Empirical Finance: Assignment 2

By Group 15

Lanlan Hou	2801069
Rick van der Ploeg	2782774
Sebastiaan van Dijen	2769505
Wa Chak Sou	2796840

School of Business and Economics

10th November 2025

QUESTION 1: Stock Prices, Returns, and Characteristics

a. Autocorrelation function

Let $r_t = \ln(P_t/P_{t-1})$ denote the daily log return of HSBC over the sample November 2015 – November 2020. The (sample) autocorrelation at lag k is

$$\rho(k) = \frac{\sum_{t=k+1}^{T}(r_t - \bar{r})(r_{t-k} - \bar{r})}{\sum_{t=1}^{T}(r_t - \bar{r})^2},$$

and its squared–return analogue uses r_t^2 in place of r_t to probe dependence in volatility.

Figure 1.1.1 and 1.1.2 report the ACFs for the raw and squared log returns, respectively, with 95% confidence bands (approximately $\pm 1.96/\sqrt{T}$ under the null of no autocorrelation).

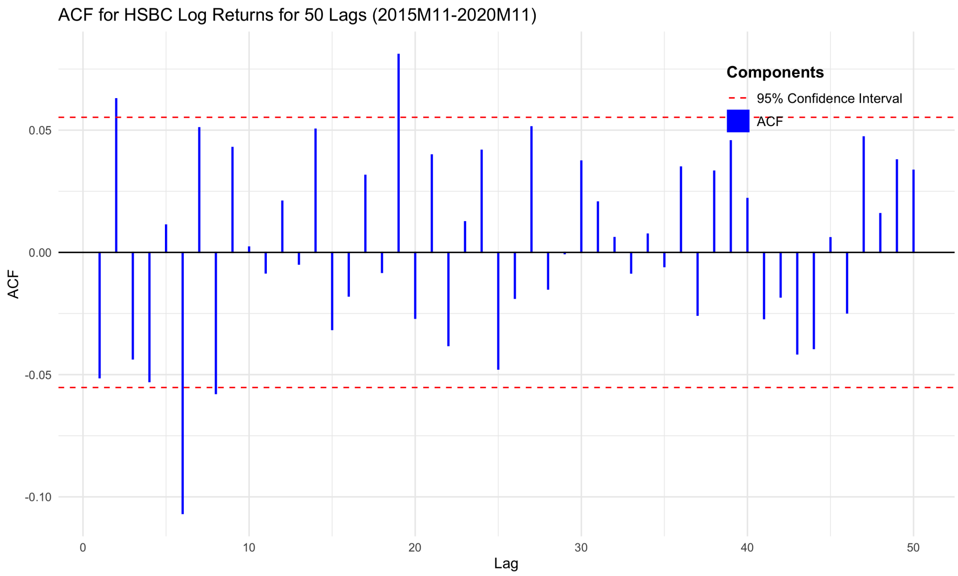


Figure 1.1.1 ACF for HSBC Log Returns for 50 lags (November 2015 to November 2020)

Returns

The ACF of r_t shows most spikes within the confidence bands, with only a few isolated lags approaching or barely exceeding the threshold. This pattern is consistent with the common stylized fact that daily equity returns have weak (if any) linear serial correlation.

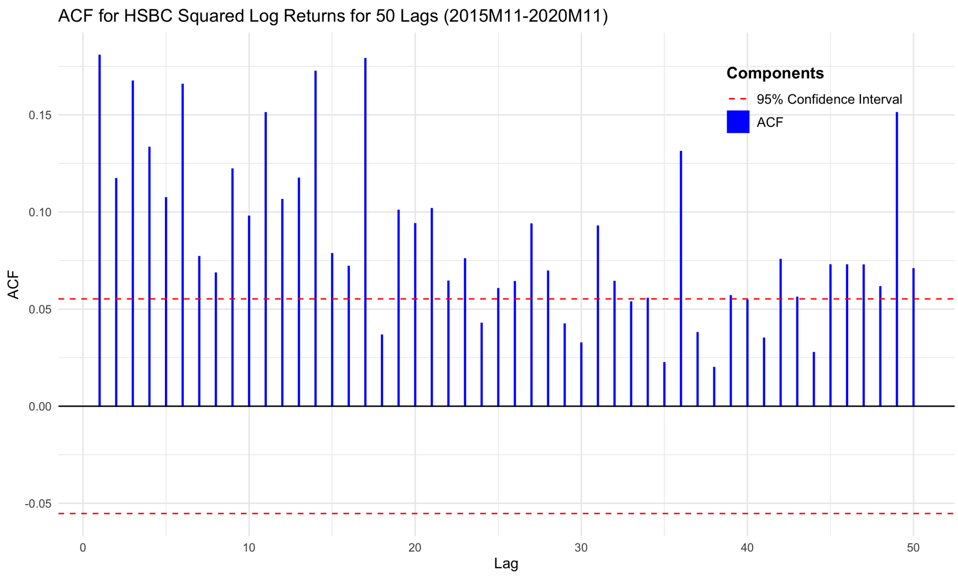


Figure 1.1.2 ACF for HSBC Squared Log Returns for 50 lags (November 2015 to November 2020)

Squared Returns

In contrast, the ACF of r_t^2 displays persistent, statistically significant autocorrelation over many lags. This indicates volatility clustering: large price movements tend to be followed by large movements (of either sign), and small by small. Such long-memory in the second moment violates the i.i.d. assumption and motivates conditional heteroskedasticity models.

b. Test for joint autocorrelation

Lags	Statistic	Critical Value (5%)	p-Value
10	38.6530	18.3070	2.920×10^{-5}
20	54.8994	31.4104	4.251×10^{-5}
30	70.0919	43.7730	4.719×10^{-5}
40	78.0982	55.7585	2.945×10^{-4}
50	91.1965	67.5048	3.348×10^{-4}

Table 1.1.1: Ljung-Box Test Results for HSBC Log Returns (November 2015 to November 2020)

Lags	Statistic	Critical Value (5%)	p-Value
10	210.0726	18.3070	0.000
20	390.2060	31.4104	0.000
30	449.6853	43.7730	0.000
40	507.1996	55.7585	0.000
50	583.4937	67.5048	0.000

Table 1.2.1: Ljung-Box Test Results for HSBC Squared Log Returns (November 2015 to November 2020)

Table 1.1.1 and 1.2.1 report the Ljung-Box test statistics for the HSBC log returns and squared returns over the period November 2015 to November 2020, using lag lengths from 10 to 50.

For the log returns (Table 1.1.1), all test statistics exceed their 5% critical values, and the associated p-values are below 0.05 for all lag lengths. This indicates significant joint autocorrelation in the return series. For the squared log returns (Table 1.2.1), the Ljung-Box statistics are extremely large, with p-values almost equal to zero for all lag lengths choices. This rejects the null hypothesis of no autocorrelation, shows clearly of volatility clustering, means that periods of high volatility tend to be followed by high volatility and same for the opposite.

Question 2: Return distributions

a. MA and EWMA volatilities

With the assumption that $\mu = 0$ which is common for daily returns, the moving average (MA) volatility is computed as the average of the past W squared observations:

$$\sigma_{t+1}^2 = \frac{1}{W-1}\sum_{w=0}^{W-1} y_{t-w}^2,$$

where W denotes the window size used to compute the MA volatitlity. The exponentially weighted moving average (EWMA) volatility is computed recursively as

$$\sigma_{t+1}^2 = (1-\lambda)y_t^2 + \lambda\sigma_t^2,$$

where $\lambda \in (0,1)$ is the de-facto weight parameter controlling the rate of decay.

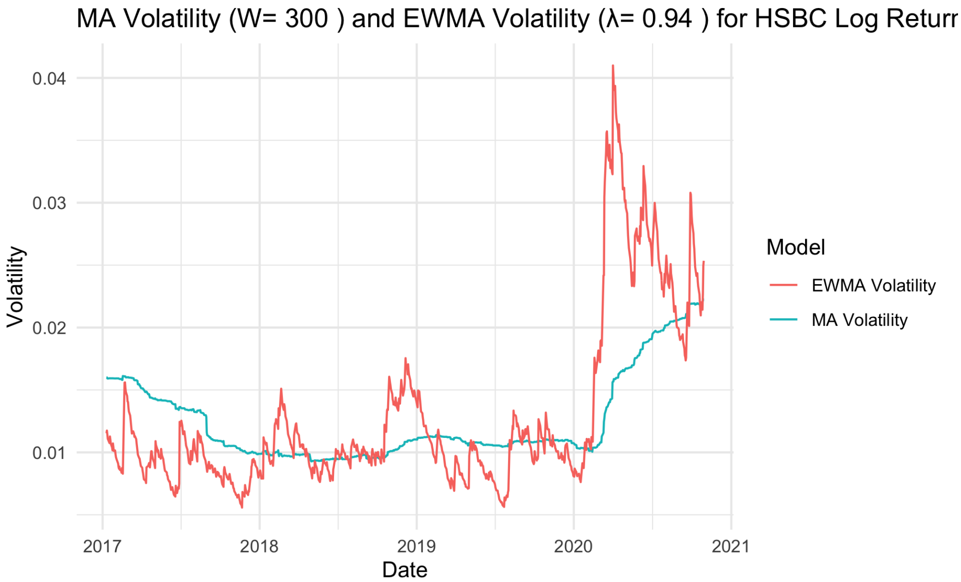
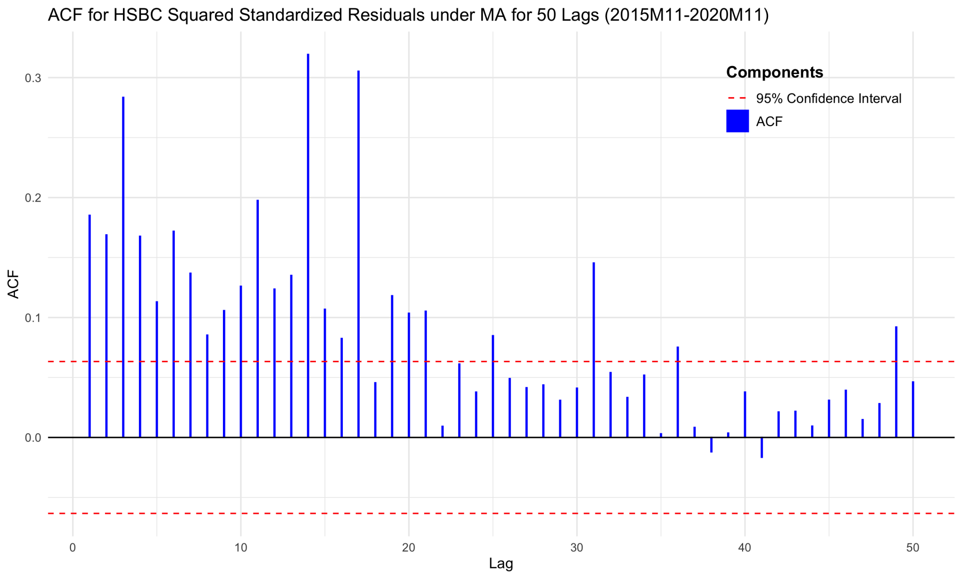


Figure 2.1: HSBC MA vs. EWMA volatility

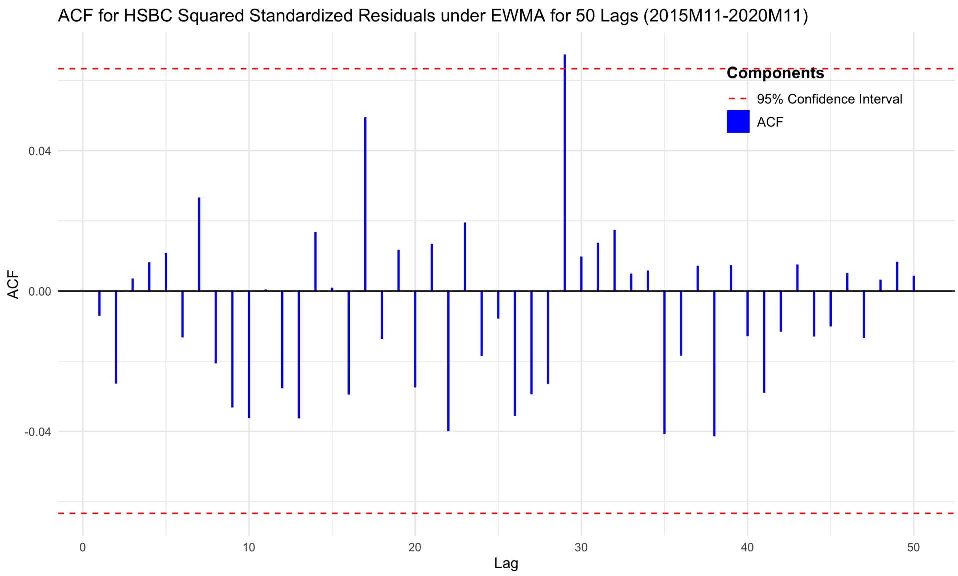
To obtain the Moving Average(MA) volatility time series, an estimation window of $W = 300$ days is being used. From Figure 2.1 we can see that the MA ($W = 300$) volatility captures all volatility but react slightly slower. Furthermore, to obtain the EWMA volatility time series , the *de-facto* weight (decay parameter) of $\lambda = 0.94$ is being applied. In contrast, the EWMA ($\lambda = 0.94$) volatility adjusts more quickly to sudden shocks, it is more sensitive to reflect market conditions. In conclusion, the EWMA model is preferable for modeling time-varying volatility.

b. Normality test of returns

i. Autocorrelation (ACF)



(a): Under MA Volatility



(b) Under EWMA Volatility

Figure 2.2.1: ACF of squared standardized residuals under different volatility models

ii. Ljung-Box (LB) test

To check the autocorrelation remains in the standardized residuals, the **Ljung-Box (LB)** test was performed. The hypotheses for the test are formulated as follows:

$$H_0 : \text{There is no autocorrelation up to lag } h,$$
$$H_1 : \text{There is autocorrelation up to lag } h.$$

The Ljung–Box test statistic is given by

$$LB_N = T(T+2)\sum_{n=1}^{N} \frac{\hat{\beta}_n^2}{T-n},$$

where

- T: the size of squared standardized residuals,
- N: the number of lags tested,
- $\hat{\rho}_k$: the significance of autocorrelation coefficients.

Under the null hypothesis of no autocorrelation, the test statistic is asymptotically chi-squared distributed with h degrees of freedom:

$$LB_N \xrightarrow{d} \chi^2_{(N)} \quad \text{as} \quad T \to \infty.$$

LB test is typically applied to the **squared standardized residuals** to detect any remaining volatility clustering. A rejection of H_0 implies that the model fails to fully capture the conditional heteroskedasticity in the data.

Lags	**Critical Value (5%)**	**MA Volatility**		**EWMA Volatility**	
		Test Stat.	p-Value	Test Stat.	p-Value
10	18.3070	257.1308	0.0000	3.5353	0.9659
20	31.4104	562.0435	0.0000	9.6379	0.9743
30	43.7730	593.7460	0.0000	17.4696	0.9665
40	55.7585	628.7140	0.0000	23.6886	0.9811
50	67.5048	644.2839	0.0000	26.2866	0.9977

Table 2.2.1: Ljung-Box test results for Squared Standardized Residuals under MA and EWMA volatility Models

Table 2.2.1 shows the Ljung-Box test results of the MA($W = 300$) and EWMA ($\lambda = 0.94$) volatility models. MA volatility results extremely large test statistics with small p-values while the EWMA volatility results very small test statistics with large p-values. This result suggests that MA model produces highly autocorrelated squared standardized residuals whereas no remaining autocorrelation in EWMA model. This may imply that the EWMA model captures conditional heteroskedasticity effectively since it reacts more rapidly to new information, while the volatility clustering is still hidden in the MA model.

iii. Jarque-Bera (JB) test

To formally test for normality, the **Jarque-Bera (JB)** test was conducted. The hypotheses for the test are formulated as follows:

$$H_0 : \text{The returns are normally distributed,}$$
$$H_1 : \text{The returns are not normally distributed.}$$

The Jarque-Bera (JB) test statistic, based on the least squares residuals, is given by

$$JB = \frac{T-k}{6}(\hat{S}^2 + \frac{1}{4}(\hat{K}-3)^2),$$

where

- k: the number of covariates (regressors),
- T: the sample size,
- $\hat{S}$: the sample skewness,
- $\hat{K}$: the sample kurtosis.

Under the null hypothesis of normality, the test statistic is asymptotically chi-squared with two degrees of freedom:

$$JB \xrightarrow{d} \chi^2_2 \quad \text{as} \quad T \to \infty.$$

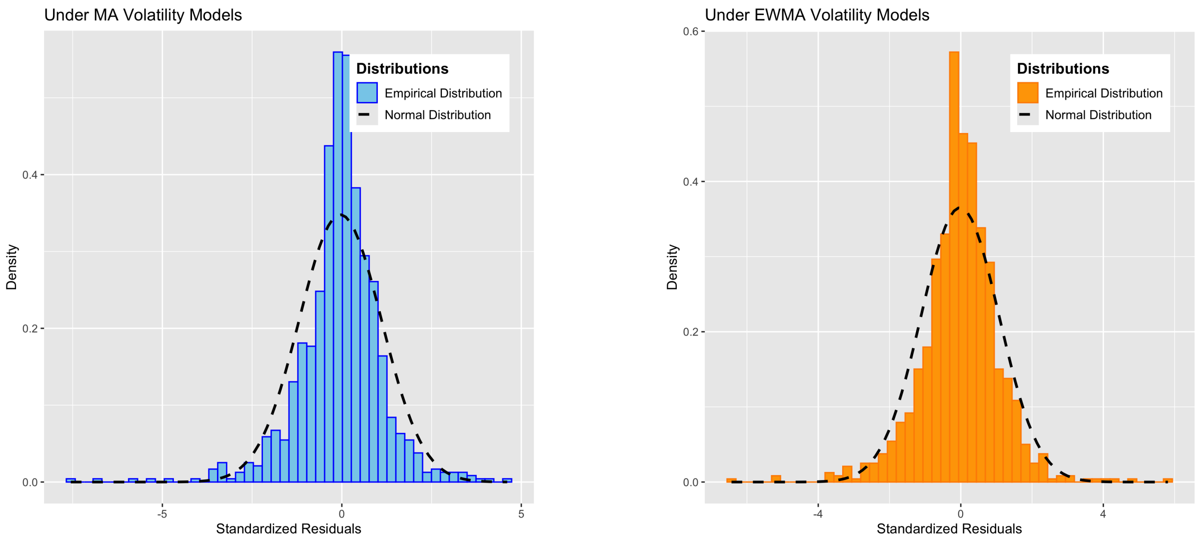
Model	Test Statistic	Critical Value (5%)	p-Value
MA	1249	5.991	$< 2.2 \times 10^{-16}$
EWMA	662.59	5.991	$< 2.2 \times 10^{-16}$

Table 2.2.2: Jarque-Bera (JB) test results for standardized residuals under the MA and EWMA models

Table 2.2.2 reveals that the standardized residuals of both MA and EWMA volatility models result extremely large Jarque–Bera statistics with p-values close to 0 in the Jarque-Bera Test. It indicates the significant deviation of normality and reject the null of the residuals comes from a normal distribution.

iv. Distributions of Residuals

Figure 2.2.2 reveal the distributions of the standardized residuals under different volatility models. The histograms reveal leptokurtic shapes with fat tails. Combining wth the tests results above, these suggest that while MA and EWMA capture conditional heteroskedasticity, they fail to fully model the heavy-tailed nature of financial returns.



(a) MA Volatility

(b) EWMA Volatility

Figure 2.2.2: Distributions of Standardized Residuals of HSBC Log Returns under Different Conditional Volatility Models

Appendix

```
library(openxlsx)
raw <- read.xlsx('student_groups_stocks.xlsx', sheet = 1)

groupNumber <- 15
nameOfStock <- raw$Stock.Name[groupNumber]
startDate <- raw$Start.Date[groupNumber]
endDate <- raw$`End.Date.(+10y)`[groupNumber]

# Compute the start date of the final five years
library(lubridate)

originalDate <- as.Date(startDate)
finalFiveYearsStartDate <- originalDate %m+% years(5)

# Extract the daily stock price from source
library(quantmod)

getSymbols(nameOfStock, src = 'yahoo', from = finalFiveYearsStartDate,
    to = endDate)
date <- index(HSBC)

# Adjusted Close
adjustedPrice <- as.numeric(HSBC[,'HSBC.Adjusted'])

# Log Returns
compoundedReturns <- numeric(length(adjustedPrice) - 1)
for(i in (2:length(adjustedPrice))){
  compoundedReturns[i-1] <- log(adjustedPrice[i] / adjustedPrice[i-1])
}
# Squared Log Returns
squaredCompoundedReturns <- compoundedReturns ^ 2

# Function to compute the ACF and plot the graph
```

```
plotACF <- function(input_list, input_maxLags, objectName, savePath) {
  ACFResult <- acf(input_list, lag.max = input_maxLags, plot = FALSE)

  ACFValues <- ACFResult$acf[-1]
  lags <- ACFResult$lag[-1]
  numberOfObservations <- length(input_list)
  confidenceBands <- 1.96 * 1 / sqrt(length(input_list))

  library(ggplot2)
  plot_df <- data.frame(lag = lags, acf = ACFValues)

  ggplot(plot_df, aes(x = lag, y = acf)) +
    geom_bar(aes(color = 'ACF'), stat = 'identity',
             fill ='blue', width = 0.05) +
    geom_hline(yintercept = 0, color = 'black') +
    geom_hline(aes(yintercept = -confidenceBands,
                   color = '95% Confidence Interval'), linetype = 'dashed')
                       +
    geom_hline(aes(yintercept = confidenceBands,
                   color = '95% Confidence Interval'), linetype = 'dashed')
                       +
    scale_color_manual(name = 'Components',
                       values = c('ACF' = 'blue',
                                  '95% Confidence Interval' = 'red')) +
    labs(title = paste('ACF for HSBC', objectName, 'for', input_maxLags
        , 'Lags (2015M11-2020M11)'),
         x = 'Lag', y = 'ACF') +
    theme_minimal() +
    theme(legend.position = c(0.95, 0.95),
          legend.justification = c("right", "top"),
          legend.title = element_text(size = 12, face = "bold"),
          legend.text = element_text(size = 10))
  ggsave(savePath, width = 10, height = 6, dpi = 300)
}

returns_result <- plotACF(compoundedReturns,
                          input_maxLags = 50,
                          objectName = 'Log Returns',
                          savePath = 'figures/returns_acf_plot.png')

squaredReturns_result <- plotACF(squaredCompoundedReturns,
                                 input_maxLags = 50,
                                 objectName = 'Squared Log Returns',
                                 savePath = 'figures/squaredReturns_acf
                                    _plot.png')
```

Figure 1: Question 1(a)

```
# Function to perform the Ljung-Box test and report the result
LBTest <- function(input_series, input_maxLags, steps){
  steppedLags <- seq(steps, input_maxLags, by = steps)

  result_df <- data.frame(Lag = integer(),
                          Statistic = numeric(),
                          Crit_Value = numeric(),
                          P_Value = numeric())

  for (lag in steppedLags) {
    lb_test <- Box.test(input_series, lag = lag, type = 'Ljung-Box')
```

```
    testStatistic <- as.numeric(lb_test$statistic)
    pValue <- as.numeric(lb_test$p.value)
    criticalValue <- qchisq(0.95, df = lag)

    newRow <- data.frame(Lag = lag,
                         Statistic = round(testStatistic, 4),
                         Critical_Value = round(criticalValue, 4),
                         P_Value = signif(pValue, 4))

    result_df <- rbind(result_df,newRow)
  }
  return(result_df)
}

returns_LBTest <- LBTest(compoundedReturns, 50, 10)
returns_LBTest
returns_LBTest <- LBTest(squaredCompoundedReturns, 50, 10)
returns_LBTest
```

Figure 2: Question 1(b)

```
# compute the MA volatility
window <- 300

hat_y <- numeric(length(compoundedReturns)-window)
MA_volatility <- numeric(length(compoundedReturns)-window)

for (t in window:length(compoundedReturns)){
  hat_y[t-window+1] <- mean(compoundedReturns[(t-window+1):(t)])
  MA_volatility[t-window+1] <- sqrt(1/(window-1) * sum(sapply(0:(window-1),
                                    function(j) (compoundedReturns[t-j])^2)))
}

# compute the EWMA volatility
lambda <- 0.94

EWMA_volatility <- numeric(length(compoundedReturns)+1)

for (t in 1:length(compoundedReturns)){
  EWMA_volatility[t+1] <- sqrt((1 - lambda) * (compoundedReturns[t]^2)
                          + lambda * EWMA_volatility[t]^2)
}

MA_df <- data.frame(MA_vol = MA_volatility,
                    EWMA_vol = EWMA_volatility[window:length(compoundedReturns)],
                    date = date[window:length(compoundedReturns)])

ggplot(MA_df, aes(x = date)) +
  geom_line(aes(y = MA_vol, color = 'MA Volatility')) +
  geom_line(aes(y = EWMA_vol, color = 'EWMA Volatility')) +
  labs(title = paste('MA Volatility (W=', window,
                     ') and EWMA Volatility (  =', lambda,
                     ') for HSBC Log Returns (2015M11-2020M11)'),
       x = 'Date', y = 'Volatility', color = 'Model') +
  theme_minimal()
ggsave('figures/ma_plot.png', dpi = 300)
```

Figure 3: Question 2(a)

```r
# calculate standrized residuals
residual <- (compoundedReturns[(window+1):length(compoundedReturns)])
residual_MA <- residual / MA_volatility[1:(length(hat_y)-1)]
residual_EWMA <- residual / EWMA_volatility[(window+1):(length(compoundedReturns))]

# residual autocorrelation checking
MA_result <- plotACF(residual_MA^2,
                     input_maxLags = 50,
                     objectName = 'Squared Standardized Residuals under MA',
                     savePath = 'figures/residuals_MA_acf_plot.png')
EWMA_result <- plotACF(residual_EWMA^2,
                       input_maxLags = 50,
                       objectName = 'Squared Standardized Residuals under EWMA',
                       savePath = 'figures/residuals_EWMA_acf_plot.png')

# Distribution of Standardized Residuals
residual_df <- data.frame(MA_vol =  residual_MA,
                          EWMA_vol = residual_EWMA)

ggplot(residual_df, aes(x = MA_vol)) +
  geom_histogram(aes(y = ..density.., color = 'Empirical Distribution'),
                 bins = 50, fill = 'skyblue') +
  stat_function(fun = dnorm,
                args = list(mean = mean(residual_MA, na.rm = TRUE),
                            sd = sd(residual_MA, na.rm = TRUE)),
                aes(color = 'Normal Distribution'), linewidth = 1, linetype = 'dashed') +
  scale_color_manual(name = "Distributions",
                     values = c('Normal Distribution' = 'black',
                                'Empirical Distribution' = 'blue')) +
  labs(title = 'Under MA Volatility Models',
       x = 'Standardized Residuals', y = 'Density') +
  theme(legend.position = c(0.95, 0.95),
        legend.justification = c("right", "top"),
        legend.title = element_text(size = 12, face = "bold"),
        legend.text = element_text(size = 10))
ggsave('figures/MA_residual_distribution.png', width = 6, height = 6, dpi = 300)

ggplot(residual_df, aes(x = EWMA_vol)) +
  geom_histogram(aes(y = ..density.., color = 'Empirical Distribution'),
                 bins = 50, fill = 'orange') +
  stat_function(fun = dnorm,
                args = list(mean = mean(residual_EWMA, na.rm = TRUE),
                            sd = sd(residual_EWMA, na.rm = TRUE)),
                aes(color = 'Normal Distribution'), linewidth = 1, linetype = 'dashed') +
  scale_color_manual(name = "Distributions",
                     values = c('Normal Distribution' = 'black',
```

```
                                      'Empirical Distribution' = 'darkorange'))+
  labs(title = 'Under EWMA Volatility Models',
       x = 'Standardized Residuals', y = 'Density') +
  theme(legend.position = c(0.95, 0.95),
          legend.justification = c("right", "top"),
          legend.title = element_text(size = 12, face = "bold"),
          legend.text = element_text(size = 10))
ggsave('figures/EWMA_residual_distribution.png', width = 6, height = 6, dpi = 300)

# Ljung-Box Test
MA_LBTest <- LBTest((residual_MA^2), 50, 10)
MA_LBTest
EWMA_LBTest <- LBTest((residual_EWMA^2), 50, 10)
EWMA_LBTest

library('tseries')

# Jarque-Bera Test
MA_JBTest <- jarque.bera.test(residual_MA)
EWMA_JBTest <- jarque.bera.test(residual_EWMA)

JBTest_result_df <- data.frame(Model = character(),
                                Statistic = numeric(),
                                Crit_Value = numeric(),
                                P_Value = numeric())
MA_newRow <- data.frame(Model = 'MA',
                        Statistic = round(MA_JBTest$statistic, 4),
                        Critical_Value = round(qchisq(0.95, df = MA_JBTest$parameter), 4),
                        P_Value = signif(MA_JBTest$p.value, 4))
EWMA_newRow <- data.frame(Model = 'EWMA',
                        Statistic = round(EWMA_JBTest$statistic, 4),
                        Critical_Value = round(qchisq(0.95, df = EWMA_JBTest$parameter), 4),
                        P_Value = signif(EWMA_JBTest$p.value, 4))

JBTest_result_df <- rbind(JBTest_result_df,MA_newRow)
JBTest_result_df <- rbind(JBTest_result_df,EWMA_newRow)
JBTest_result_df
```

Figure 4: Question 2(b)