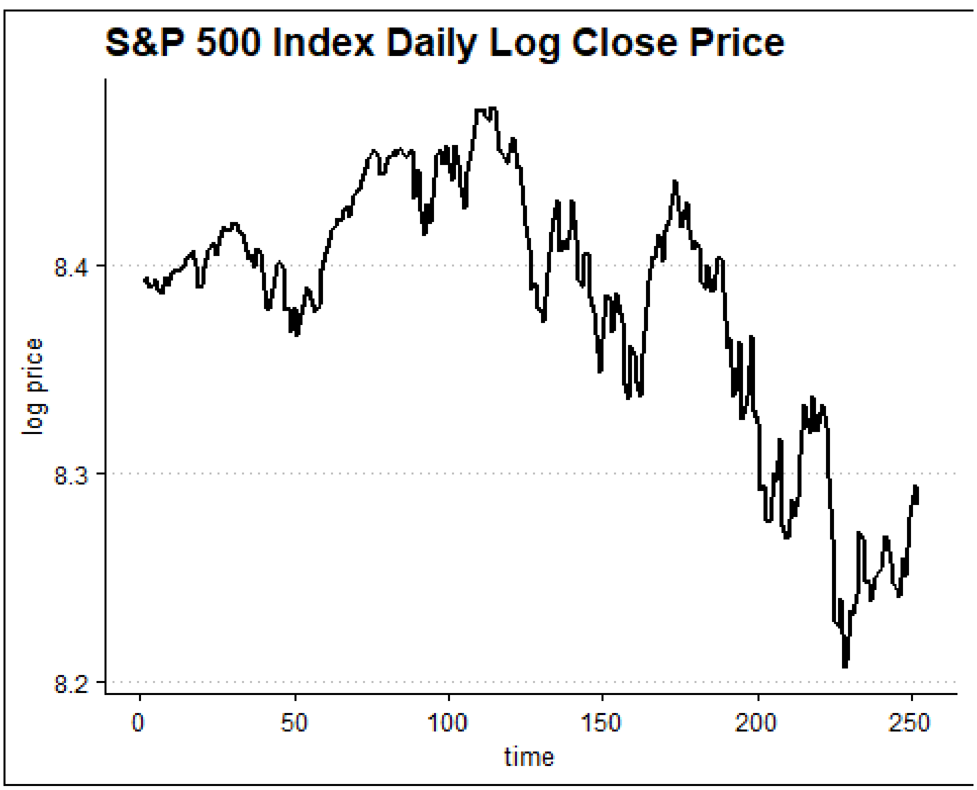
1. Colin Bennett and Miguel A. Gil (2012), *Measuring Historical Volatility*, Santander Global Banking and Markets
2. JJ (2016), *MAE and RMSE — Which Metric is Better?* Medium https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d
3. MarketWatch (Main Data Source), Accessed: July 24th, 2022 https://www.marketwatch.com/investing/index/spx/download-data
4. Michael J. Sharpe, *Log-normal Model for Stock Price*, Mathematics Department, UCSD https://mathweb.ucsd.edu/~msharpe/stockgrowth.pdf
5. NIST/SEMATECH (2003), *EWMA Control Charts*, Engineering Statistics Handbook, 6.2.3.4 https://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm
6. Peter Christoffersen (2011), *Elements of Financial Risk Management*, Elsevier Science & Technology, Charpter 4, pp. 67-92
7. Rudi Kurniawan (2015), *Modeling Volatility in the Indonesian Stock Market: An Exercise Using GARCH Model*, pp. 6-9
8. Tim Plaehn (2019), *Is the Total Stock Market More Stable Than the S&P 500?* ZACKS Finance https://finance.zacks.com/total-stock-market-stable-sp-500-10425.html

APPENDIX

```
library(tidyverse)
library(zoo)
library(ggthemes)
library(gridExtra)

data = read.csv("SP500.csv")
```

- Log close Price

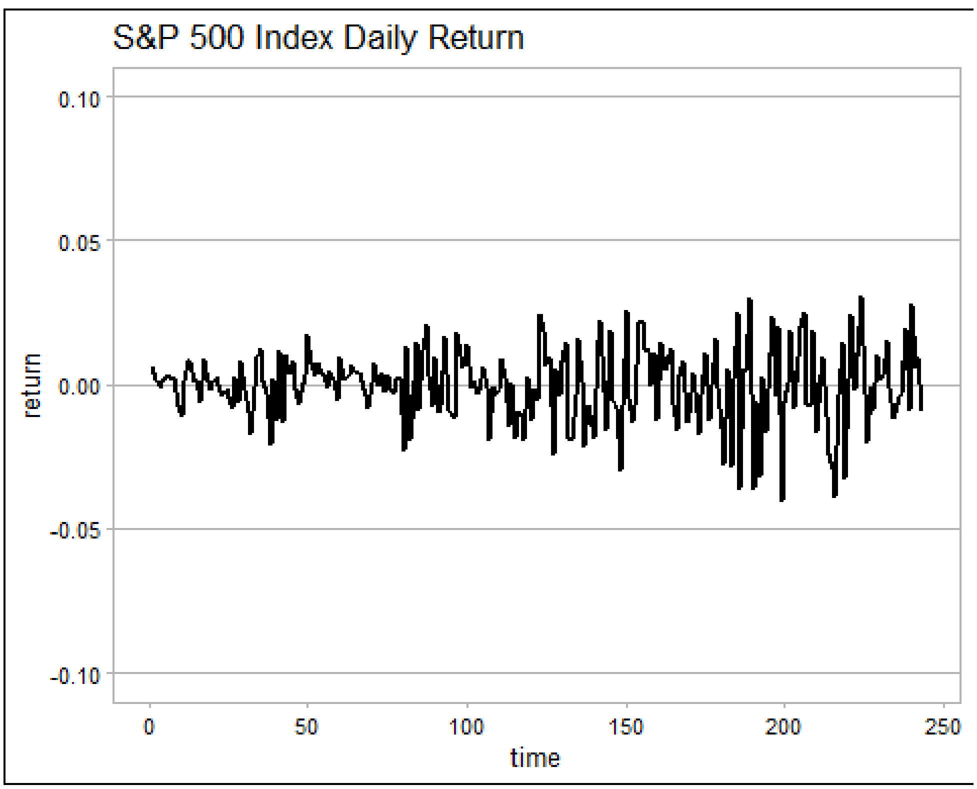
```
data = data %>% select(Date, Close)
data$Close = as.numeric(gsub(",", "", data$Close, fixed=TRUE))
data = data %>% mutate(Close.lead = lead(Close,1)) %>%
filter(is.na(Close.lead)==FALSE)
# Log Close Price
data = data %>% mutate(Log.Close = log(Close))
# Return
data = data %>% mutate(return = Close/Close.lead-1)

ggplot()+
  geom_line(aes(1:nrow(data),rev(data$Log.Close)),size=1)+
  labs(x="time", y="log price", title="S&P 500 Index Daily Log Close
Price")+
  theme_clean()
```



```
ggsave("Daily Log Close Price.png",width=8,height=4.5)
```

- Return

```
# Volatility
window = 10
data$volatility = append(rollapply(data$return, width=window, FUN=sd),
rep(0,window-1))
data = data %>% filter(volatility!=0)

# Split
test_samples = 50
test = data[1:test_samples,]
train = data[(test_samples+1):nrow(data),]

ggplot(data = data)+
  geom_line(aes(1:nrow(data),rev(return)),size=1)+
  ylim(-0.1,0.1)+
  labs(x="time", y="return", title="S&P 500 Index Daily Return")+
  theme_calc()
```




```
ggsave("Daily Return.png",width=8,height=4.5)
```

- Constant Volatility

```
# Estimators
vol_train = sd(train$return)
```

```
mu_train = mean(train$return)
vol_const = data$volatility[test_samples+1]

lognormal = function(para) {
  mu = para[1]
  loglike = 0
  s = vol_train

  for (i in (nrow(train)-1):1) {
    loglike = loglike +
log(dnorm(log(train$Close[i]/train$Close[i+1])+0.5*s^2,mu,s))
  }
  return(-loglike)
}

para = -0.01
mu_mle = optim(para,lognormal,gr=NULL,method="BFGS")$par

set.seed(42)
test$noise.const = rnorm(nrow(test),0,vol_const)
test$return.pred.const = mu_train + test$noise.const
rmse_const = sqrt(sum((test$return.pred.const-
test$return)^2)/(nrow(test)-1))
```

- Historical

```
set.seed(42)
for (i in seq(1,nrow(test))) {test$noise.hist[i] =
rnorm(1,0,test$volatility[i])}
test$return.pred.hist = mu_mle + test$noise.hist
rmse_hist = sqrt(sum((test$return.pred.hist-
test$return)^2)/(nrow(test)-1))
```

- EWMA

```
# Estimators
lambda = 0.75

ewma = function(i,l,w) {
  v = l^w * data$volatility[i+w]^2
  for (j in seq(1,w)){
    v = v + (1-l)*l^(j-1)*data$return[i+j]^2
  }
  sqrt(v)
}

for (i in seq(1,nrow(test))) {test$vol.ewma[i] = ewma(i,lambda,window)}

for (i in seq(1,nrow(train))) {train$vol.ewma[i] =
ewma(test_samples+i,lambda,window)}
```

```
data$vol.ewma = append(test$vol.ewma,train$vol.ewma)

#test$vol.ewma = sqrt(lead(lambda*test$volatility^2 + (1-
lambda)*test$return^2, 1))
#test$vol.ewma[nrow(test)] = sqrt(lambda*train$volatility[1]^2 + (1-
lambda)*train$return[1]^2)

set.seed(42)
for (i in seq(1,nrow(test))) {test$noise.ewma[i] =
rnorm(1,0,test$vol.ewma[i])}
test$return.pred.ewma = mu_mle + test$noise.ewma
rmse_ewma = sqrt(sum((test$return.pred.ewma-
test$return)^2)/(nrow(test)-1))
```

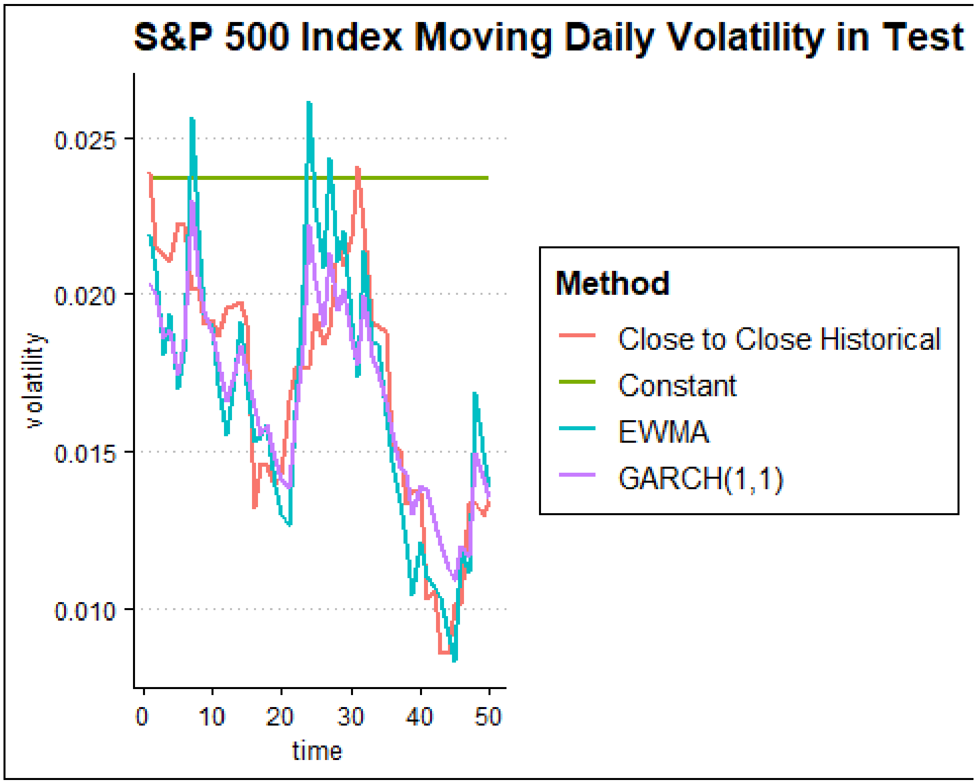
- GARCH(1,1)

```
garch_loglike = function(para) {
  w = para[1]
  a = para[2]
  b = para[3]
  loglike = 0
  s2 = 0

  for (i in (nrow(train)-1):1) {
    s2 = w + a*train$return[i+1]^2 + b*s2
    loglike = loglike + log(dnorm(train$return[i],0,sqrt(s2)))
  }
  return(-loglike)
}

para_mle = optim(c(0.1,0.2,0.75),garch_loglike)

y = head(train$volatility,-1)^2
x1 = train$return[2:nrow(train)]^2
x2 = train$volatility[2:nrow(train)]^2

model_garch = lm(y~x1+x2)

omega = para_mle$par[1]
alpha = para_mle$par[2]
beta = para_mle$par[3]

#omega = model_garch$coefficients[1]
#alpha = model_garch$coefficients[2]
#beta = model_garch$coefficients[3]

#omega = para_mle$par[1]
#alpha = para_mle$par[2]
#beta = 0.75
```

```
garch = function(i,w,a,b,l) {
  v = data$volatility[i+l]^2
  for (j in l:1) {
    v = w + a*data$return[i+j]^2 + b*v
    s = sqrt(v)
  }
  return (s)
}

for (i in seq(1,nrow(test))) {test$vol.garch[i] =
garch(i,omega,alpha,beta,window)}

for (i in seq(1,nrow(train))) {train$vol.garch[i] =
garch(test_samples+i,omega,alpha,beta,window)}

data$vol.garch = append(test$vol.garch,train$vol.garch)

set.seed(42)
for (i in seq(1,nrow(test))) {test$noise.garch[i] =
rnorm(1,0,test$vol.garch[i])}
test$return.pred.garch = mu_mle + test$noise.garch
rmse_garch = sqrt(sum((test$return.pred.garch-
test$return)^2)/(nrow(test)-1))
```

- Plot

```
ggplot(data=test)+

geom_line(aes(x=seq(1,nrow(test)),y=rep(vol_const,nrow(test)),group=1,c
olor="Constant"),size=1)+

geom_line(aes(x=seq(1,nrow(test)),y=rev(volatility),group=1,color="Clos
e to Close Historical"),size=1)+

geom_line(aes(x=seq(1,nrow(test)),y=rev(vol.ewma),group=1,color="EWMA")
,size=1)+

geom_line(aes(x=seq(1,nrow(test)),y=rev(vol.garch),group=1,color="GARCH
(1,1)"),size=1)+
  labs(y="volatility",x="time", title="S&P 500 Index Moving Daily
Volatility in Test Dataset", color="Method")+
  theme_clean()
```

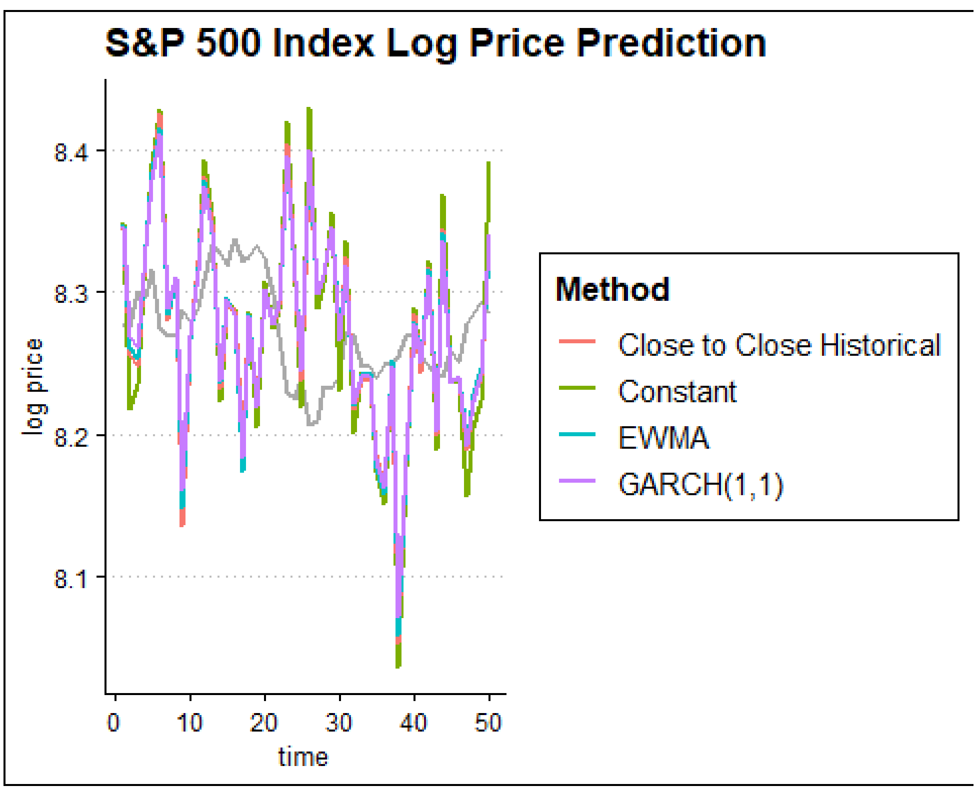



```
ggsave("Volatility by 4 Methods.png",width=8,height=4.5)
```

- Log Stock Price Prediction

```
log_price_pred = function(w,m,s) {
  set.seed(42)
  log_price_pred_list = c()
  for (i in 1:test_samples) {
    vol = data[i+w,s]
    if (s == "constant") {
      vol = vol_const
    }
    log_price = data$Log.Close[i+w]
    for (j in 1:w) {
      log_price = log_price + (m-0.5*vol^2) + rnorm(1,0,vol)
    }
    log_price_pred_list = append(log_price_pred_list,log_price)
  }
  log_price_pred_list
}

lp = data$Log.Close[1:test_samples]
# Constant
lp_pred_const= log_price_pred(window,mu_mle,"constant")
rmse_log_const = sqrt(sum((lp-lp_pred_const)^2)/(length(lp)-1))
# Historical
lp_pred_hist= log_price_pred(window,mu_mle,"volatility")
```

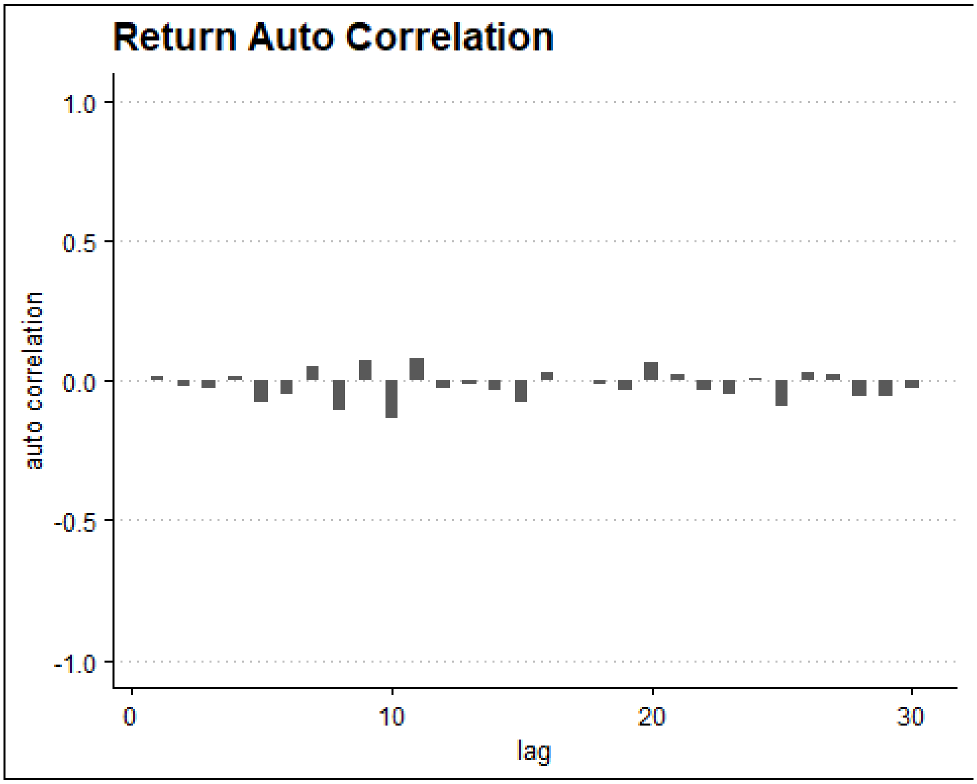
```
rmse_log_hist = sqrt(sum((lp-lp_pred_hist)^2)/(length(lp)-1))
# EWMA
lp_pred_ewma= log_price_pred(window,mu_mle,"vol.ewma")
rmse_log_ewma = sqrt(sum((lp-lp_pred_ewma)^2)/(length(lp)-1))
# GARCH
lp_pred_garch= log_price_pred(window,mu_mle,"vol.garch")
rmse_log_garch = sqrt(sum((lp-lp_pred_garch)^2)/(length(lp)-1))

ggplot()+
  geom_line(aes(1:length(lp),rev(lp)),color="dark grey",size=1)+

geom_line(aes(1:length(lp),rev(lp_pred_const),color="Constant"),size=1)
+
  geom_line(aes(1:length(lp),rev(lp_pred_hist),color="Close to Close
Historical"),size=1)+
  geom_line(aes(1:length(lp),rev(lp_pred_ewma),color="EWMA"),size=1)+

geom_line(aes(1:length(lp),rev(lp_pred_garch),color="GARCH(1,1)"),size=
1)+
  labs(y="log price",x="time",title="S&P 500 Index Log Price
Prediction",color="Method")+
  theme_clean()
```



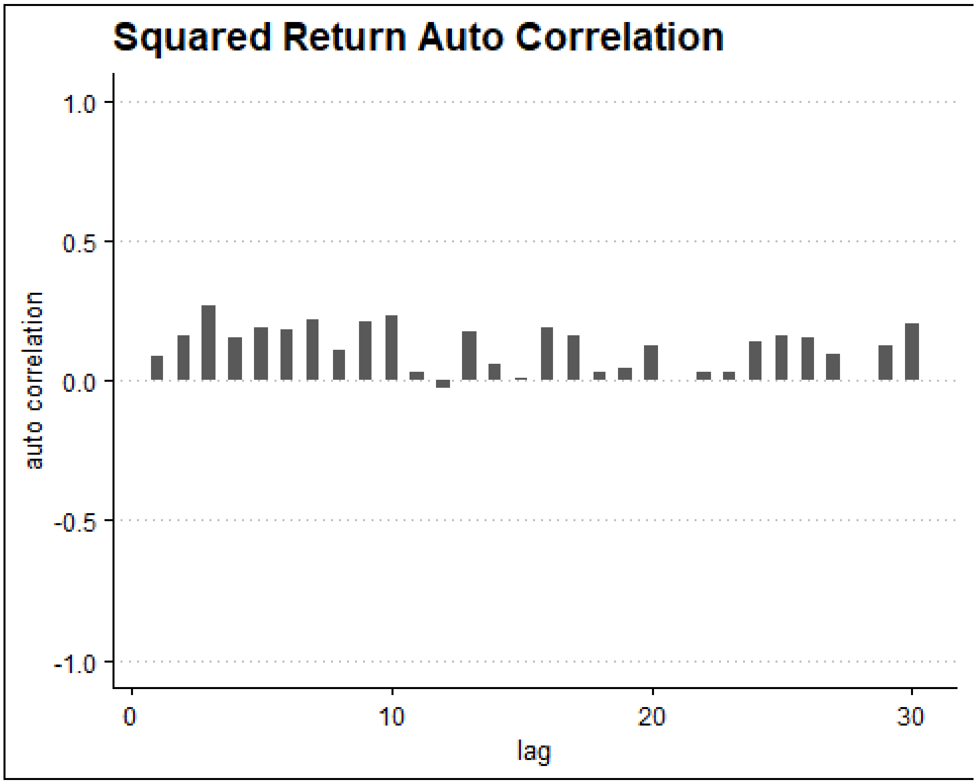

```
ggsave("Log Price Prediction.png",width=8,height=4.5)
```

- Auto Correlation

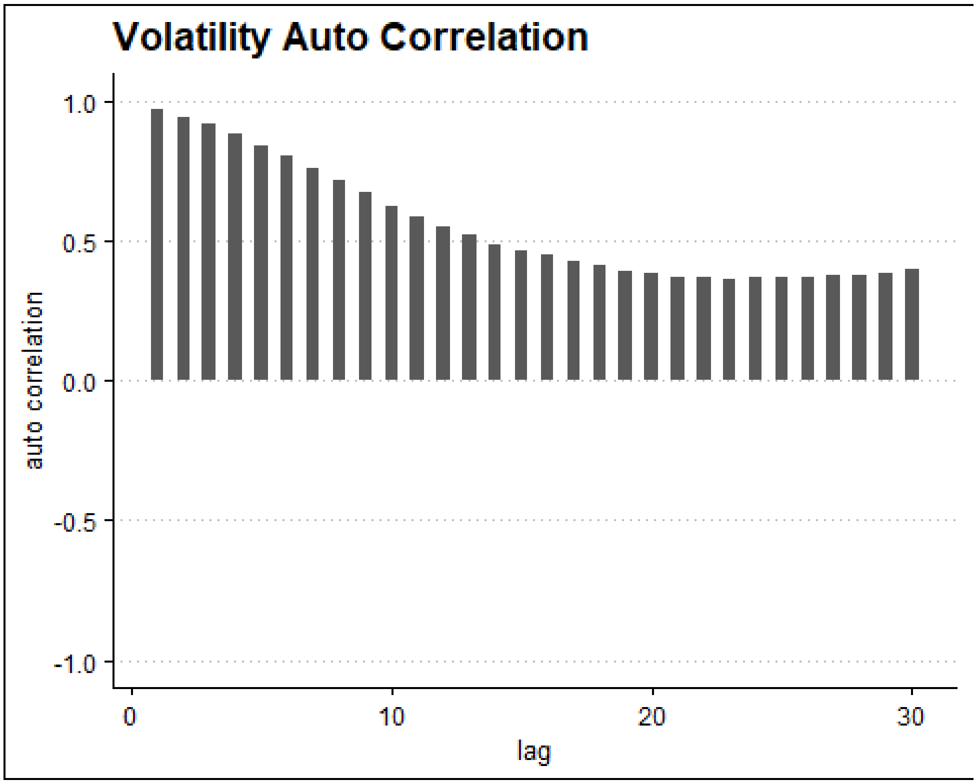
```r
auto_corr = function(lag,sample) {
  if (lag != 0) {
    sample.lag = lag(sample,lag)
    x = tail(sample,-lag)
    y = tail(sample.lag,-lag)
    return (cor(x,y))
  } else {
    return (1)
  }
}

ac_return = c()
ac_r2 = c()
ac_vol = c()

lag_max = 30

for (i in 1:lag_max) {
  ac_return = append(ac_return,auto_corr(i,data$return))
}

for (i in 1:lag_max) {
  ac_r2 = append(ac_r2,auto_corr(i,data$return^2))
}

for (i in 1:lag_max) {
  ac_vol = append(ac_vol,auto_corr(i,data$volatility))
}

ggplot()+
  geom_col(aes(1:lag_max,ac_return),width=0.5)+
  ylim(-1,1)+
  labs(title="Return Auto Correlation",y="auto correlation", x="lag")+
  theme_clean()
```




```
ggsave("AC_Return.png",width=8,height=4.5)

ggplot()+
  geom_col(aes(1:lag_max,ac_r2),width=0.5)+
  ylim(-1,1)+
  labs(title="Squared Return Auto Correlation",y="auto correlation",
x="lag")+
  theme_clean()
```




```
ggsave("AC_Squared Return.png",width=8,height=4.5)

ggplot()+
  geom_col(aes(1:lag_max,ac_vol),width=0.5)+
  ylim(-1,1)+
  labs(title="Volatility Auto Correlation",y="auto correlation",
x="lag")+
  theme_clean()
```




```
ggsave("AC_Volatility.png",width=8,height=4.5)
```

- RMSE Table

```
png("Return RMSE.png", width=400, height=100)
p =
round(data.frame(Constant=rmse_const,Historical=rmse_hist,EWMA=rmse_ewm
a,GARCH=rmse_garch),5)
row.names(p) = "Return RMSE"
p = tableGrob(p)
grid.arrange(p)

png("Log Price RMSE.png", width=400, height=100)
q =
round(data.frame(Constant=rmse_log_const,Historical=rmse_log_hist,EWMA=
rmse_log_ewma,GARCH=rmse_log_garch),5)
row.names(q) = "Log Price RMSE"
q = tableGrob(q)
grid.arrange(q)
```