CSI 300 Index Volatility Forcasting

Sean Xiang

2017/8/22

In Finance, Volatility(symbol) is the degree of variation of a trading price series over time as measured by the standard deviation of logarithmic returns. Over this project different models are used to calculate rolling volatility that intends to approximate future volatility over the upcoming trading periods(T = 30, 90,ect.) Graphs were generated and an error index was invented to measure the deviation of each model both visually and quantitatively. In addition, The financial market was brokedown into three parts based on market trends: bull market, bear market, flat market, and each of which was given model comparison and analysis. Finally, A time-series Garch(1,1) model was implemented to forecast future monthly volatilities; The forecasted result is pretty similar to realized one. The following is our agenda:

1. Introduction
2. Simple Moving Average (SMA
3. OHLC Model(Parkinson High-Low Volatility Model, German-Klass Model, Rogers-Satchel Model, Yang-Zhang Model)
4. Weighted Moving Average (WMA, EWMA
5. Perfomance Analytics (Error Index, Bull Market, bear Market, Flat Market)
6. Time Series / ARCH Garch(1,1) Model
7. Further Improvement

1. Introduction

Volatility refers to the amount of uncertainty or risk about the size of changes in a security's value. A higher volatility means that a security's value can potentially be spread out over a larger range of values. This means that the price of the security can change dramatically over a short time period in either direction. A lower volatility means that a security's value does not fluctuate dramatically, but changes in value at a steady pace over a period of time.

2. Simple Moving Average (SMA)

A simple moving average (SMA) is an arithmetic moving average calculated by adding the closing price of the security for a number of time periods and then dividing this total by the number of time periods.In financial applications a simple moving average (SMA) is the unweighted mean of the previous n data.

Data Collecting

Source: Wind Financial Terminal

Data: Historical daily return on CSI 300 index, which tracks the Shanghai and Shenzhen Markets

Time: 2007/7/4 - 2017/8/4

Indicator(s): Opening price, High price, Low price, Closing price, Volume

```
library(ggplot2)
library(tidyr)
library(ggthemes)
library(lubridate)
library(zoo)
```

```r
library(xts)
library(magrittr)
library(roll)
library(TTR)
library(quantmod)
library(base)
```

```r
library(readxl)
Index_Data <- read_excel("~/Documents/Bu Academics/Rising Senior Summer/htf /Project.xlsx", col_names =
names(Index_Data)[1] <- "Date"
names(Index_Data)[2] <- "PerChange"
names(Index_Data)[3] <- "Open"
names(Index_Data)[4] <- "Close"
names(Index_Data)[5] <- "High"
names(Index_Data)[6] <- "Low"
names(Index_Data)[7] <- "Vol"
View(Index_Data)
```

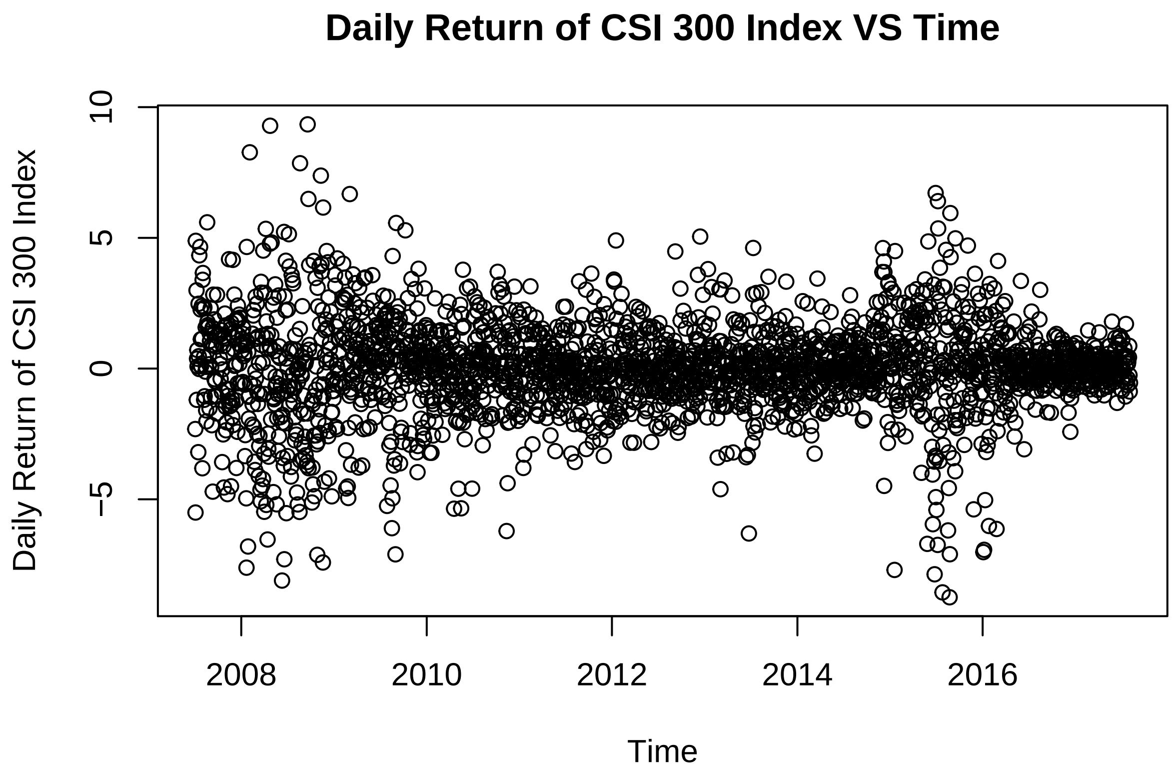
Summary Statistics

```r
summary(Index_Data[-1])
```

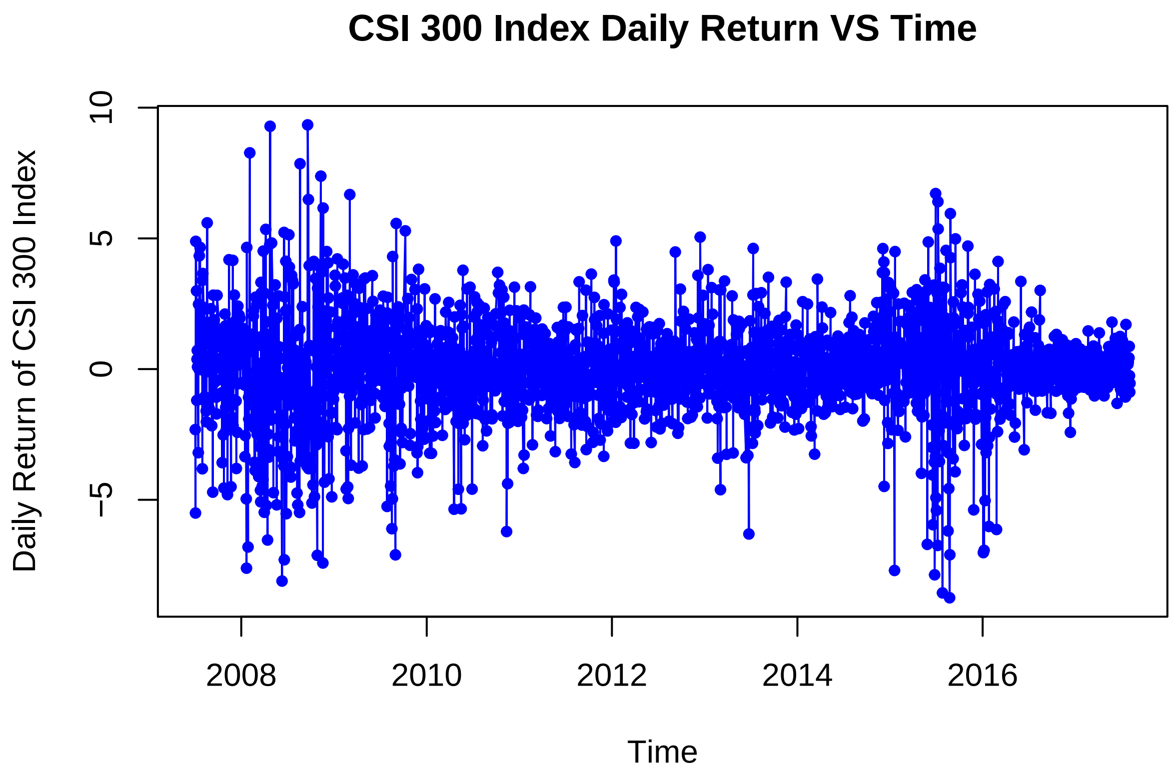
```
##    PerChange             Open          Close           High     
##  Min.   :-8.74769   Min.   :1615   Min.   :1628   Min.   :1648  
##  1st Qu.:-0.78333   1st Qu.:2452   1st Qu.:2452   1st Qu.:2473  
##  Median : 0.06635   Median :3042   Median :3044   Median :3073  
##  Mean   : 0.01543   Mean   :3078   Mean   :3081   Mean   :3111  
##  3rd Qu.: 0.88119   3rd Qu.:3440   3rd Qu.:3445   3rd Qu.:3467  
##  Max.   : 9.34198   Max.   :5862   Max.   :5877   Max.   :5892  
##       Low            Vol           
##  Min.   :1607   Min.   :1.762e+09  
##  1st Qu.:2433   1st Qu.:4.712e+09  
##  Median :3011   Median :7.011e+09  
##  Mean   :3045   Mean   :9.677e+09  
##  3rd Qu.:3415   3rd Qu.:1.041e+10  
##  Max.   :5816   Max.   :6.864e+10
```

```r
theme_update(plot.title = element_text(hjust = 0.5))
# In light summary statistics CSI 300 index has the biggest daily return in 2018/9/19 with a staggering
```

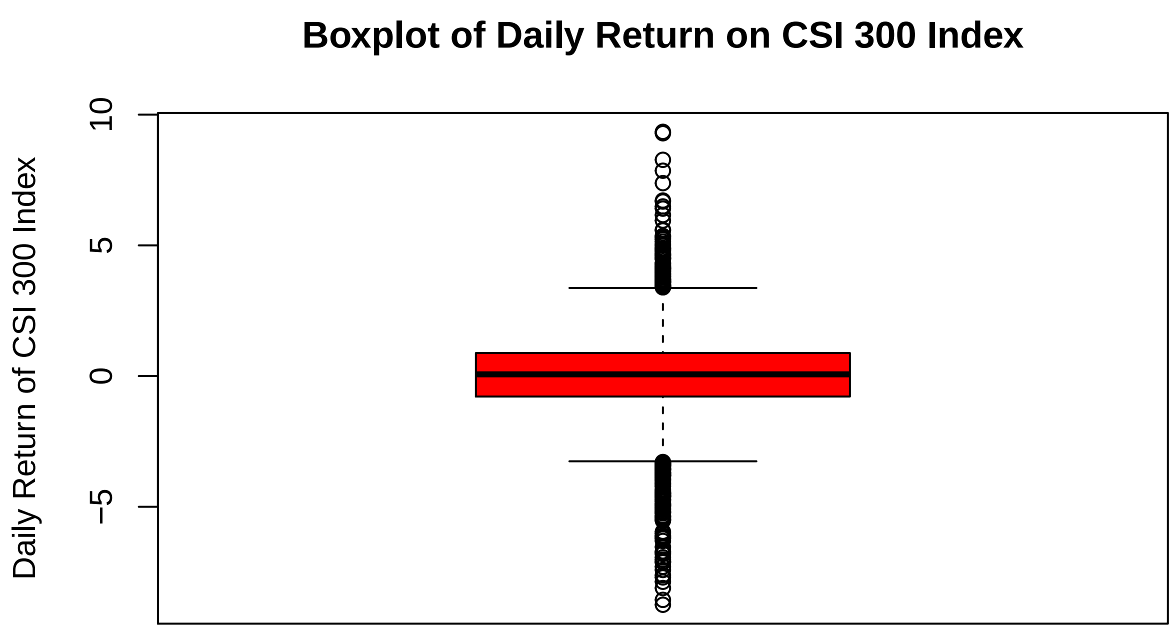
```r
plot(Index_Data$Date,Index_Data$PerChange,xlab = "Time",ylab = "Daily Return of CSI 300 Index", main =
```



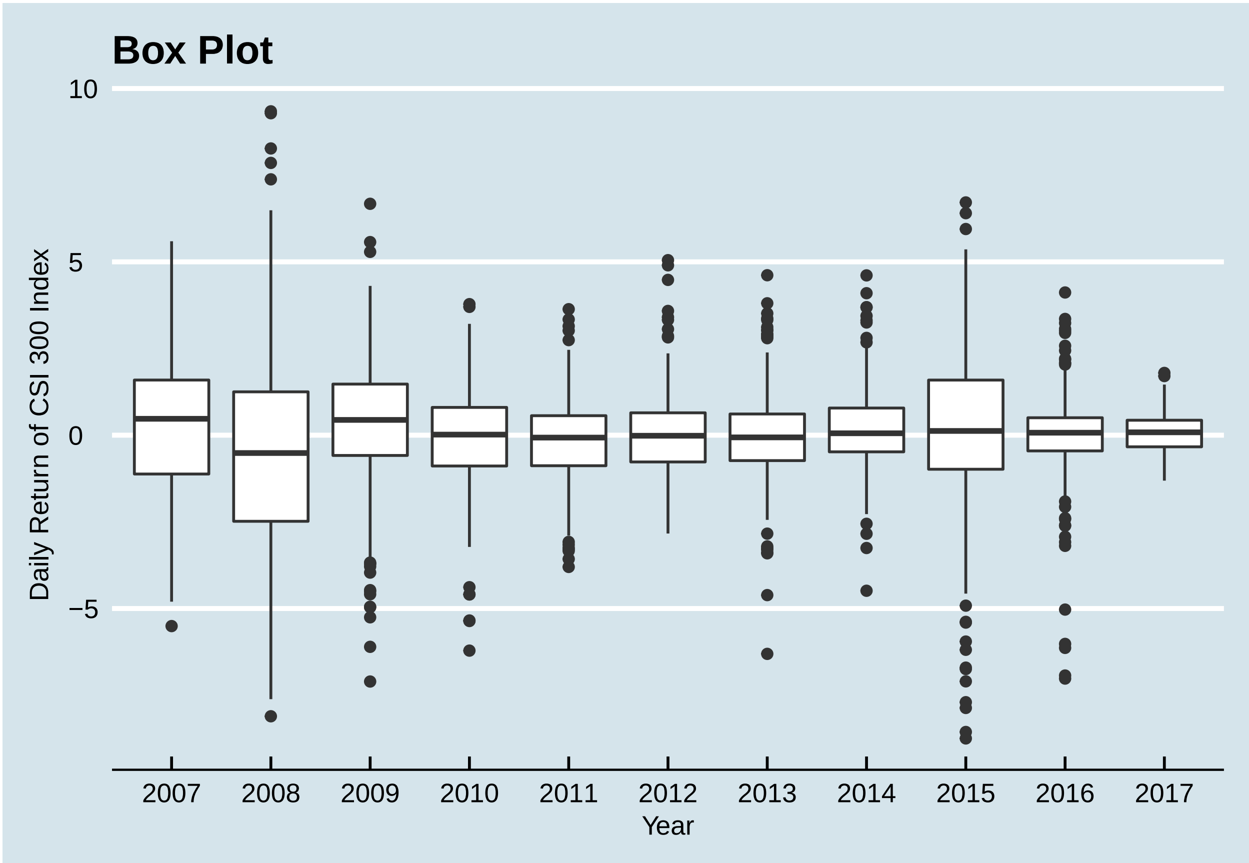

```r
plot(Index_Data$Date, Index_Data$PerChange, type = 'o', xlab = "Time", ylab = "Daily Return of CSI 300 I
```




```r
boxplot(Index_Data$PerChange, col = "red", border = "black",ylab = "Daily Return of CSI 300 Index", mai
```




```
ggplot(data = Index_Data, aes(x = as.character(year(Index_Data$Date)), y = Index_Data$PerChange)) + xlal
```




```
Data <- as.matrix(Index_Data$PerChange) / 100

# Calculate moving var
Moving_variance <- roll_var(Data,300)
Moving_variance <- Moving_variance %>% na.omit()    # omit NA
```

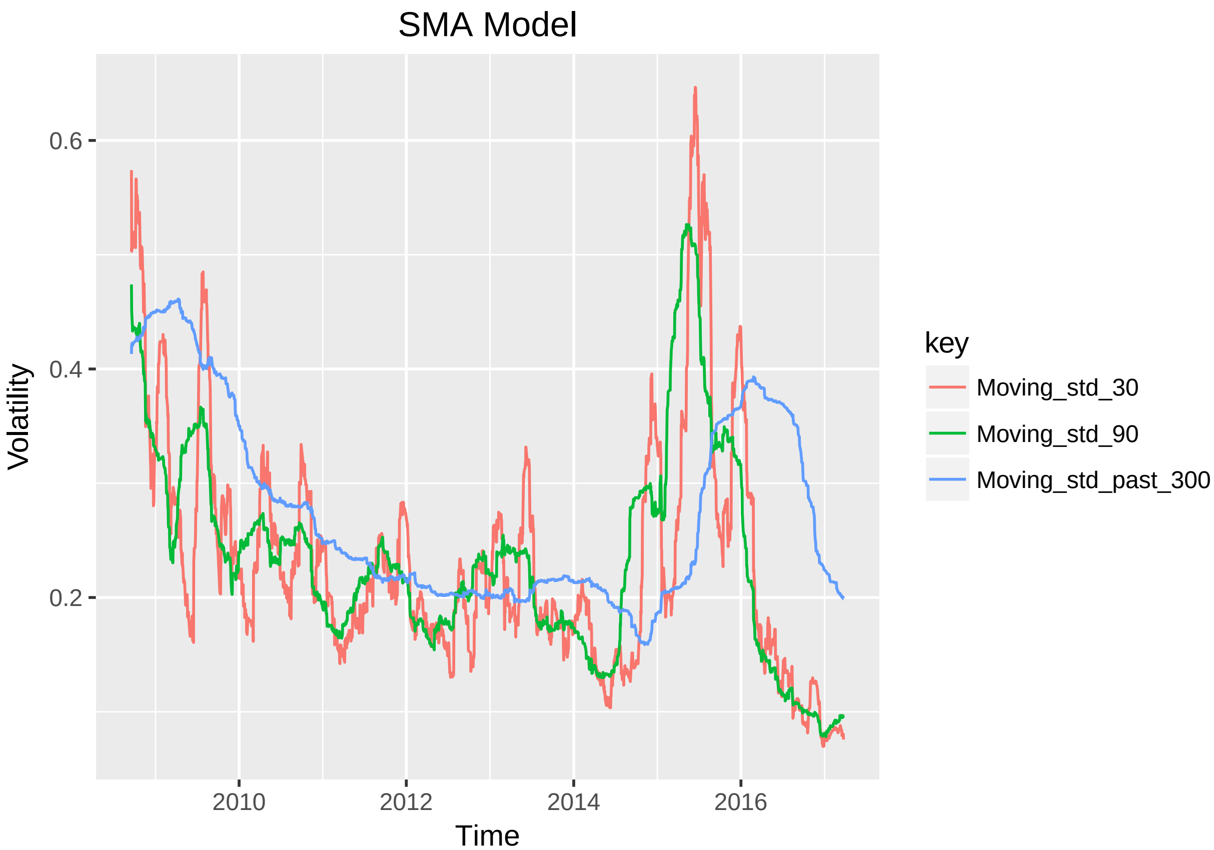
```r
# Calculate moving std
Moving_std_past_300<- Moving_variance %>% sqrt() * sqrt(240)
```

Next we use std as a proxy of volatility to speculate CSI 300 Index Volatility in next 30 and 90 trading days repsectively.

```r
# Moving average in the next 90 days
Data_testing <- as.matrix(Data[c(301:length(Data))])

# Moving std in the next 30 days
Moving_std_30 <- roll_var(Data_testing, 30) %>% na.omit() %>% sqrt()
Moving_std_30 <- Moving_std_30 * sqrt(240)

# Moving std in the next 90 days
Moving_std_90 <- roll_var(Data_testing, 90) %>% na.omit() %>% sqrt()
Moving_std_90 <- Moving_std_90 * sqrt(240)
```

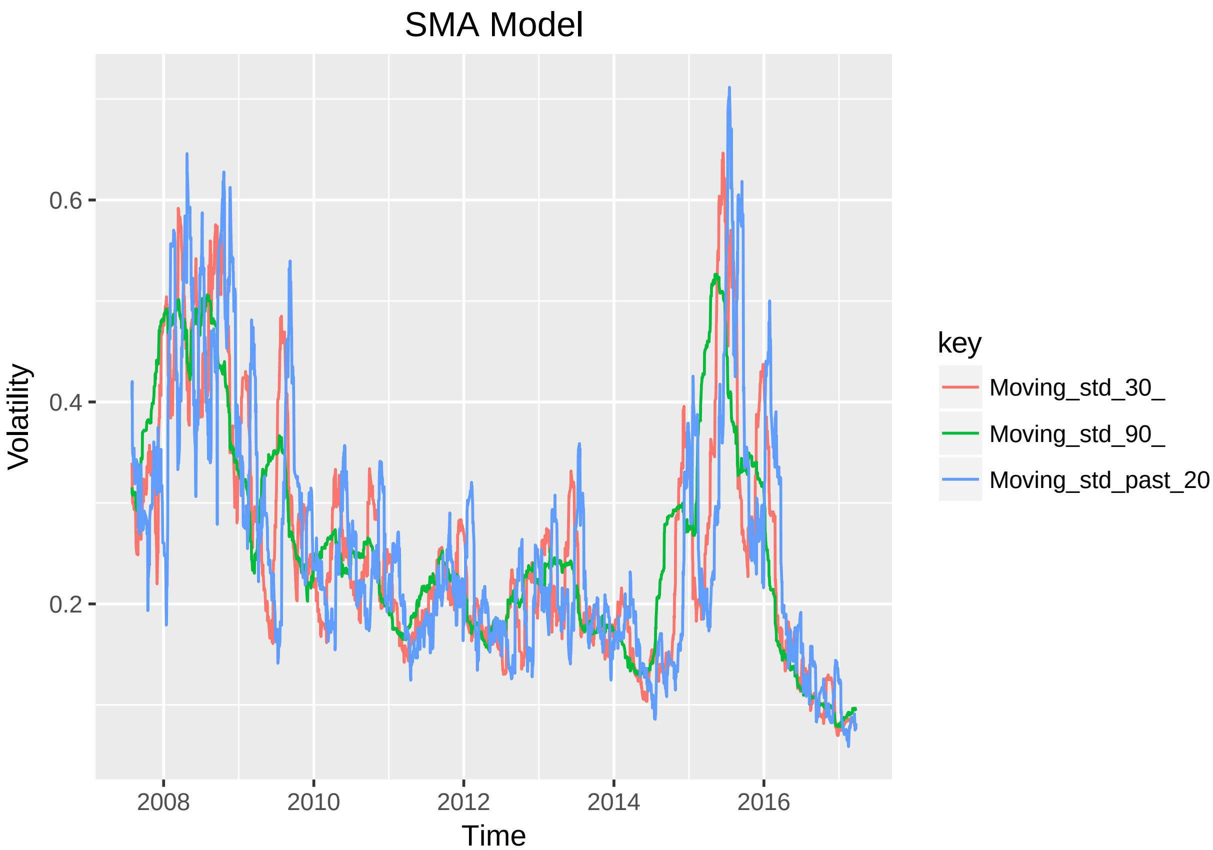
Now, put these time series financial data into perspective. Since each Moving_std have different width, we take a standard width as a benchmark.

```r
# N = 300
sta_width <- length(Moving_std_90)
Moving_time <- Index_Data[300:(300 + sta_width-1),1]
Moving_std_past_300<- as.matrix(Moving_std_past_300[1:sta_width])
Moving_std_30 <- as.matrix(Moving_std_30[1:sta_width])
Moving_std_table <- data.frame(cbind(Moving_time, Moving_std_past_300, Moving_std_30, Moving_std_90))
colnames(Moving_std_table)<- c("Time", "Moving_std_past_300", "Moving_std_30", "Moving_std_90")

# Generate Time_series Graph
Moving_std_table %>%
  gather(key, Volatility, Moving_std_past_300, Moving_std_30, Moving_std_90) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("SMA Model")
```



As we can tell from the graph the forecasted future volatility curve, from which uses past 300 trading days daily return to calculate, seems too smooth compared to the next 30 days and next 90 days volatility curves benchmark. It might be a good idea to take past 20 trading days instead to make predicted volatility curve more oscillating.

```
# N = 20
Moving_std_past_20<- roll_sd(Data, 20) %>% na.omit() * sqrt(240)

Data_testing_ <- as.matrix(Data[c(21:length(Data))])

# Moving std in the next 30 days
Moving_std_30_ <- roll_sd(Data_testing_, 30) %>% na.omit() * sqrt(240)

# Moving std in the next 90 days
Moving_std_90_<- roll_sd(Data_testing_, 90) %>% na.omit() * sqrt(240)

sta_width_ <- length(Moving_std_90_)
Moving_time_ <- Index_Data[20:(20 + sta_width_- 1),1]
Moving_std_past_20<- as.matrix(Moving_std_past_20[1:sta_width_])
Moving_std_30_ <- as.matrix(Moving_std_30_[1:sta_width_])
Moving_std_table_ <- data.frame(cbind(Moving_time_, Moving_std_past_20, Moving_std_30_, Moving_std_90_))
colnames(Moving_std_table_)<- c("Time", "Moving_std_past_20", "Moving_std_30_", "Moving_std_90_")
Moving_std_table_ %>%
  gather(key, Volatility, Moving_std_past_20, Moving_std_30_, Moving_std_90_) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("SMA Model")
```



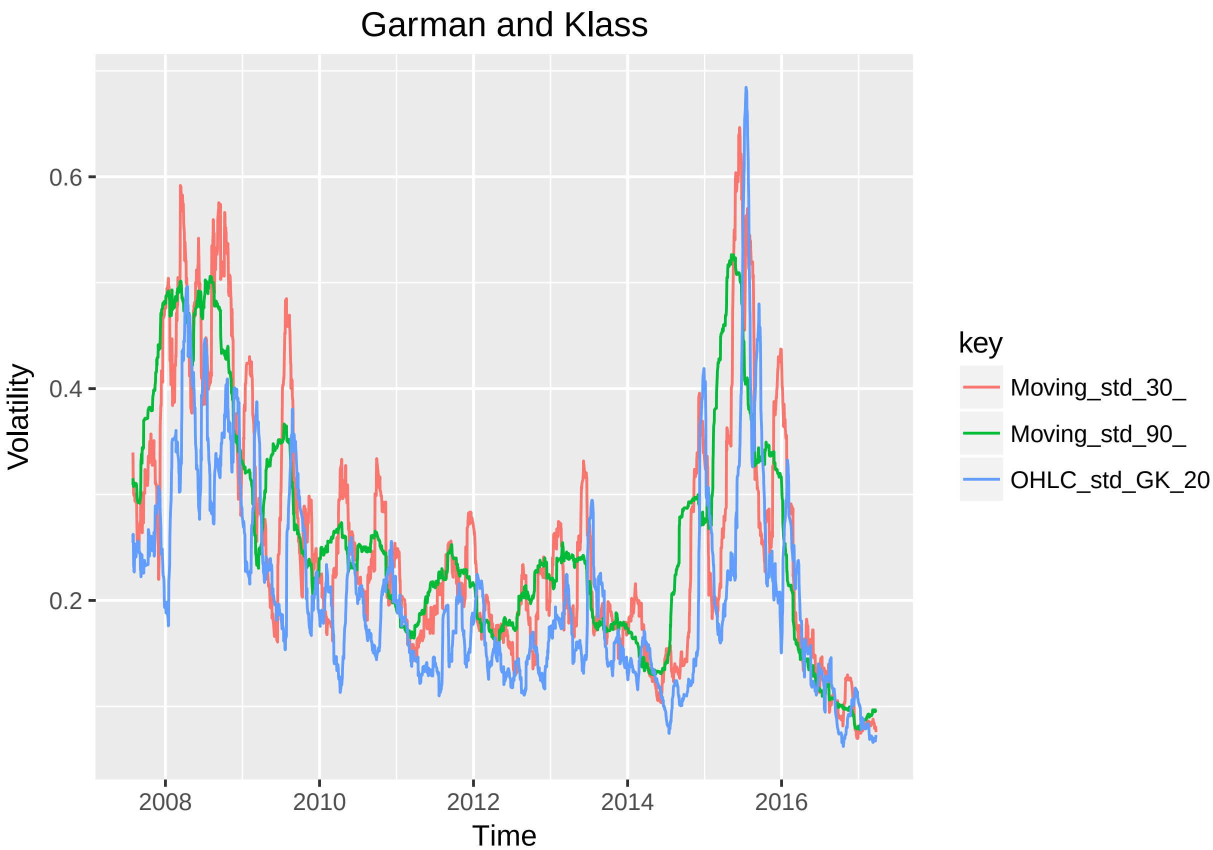
As we can tell from the graph the forecasted future volatility curve using past 20 days CSI 300 Index daily return solves the problem that curve being too smooth. It might be a great idea to put weights to get better estimated result.

3. OHLC Model

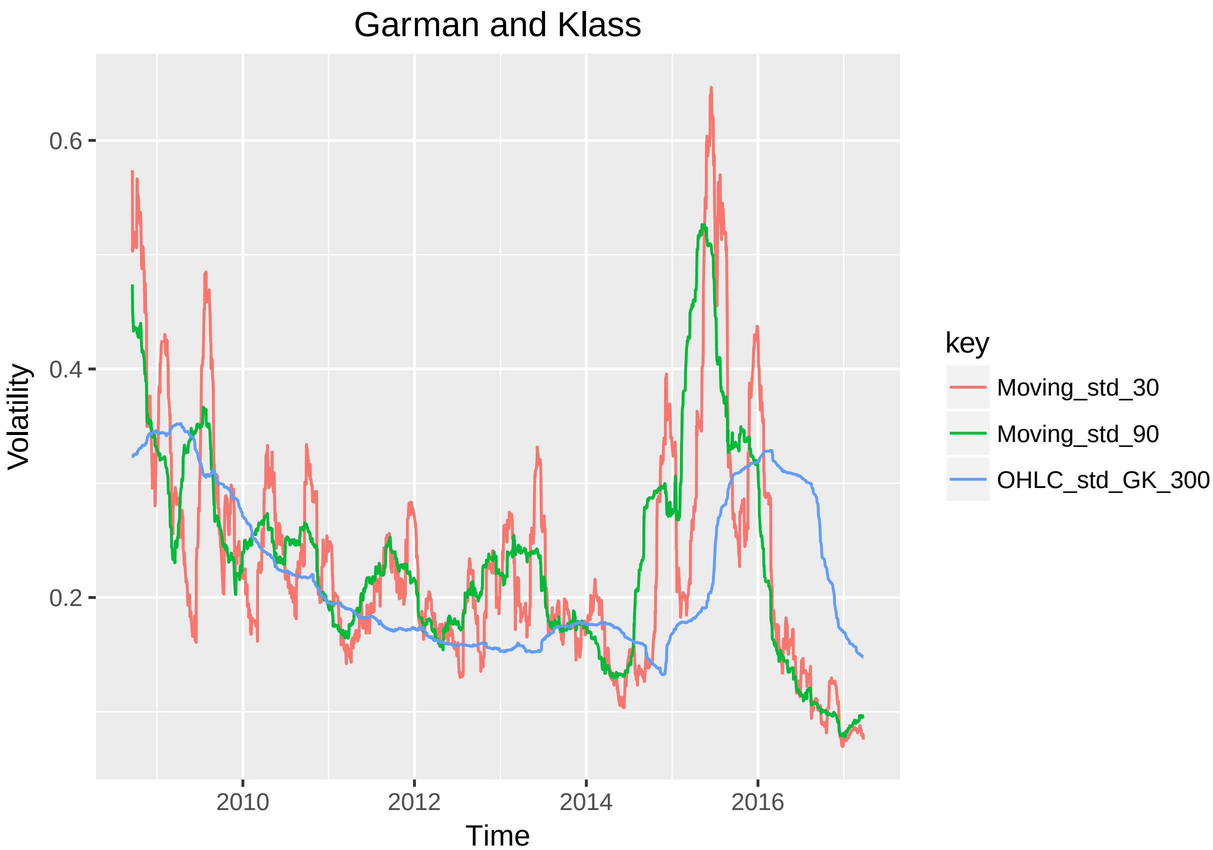
OHLC model, short for "Opening Price, High Price, Low Price, Closing Price", was developed to estimate market volatility.

OHLC Volatility: Garman and Klass (calc="garman.klass") The Garman and Klass estimator for estimating historical volatility assumes Brownian motion with zero drift and no opening jumps (i.e. the opening = close of the previous period). This estimator is 7.4 times more efficient than the close-to-close estimator.

```
# N = 20
e <- exp(1)
pt <- log(Index_Data$High / Index_Data$Low, base = e)
qt <- log(Index_Data$Close / Index_Data$Open, base = e)

OHLC_std <- sqrt(1/20 * runSum(0.5 * pt^2 - (2*log(2)-1) * qt^2, 20)) * sqrt(240)
OHLC_std_GK_20 <- as.matrix(OHLC_std[20:(20 + sta_width_- 1)])
Moving_OHLC_table_ <- data.frame(cbind(Moving_time_, OHLC_std_GK_20, Moving_std_30_, Moving_std_90_))
colnames(Moving_OHLC_table_)<- c("Time", "OHLC_std_GK_20", "Moving_std_30_", "Moving_std_90_")

Moving_OHLC_table_ %>%
  gather(key, Volatility, OHLC_std_GK_20, Moving_std_30_, Moving_std_90_) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line()+ ggtitle("Garman and Klass")
```

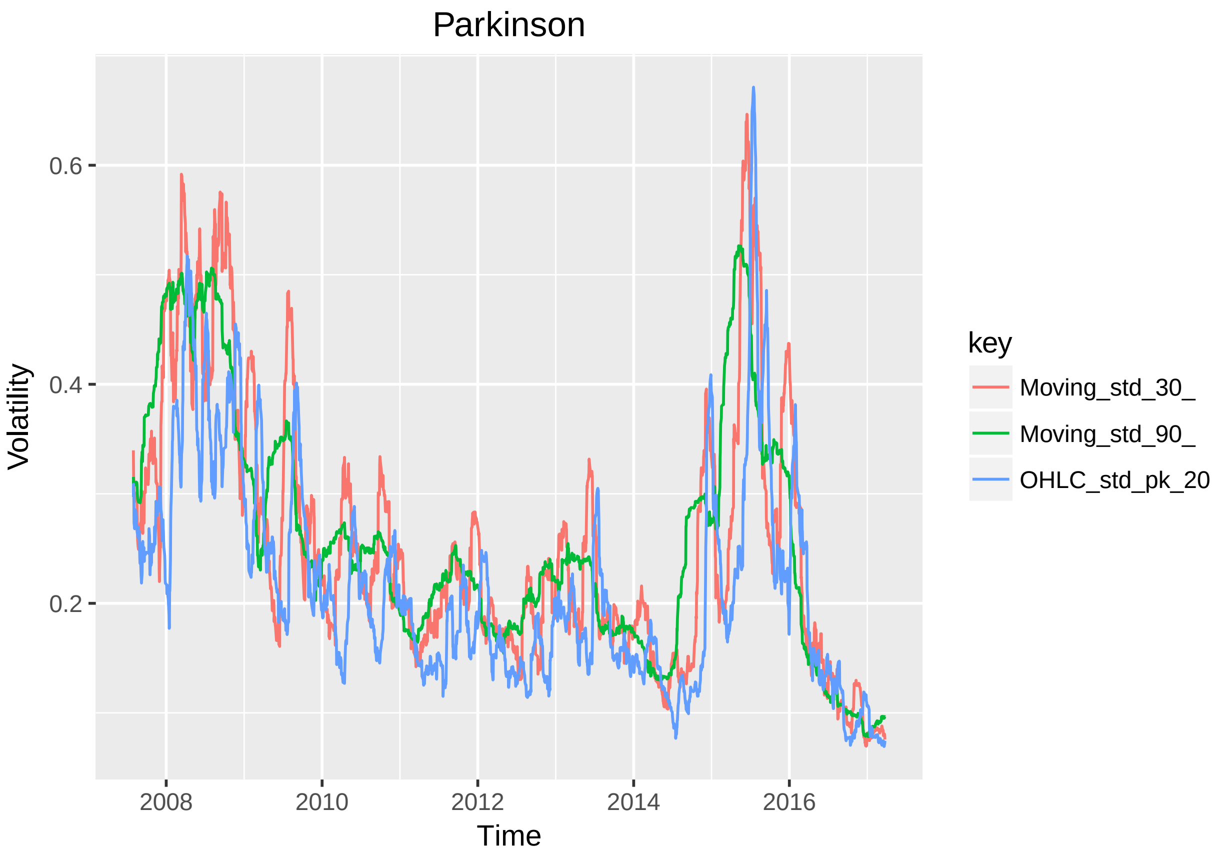



```r
# N = 300
OHLC_std <- sqrt(1/300 * runSum(0.5 * pt^2 - (2*log(2)-1) * qt^2, 300)) * sqrt(240)
OHLC_std_GK_300 <- as.matrix(OHLC_std[300:(300 + sta_width - 1)])
Moving_OHLC_table <- data.frame(cbind(Moving_time, OHLC_std_GK_300, Moving_std_30, Moving_std_90))
colnames(Moving_OHLC_table)<- c("Time", "OHLC_std_GK_300", "Moving_std_30", "Moving_std_90")

Moving_OHLC_table %>%
  gather(key, Volatility, OHLC_std_GK_300, Moving_std_30, Moving_std_90) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("Garman and Klass")
```



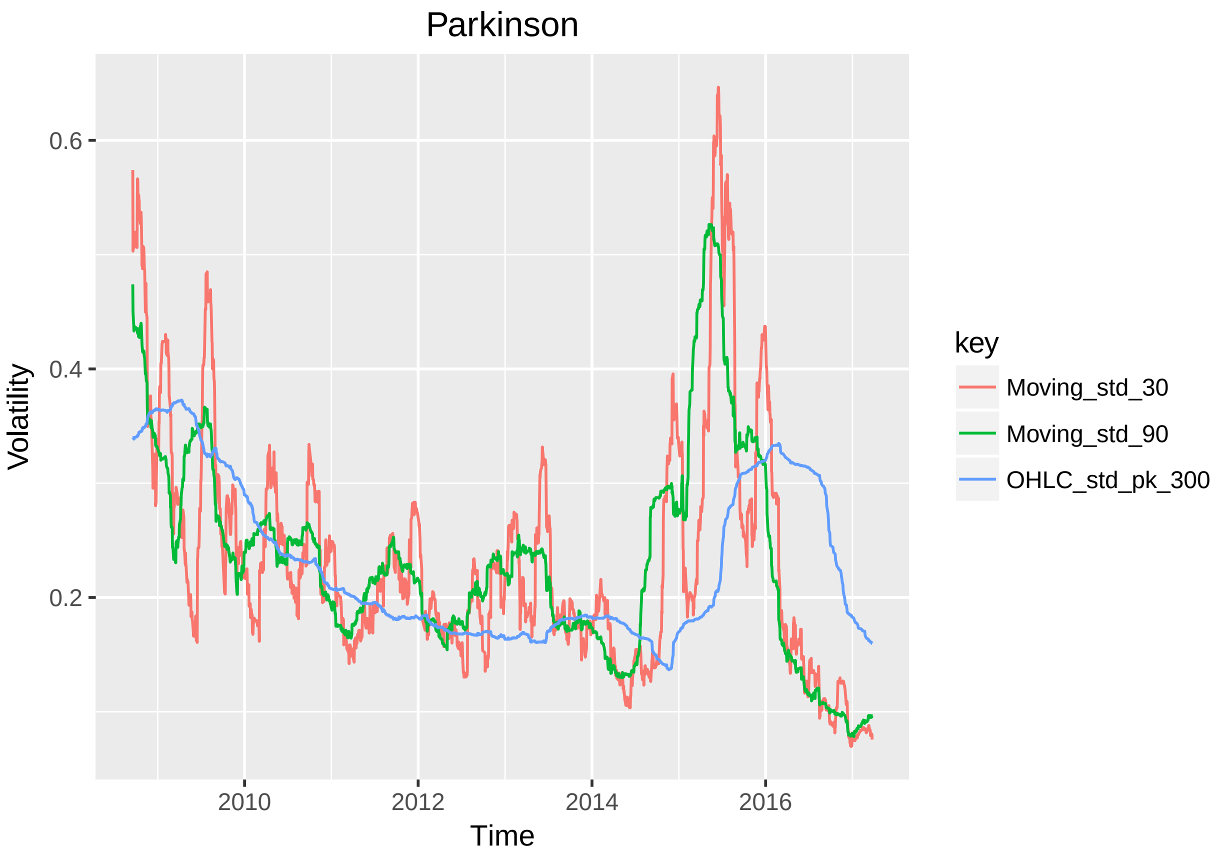
High-Low Volatility: Parkinson (calc="parkinson") The Parkinson formula for estimating the historical volatility of an underlying based on high and low prices.Empirically, as sample size amounts to over two hundred, the ratio of Parkinson Variance and True Variance approaches one.

```
# N = 20
OHLC_std <- sqrt(1/(4*20*log(2)) * runSum(pt^2, 20))* sqrt(240)
OHLC_std_pk_20<- as.matrix(OHLC_std[20:(20 + sta_width_- 1)])
Moving_OHLC_table_ <- data.frame(cbind(Moving_time_, OHLC_std_pk_20, Moving_std_30_, Moving_std_90_))
colnames(Moving_OHLC_table_)<- c("Time", "OHLC_std_pk_20", "Moving_std_30_", "Moving_std_90_")

Moving_OHLC_table_ %>%
  gather(key, Volatility, OHLC_std_pk_20, Moving_std_30_, Moving_std_90_) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("Parkinson")
```



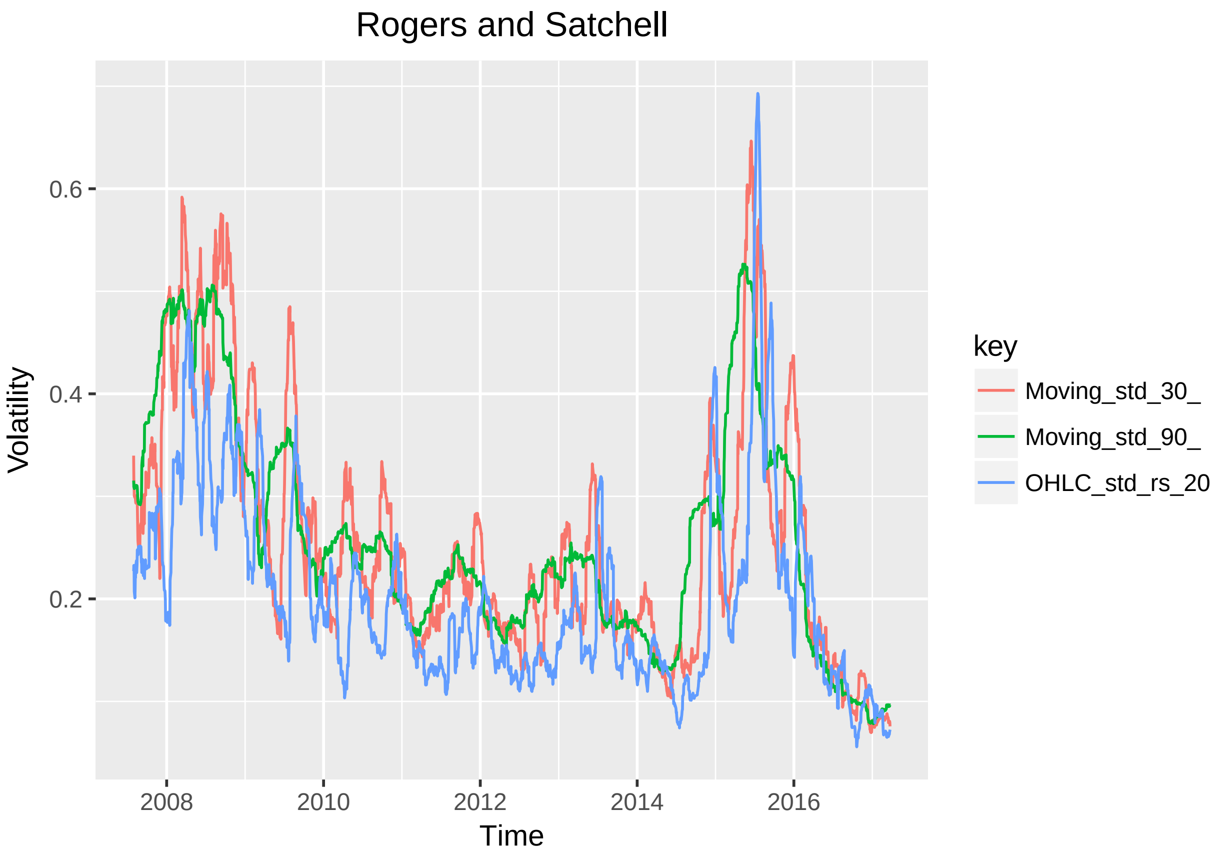
Now, try N = 300

```
OHLC_std <- sqrt(1/(4*300*log(2)) * runSum(pt^2, 300))* sqrt(240)
OHLC_std_pk_300 <- as.matrix(OHLC_std[300:(300 + sta_width - 1)])
Moving_OHLC_table <- data.frame(cbind(Moving_time, OHLC_std_pk_300, Moving_std_30, Moving_std_90))
colnames(Moving_OHLC_table)<- c("Time", "OHLC_std_pk_300", "Moving_std_30", "Moving_std_90")

Moving_OHLC_table %>%
  gather(key, Volatility, OHLC_std_pk_300, Moving_std_30, Moving_std_90) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("Parkinson")
```

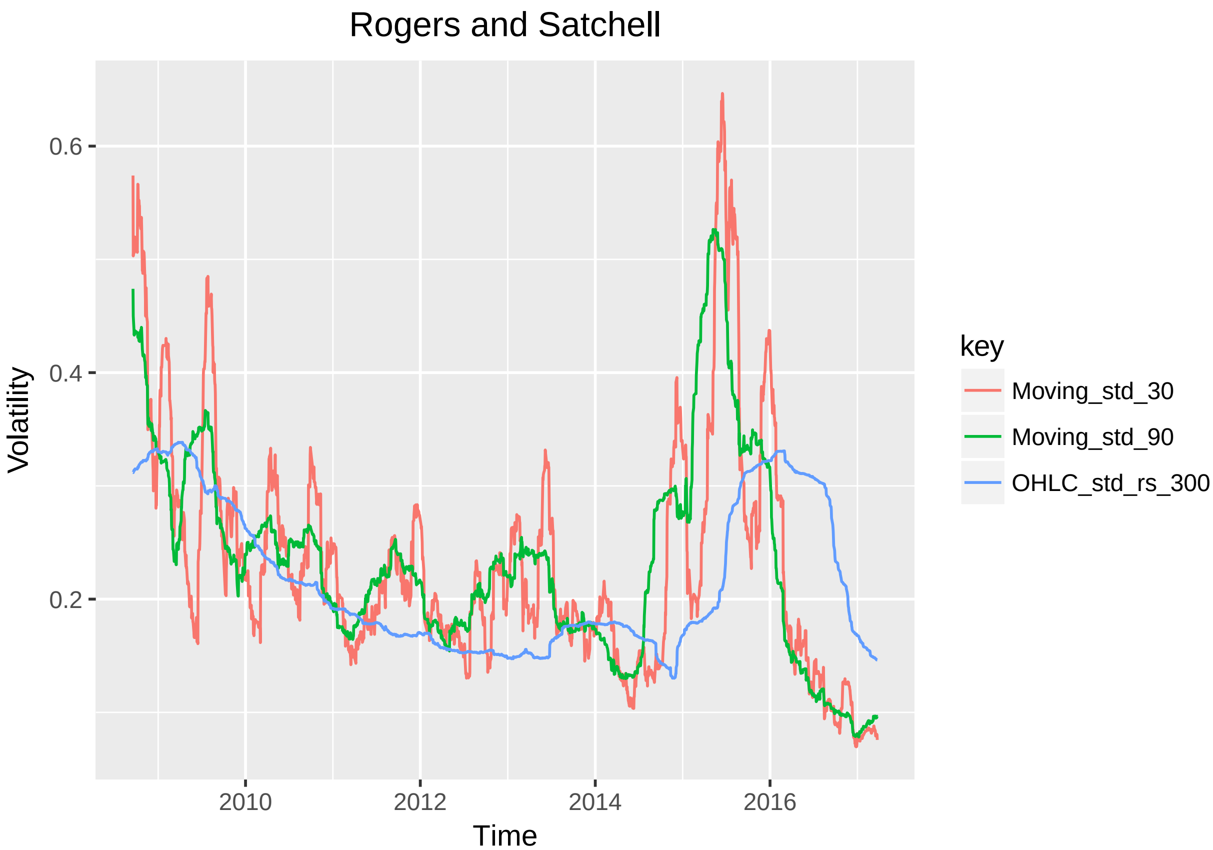


OHLC Volatility: Rogers and Satchell (calc="rogers.satchell") The Roger and Satchell historical volatility estimator allows for non-zero drift, but assumed no opening jump.

```
# N = 20
hc <- log(Index_Data$High / Index_Data$Close, base = e)
ho <- log(Index_Data$High / Index_Data$Open, base = e)
lc <- log(Index_Data$Low / Index_Data$Close, base = e)
lo <- log(Index_Data$Low / Index_Data$Open, base = e)
OHLC_std_rs_20 <- sqrt(1/20 * runSum(hc * ho + lc * lo, 20)) * sqrt(240)

OHLC_std_rs_20 <- as.matrix(OHLC_std_rs_20[20:(20 + sta_width_- 1)])
Moving_OHLC_table_ <- data.frame(cbind(Moving_time_, OHLC_std_rs_20, Moving_std_30_, Moving_std_90_))
colnames(Moving_OHLC_table_)<- c("Time", "OHLC_std_rs_20", "Moving_std_30_", "Moving_std_90_")

Moving_OHLC_table_ %>%
  gather(key, Volatility, OHLC_std_rs_20, Moving_std_30_, Moving_std_90_) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("Rogers and Satchell")
```

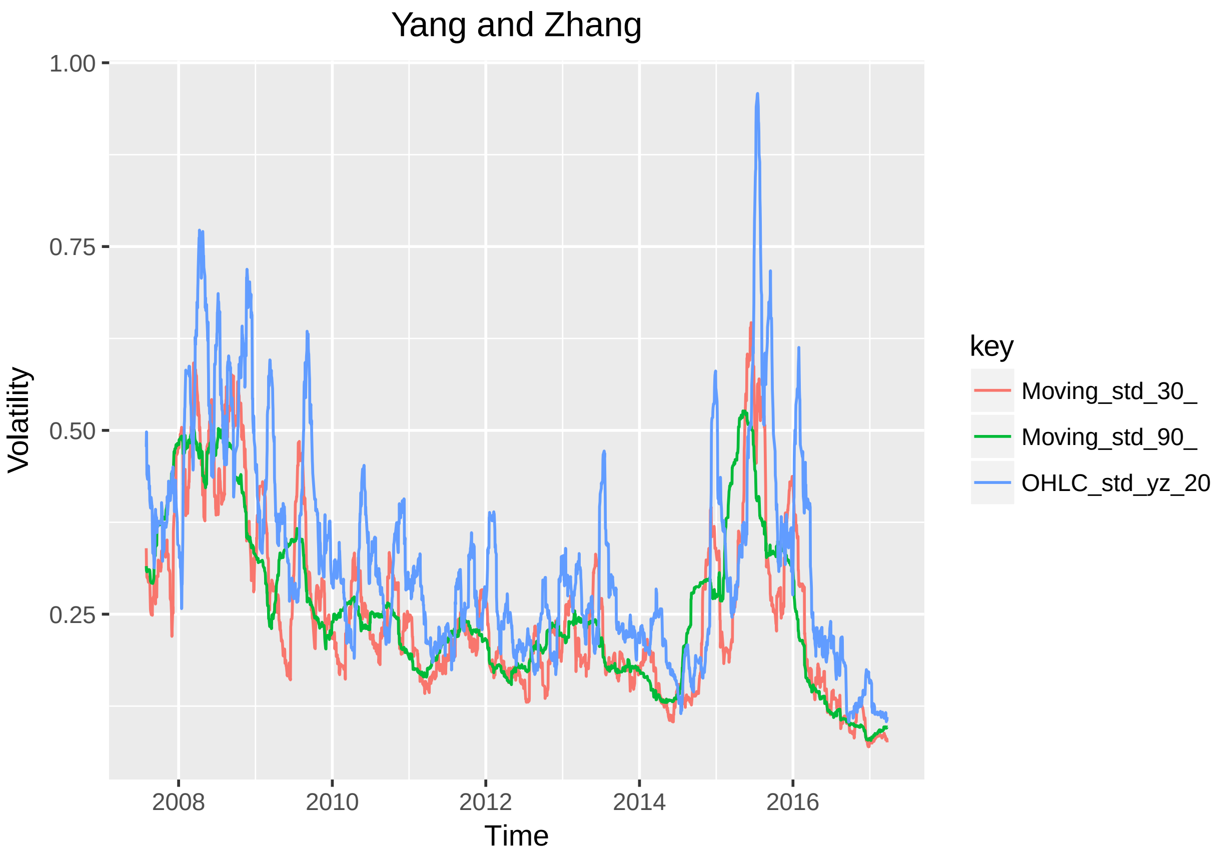



```r
# N = 300
OHLC_std_rs_300 <- sqrt(1/300 * runSum(hc * ho + lc * lo, 300)) * sqrt(240)

OHLC_std_rs_300 <- as.matrix(OHLC_std_rs_300[300:(300 + sta_width - 1)])
Moving_OHLC_table <- data.frame(cbind(Moving_time, OHLC_std_rs_300, Moving_std_30, Moving_std_90))
colnames(Moving_OHLC_table)<- c("Time", "OHLC_std_rs_300", "Moving_std_30", "Moving_std_90")

Moving_OHLC_table %>%
  gather(key, Volatility, OHLC_std_rs_300, Moving_std_30, Moving_std_90) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("Rogers and Satchell")
```

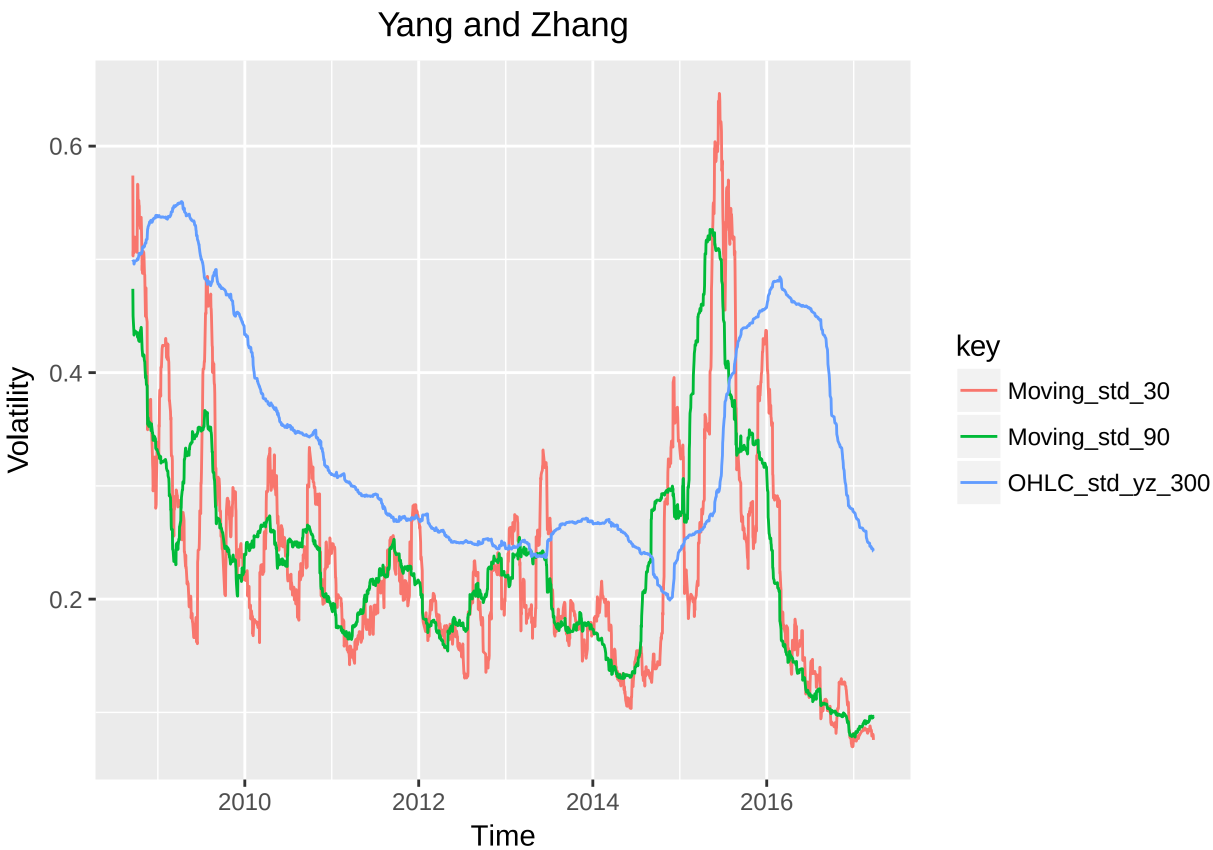


OHLC Volatility: Yang and Zhang (calc="yang.zhang") The Yang and Zhang historical volatility estimator has minimum estimation error, and is independent of drift and opening gaps. It can be interpreted as a weighted average of the Rogers and Satchell estimator, the close-open volatility, and the open-close volatility.

```
# N = 20
s2o <- 1/19 * runSum(log(Index_Data$Open/lag(Index_Data$Close,1))^2, n=20)
s2c <- 1/19 * runSum(log(Index_Data$Close/lag(Index_Data$Open,1))^2, n=20)
s2rs <- 1/19 * runSum(hc * ho + lc * lo, n=20)
k <- 0.34 / (2 * 20 / 19)
OHLC_std_yz_20 <- sqrt(s2o + k*s2c + (1-k)*s2rs) * sqrt(240)
OHLC_std_yz_20 <- as.matrix(OHLC_std_yz_20[20:(20 + sta_width_- 1)])

Moving_OHLC_table_ <- data.frame(cbind(Moving_time_, OHLC_std_yz_20, Moving_std_30_, Moving_std_90_))
colnames(Moving_OHLC_table_)<- c("Time", "OHLC_std_yz_20", "Moving_std_30_", "Moving_std_90_")

Moving_OHLC_table_ %>%
  gather(key, Volatility, OHLC_std_yz_20, Moving_std_30_, Moving_std_90_) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("Yang and Zhang")
```




```
# N = 300
s2o <- 1/299 * runSum(log(Index_Data$Open/lag(Index_Data$Close,1))^2, n=300)
s2c <- 1/299 * runSum(log(Index_Data$Close/lag(Index_Data$Open,1))^2, n=300)
s2rs <- 1/299 * runSum(hc * ho + lc * lo, n=300)
k <- 0.34 / (2 * 300 / 299)
OHLC_std_yz_300 <- sqrt(s2o + k*s2c + (1-k)*s2rs) * sqrt(240)
OHLC_std_yz_300 <- as.matrix(OHLC_std_yz_300[300:(300 + sta_width - 1)])

Moving_OHLC_table <- data.frame(cbind(Moving_time, OHLC_std_yz_300, Moving_std_30, Moving_std_90))
colnames(Moving_OHLC_table)<- c("Time", "OHLC_std_yz_300", "Moving_std_30", "Moving_std_90")

Moving_OHLC_table %>%
  gather(key, Volatility, OHLC_std_yz_300, Moving_std_30, Moving_std_90) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) +
  geom_line() + ggtitle("Yang and Zhang")
```

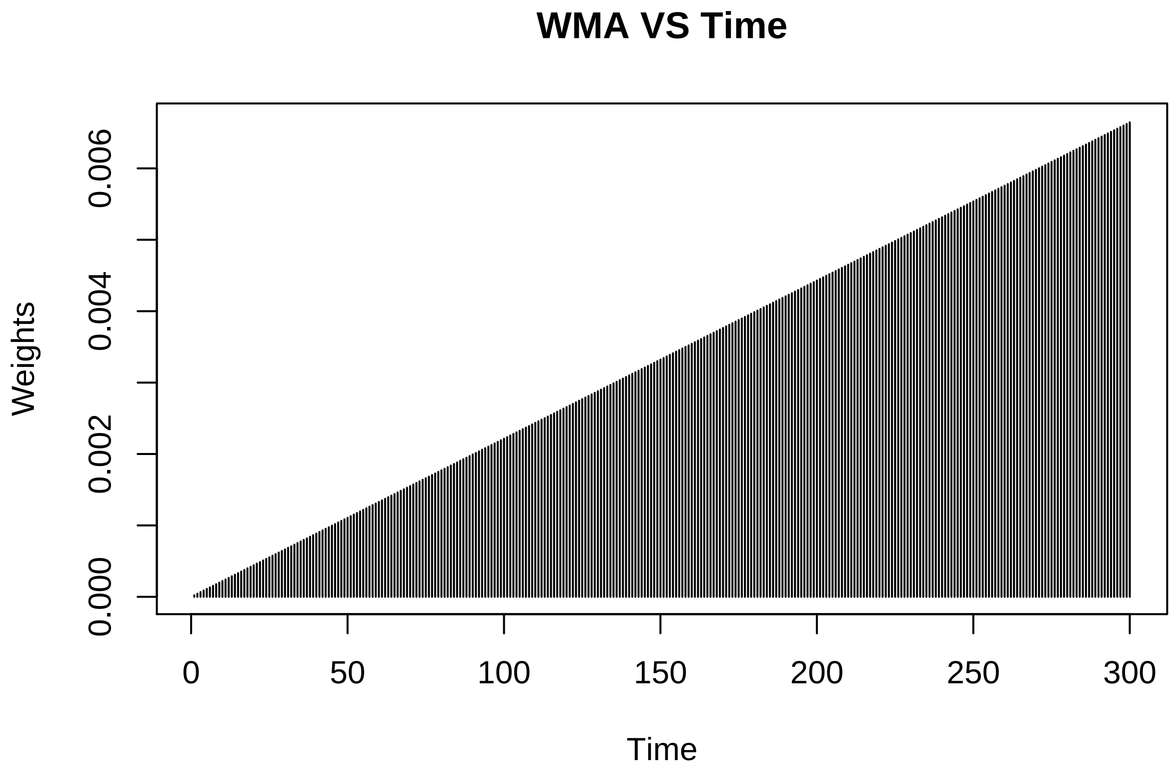


4. Weighted Moving Average (WMA, EWMA)

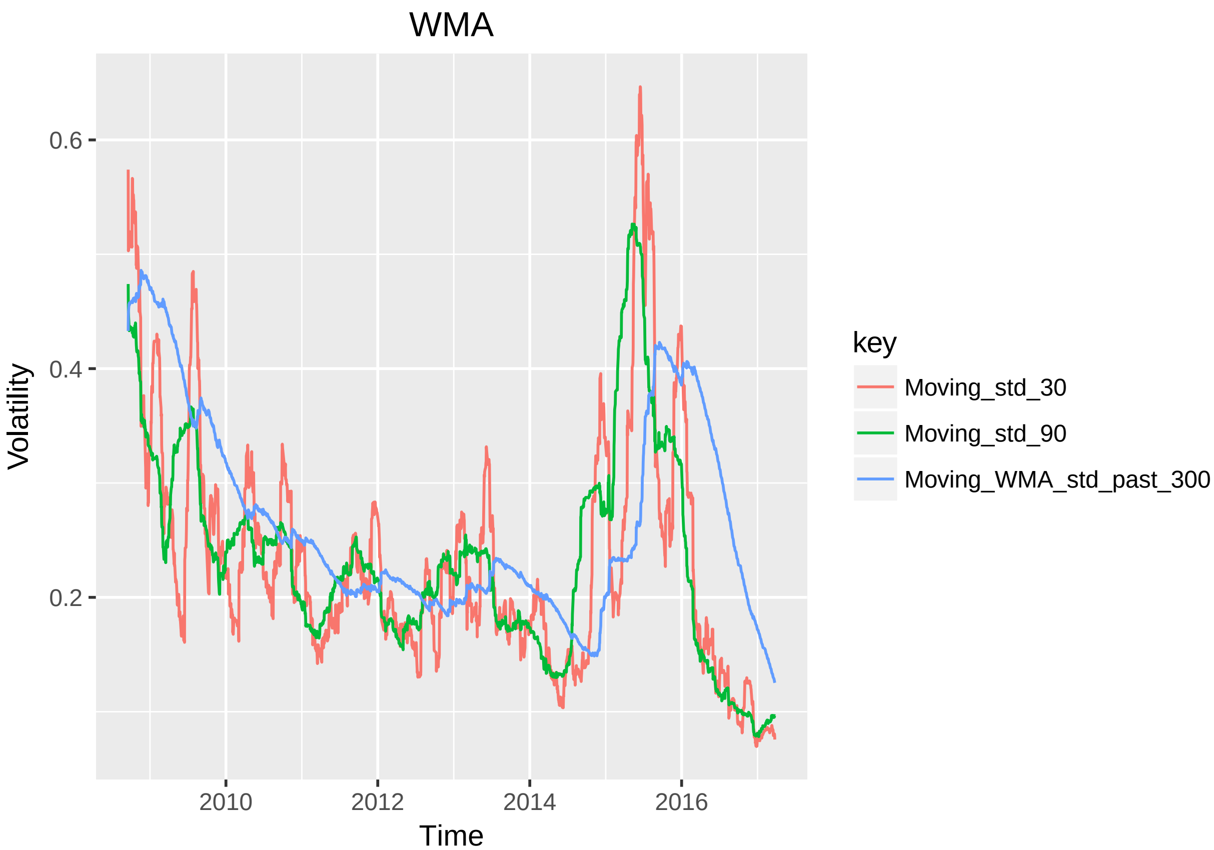
A weighted average is an average that has multiplying factors to give different weights to data at different positions in the sample window. In technical analysis of financial data, a weighted moving average (WMA) has the specific meaning of weights that decrease in arithmetical progression.In an n-day WMA the latest day has weight n, the second latest n − 1, etc., down to one.

N = 300

```
# WMA
wt_WMA_300 <- c(1:300)

# plot WMA
plot(wt_WMA_300 / sum(wt_WMA_300), xlab = "Time", ylab = "Weights", main = "WMA VS Time", type = "h")
```

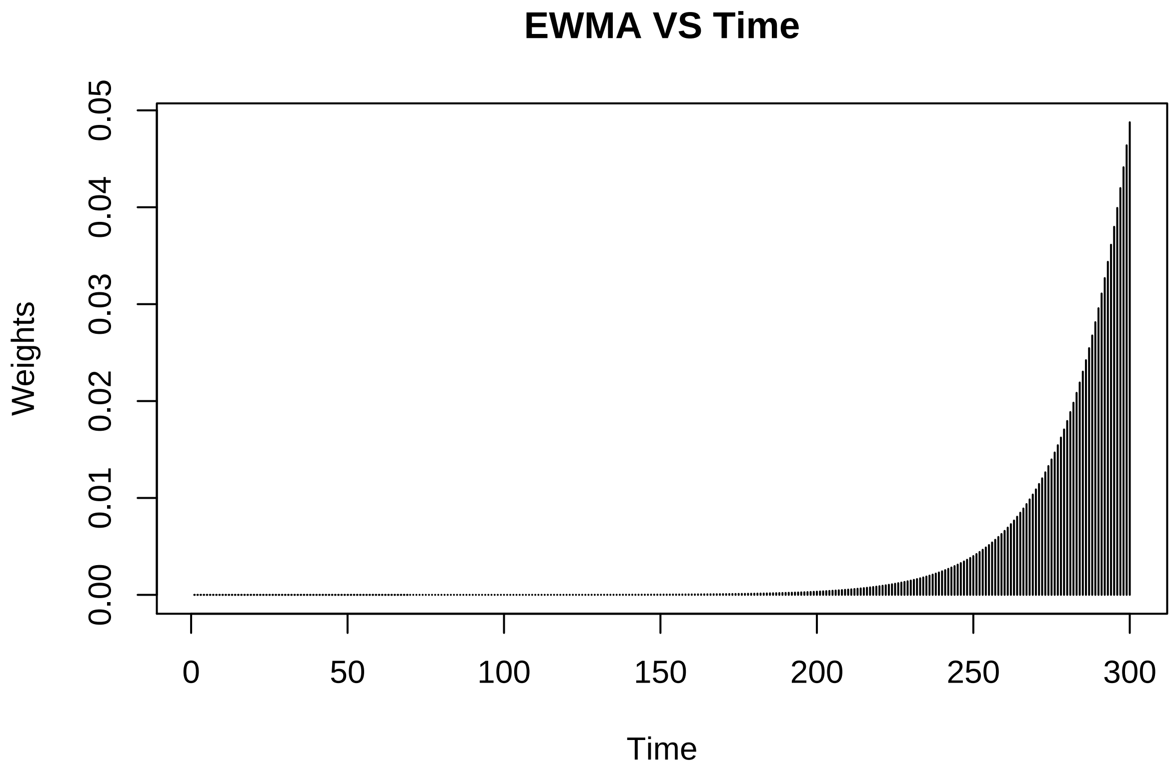



```
Moving_WMA_std_past_300 <- roll_sd(Data, width = 300, weights = wt_WMA_300) %>% na.omit() * sqrt(240)
Moving_WMA_std_past_300 <- as.matrix(Moving_WMA_std_past_300[1:sta_width])

Moving_WMA_std_table_300 <- data.frame(cbind(Moving_time, Moving_WMA_std_past_300, Moving_std_30, Movin
colnames(Moving_WMA_std_table_300)<- c("Time", "Moving_WMA_std_past_300", "Moving_std_30", "Moving_std_

Moving_WMA_std_table_300 %>%
  gather(key, Volatility, Moving_WMA_std_past_300, Moving_std_30, Moving_std_90) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) + geom_line()+ ggtitle("WMA")
```

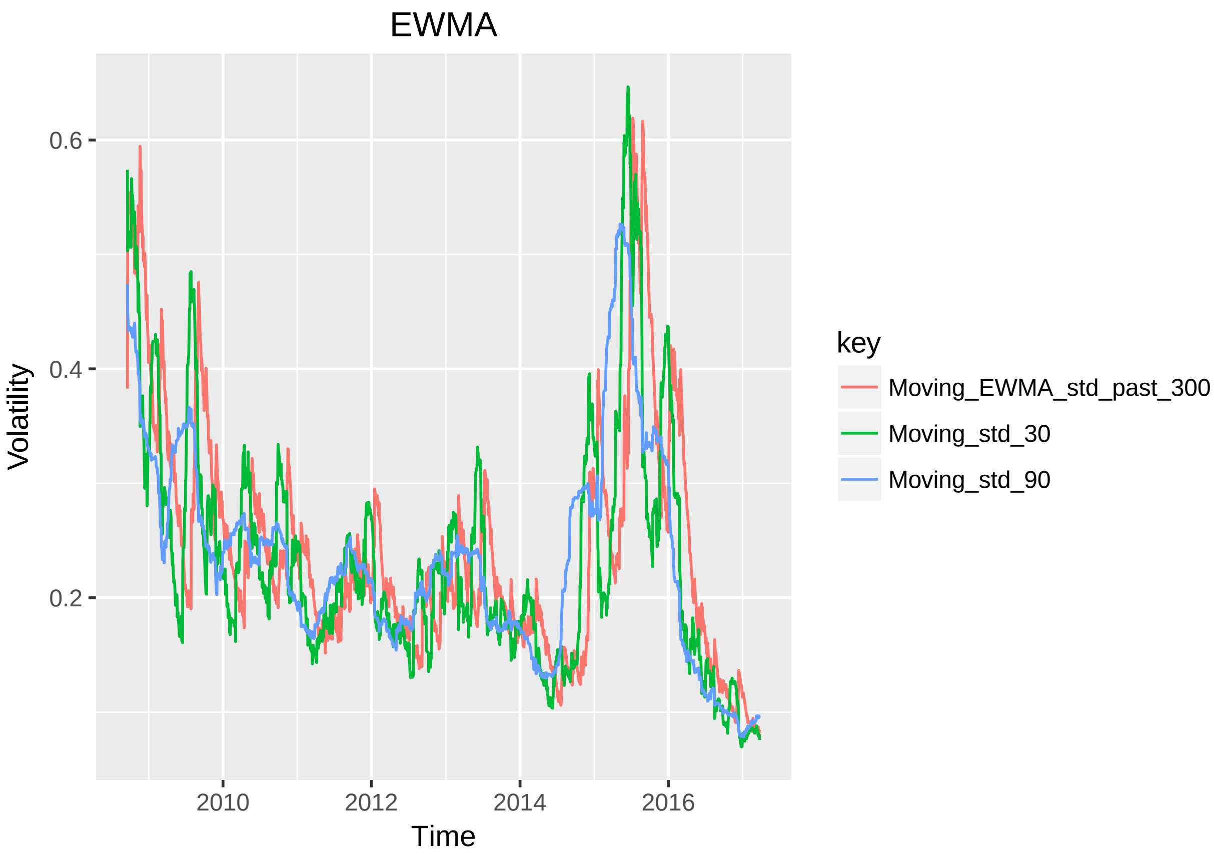


An exponential moving average (EMA), also known as an exponentially weighted moving average (EWMA), is a type of infinite impulse response filter that applies weighting factors which decrease exponentially. The weighting for each older datum decreases exponentially, never reaching zero.

```
# EWMA
lamda_300 <- -0.05    # When n becomes more nagetive, volatility gets larger
wt_EWMA_300 <- sort(e^(lamda_300 * wt_WMA_300), decreasing = FALSE)

# Plot EWMA
plot(wt_EWMA_300 / sum(wt_EWMA_300), xlab = "Time", ylab = "Weights", main = "EWMA VS Time", type = "h")
```

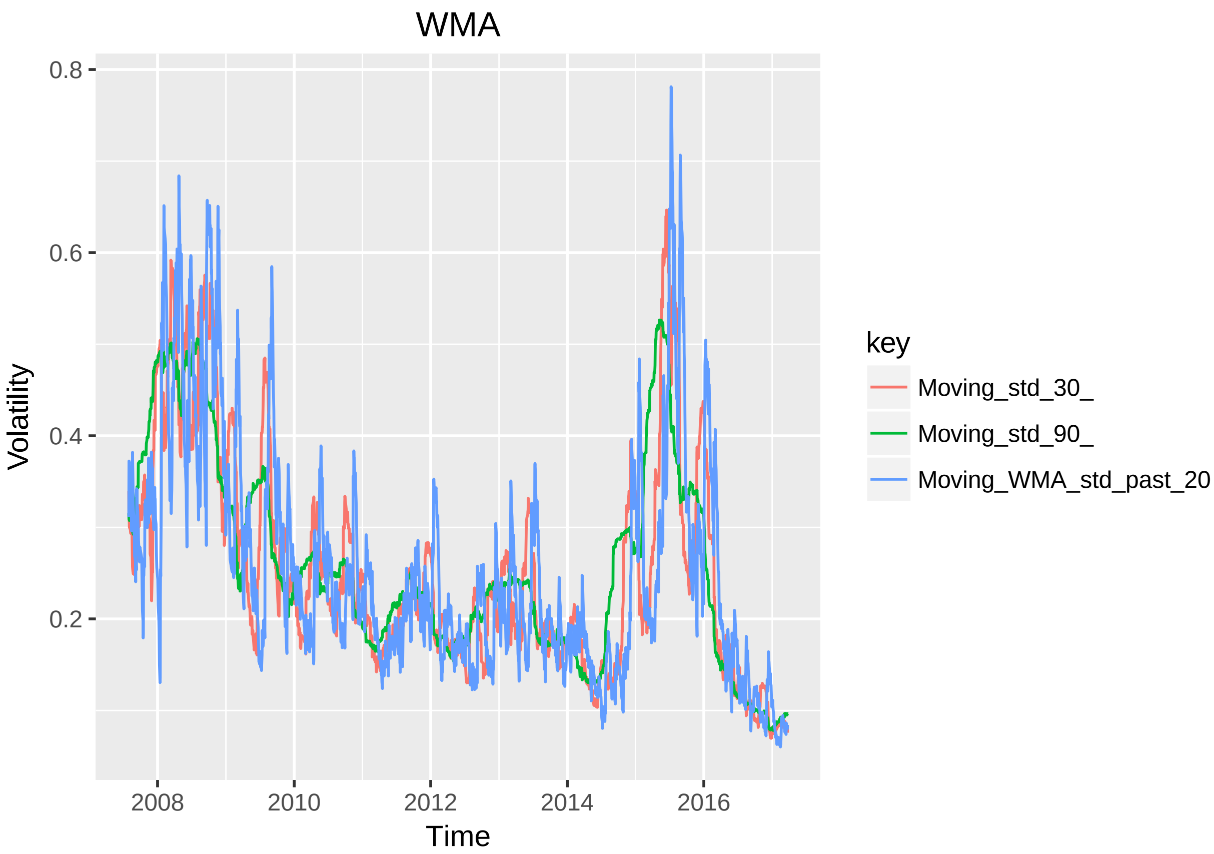



```
Moving_EWMA_std_past_300 <- roll_sd(Data, width = 300, weights = wt_EWMA_300) %>% na.omit() * sqrt(240)
Moving_EWMA_std_past_300 <- as.matrix(Moving_EWMA_std_past_300[1:sta_width])
Moving_EWMA_std_table_300 <- data.frame(cbind(Moving_time, Moving_EWMA_std_past_300, Moving_std_30, Movi
colnames(Moving_EWMA_std_table_300)<- c("Time", "Moving_EWMA_std_past_300", "Moving_std_30", "Moving_std

Moving_EWMA_std_table_300 %>%
  gather(key, Volatility, Moving_EWMA_std_past_300, Moving_std_30, Moving_std_90) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) + geom_line() + ggtitle("EWMA")
```

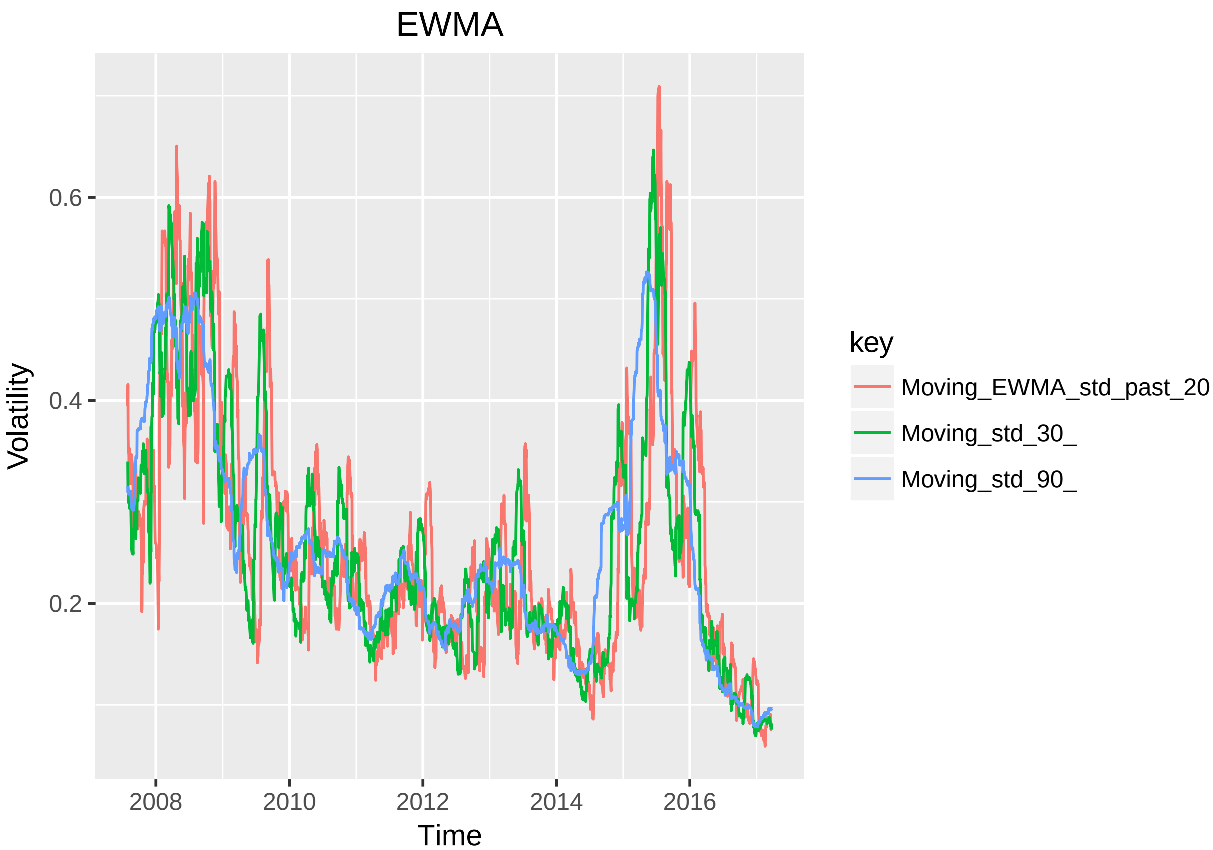


N = 20

```
# WMA
wt_WMA_20 <- c(1:20)
Moving_WMA_std_past_20 <- roll_sd(Data, width = 20, weights = wt_WMA_20) %>% na.omit() * sqrt(240)

Moving_WMA_std_past_20 <- as.matrix(Moving_WMA_std_past_20[1:sta_width_])

Moving_WMA_std_table_20 <- data.frame(cbind(Moving_time_, Moving_WMA_std_past_20, Moving_std_30_, Movin
colnames(Moving_WMA_std_table_20)<- c("Time", "Moving_WMA_std_past_20", "Moving_std_30_", "Moving_std_9

Moving_WMA_std_table_20 %>%
  gather(key, Volatility, Moving_WMA_std_past_20, Moving_std_30_, Moving_std_90_) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) + geom_line() + ggtitle("WMA")
```




```
# EWMA
lamda_20 <- -0.01  # When n becomes more nagetive, volatility gets larger
wt_EWMA_20 <- sort(e^(lamda_20 * wt_WMA_20), decreasing = FALSE)
Moving_EWMA_std_past_20 <- roll_sd(Data, width = 20, weights = wt_EWMA_20) %>% na.omit() * sqrt(240)

Moving_EWMA_std_past_20 <- as.matrix(Moving_EWMA_std_past_20[1:sta_width_])
Moving_EWMA_std_table_20 <- data.frame(cbind(Moving_time_, Moving_EWMA_std_past_20, Moving_std_30_, Movi
colnames(Moving_EWMA_std_table_20)<- c("Time", "Moving_EWMA_std_past_20", "Moving_std_30_", "Moving_std
Moving_EWMA_std_table_20 %>%
  gather(key, Volatility, Moving_EWMA_std_past_20, Moving_std_30_, Moving_std_90_) %>%
  ggplot(aes(x = Time, y = Volatility, colour = key)) + geom_line() + ggtitle("EWMA")
```



5. Perfomance Analytics

In this section an error index is introduced to measure perfomance of each model. The financial market is brokedown into three parts based on market trends: bull, bear, and flat market, and each of which is analyzed repectively.

```
perf<- function(sr1, sr2, num){
  if (length(sr1) == length(sr2)){
    if (num == 1){
      return(100 * mean(abs(sr1 - sr2)))
    } else if (num == 2){
      return(100 * sd(abs(sr1 - sr2)))
    } else if (num == 3){
      return(100 * mean(sr1 - sr2))
    } else {
      return(100 * sd(sr1 - sr2))
    }
  }
  else{
    warning("Two time-series seqeunces have different length")
  }
}

addnames <- function(fra){
  rownames(fra) <- c("Past 20 days", "Past 300 days")
  colnames(fra) <- c("SMA", "WMA","EWMA","Garman and Klass","Parkinson","Rogers and Satchell","Yang and
```

```r
  return(fra)
}

compareMarket <- function(fra){
    rownames(fra) <- c("Bull Market", "Bear Market", "Flat Market", "Overall")
  colnames(fra) <- c("SMA", "WMA","EWMA","Garman and Klass","Parkinson","Rogers and Satchell","Yang and
  return(fra)
}
```

Future 30 days

```r
# N = 20
Vec_30_past20_1 <- round(c(perf(Moving_std_past_20, Moving_std_30_, 1),
                    perf(Moving_WMA_std_past_20, Moving_std_30_, 1),
                    perf(Moving_EWMA_std_past_20,Moving_std_30_, 1),
                    perf(OHLC_std_GK_20,Moving_std_30_, 1),
                    perf(OHLC_std_pk_20,Moving_std_30_, 1),
                    perf(OHLC_std_rs_20,Moving_std_30_, 1),
                    perf(OHLC_std_yz_20,Moving_std_30_, 1)),digits = 2)

Vec_30_past20_2 <- round(c(perf(Moving_std_past_20, Moving_std_30_, 2),
                    perf(Moving_WMA_std_past_20, Moving_std_30_, 2),
                    perf(Moving_EWMA_std_past_20,Moving_std_30_, 2),
                    perf(OHLC_std_GK_20,Moving_std_30_, 2),
                    perf(OHLC_std_pk_20,Moving_std_30_, 2),
                    perf(OHLC_std_rs_20,Moving_std_30_, 2),
                    perf(OHLC_std_yz_20,Moving_std_30_, 2)),digits = 2)

Vec_30_past20_3 <- round(c(perf(Moving_std_past_20, Moving_std_30_, 3),
                    perf(Moving_WMA_std_past_20, Moving_std_30_, 3),
                    perf(Moving_EWMA_std_past_20,Moving_std_30_, 3),
                    perf(OHLC_std_GK_20,Moving_std_30_, 3),
                    perf(OHLC_std_pk_20,Moving_std_30_, 3),
                    perf(OHLC_std_rs_20,Moving_std_30_, 3),
                    perf(OHLC_std_yz_20,Moving_std_30_, 3)),digits = 2)

Vec_30_past20_4 <- round(c(perf(Moving_std_past_20, Moving_std_30_, 4),
                    perf(Moving_WMA_std_past_20, Moving_std_30_, 4),
                    perf(Moving_EWMA_std_past_20,Moving_std_30_, 4),
                    perf(OHLC_std_GK_20,Moving_std_30_, 4),
                    perf(OHLC_std_pk_20,Moving_std_30_, 4),
                    perf(OHLC_std_rs_20,Moving_std_30_, 4),
                    perf(OHLC_std_yz_20,Moving_std_30_, 4)),digits = 2)

# N = 300
Vec_30_past300_1 <- round(c(perf(Moving_std_past_300,Moving_std_30, 1),
                    perf(Moving_WMA_std_past_300,Moving_std_30, 1),
                    perf(Moving_EWMA_std_past_300,Moving_std_30, 1),
                    perf(OHLC_std_GK_300,Moving_std_30, 1),
                    perf(OHLC_std_pk_300,Moving_std_30, 1),
                    perf(OHLC_std_rs_300,Moving_std_30, 1),
                    perf(OHLC_std_yz_300,Moving_std_30, 1)), digits = 2)

Vec_30_past300_2 <- round(c(perf(Moving_std_past_300,Moving_std_30, 2),
```

```r
                    perf(Moving_WMA_std_past_300,Moving_std_30, 2),
                    perf(Moving_EWMA_std_past_300,Moving_std_30, 2),
                    perf(OHLC_std_GK_300,Moving_std_30, 2),
                    perf(OHLC_std_pk_300,Moving_std_30, 2),
                    perf(OHLC_std_rs_300,Moving_std_30, 2),
                    perf(OHLC_std_yz_300,Moving_std_30, 2)), digits = 2)

Vec_30_past300_3 <- round(c(perf(Moving_std_past_300,Moving_std_30, 3),
                    perf(Moving_WMA_std_past_300,Moving_std_30, 3),
                    perf(Moving_EWMA_std_past_300,Moving_std_30, 3),
                    perf(OHLC_std_GK_300,Moving_std_30, 3),
                    perf(OHLC_std_pk_300,Moving_std_30, 3),
                    perf(OHLC_std_rs_300,Moving_std_30, 3),
                    perf(OHLC_std_yz_300,Moving_std_30, 3)), digits = 2)

Vec_30_past300_4 <- round(c(perf(Moving_std_past_300,Moving_std_30, 4),
                    perf(Moving_WMA_std_past_300,Moving_std_30, 4),
                    perf(Moving_EWMA_std_past_300,Moving_std_30, 4),
                    perf(OHLC_std_GK_300,Moving_std_30, 4),
                    perf(OHLC_std_pk_300,Moving_std_30, 4),
                    perf(OHLC_std_rs_300,Moving_std_30, 4),
                    perf(OHLC_std_yz_300,Moving_std_30, 4)), digits = 2)


Vec_30_1 <- data.frame(rbind(Vec_30_past20_1,Vec_30_past300_1))
Vec_30_2 <- data.frame(rbind(Vec_30_past20_2,Vec_30_past300_2))
Vec_30_3 <- data.frame(rbind(Vec_30_past20_3,Vec_30_past300_3))
Vec_30_4 <- data.frame(rbind(Vec_30_past20_4,Vec_30_past300_4))

Vec_30_1 <- addnames(Vec_30_1)
Vec_30_2 <- addnames(Vec_30_2)
Vec_30_3 <- addnames(Vec_30_3)
Vec_30_4 <- addnames(Vec_30_4)

View(Vec_30_1)
View(Vec_30_3)
```

Future 90 days

```r
# N = 20
Vec_90_past20_1 <- round(c(perf(Moving_std_past_20, Moving_std_90_, 1),
                    perf(Moving_WMA_std_past_20, Moving_std_90_, 1),
                    perf(Moving_EWMA_std_past_20,Moving_std_90_, 1),
                    perf(OHLC_std_GK_20,Moving_std_90_, 1),
                    perf(OHLC_std_pk_20,Moving_std_90_, 1),
                    perf(OHLC_std_rs_20,Moving_std_90_, 1),
                    perf(OHLC_std_yz_20,Moving_std_90_, 1)), digits = 2)

Vec_90_past20_2 <- round(c(perf(Moving_std_past_20, Moving_std_90_, 2),
                    perf(Moving_WMA_std_past_20, Moving_std_90_, 2),
                    perf(Moving_EWMA_std_past_20,Moving_std_90_, 2),
                    perf(OHLC_std_GK_20,Moving_std_90_, 2),
                    perf(OHLC_std_pk_20,Moving_std_90_, 2),
                    perf(OHLC_std_rs_20,Moving_std_90_, 2),
```

```r
                   perf(OHLC_std_yz_20,Moving_std_90_, 2)), digits = 2)

Vec_90_past20_3 <- round(c(perf(Moving_std_past_20, Moving_std_90_, 3),
                   perf(Moving_WMA_std_past_20, Moving_std_90_, 3),
                   perf(Moving_EWMA_std_past_20,Moving_std_90_, 3),
                   perf(OHLC_std_GK_20,Moving_std_90_, 3),
                   perf(OHLC_std_pk_20,Moving_std_90_, 3),
                   perf(OHLC_std_rs_20,Moving_std_90_, 3),
                   perf(OHLC_std_yz_20,Moving_std_90_, 3)), digits = 2)

Vec_90_past20_4 <- round(c(perf(Moving_std_past_20, Moving_std_90_, 4),
                   perf(Moving_WMA_std_past_20, Moving_std_90_, 4),
                   perf(Moving_EWMA_std_past_20,Moving_std_90_, 4),
                   perf(OHLC_std_GK_20,Moving_std_90_, 4),
                   perf(OHLC_std_pk_20,Moving_std_90_, 4),
                   perf(OHLC_std_rs_20,Moving_std_90_, 4),
                   perf(OHLC_std_yz_20,Moving_std_90_, 4)), digits = 2)

# N = 300
Vec_90_past300_1 <- round(c(perf(Moving_std_past_300,Moving_std_90, 1),
                   perf(Moving_WMA_std_past_300,Moving_std_90, 1),
                   perf(Moving_EWMA_std_past_300,Moving_std_90, 1),
                   perf(OHLC_std_GK_300,Moving_std_90, 1),
                   perf(OHLC_std_pk_300,Moving_std_90, 1),
                   perf(OHLC_std_rs_300,Moving_std_90, 1),
                   perf(OHLC_std_yz_300,Moving_std_90, 1)), digits = 2)

Vec_90_past300_2 <- round(c(perf(Moving_std_past_300,Moving_std_90, 2),
                   perf(Moving_WMA_std_past_300,Moving_std_90, 2),
                   perf(Moving_EWMA_std_past_300,Moving_std_90, 2),
                   perf(OHLC_std_GK_300,Moving_std_90, 2),
                   perf(OHLC_std_pk_300,Moving_std_90, 2),
                   perf(OHLC_std_rs_300,Moving_std_90, 2),
                   perf(OHLC_std_yz_300,Moving_std_90, 2)), digits = 2)

Vec_90_past300_3 <- round(c(perf(Moving_std_past_300,Moving_std_90, 3),
                   perf(Moving_WMA_std_past_300,Moving_std_90, 3),
                   perf(Moving_EWMA_std_past_300,Moving_std_90, 3),
                   perf(OHLC_std_GK_300,Moving_std_90, 3),
                   perf(OHLC_std_pk_300,Moving_std_90, 3),
                   perf(OHLC_std_rs_300,Moving_std_90, 3),
                   perf(OHLC_std_yz_300,Moving_std_90, 3)), digits = 2)

Vec_90_past300_4 <- round(c(perf(Moving_std_past_300,Moving_std_90, 4),
                   perf(Moving_WMA_std_past_300,Moving_std_90, 4),
                   perf(Moving_EWMA_std_past_300,Moving_std_90, 4),
                   perf(OHLC_std_GK_300,Moving_std_90, 4),
                   perf(OHLC_std_pk_300,Moving_std_90, 4),
                   perf(OHLC_std_rs_300,Moving_std_90, 4),
                   perf(OHLC_std_yz_300,Moving_std_90, 4)), digits = 2)

Vec_90_1 <- data.frame(rbind(Vec_90_past20_1,Vec_90_past300_1))
Vec_90_2 <- data.frame(rbind(Vec_90_past20_2,Vec_90_past300_2))
```

```r
Vec_90_3 <- data.frame(rbind(Vec_90_past20_3,Vec_90_past300_3))
Vec_90_4 <- data.frame(rbind(Vec_90_past20_4,Vec_90_past300_4))

Vec_90_1 <- addnames(Vec_90_1)
Vec_90_2 <- addnames(Vec_90_2)
Vec_90_3 <- addnames(Vec_90_3)
Vec_90_4 <- addnames(Vec_90_4)

print(Vec_90_1)
```

```
##                 SMA WMA EWMA Garman and Klass Parkinson Rogers and Satchell
## Past 20 days   6.96 7.2 6.96             7.42      7.04                7.74
## Past 300 days  8.52 7.5 6.27             6.71      6.73                6.81
##               Yang and Zhang
## Past 20 days            9.51
## Past 300 days          12.89
```

```r
print(Vec_90_3)
```

```
##                  SMA   WMA  EWMA Garman and Klass Parkinson
## Past 20 days   -0.21 -0.39 -0.22            -5.21     -4.25
## Past 300 days   3.97  3.16  1.01            -1.49     -0.51
##               Rogers and Satchell Yang and Zhang
## Past 20 days                -5.56           6.78
## Past 300 days               -1.80          10.35
```

EWMA seems to be the most stable model among all possible models that forecast market volatility. Next step I would like to see whether the result remains unchanged if the financial market is brokendown into three parts based on market trends(Bull, Bear, and Falt Market), and under each of which an analysis is given.

```r
# Bull Market (2014/11/20 - 2015/6/9)
BullM_20 <- function(fk){
  return(fk[1754:1911])
}

# Bear Market (2015/6/9 - 2015/8/26)
BearM_20 <- function(fk){
  return(fk[1911:1966])
}

# Flat Market (2014/4/28 - 2014/7/17)
FlatM_20 <- function(fk){
  return(fk[1638:1693])
}
```

Since the trading period of each market trend spans only a few months, it is more reasonable to take N = 20 to conduct analysis instead of N = 300.

Future 30 days

```r
# N = 20, Bull Market
Vec_30_past20_Bull_1 <- round(c(perf(BullM_20(Moving_std_past_20), BullM_20(Moving_std_30_), 1),
                   perf(BullM_20(Moving_WMA_std_past_20), BullM_20(Moving_std_30_), 1),
                   perf(BullM_20(Moving_EWMA_std_past_20),BullM_20(Moving_std_30_), 1),
                   perf(BullM_20(OHLC_std_GK_20),BullM_20(Moving_std_30_), 1),
                   perf(BullM_20(OHLC_std_pk_20),BullM_20(Moving_std_30_), 1),
```

```r
                  perf(BullM_20(OHLC_std_rs_20),BullM_20(Moving_std_30_), 1),
                  perf(BullM_20(OHLC_std_yz_20),BullM_20(Moving_std_30_), 1)),digits = 2)

Vec_30_past20_Bull_2<- round(c(perf(BullM_20(Moving_std_past_20), BullM_20(Moving_std_30_), 2),
                  perf(BullM_20(Moving_WMA_std_past_20), BullM_20(Moving_std_30_), 2),
                  perf(BullM_20(Moving_EWMA_std_past_20),BullM_20(Moving_std_30_), 2),
                  perf(BullM_20(OHLC_std_GK_20),BullM_20(Moving_std_30_), 2),
                  perf(BullM_20(OHLC_std_pk_20),BullM_20(Moving_std_30_), 2),
                  perf(BullM_20(OHLC_std_rs_20),BullM_20(Moving_std_30_), 2),
                  perf(BullM_20(OHLC_std_yz_20),BullM_20(Moving_std_30_), 2)),digits = 2)

Vec_30_past20_Bull_3 <- round(c(perf(BullM_20(Moving_std_past_20), BullM_20(Moving_std_30_), 3),
                  perf(BullM_20(Moving_WMA_std_past_20), BullM_20(Moving_std_30_), 3),
                  perf(BullM_20(Moving_EWMA_std_past_20),BullM_20(Moving_std_30_), 3),
                  perf(BullM_20(OHLC_std_GK_20),BullM_20(Moving_std_30_), 3),
                  perf(BullM_20(OHLC_std_pk_20),BullM_20(Moving_std_30_), 3),
                  perf(BullM_20(OHLC_std_rs_20),BullM_20(Moving_std_30_), 3),
                  perf(BullM_20(OHLC_std_yz_20),BullM_20(Moving_std_30_), 3)),digits = 2)

Vec_30_past20_Bull_4 <- round(c(perf(BullM_20(Moving_std_past_20), BullM_20(Moving_std_30_), 4),
                  perf(BullM_20(Moving_WMA_std_past_20), BullM_20(Moving_std_30_), 4),
                  perf(BullM_20(Moving_EWMA_std_past_20),BullM_20(Moving_std_30_), 4),
                  perf(BullM_20(OHLC_std_GK_20),BullM_20(Moving_std_30_), 4),
                  perf(BullM_20(OHLC_std_pk_20),BullM_20(Moving_std_30_), 4),
                  perf(BullM_20(OHLC_std_rs_20),BullM_20(Moving_std_30_), 4),
                  perf(BullM_20(OHLC_std_yz_20),BullM_20(Moving_std_30_), 4)),digits = 2)


# N = 20, Bear Market
Vec_30_past20_Bear_1 <- round(c(perf(BearM_20(Moving_std_past_20), BearM_20(Moving_std_30_), 1),
                  perf(BearM_20(Moving_WMA_std_past_20), BearM_20(Moving_std_30_), 1),
                  perf(BearM_20(Moving_EWMA_std_past_20),BearM_20(Moving_std_30_), 1),
                  perf(BearM_20(OHLC_std_GK_20),BearM_20(Moving_std_30_), 1),
                  perf(BearM_20(OHLC_std_pk_20),BearM_20(Moving_std_30_), 1),
                  perf(BearM_20(OHLC_std_rs_20),BearM_20(Moving_std_30_), 1),
                  perf(BearM_20(OHLC_std_yz_20),BearM_20(Moving_std_30_), 1)),digits = 2)

Vec_30_past20_Bear_2 <- round(c(perf(BearM_20(Moving_std_past_20), BearM_20(Moving_std_30_), 2),
                  perf(BearM_20(Moving_WMA_std_past_20), BearM_20(Moving_std_30_), 2),
                  perf(BearM_20(Moving_EWMA_std_past_20),BearM_20(Moving_std_30_), 2),
                  perf(BearM_20(OHLC_std_GK_20),BearM_20(Moving_std_30_), 2),
                  perf(BearM_20(OHLC_std_pk_20),BearM_20(Moving_std_30_), 2),
                  perf(BearM_20(OHLC_std_rs_20),BearM_20(Moving_std_30_), 2),
                  perf(BearM_20(OHLC_std_yz_20),BearM_20(Moving_std_30_), 2)),digits = 2)

Vec_30_past20_Bear_3 <- round(c(perf(BearM_20(Moving_std_past_20), BearM_20(Moving_std_30_), 3),
                  perf(BearM_20(Moving_WMA_std_past_20), BearM_20(Moving_std_30_), 3),
                  perf(BearM_20(Moving_EWMA_std_past_20),BearM_20(Moving_std_30_), 3),
                  perf(BearM_20(OHLC_std_GK_20),BearM_20(Moving_std_30_), 3),
                  perf(BearM_20(OHLC_std_pk_20),BearM_20(Moving_std_30_), 3),
                  perf(BearM_20(OHLC_std_rs_20),BearM_20(Moving_std_30_), 3),
                  perf(BearM_20(OHLC_std_yz_20),BearM_20(Moving_std_30_), 3)),digits = 2)
```

```r
Vec_30_past20_Bear_4 <- round(c(perf(BearM_20(Moving_std_past_20), BearM_20(Moving_std_30_), 4),
                    perf(BearM_20(Moving_WMA_std_past_20), BearM_20(Moving_std_30_), 4),
                    perf(BearM_20(Moving_EWMA_std_past_20),BearM_20(Moving_std_30_), 4),
                    perf(BearM_20(OHLC_std_GK_20),BearM_20(Moving_std_30_), 4),
                    perf(BearM_20(OHLC_std_pk_20),BearM_20(Moving_std_30_), 4),
                    perf(BearM_20(OHLC_std_rs_20),BearM_20(Moving_std_30_), 4),
                    perf(BearM_20(OHLC_std_yz_20),BearM_20(Moving_std_30_), 4)),digits = 2)

# N = 20, Flat Market
Vec_30_past20_Flat_1 <- round(c(perf(FlatM_20(Moving_std_past_20), FlatM_20(Moving_std_30_), 1),
                    perf(FlatM_20(Moving_WMA_std_past_20), FlatM_20(Moving_std_30_), 1),
                    perf(FlatM_20(Moving_EWMA_std_past_20),FlatM_20(Moving_std_30_), 1),
                    perf(FlatM_20(OHLC_std_GK_20),FlatM_20(Moving_std_30_), 1),
                    perf(FlatM_20(OHLC_std_pk_20),FlatM_20(Moving_std_30_), 1),
                    perf(FlatM_20(OHLC_std_rs_20),FlatM_20(Moving_std_30_), 1),
                    perf(FlatM_20(OHLC_std_yz_20),FlatM_20(Moving_std_30_), 1)),digits = 2)

Vec_30_past20_Flat_2 <- round(c(perf(FlatM_20(Moving_std_past_20), FlatM_20(Moving_std_30_), 2),
                    perf(FlatM_20(Moving_WMA_std_past_20), FlatM_20(Moving_std_30_), 2),
                    perf(FlatM_20(Moving_EWMA_std_past_20),FlatM_20(Moving_std_30_), 2),
                    perf(FlatM_20(OHLC_std_GK_20),FlatM_20(Moving_std_30_), 2),
                    perf(FlatM_20(OHLC_std_pk_20),FlatM_20(Moving_std_30_), 2),
                    perf(FlatM_20(OHLC_std_rs_20),FlatM_20(Moving_std_30_), 2),
                    perf(FlatM_20(OHLC_std_yz_20),FlatM_20(Moving_std_30_), 2)),digits = 2)

Vec_30_past20_Flat_3 <- round(c(perf(FlatM_20(Moving_std_past_20), FlatM_20(Moving_std_30_), 3),
                    perf(FlatM_20(Moving_WMA_std_past_20), FlatM_20(Moving_std_30_), 3),
                    perf(FlatM_20(Moving_EWMA_std_past_20),FlatM_20(Moving_std_30_), 3),
                    perf(FlatM_20(OHLC_std_GK_20),FlatM_20(Moving_std_30_), 3),
                    perf(FlatM_20(OHLC_std_pk_20),FlatM_20(Moving_std_30_), 3),
                    perf(FlatM_20(OHLC_std_rs_20),FlatM_20(Moving_std_30_), 3),
                    perf(FlatM_20(OHLC_std_yz_20),FlatM_20(Moving_std_30_), 3)),digits = 2)

Vec_30_past20_Flat_4 <- round(c(perf(FlatM_20(Moving_std_past_20), FlatM_20(Moving_std_30_), 4),
                    perf(FlatM_20(Moving_WMA_std_past_20), FlatM_20(Moving_std_30_), 4),
                    perf(FlatM_20(Moving_EWMA_std_past_20),FlatM_20(Moving_std_30_), 4),
                    perf(FlatM_20(OHLC_std_GK_20),FlatM_20(Moving_std_30_), 4),
                    perf(FlatM_20(OHLC_std_pk_20),FlatM_20(Moving_std_30_), 4),
                    perf(FlatM_20(OHLC_std_rs_20),FlatM_20(Moving_std_30_), 4),
                    perf(FlatM_20(OHLC_std_yz_20),FlatM_20(Moving_std_30_), 4)),digits = 2)


MarketVec_30_1 <- data.frame(rbind(Vec_30_past20_Bull_1,Vec_30_past20_Bear_1,Vec_30_past20_Flat_1, Vec_3
MarketVec_30_2 <- data.frame(rbind(Vec_30_past20_Bull_2,Vec_30_past20_Bear_2,Vec_30_past20_Flat_2, Vec_3
MarketVec_30_3 <- data.frame(rbind(Vec_30_past20_Bull_3,Vec_30_past20_Bear_3,Vec_30_past20_Flat_3, Vec_3
MarketVec_30_4 <- data.frame(rbind(Vec_30_past20_Bull_4,Vec_30_past20_Bear_4,Vec_30_past20_Flat_4, Vec_3

MarketVec_30_1 <- compareMarket(MarketVec_30_1)
MarketVec_30_2 <- compareMarket(MarketVec_30_2)
MarketVec_30_3 <- compareMarket(MarketVec_30_3)
MarketVec_30_4 <- compareMarket(MarketVec_30_4)

print(MarketVec_30_1)
```

```
##                SMA   WMA  EWMA Garman and Klass Parkinson
## Bull Market 11.44 10.61 11.32            11.47     10.96
## Bear Market 11.70 13.45 11.79            14.18     13.31
## Flat Market  2.71  2.78  2.72             2.70      2.54
## Overall      6.58  6.74  6.57             6.71      6.31
##              Rogers and Satchell Yang and Zhang
## Bull Market                11.65           9.90
## Bear Market                14.22          19.39
## Flat Market                 2.98           4.70
## Overall                     7.04           9.35
```

```r
print(MarketVec_30_3)
```

```
##                SMA   WMA  EWMA Garman and Klass Parkinson
## Bull Market -6.26 -5.87 -6.20            -9.05     -8.86
## Bear Market -1.09 -1.22 -1.07            -6.36     -6.34
## Flat Market -0.08 -0.48 -0.12            -1.83     -1.58
## Overall      0.06 -0.12  0.05            -4.94     -3.98
##              Rogers and Satchell Yang and Zhang
## Bull Market                -8.87           1.95
## Bear Market                -5.78          15.57
## Flat Market                -1.66           3.95
## Overall                    -5.29           7.05
```

Future 90 days

```r
# N = 20, Bull Market
Vec_90_past20_Bull_1 <- round(c(perf(BullM_20(Moving_std_past_20), BullM_20(Moving_std_90_), 1),
                   perf(BullM_20(Moving_WMA_std_past_20), BullM_20(Moving_std_90_), 1),
                   perf(BullM_20(Moving_EWMA_std_past_20),BullM_20(Moving_std_90_), 1),
                   perf(BullM_20(OHLC_std_GK_20),BullM_20(Moving_std_90_), 1),
                   perf(BullM_20(OHLC_std_pk_20),BullM_20(Moving_std_90_), 1),
                   perf(BullM_20(OHLC_std_rs_20),BullM_20(Moving_std_90_), 1),
                   perf(BullM_20(OHLC_std_yz_20),BullM_20(Moving_std_90_), 1)),digits = 2)

Vec_90_past20_Bull_2<- round(c(perf(BullM_20(Moving_std_past_20), BullM_20(Moving_std_90_), 2),
                   perf(BullM_20(Moving_WMA_std_past_20), BullM_20(Moving_std_90_), 2),
                   perf(BullM_20(Moving_EWMA_std_past_20),BullM_20(Moving_std_90_), 2),
                   perf(BullM_20(OHLC_std_GK_20),BullM_20(Moving_std_90_), 2),
                   perf(BullM_20(OHLC_std_pk_20),BullM_20(Moving_std_90_), 2),
                   perf(BullM_20(OHLC_std_rs_20),BullM_20(Moving_std_90_), 2),
                   perf(BullM_20(OHLC_std_yz_20),BullM_20(Moving_std_90_), 2)),digits = 2)

Vec_90_past20_Bull_3 <- round(c(perf(BullM_20(Moving_std_past_20), BullM_20(Moving_std_90_), 3),
                   perf(BullM_20(Moving_WMA_std_past_20), BullM_20(Moving_std_90_), 3),
                   perf(BullM_20(Moving_EWMA_std_past_20),BullM_20(Moving_std_90_), 3),
                   perf(BullM_20(OHLC_std_GK_20),BullM_20(Moving_std_90_), 3),
                   perf(BullM_20(OHLC_std_pk_20),BullM_20(Moving_std_90_), 3),
                   perf(BullM_20(OHLC_std_rs_20),BullM_20(Moving_std_90_), 3),
                   perf(BullM_20(OHLC_std_yz_20),BullM_20(Moving_std_90_), 3)),digits = 2)

Vec_90_past20_Bull_4 <- round(c(perf(BullM_20(Moving_std_past_20), BullM_20(Moving_std_90_), 4),
                   perf(BullM_20(Moving_WMA_std_past_20), BullM_20(Moving_std_90_), 4),
                   perf(BullM_20(Moving_EWMA_std_past_20),BullM_20(Moving_std_90_), 4),
                   perf(BullM_20(OHLC_std_GK_20),BullM_20(Moving_std_90_), 4),
                   perf(BullM_20(OHLC_std_pk_20),BullM_20(Moving_std_90_), 4),
```

```r
                    perf(BullM_20(OHLC_std_rs_20),BullM_20(Moving_std_90_), 4),
                    perf(BullM_20(OHLC_std_yz_20),BullM_20(Moving_std_90_), 4)),digits = 2)


# N = 20, Bear Market
Vec_90_past20_Bear_1 <- round(c(perf(BearM_20(Moving_std_past_20), BearM_20(Moving_std_90_), 1),
                    perf(BearM_20(Moving_WMA_std_past_20), BearM_20(Moving_std_90_), 1),
                    perf(BearM_20(Moving_EWMA_std_past_20),BearM_20(Moving_std_90_), 1),
                    perf(BearM_20(OHLC_std_GK_20),BearM_20(Moving_std_90_), 1),
                    perf(BearM_20(OHLC_std_pk_20),BearM_20(Moving_std_90_), 1),
                    perf(BearM_20(OHLC_std_rs_20),BearM_20(Moving_std_90_), 1),
                    perf(BearM_20(OHLC_std_yz_20),BearM_20(Moving_std_90_), 1)),digits = 2)

Vec_90_past20_Bear_2 <- round(c(perf(BearM_20(Moving_std_past_20), BearM_20(Moving_std_90_), 2),
                    perf(BearM_20(Moving_WMA_std_past_20), BearM_20(Moving_std_90_), 2),
                    perf(BearM_20(Moving_EWMA_std_past_20),BearM_20(Moving_std_90_), 2),
                    perf(BearM_20(OHLC_std_GK_20),BearM_20(Moving_std_90_), 2),
                    perf(BearM_20(OHLC_std_pk_20),BearM_20(Moving_std_90_), 2),
                    perf(BearM_20(OHLC_std_rs_20),BearM_20(Moving_std_90_), 2),
                    perf(BearM_20(OHLC_std_yz_20),BearM_20(Moving_std_90_), 2)),digits = 2)

Vec_90_past20_Bear_3 <- round(c(perf(BearM_20(Moving_std_past_20), BearM_20(Moving_std_90_), 3),
                    perf(BearM_20(Moving_WMA_std_past_20), BearM_20(Moving_std_90_), 3),
                    perf(BearM_20(Moving_EWMA_std_past_20),BearM_20(Moving_std_90_), 3),
                    perf(BearM_20(OHLC_std_GK_20),BearM_20(Moving_std_90_), 3),
                    perf(BearM_20(OHLC_std_pk_20),BearM_20(Moving_std_90_), 3),
                    perf(BearM_20(OHLC_std_rs_20),BearM_20(Moving_std_90_), 3),
                    perf(BearM_20(OHLC_std_yz_20),BearM_20(Moving_std_90_), 3)),digits = 2)

Vec_90_past20_Bear_4 <- round(c(perf(BearM_20(Moving_std_past_20), BearM_20(Moving_std_90_), 4),
                    perf(BearM_20(Moving_WMA_std_past_20), BearM_20(Moving_std_90_), 4),
                    perf(BearM_20(Moving_EWMA_std_past_20),BearM_20(Moving_std_90_), 4),
                    perf(BearM_20(OHLC_std_GK_20),BearM_20(Moving_std_90_), 4),
                    perf(BearM_20(OHLC_std_pk_20),BearM_20(Moving_std_90_), 4),
                    perf(BearM_20(OHLC_std_rs_20),BearM_20(Moving_std_90_), 4),
                    perf(BearM_20(OHLC_std_yz_20),BearM_20(Moving_std_90_), 4)),digits = 2)

# N = 20, Flat Market
Vec_90_past20_Flat_1 <- round(c(perf(FlatM_20(Moving_std_past_20), FlatM_20(Moving_std_90_), 1),
                    perf(FlatM_20(Moving_WMA_std_past_20), FlatM_20(Moving_std_90_), 1),
                    perf(FlatM_20(Moving_EWMA_std_past_20),FlatM_20(Moving_std_90_), 1),
                    perf(FlatM_20(OHLC_std_GK_20),FlatM_20(Moving_std_90_), 1),
                    perf(FlatM_20(OHLC_std_pk_20),FlatM_20(Moving_std_90_), 1),
                    perf(FlatM_20(OHLC_std_rs_20),FlatM_20(Moving_std_90_), 1),
                    perf(FlatM_20(OHLC_std_yz_20),FlatM_20(Moving_std_90_), 1)),digits = 2)

Vec_90_past20_Flat_2 <- round(c(perf(FlatM_20(Moving_std_past_20), FlatM_20(Moving_std_90_), 2),
                    perf(FlatM_20(Moving_WMA_std_past_20), FlatM_20(Moving_std_90_), 2),
                    perf(FlatM_20(Moving_EWMA_std_past_20),FlatM_20(Moving_std_90_), 2),
                    perf(FlatM_20(OHLC_std_GK_20),FlatM_20(Moving_std_90_), 2),
                    perf(FlatM_20(OHLC_std_pk_20),FlatM_20(Moving_std_90_), 2),
                    perf(FlatM_20(OHLC_std_rs_20),FlatM_20(Moving_std_90_), 2),
                    perf(FlatM_20(OHLC_std_yz_20),FlatM_20(Moving_std_90_), 2)),digits = 2)
```

```r
Vec_90_past20_Flat_3 <- round(c(perf(FlatM_20(Moving_std_past_20), FlatM_20(Moving_std_90_), 3),
                    perf(FlatM_20(Moving_WMA_std_past_20), FlatM_20(Moving_std_90_), 3),
                    perf(FlatM_20(Moving_EWMA_std_past_20),FlatM_20(Moving_std_90_), 3),
                    perf(FlatM_20(OHLC_std_GK_20),FlatM_20(Moving_std_90_), 3),
                    perf(FlatM_20(OHLC_std_pk_20),FlatM_20(Moving_std_90_), 3),
                    perf(FlatM_20(OHLC_std_rs_20),FlatM_20(Moving_std_90_), 3),
                    perf(FlatM_20(OHLC_std_yz_20),FlatM_20(Moving_std_90_), 3)),digits = 2)

Vec_90_past20_Flat_4 <- round(c(perf(FlatM_20(Moving_std_past_20), FlatM_20(Moving_std_90_), 4),
                    perf(FlatM_20(Moving_WMA_std_past_20), FlatM_20(Moving_std_90_), 4),
                    perf(FlatM_20(Moving_EWMA_std_past_20),FlatM_20(Moving_std_90_), 4),
                    perf(FlatM_20(OHLC_std_GK_20),FlatM_20(Moving_std_90_), 4),
                    perf(FlatM_20(OHLC_std_pk_20),FlatM_20(Moving_std_90_), 4),
                    perf(FlatM_20(OHLC_std_rs_20),FlatM_20(Moving_std_90_), 4),
                    perf(FlatM_20(OHLC_std_yz_20),FlatM_20(Moving_std_90_), 4)),digits = 2)


MarketVec_90_1 <- data.frame(rbind(Vec_90_past20_Bull_1,Vec_90_past20_Bear_1,Vec_90_past20_Flat_1, Vec_9
MarketVec_90_2 <- data.frame(rbind(Vec_90_past20_Bull_2,Vec_90_past20_Bear_2,Vec_90_past20_Flat_2, Vec_9
MarketVec_90_3 <- data.frame(rbind(Vec_90_past20_Bull_3,Vec_90_past20_Bear_3,Vec_90_past20_Flat_3, Vec_9
MarketVec_90_4 <- data.frame(rbind(Vec_90_past20_Bull_4,Vec_90_past20_Bear_4,Vec_90_past20_Flat_4, Vec_9

MarketVec_90_1 <- compareMarket(MarketVec_90_1)
MarketVec_90_2 <- compareMarket(MarketVec_90_2)
MarketVec_90_3 <- compareMarket(MarketVec_90_3)
MarketVec_90_4 <- compareMarket(MarketVec_90_4)

print(MarketVec_90_1)
```

```
##                SMA   WMA  EWMA Garman and Klass Parkinson
## Bull Market 16.13 16.17 16.10            17.95     17.42
## Bear Market 15.36 15.89 15.45            13.21     12.69
## Flat Market  1.85  2.06  1.86             2.72      2.61
## Overall      6.96  7.20  6.96             7.42      7.04
##             Rogers and Satchell Yang and Zhang
## Bull Market               18.13          14.05
## Bear Market               13.08          27.61
## Flat Market                2.58           3.64
## Overall                    7.74           9.51
```

```r
print(MarketVec_90_3)
```

```
##                 SMA    WMA   EWMA Garman and Klass Parkinson
## Bull Market -12.08 -11.69 -12.03           -14.87    -14.69
## Bear Market  10.88  10.75  10.90             5.61      5.63
## Flat Market  -0.90  -1.30  -0.94            -2.65     -2.40
## Overall      -0.21  -0.39  -0.22            -5.21     -4.25
##             Rogers and Satchell Yang and Zhang
## Bull Market              -14.70          -3.88
## Bear Market                6.19          27.54
## Flat Market               -2.47           3.13
## Overall                   -5.56           6.78
```

Summary:

3. Parkinson model systematically underestimates volatility
4. EWMA and Parkinson models have the smallest index error in both bear market and flat market
5. Yang and Zhang model has the smallest index error in bull market, but such model perform poorly (oftentimes overestamite volatility) in other market trends

6. Time Series / ARCH Garch(1,1) Model

A time series is a series of data points indexed (or listed or graphed) in time order. Autoregressive conditional heteroskedasticity (ARCH) is the condition that there are one or more data points in a series for which the variance of the current error term or innovation is a function of the actual sizes of the previous time periods' error terms: often the variance is related to the squares of the previous innovations. If an autoregressive moving average model (ARMA model) is assumed for the error variance, the model is a generalized autoregressive conditional heteroscedasticity (GARCH) model. In that case, the GARCH (p, q) model (where p is the order of the GARCH terms sigmaˆ2 and q is the order of the ARCH terms epsilonˆ2). In the following I will use Garch(1,1) model to forecast monthly volatility in 2017 using historical daily return on CSI 300 Index from Wind Financial Terminal.

Data Collecting

Source: Wind Financial Terminal

Data: Historical daily return on CSI 300 index, which tracks the Shanghai and Shenzhen Markets

Time: 2007/7/4 - 2017/8/4

Indicator(s): Closing Price(Index)

```
library(readxl)
library(quantmod)
library(xts)
library(zoo)
library(timeDate)
library(timeSeries)
library(vars)
library(tseries)
library(fBasics)
library(fGarch)
library(TTR)
library(stats)
library(graphics)
library(magrittr)
library(ggplot2)
library(tidyr)
```

```
HS <- read_excel("~/Documents/Bu Academics/Rising Senior Summer/htf /Index300.xlsx", col_names = FALSE)
colnames(HS) <- c("Date", "Index")
HS <- xts(HS$Index, order.by = as.Date(HS$Date))
colnames(HS) <- "HS_Index"
dim(HS)
```

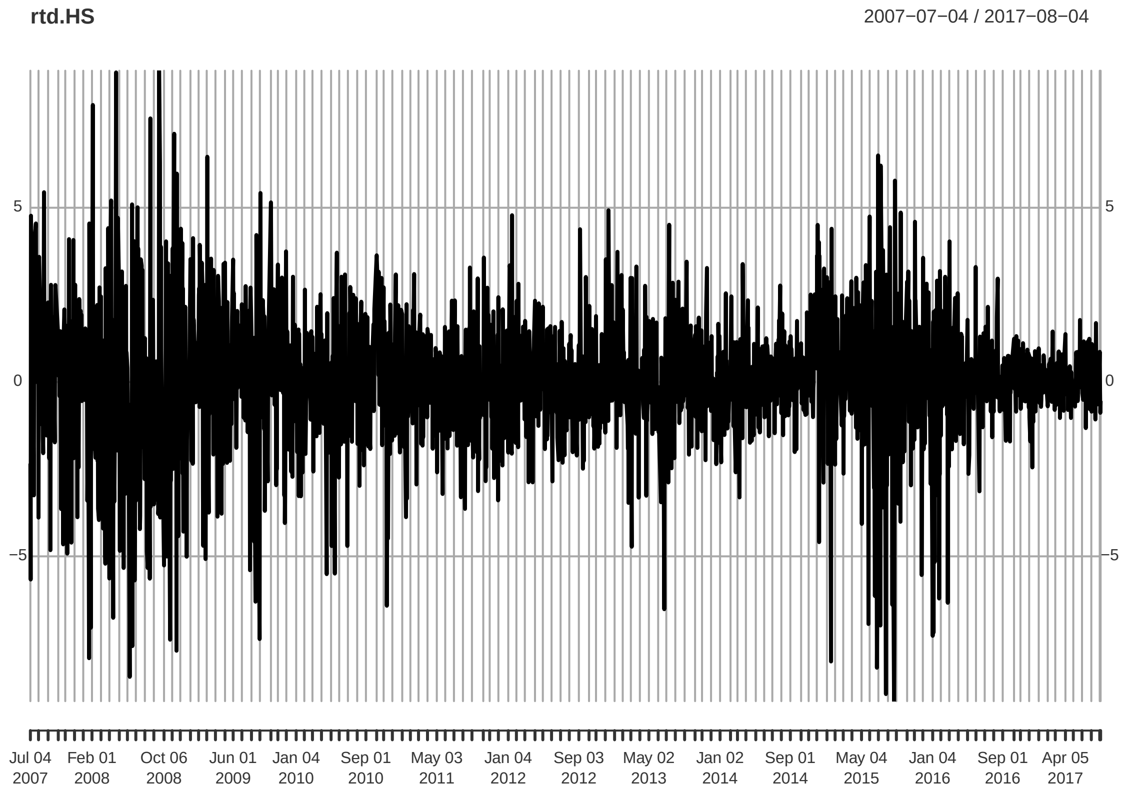
```
## [1] 2458    1
```

```
names(HS)
```

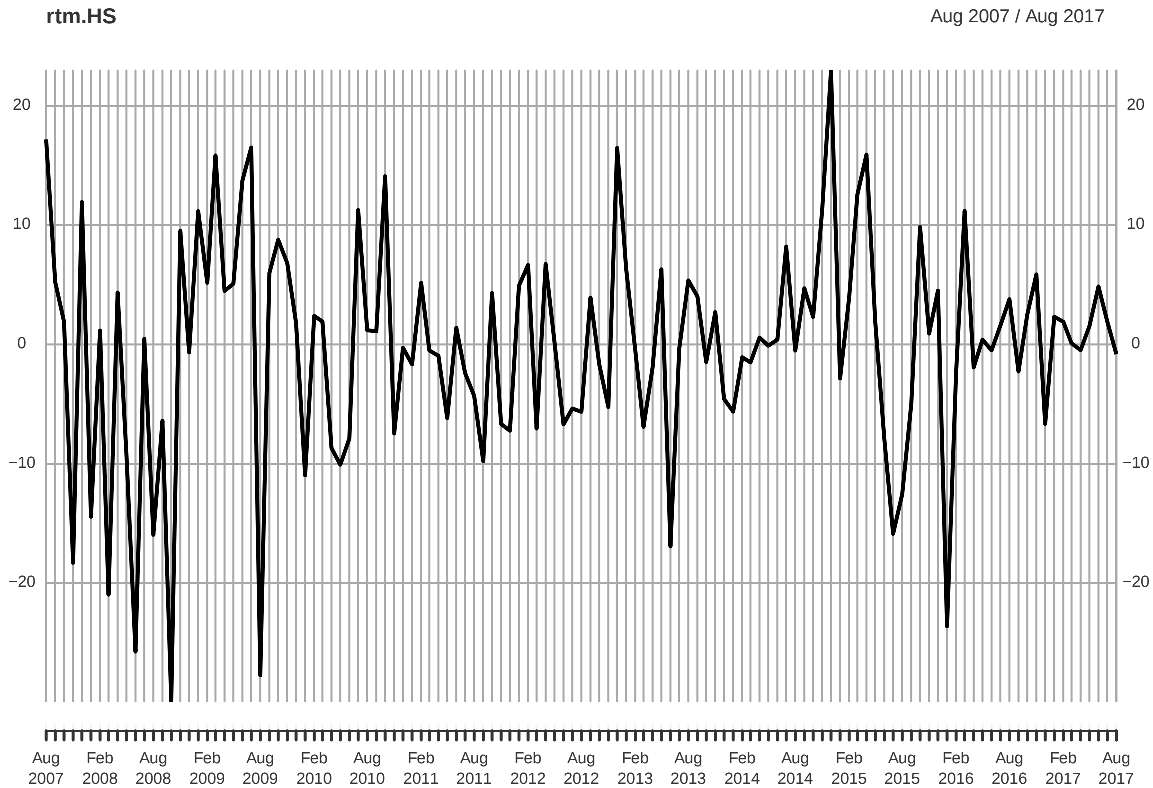
```
## [1] "HS_Index"
```

```r
chartSeries(HS, theme = 'white')
```


```r
# Daily return
rtd.HS <- diff(log(HS$HS_Index)) * 100
rtd.HS <- rtd.HS[-1,]
plot(rtd.HS)
```




```
# Monthly return (Assume there are 20 trading days in a month)
ptm.HS <- to.monthly(HS)$HS.Close
colnames(ptm.HS) <- "HS.Adjusted"
rtm.HS <- diff(log(ptm.HS)) * 100
rtm.HS <- rtm.HS[-1,]
plot(rtm.HS)
```



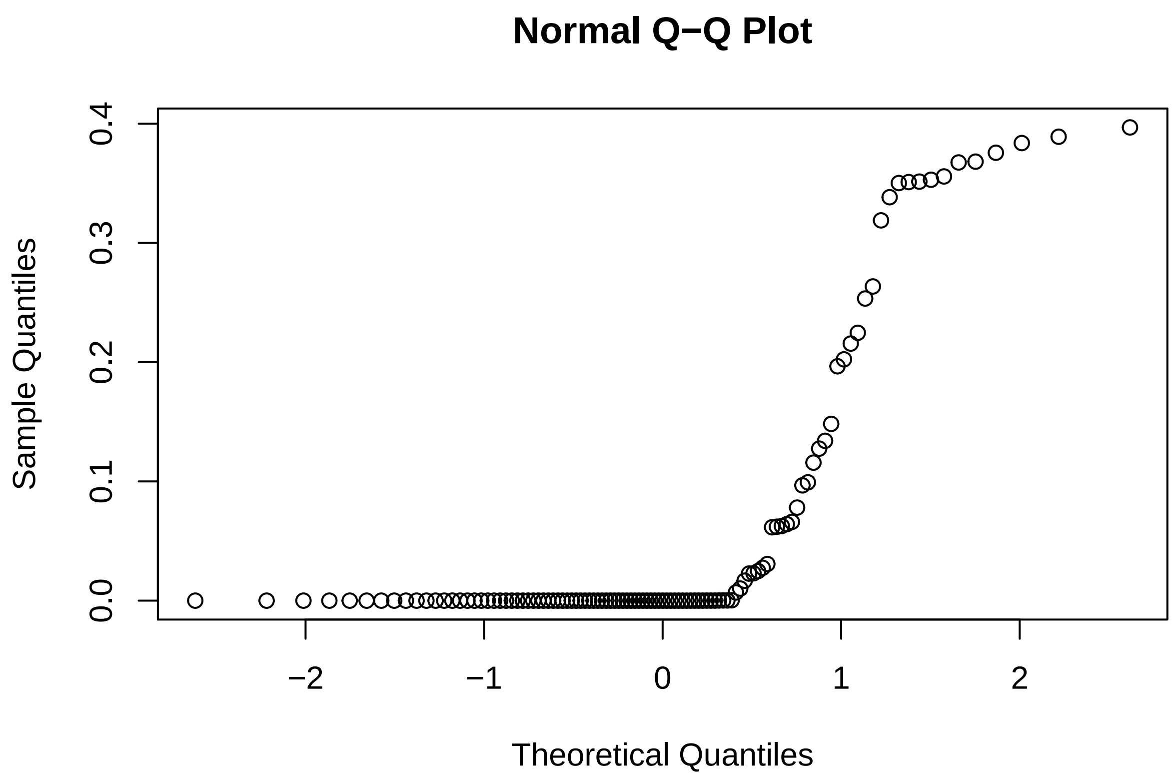

```r
# Insample and Outsample
ind.outsample <- sub(' ','',substr(index(rtm.HS), 4, 8)) %in% '2017'
ind.insample <- !ind.outsample
rtm.insample <- rtm.HS[ind.insample]
rtm.outsample <- rtm.HS[ind.outsample]
rtm.outsample <- rtm.outsample[-8]

# Check skewness and Kurtosis to validate time-series model
skewness(rtm.insample)
```

```
## [1] -0.5955349
## attr(,"method")
## [1] "moment"
```

```r
# s > 0 It is positive skewed, not bell curve shaped
kurtosis(rtm.insample)
```

```
## [1] 0.9210539
## attr(,"method")
## [1] "excess"
```

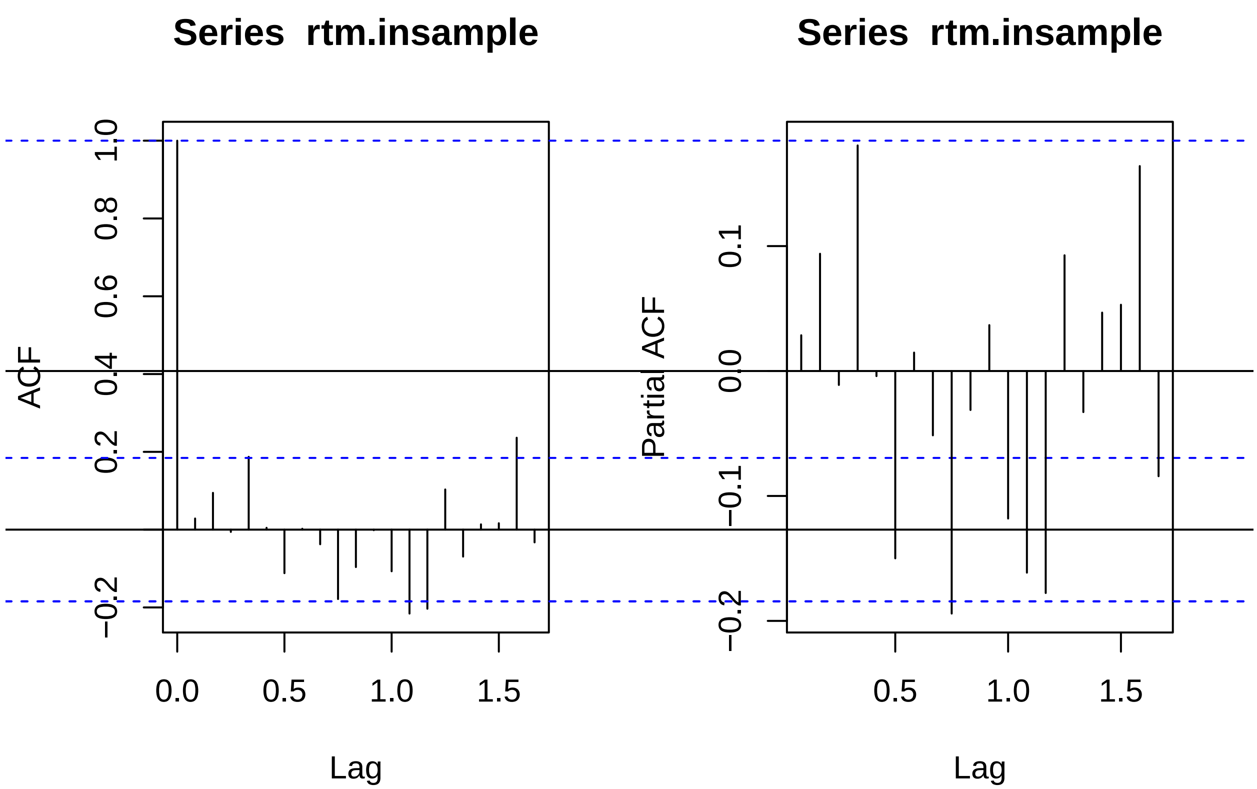
```r
# K > 3
qqnorm(dnorm(rtm.insample))
```




```
# not bell curve shaped

# Angmented Dicky-Fuller test(whether a unit root is presented in a time series sample)
adf.test(rtm.insample)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  rtm.insample
## Dickey-Fuller = -3.8607, Lag order = 4, p-value = 0.01851
## alternative hypothesis: stationary
```

```
# p-value = 0.01, we reject the null hypothesis and conclude that time-series is stationary

par(mfrow=c(1,2))
# ACF,
acf(rtm.insample)
# PACF
pacf(rtm.insample)
```




```r
# Ljung-Box-Pierce
Box.test(rtm.insample,lag = 12, type = 'Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  rtm.insample
## X-squared = 13.653, df = 12, p-value = 0.3234
```

```r
# P > 0.05, fail to reject the null hypothesis
Box.test(rtm.insample^2,lag = 12, type = 'Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  rtm.insample^2
## X-squared = 27.071, df = 12, p-value = 0.007548
```

```r
# p < 0.05, reject the null hypothesis


# Build GARCH Model
GARCH.model_1 <- garchFit(~garch(1,1), data=rtm.insample, trace=FALSE)
summary(GARCH.model_1)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = rtm.insample, trace = FALSE)
```

```
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fef9c9ee6b0>
##  [data = rtm.insample]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##       mu     omega    alpha1     beta1
## -0.43444   6.57960   0.16885   0.75770
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu       -0.4344      0.7533   -0.577    0.564
## omega     6.5796      4.3986    1.496    0.135
## alpha1    0.1688      0.1133    1.490    0.136
## beta1     0.7577      0.1158    6.543 6.04e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  -409.1291    normalized:  -3.620611
##
## Description:
##  Mon Sep 11 07:06:46 2017 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  2.329869  0.3119432
##  Shapiro-Wilk Test  R    W      0.9815609 0.1215334
##  Ljung-Box Test     R    Q(10)  11.16754  0.3446152
##  Ljung-Box Test     R    Q(15)  25.30879  0.04594703
##  Ljung-Box Test     R    Q(20)  35.38641  0.01814024
##  Ljung-Box Test     R^2  Q(10)  5.121961  0.8828833
##  Ljung-Box Test     R^2  Q(15)  8.717712  0.8918088
##  Ljung-Box Test     R^2  Q(20)  11.99099  0.9163857
##  LM Arch Test       R    TR^2   3.521657  0.9906065
##
## Information Criterion Statistics:
##      AIC      BIC      SIC     HQIC
## 7.312019 7.408564 7.309625 7.351196
```

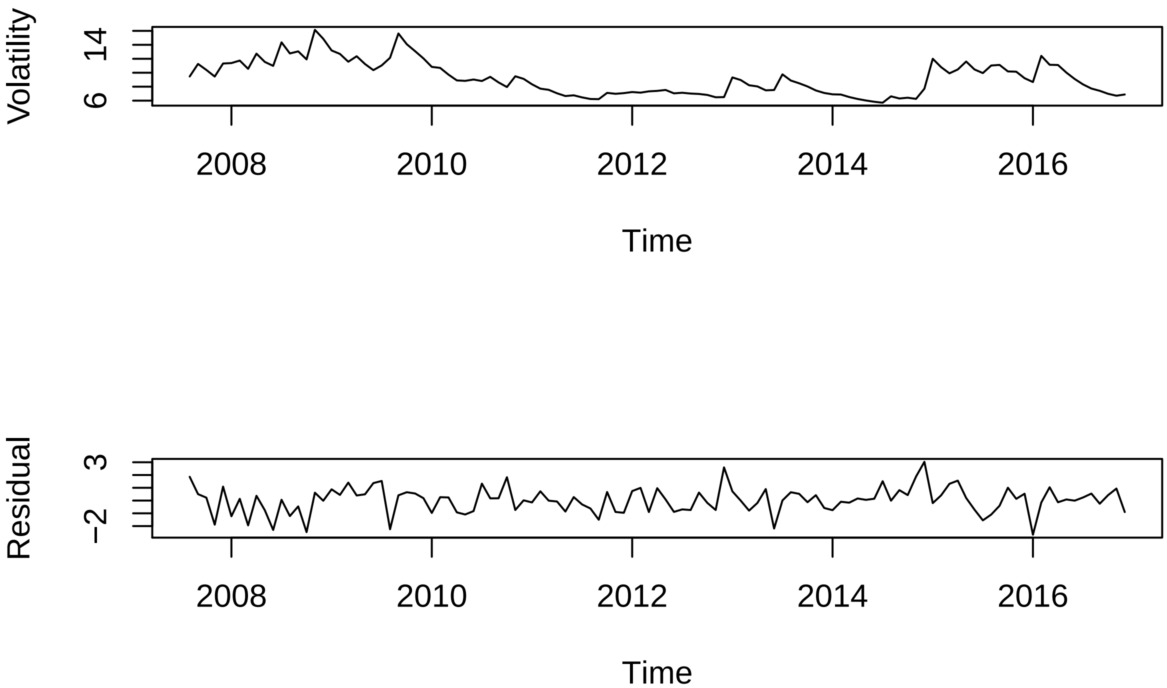
```
GARCH.model_2 <- garchFit(~garch(1,1), data=rtm.insample, cond.dist='std', trace=FALSE)
summary(GARCH.model_2)
```

```
##
## Title:
##  GARCH Modelling
##
```

```
## Call:
##  garchFit(formula = ~garch(1, 1), data = rtm.insample, cond.dist = "std",
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fef9daf4fb8>
##  [data = rtm.insample]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##       mu     omega    alpha1     beta1     shape
## -0.23356   5.38342   0.13994   0.79733   7.38038
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu       -0.2336      0.7586   -0.308    0.758
## omega     5.3834      4.6531    1.157    0.247
## alpha1    0.1399      0.1074    1.303    0.193
## beta1     0.7973      0.1163    6.854 7.19e-12 ***
## shape     7.3804      5.2337    1.410    0.158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  -407.971    normalized:  -3.610363
##
## Description:
##  Mon Sep 11 07:06:46 2017 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  2.525962  0.2828098
##  Shapiro-Wilk Test  R    W      0.9810417 0.109466
##  Ljung-Box Test     R    Q(10)  11.00146  0.3574045
##  Ljung-Box Test     R    Q(15)  25.05264  0.04924091
##  Ljung-Box Test     R    Q(20)  34.99325  0.02014023
##  Ljung-Box Test     R^2  Q(10)  4.985329  0.8921559
##  Ljung-Box Test     R^2  Q(15)  8.396015  0.9069268
##  Ljung-Box Test     R^2  Q(20)  11.78451  0.9232932
##  LM Arch Test       R    TR^2   3.451132  0.9914341
##
## Information Criterion Statistics:
##      AIC      BIC      SIC     HQIC
## 7.309221 7.429902 7.305522 7.358192
```

```
# plot(GARCH.model_1) Uncomment this if you want to plot model
# plot(GARCH.model_2) Uncomment this if you want to plot model
```

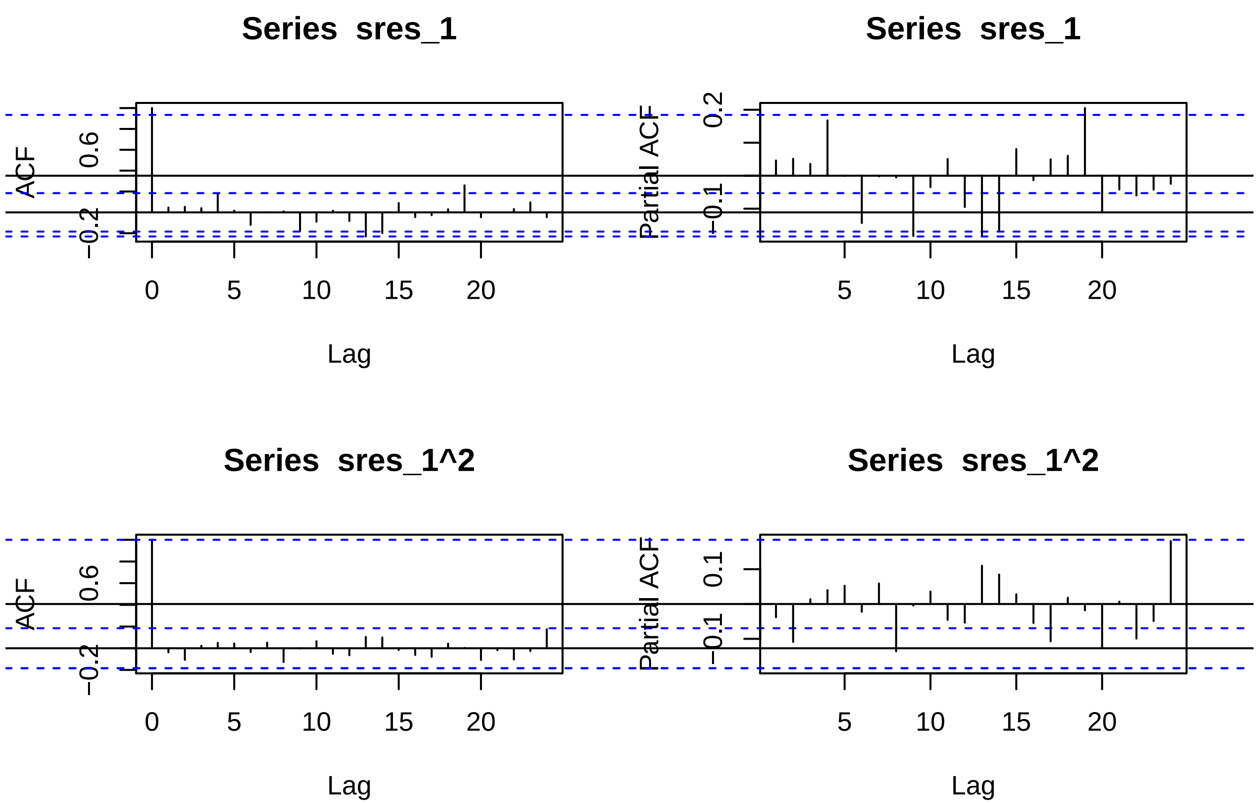
```r
# Select Volatility
vol_1 <- fBasics::volatility(GARCH.model_1)   # Get Volatility from Garch(1,1)
sres_1 <- residuals(GARCH.model_1, standardize = TRUE) # Get residual
vol_1.ts <- ts(vol_1, frequency=12, start=c(2007, 8))
sres_1.ts <- ts(sres_1, frequency=12, start=c(2007, 8))
par(mfcol=c(2,1))
plot(vol_1.ts, xlab='Time', ylab='Volatility')
plot(sres_1.ts, xlab='Time', ylab='Residual')
```



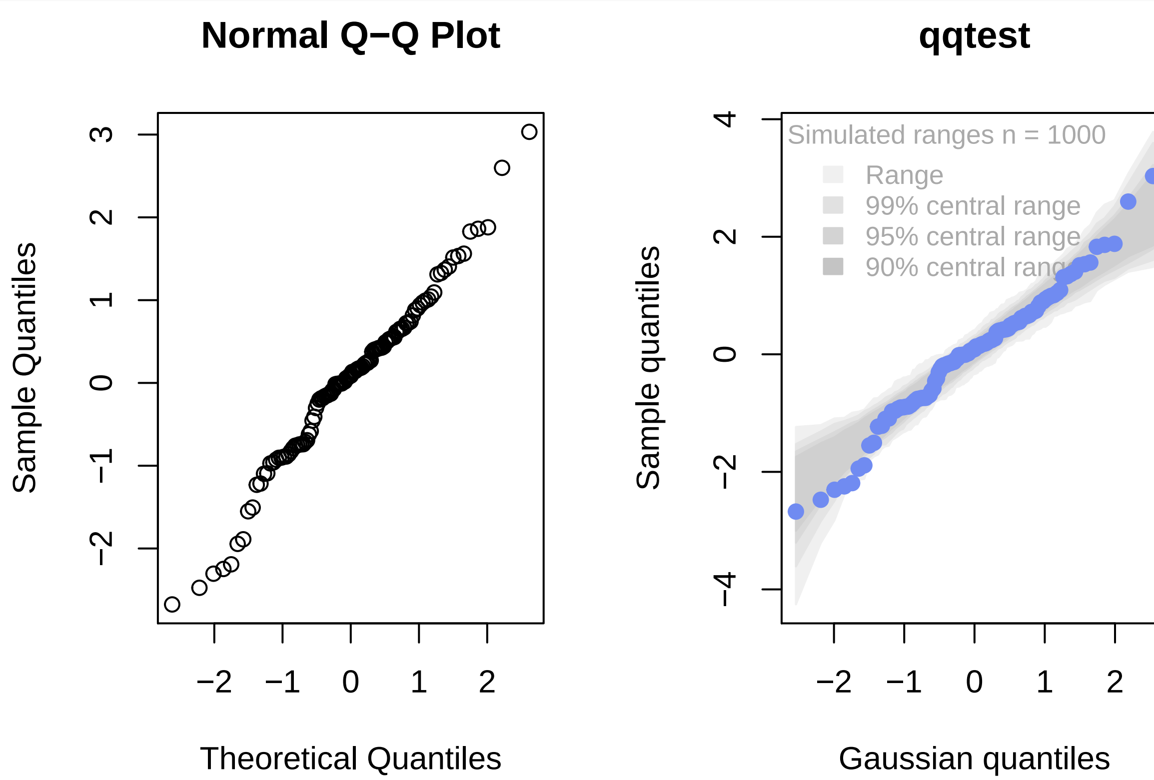

```r
# Model Validation
library(qqtest)
```

```
##
## Attaching package: 'qqtest'

## The following object is masked from 'package:MASS':
##
##     bacteria
```

```r
par(mfrow = c(2,2))
acf(sres_1, lag=24)
pacf(sres_1, lag=24)
acf(sres_1^2, lag=24)
pacf(sres_1^2, lag=24)
```



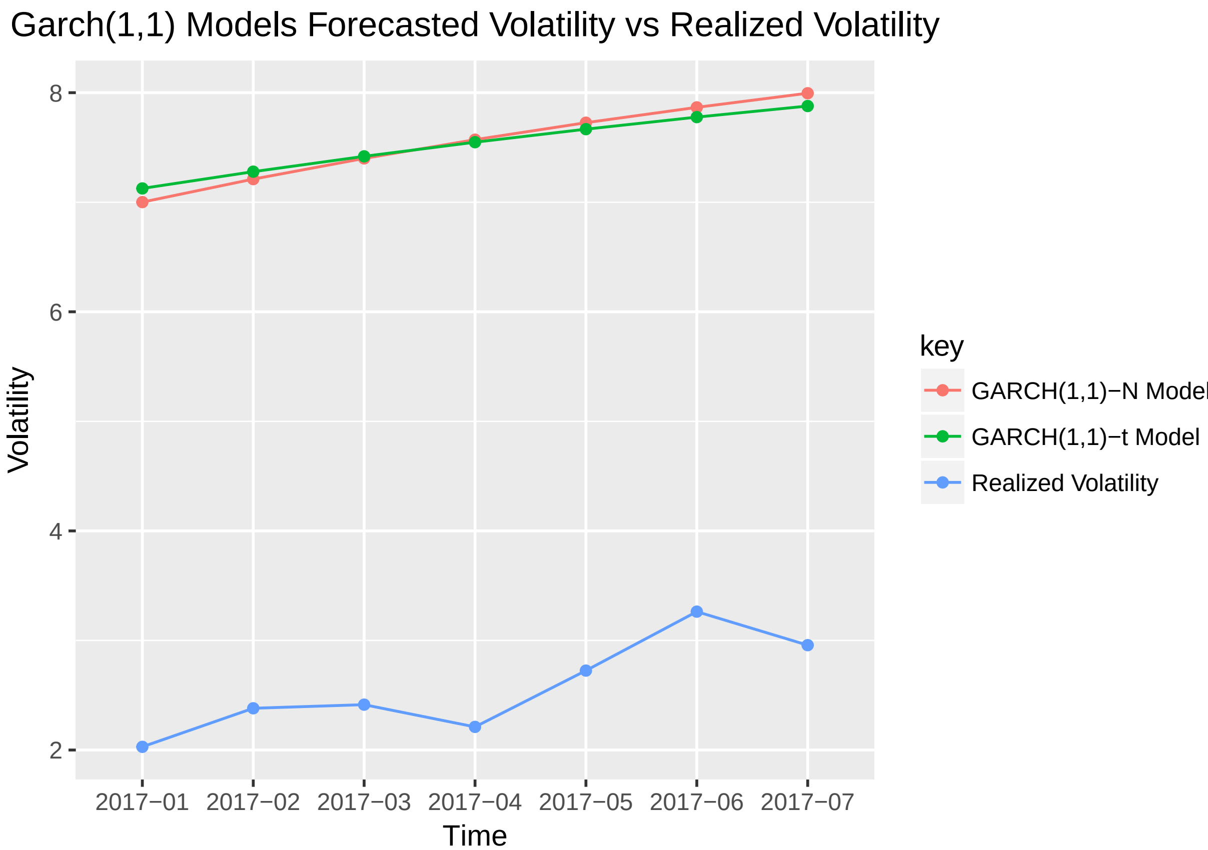

```r
par(mfrow = c(1,2))
qqnorm(sres_1)
qqtest(sres_1)
```



```r
# Volatility Forecasting
pred.model_1 <- predict(GARCH.model_1, n.ahead = 7, trace = FALSE, mse = 'cond', plot=FALSE)
pred.model_2 <- predict(GARCH.model_2, n.ahead = 7, trace = FALSE, mse = 'cond', plot=FALSE)

predVol_1 <- as.matrix(pred.model_1$standardDeviation)
predVol_2 <- as.matrix(pred.model_2$standardDeviation)

et <- abs(rtm.outsample - mean(rtm.outsample))
rtd.HS.2017 <- rtd.HS['2017'] %>% na.omit()
rv <- sqrt(aggregate(rtd.HS.2017^2, by=substr(index(rtd.HS.2017), 1, 7), sum))[-8]
Time <- as.matrix(index(rv))[-8]


predVol_ <- data.frame(round(cbind(predVol_1,predVol_2, as.numeric(et),as.numeric(rv)), digits=3))

predVol <- cbind(Time, predVol_)

rownames(predVol) <- index(rv)
colnames(predVol) <- c('Time','GARCH(1,1)-N Model',
                       'GARCH(1,1)-t Model',
                       'Residual', 'Realized Volatility')
print(predVol)
```

```
##            Time GARCH(1,1)-N Model GARCH(1,1)-t Model Residual
## 2017-01 2017-01              7.001              7.126    0.589
## 2017-02 2017-02              7.211              7.279    0.160
## 2017-03 2017-03              7.400              7.419    1.643
## 2017-04 2017-04              7.571              7.548    2.209
## 2017-05 2017-05              7.726              7.667    0.204
## 2017-06 2017-06              7.866              7.777    3.123
## 2017-07 2017-07              7.995              7.878    0.183
##         Realized Volatility
## 2017-01               2.029
## 2017-02               2.381
## 2017-03               2.414
## 2017-04               2.211
## 2017-05               2.725
## 2017-06               3.263
## 2017-07               2.957
```

```r
# Model Comparison
predVol %>% gather(key, Volatility, c(c(2:3),5)) %>% ggplot(aes(x = Time, y = Volatility, colour = key,
  "Garch(1,1) Models Forecasted Volatility vs Realized Volatility")
```



Garch(1,1) model seems to overestimate monthly volatility. A possible explanation can be CSI 300 index is relatively stable (Flat) in the year of 2017, and therefore monthly volatility is smaller than expectations whereas historical daily return of CSI 300 is much more fluctuating, which means its forcatsed volatility results could be much bigger than realized ones.

7. Further Improvement

1. EWMA model can be very tricky because different half-life may impact how fast the weight of exponential moving average decreases as time decrements from n to 1. Choosing the optimal half life that will closely approximate forecasted volatility to realized one is at heart of EWMA modeling.
2. Since forecsted volatility curve using past 300 days or 20 days daily return of CSI 300 index are either too smooth or to choopy, I am sure that weight these forcasted volatility directly would be an option worth considering in the future in order to get better estimated result.
3. As we demonstrated in the project empirical evidence has shown there does not exsit a standard model that can help us straightly get the answer. It is basically a trial and error problem sovling process until the most optimal outcome is acheived. We oftentimes need to break down the market based on the market trends and give analysis to every each of them because a model comes in handy in one season may perform poorly in the other.