

Clarity Kummer; 461 Final

Due 2024-12-14

Original Time Series Observations and Comments

Shown below in “**Walmart (WMT): Adjusted Close Prices (01/01/2020 - 12/06/2023)**” is a time series plot of Walmart’s daily adjusted closing stock price from January 1, 2020, to December 6, 2023. The plot reveals a general upward trend, indicating a **non-constant mean** that increases over time. However, during **January 2021 through the first few months of 2022**, the series remains relatively stagnant, neither substantially increasing nor decreasing. Just prior to **July 2022**, the series peaked at approximately 52 before sharply declining to around 38. This decline marks the lowest value the series reaches after 2020 and represents the most dramatic, short-term fluctuation observed during the time frame. Following this sharp decline, the series demonstrates a strong and consistent upward trend, which persists throughout the remainder of the period. The series demonstrates volatility cluster, where significant changes in the series are followed by substantial changes of either sign, and small changes tend to be followed by small changes of either sign. Two notable periods of volatility clustering are **between January 2020 and July 2020**, and between **January 2022 and July 2022**.



Stationary Tests On Original Data

To confirm that the original data series is in fact non-stationary, I conducted an ADF Test and a KPSS Test. For a stationary time series, an ADF Test will compute a p-value **less than** a standard significance level of 0.05. For a stationary time series, a KPSS Test will compute a p-value **greater than** a standard significance level of 0.05. **As shown in Appendix 1:** the ADF test computed a p-value of 0.07851, the KPSS test computed a p-value of 0.01. Thus, both test confirm that original series is in fact, not stationary.

Transforming Original Series Into Stationary Series

Before modeling the series or proceeding with the current analysis, it is essential to transform the series to achieve stationarity. To do this, we first take the logarithm of the Walmart daily adjusted closing stock price and then apply differencing to the series. Differencing can be repeated as many times as necessary to achieve stationarity. As demonstrated in the “**WMT: Log Return (d=1)**” plot below, after taking the log of the data and taking the first difference, the series no longer exhibits a general upward trend and appears stabilized around

zero. **As shown in Appendix 2:** the ADF and KPSS test reveal the series is stationary under these transformations, yielding a p-value of 0.01 and 0.10. However, it is evident that this transformation did not fully stabilize the variance of the series. While the mean is now stationary, the series remains highly variable as there are peaks of volatility ever evident in **“WMT: Log Return (d=1)”**.



Identifying orders p and q of ARIMA

Now that we have achieved stationarity, the next step is to identify a suitable model to describe the series. The model identification process begins by determining the orders of the ARIMA(p,d,q) model, which I tentatively assume describes the series. An ARIMA(p,d,q) model is a generalized ARMA(p,q) model for non-stationary data. Once an ARIMA(p,d,q) process is differenced 'd' times to achieve stationarity, the model becomes an ARMA(p,q) process. Since we have established above that the first difference of the series (d=1) achieves stationarity, the current objective is to determine the appropriate order of the stationary ARMA(p,q) process that best fits the transformed series. **As shown in Appendix 3:** the ACF and PACF plots both demonstrate that for multiple lags there are significant auto-correlations. The ACF plot, could be taken to suggest an MA(1) model, however neither plot demonstrates a clear cut off in significant auto-correlations beyond a certain lag. Thus, it is difficult to use these plots as metrics for choosing the orders 'p' and/ or 'q' for the series. Additionally, neither plot shows a clear "tail off" pattern. Thus, I will use the auto.arima() function to determine a tentative model for the given series.

ARMA(0,1)

The auto.arima() function suggested an MA(1) model for the data. The full model output summary is provided in **Appendix 4**.

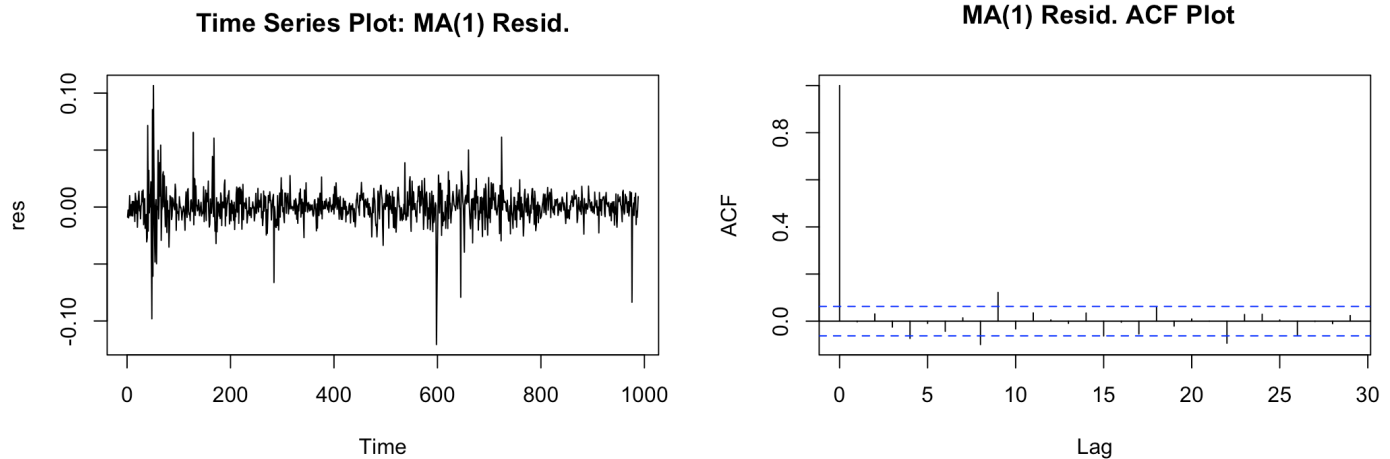
The estimated parameters for an MA(1) Model provided by auto.arima():

$$\hat{\theta} = -0.0676$$

$$SE(\hat{\theta}) = 0.0309$$

Additionally, the AIC and BIC were computed as -5498.71 and -5488.91, respectively. The time series plot of the residuals from this MA(1) model are shown below in **“Time Series Plot: MA(1) Resid.”** The residuals, demonstrate some strong variation, yet, for the most part appear to be centered around 0. Additionally, there do not appear to be any significant trends. Conducting a Box-Ljung test on the residuals from the fitted MA(1) model yielded a p-value of 0.95, indicating that the residuals resemble weak white noise. However, the residuals do not exhibit strong white noise, as expected under a well-specified and efficiently estimated model. This issue is

evident in the residual ACF plot, " **MA(1) Resid. ACF Plot**" shown below. The ACF plots of the residuals reveals significant autocorrelations at multiple lags. This behavior is also present in the ACF plots of the squared residuals and the absolute residuals, (shown in **Appendix 5**) suggesting that the residuals exhibit patterns inconsistent with strong white noise. These findings indicate that while the MA(1) model may adequately capture the mean structure of the series, it fails to account for volatility clustering or heteroskedasticity in the data. Therefore, to model the residual structure more effectively, we need to consider fitting a GARCH(m,s) model.



ARMA(0,1)+GARCH(1,1)

Fitting a GARCH(m,s) model will provide a means to capture the conditional heteroskedasticity of the errors evident in the ACF plots of the MA(1) model. I first consider an ARMA(0,1)+GARCH(1,1) model. A GARCH(1,1) model serves as a good baseline, and the ARMA(0,1) was shown to be a good model to capture the mean function.

After fitting an ARMA(0,1)+GARCH(1,1), the estimated coefficients are as follows:

$$\begin{aligned}\hat{\mu} &= 2.100e - 04; SE(\hat{\mu}) = 3.815e - 04 \\ \hat{\theta}_1 &= -4.778e - 02; SE(\hat{\theta}_1) = 3.790e - 02 \\ \hat{\alpha}_0 &= 1.975e - 05; SE(\hat{\alpha}_0) = 5.859e - 06 \\ \hat{\alpha}_1 &= 9.948e - 02; SE(\hat{\alpha}_1) = 2.179e - 02 \\ \hat{\beta}_1 &= 8.071e - 01; SE(\hat{\beta}_1) = 4.269e - 02\end{aligned}$$

Given the values of the estimated parameters, we may first observe that $\hat{\mu}$ is insignificantly different from 0. We expect and assume that the mean of the model is centered around 0, so this estimate is in line and confirms / validates such assumptions. The estimate for $\hat{\theta}_1$ is insignificantly different from 0. Thus, this seems to be an unnecessary parameter in the model, and serves to suggest fitting a GARCH(1,1) without the additional MA(1) model. $\hat{\alpha}_0, \hat{\alpha}_1, \hat{\beta}_1$ are all significant parameters in the given model. The ARMA(0,1)+GARCH(1,1) provided an AIC and BIC of -5.772180 and -5.747404, respectively. **Shown in Appendix 6** are the ACF plots of the residuals. Under all residual transformations, the residuals appear to behave like strong white noise. That is, the residuals demonstrate no significant auto-correlations for all lags k. The model, however, severely deviates from a normal distribution, as evident in the QQ-Plot (**See Appendix 6**), and by a Kolmogorov-Smirnov (KS) test providing a p-value = 0.0001335. However, before continuing with the ARMA(0,1)+GARCH(1,1) discussion, I consider the effects of fitting just a GARCH(1,1) model. (**Please see Appendix 6 for detailed output / plots for this model**)

Because $\hat{\theta}_1$ was insignificantly different from 0, the inclusion of the ARMA(p,q) model is suggests insignificance. Thus, I proceed to fit just a GARCH(1,1) model and discuss the results below.

GRACH(1,1)

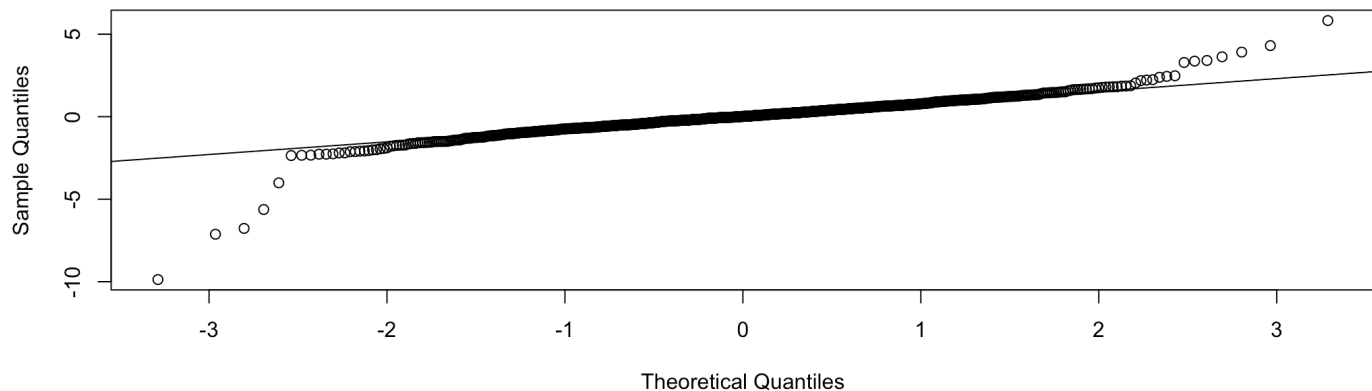
After fitting a GARCH(1,1) model to the data, the estimated parameters are:

$$\begin{aligned}\hat{\mu} &= 1.945e - 04; SE(\hat{\mu}) = 3.998e - 04 \\ \hat{\alpha}_0 &= 2.023e - 05; SE(\hat{\alpha}_0) = 6.026e - 06 \\ \hat{\alpha}_1 &= 9.928e - 02; SE(\hat{\alpha}_1) = 2.194e - 02 \\ \hat{\beta}_1 &= 8.047e - 01; SE(\hat{\beta}_1) = 4.381e - 02\end{aligned}$$

Based on these observations, it is evident that all parameters in the GARCH(1,1) model are significant.

The estimated parameters $\hat{\alpha}_0, \hat{\alpha}_1, \hat{\beta}_1$, which were significant in the ARMA(0,1)+GARCH(1,1) model, remain significant in the GARCH(1,1) model. Furthermore, their values are nearly identical in both models. Thus, because of its parsimony the GARCH(1,1) suggesting a better choice model. The AIC and BIC values for both models, ARMA(0,1)+GARCH(1,1) and GARCH(1,1), are essentially the same, further supporting the insignificance of the ARMA(0,1) component. Thus, the GARCH(1,1) model is more appropriate for describing the data. Additionally, the residuals of the GARCH(1,1) model still demonstrate the behavior of strong white noise, which is expected for an adequately specified and efficiently estimated model (**Shown in Appendix 7**). However, the QQ-plot, shown below in “**GARCH(1,1) QQ-Plot**”, reveals that the residuals still do not follow a normal distribution. While the GARCH(1,1) model adequately captures the mean and variance structure of the series, it fails to account for the true underlying distribution of the data. To address this, we need to refit the model using an appropriate error distribution that better reflects the data’s behavior. (**Please see Appendix 7 for detailed output from the GARCH(1,1) Model**)

GARCH(1,1) QQ-Plot



GARCH(1,1) with Conditional Distribution.

Considering a conditional distribution, specifically, a standardized student t distribution for the distribution of the errors significantly improved the model.

The Estimated Parameters for the GARCH(1,1) model with standardized student t distribution:

$$\hat{\mu} = 4.760e - 04; SE(\hat{\mu}) = 3.198e - 04$$

$$\hat{\alpha}_0 = 1.635e - 05; SE(\hat{\alpha}_0) = 6.616e - 06$$

$$\hat{\alpha}_1 = 1.332e - 01; SE(\hat{\alpha}_1) = 4.542e - 02$$

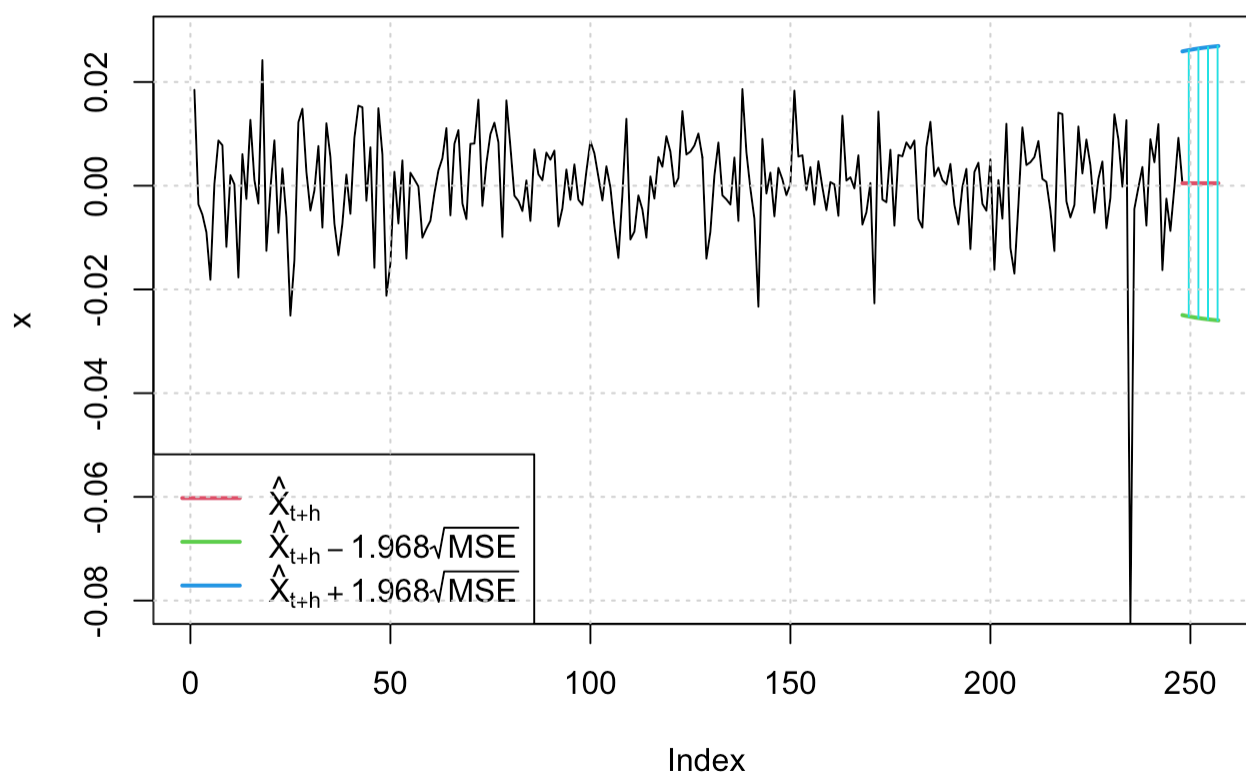
$$\hat{\beta}_1 = 7.815e - 01; SE(\hat{\beta}_1) = 6.624e - 02$$

To evaluate the accuracy of a **standardized Student's t-distribution** in modeling the residuals, I generated a random sample from a standardized Student's t-distribution. I then compared this sample with the residuals' distribution using the **Kolmogorov-Smirnov (KS) test**. The resulting **p-value** was 0.5806, which is greater than **0.05**, indicating that the residuals' distribution and the standardized Student's t-distribution are statistically similar. When viewing the output summary for the model, the residual tests nearly all provide p-values > 0.05. Thus, providing evidence that the model is adequate and the residuals are as we expect them to be; strong white noise, following a standardized student's t-distribution. The full model outputs are included in **Appendix 8**. Additionally, when adjusting the residual distribution, the AIC value and BIC values decreased to -6.042261 and -6.017485, respectively. Thus, indicating that considering this conditional distribution is beneficial and leads to a model that is better able to fit the data. Further, the residuals still behave like strong white noise, as demonstrated by the lack of significant auto-correlations in the ACF plots under all transformations. The ACF plots are shown in **Appendix 8**. We conclude that the GARCH(1,1) model with a standardized Student's t-distribution provides the best fit for the data.

Forecasting

Below is an image of the 10-steps ahead forecast. Such forecast demonstrate the accuracy of the GARCH(1,1) model with a standardized student's t-distribution. The detailed forecast values are found in **Appendix 9**.

Prediction with confidence intervals



Appendix

Appendix 1

Stationary Test Results

Test	P.Value
ADF Test	0.0785144
KPSS Test	0.0100000

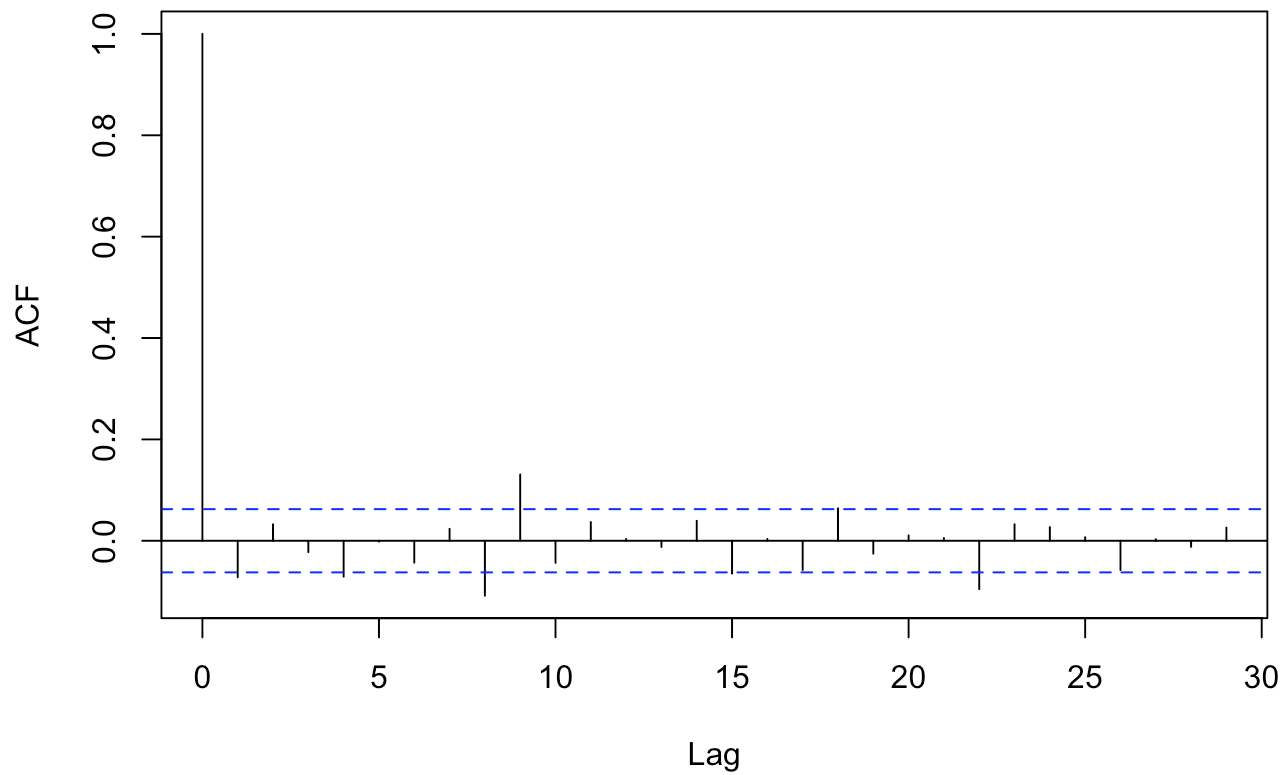
Appendix 2

Stationary Test Results Differenced Log Returns (d=1)

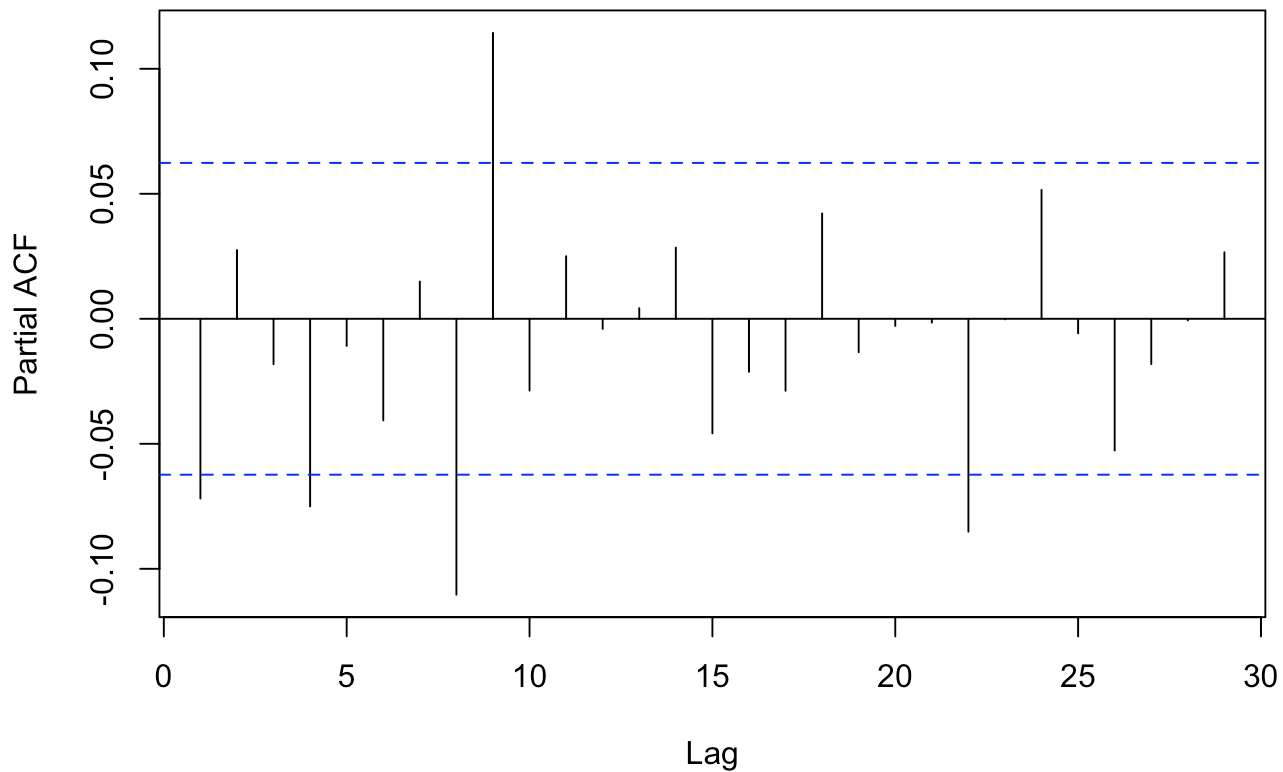
Test	P.Value
ADF Test	0.01
KPSS Test	0.10

Appendix 3

ACF Plot, Differenced Log Returns



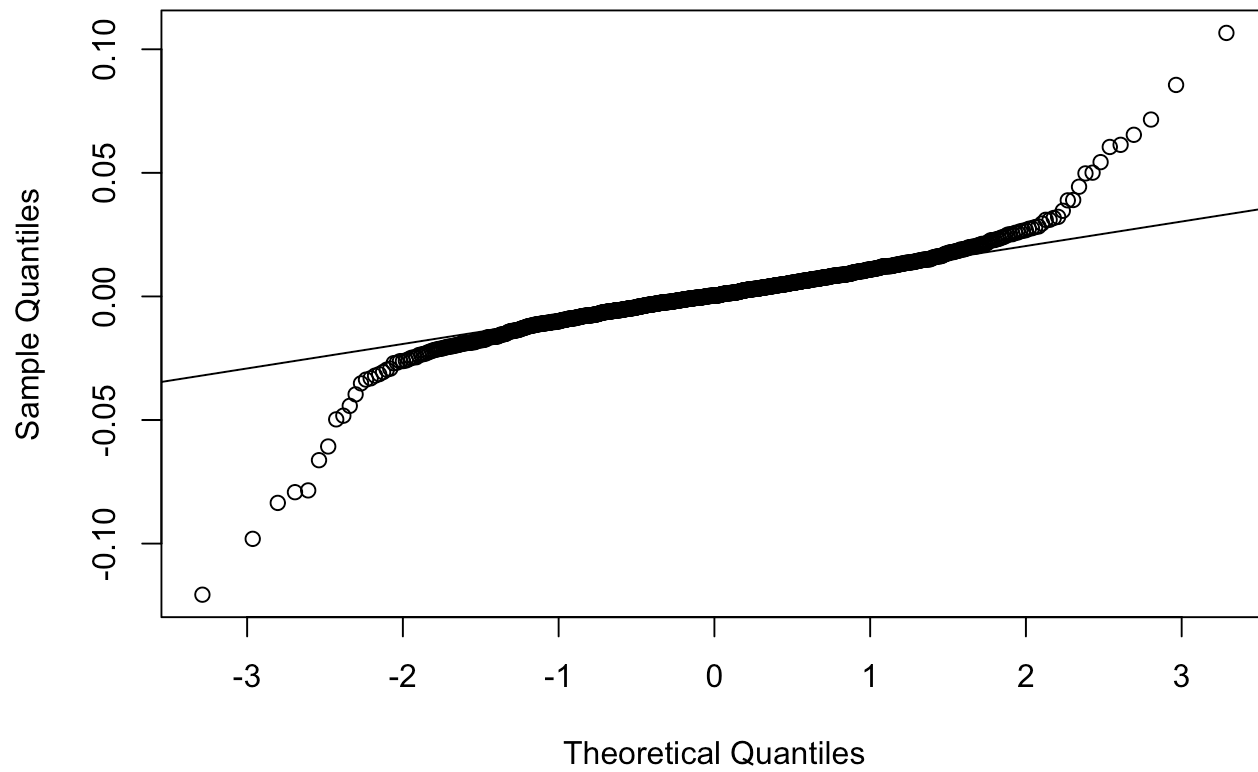
PACF Plot, Differenced Log Returns



Appendix 4

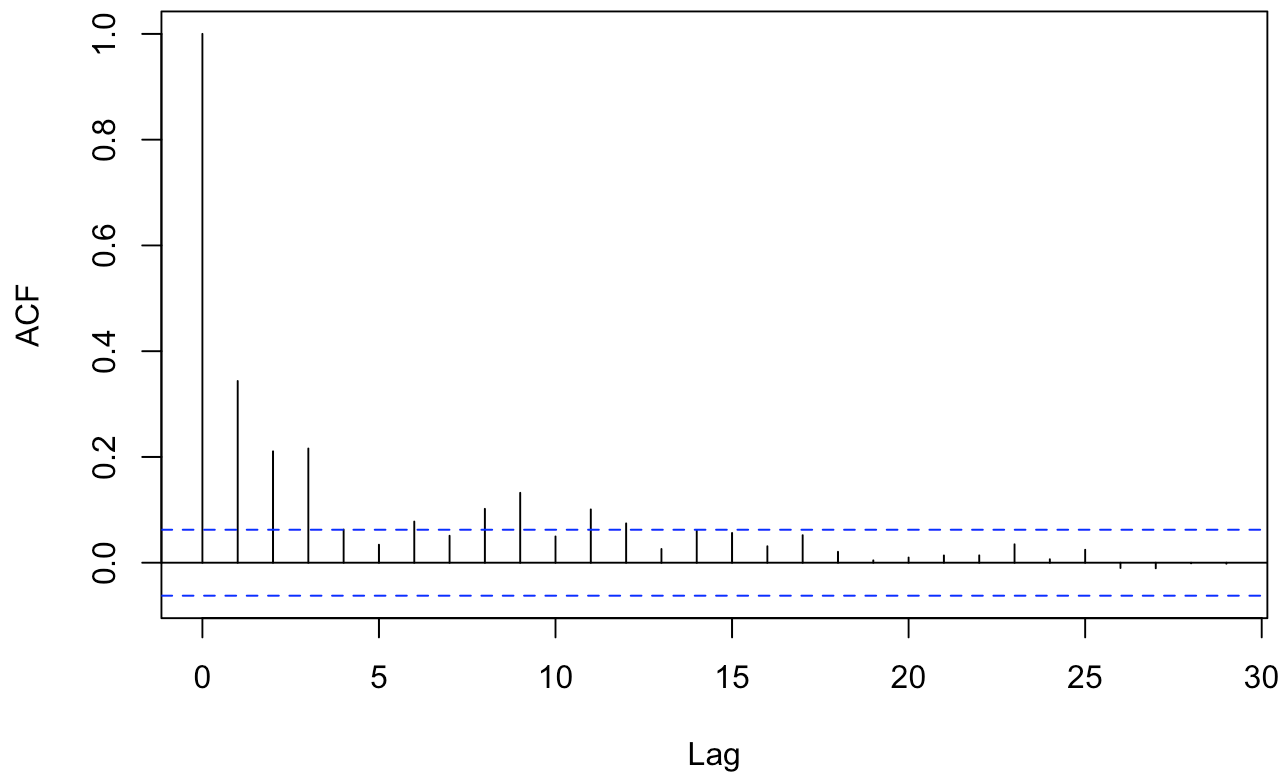
```
## Series: diff.log.return
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##      ma1
##      -0.0676
## s.e.    0.0309
##
## sigma^2 = 0.0002234: log likelihood = 2751.35
## AIC=-5498.71   AICc=-5498.69   BIC=-5488.91
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.0003563338 0.01494003 0.00960361 96.80728 110.6965 0.6889828
##              ACF1
## Training set -0.00259078
```

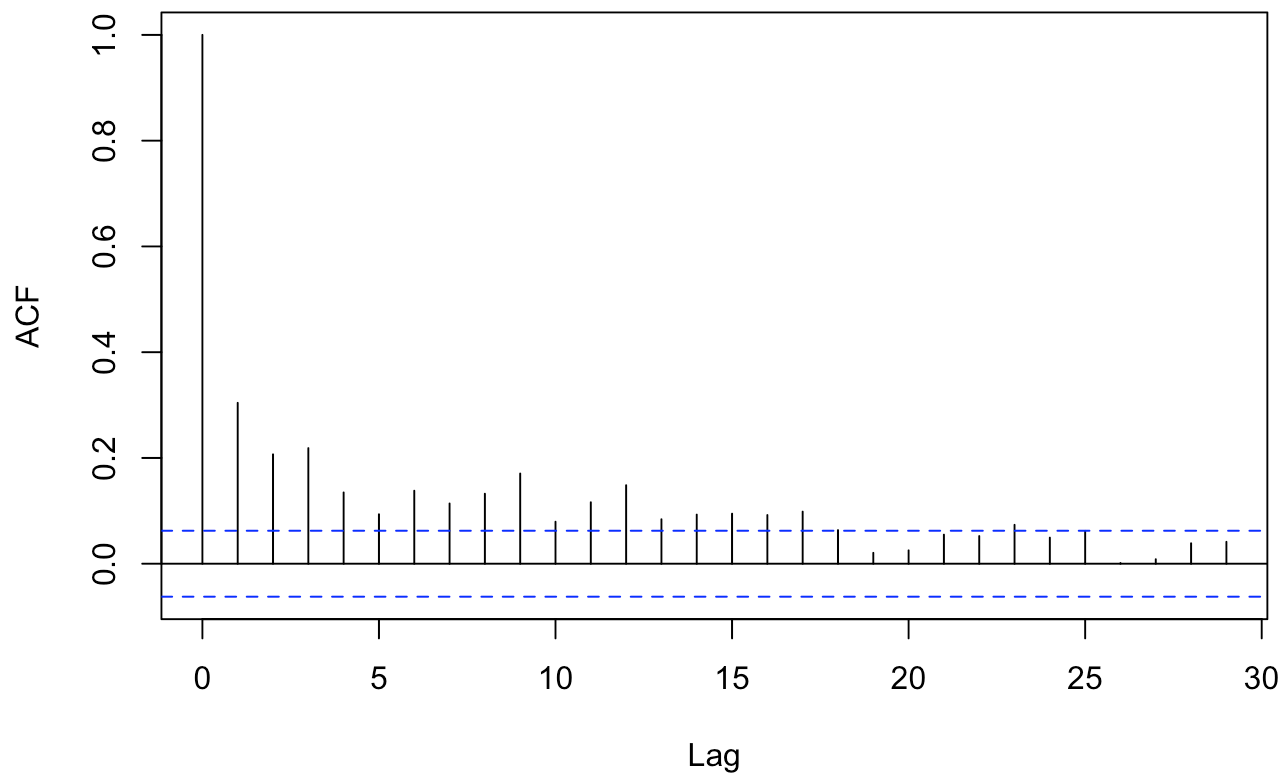

QQ Plot; MA(1) Resid.



Appendix 5

MA(1) Sq. Resid. ACF Plot



MA(1) Abs.Value Resid. ACF Plot

Appendix 6

```
##
## Title:
##   GARCH Modelling
##
## Call:
##   garchFit(formula = ~arma(0, 1) + garch(1, 1), data = diff.log.return,
##     trace = FALSE)
##
## Mean and Variance Equation:
##   data ~ arma(0, 1) + garch(1, 1)
## <environment: 0x133b35bd8>
##   [data = diff.log.return]
##
## Conditional Distribution:
##   norm
##
## Coefficient(s):
##           mu           ma1           omega           alpha1           beta1
## 2.1003e-04 -4.7778e-02  1.9746e-05  9.9480e-02  8.0706e-01
##
## Std. Errors:
##   based on Hessian
##
## Error Analysis:
##           Estimate Std. Error t value Pr(>|t|)
## mu      2.100e-04  3.815e-04   0.550 0.581991
## ma1     -4.778e-02  3.790e-02  -1.261 0.207482
## omega    1.975e-05  5.859e-06   3.370 0.000751 ***
## alpha1   9.948e-02  2.179e-02   4.565 4.99e-06 ***
## beta1    8.071e-01  4.269e-02  18.903 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 2856.457    normalized: 2.89115
##
## Description:
##   Tue Dec 17 18:37:45 2024 by user:
##
##
## Standardised Residuals Tests:
##
##           Statistic    p-Value
## Jarque-Bera Test    R    Chi^2 1.134770e+04 0.0000000
## Shapiro-Wilk Test    R    W      8.783147e-01 0.0000000
## Ljung-Box Test      R    Q(10) 7.569599e+00 0.6707968
## Ljung-Box Test      R    Q(15) 8.592182e+00 0.8978581
## Ljung-Box Test      R    Q(20) 1.272901e+01 0.8887059
## Ljung-Box Test      R^2  Q(10) 1.207276e+00 0.9995946
## Ljung-Box Test      R^2  Q(15) 1.813512e+00 0.9999846
## Ljung-Box Test      R^2  Q(20) 2.078270e+00 0.9999998
```

```
## LM Arch Test      R      TR^2    1.281666e+00 0.9999443
```

```
##
```

```
## Information Criterion Statistics:
```

```
##      AIC      BIC      SIC      HQIC
```

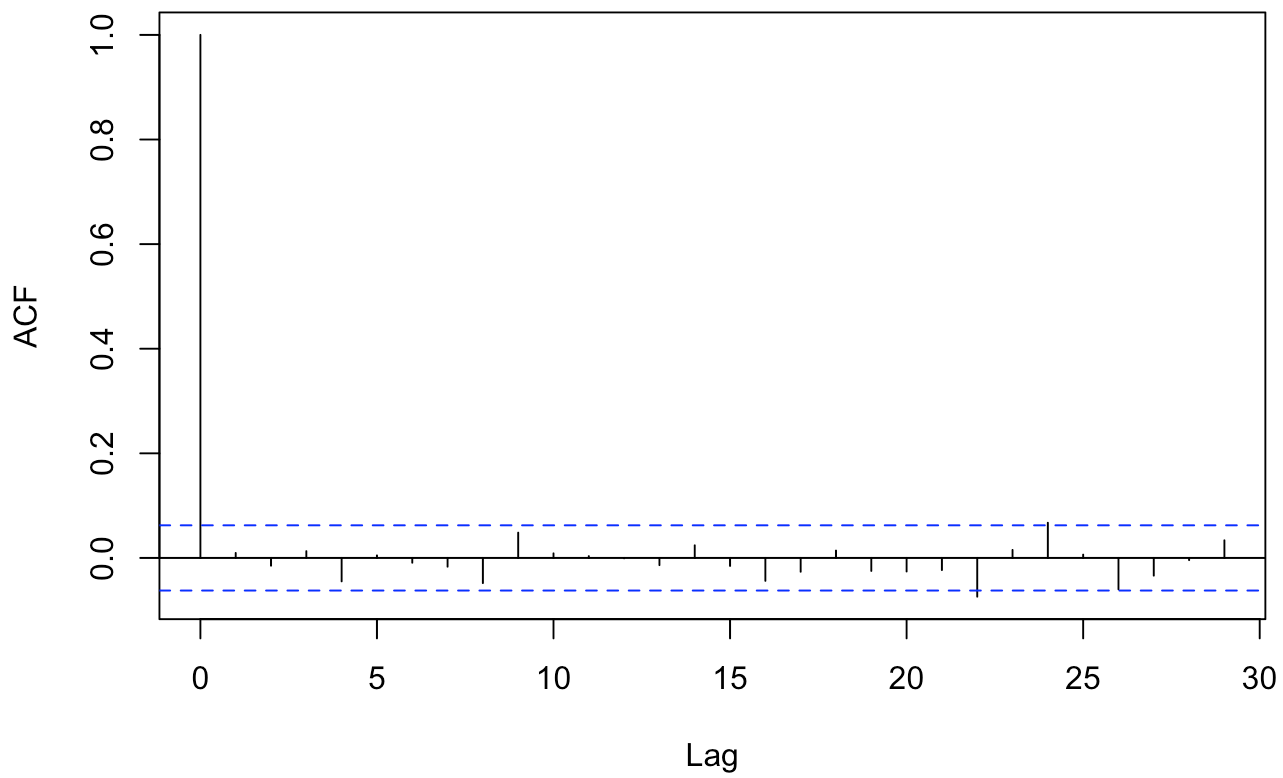
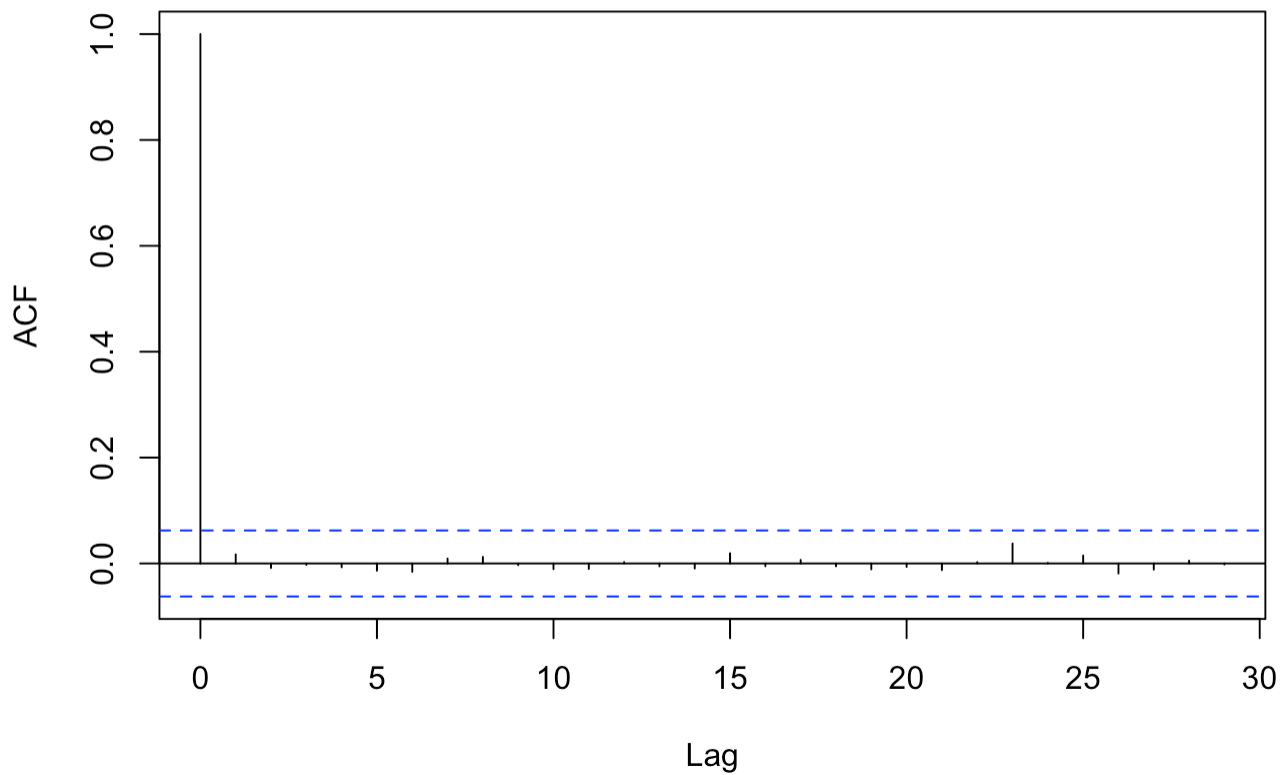
```
## -5.772179 -5.747404 -5.772230 -5.762757
```

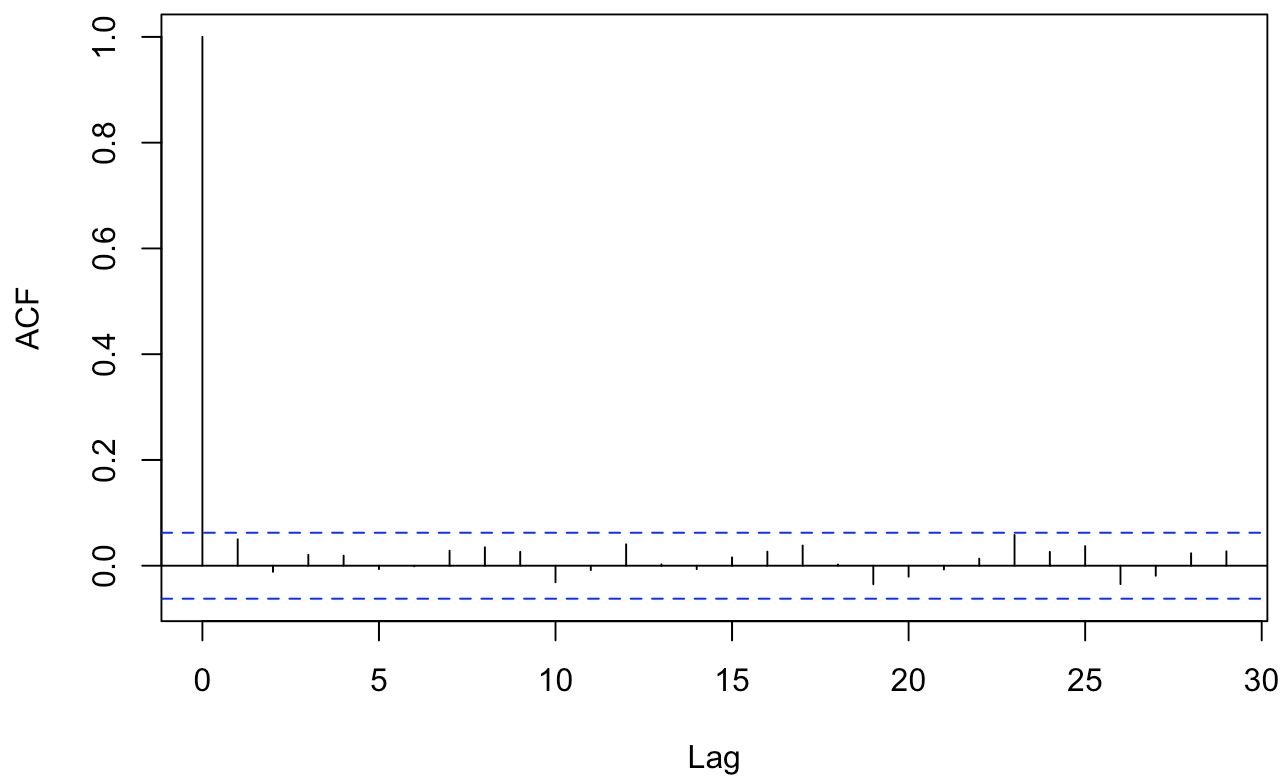
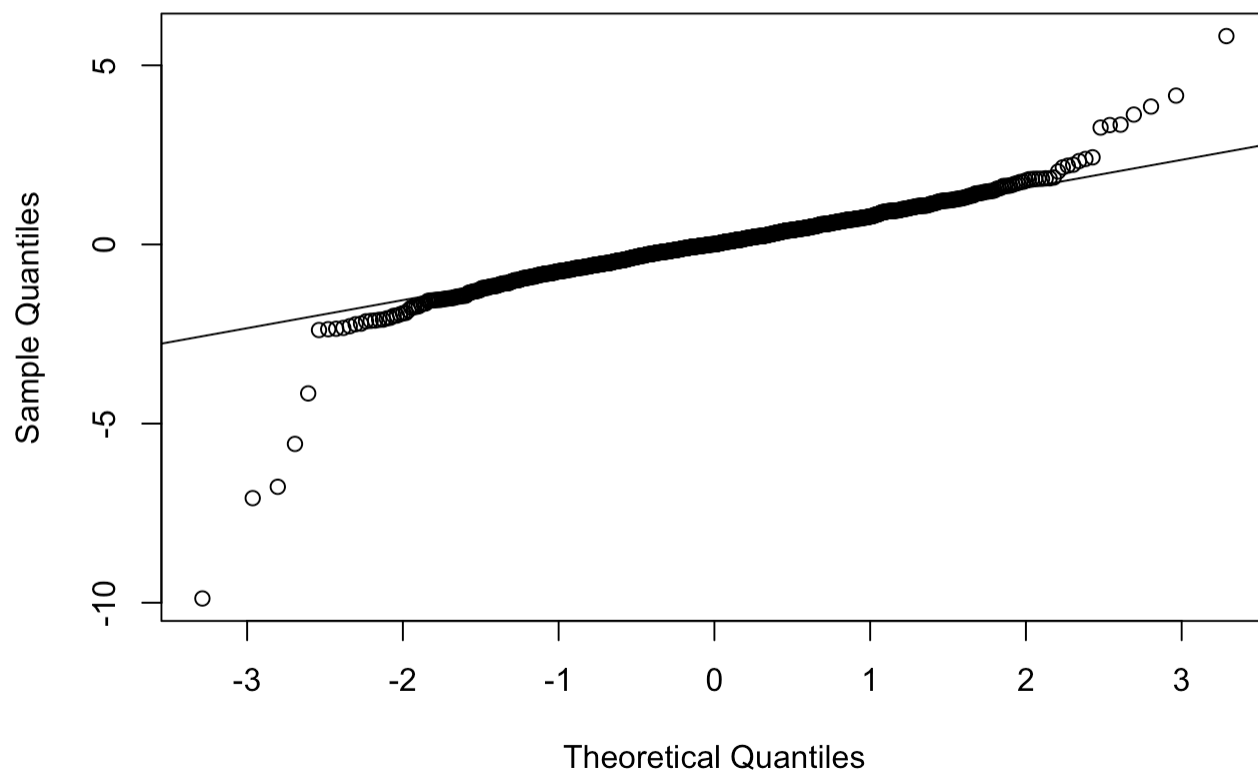
```
##      AIC
```

```
## -5.772179
```

```
##      BIC
```

```
## -5.747404
```

ACF Plot: Residuals ARMA(0,1)+GARCH(1,1)**ACF Plot: Resids. Squared ARMA(0,1)+GARCH(1,1)**

ACF Plot: Abs. Val. Resids. ARMA(0,1)+GARCH(1,1)**QQ-PLOT ARMA(0,1)+GARCH(1,1)**

```
##  
## Asymptotic one-sample Kolmogorov-Smirnov test  
##  
## data:  res  
## D = 0.069754, p-value = 0.0001335  
## alternative hypothesis: two-sided
```


Appendix 7

```
##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~garch(1, 1), data = diff.log.return, trace = FALSE)
##
## Mean and Variance Equation:
## data ~ garch(1, 1)
## <environment: 0x133de5190>
## [data = diff.log.return]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##          mu          omega      alpha1      beta1
## 1.9450e-04  2.0232e-05  9.9277e-02  8.0469e-01
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate  Std. Error  t value Pr(>|t|)
## mu      1.945e-04   3.998e-04    0.486 0.626618
## omega   2.023e-05   6.026e-06    3.358 0.000786 ***
## alpha1  9.928e-02   2.194e-02    4.525 6.03e-06 ***
## beta1   8.047e-01   4.381e-02   18.370 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 2855.395      normalized: 2.890076
##
## Description:
## Tue Dec 17 18:37:45 2024 by user:
##
##
## Standardised Residuals Tests:
##
##              Statistic    p-Value
## Jarque-Bera Test  R    Chi^2  1.143631e+04 0.0000000
## Shapiro-Wilk Test  R    W      8.772439e-01 0.0000000
## Ljung-Box Test    R    Q(10)  8.515525e+00 0.5786134
## Ljung-Box Test    R    Q(15)  9.617645e+00 0.8430773
## Ljung-Box Test    R    Q(20)  1.338601e+01 0.8602146
## Ljung-Box Test    R^2  Q(10)  1.130237e+00 0.9996991
## Ljung-Box Test    R^2  Q(15)  1.734017e+00 0.9999886
## Ljung-Box Test    R^2  Q(20)  1.985906e+00 0.9999999
## LM Arch Test      R    TR^2   1.185283e+00 0.9999637
##
```

Information Criterion Statistics:

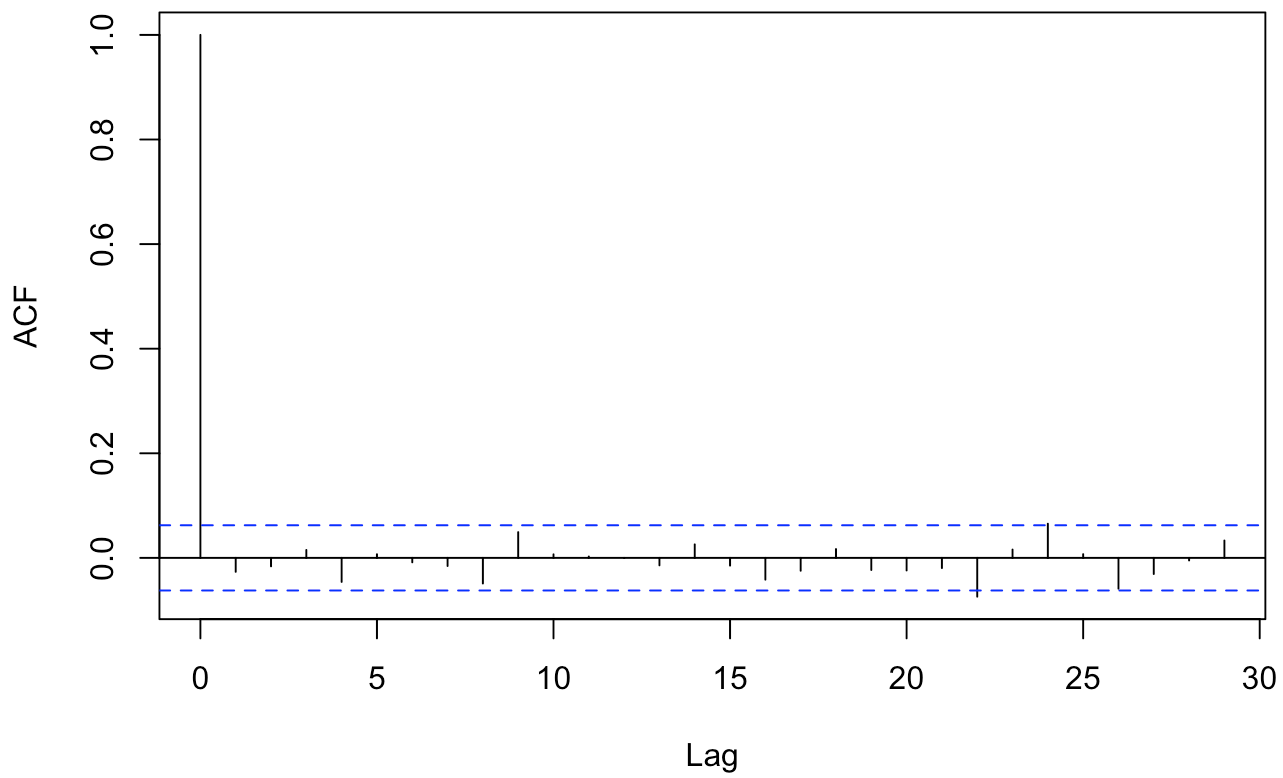
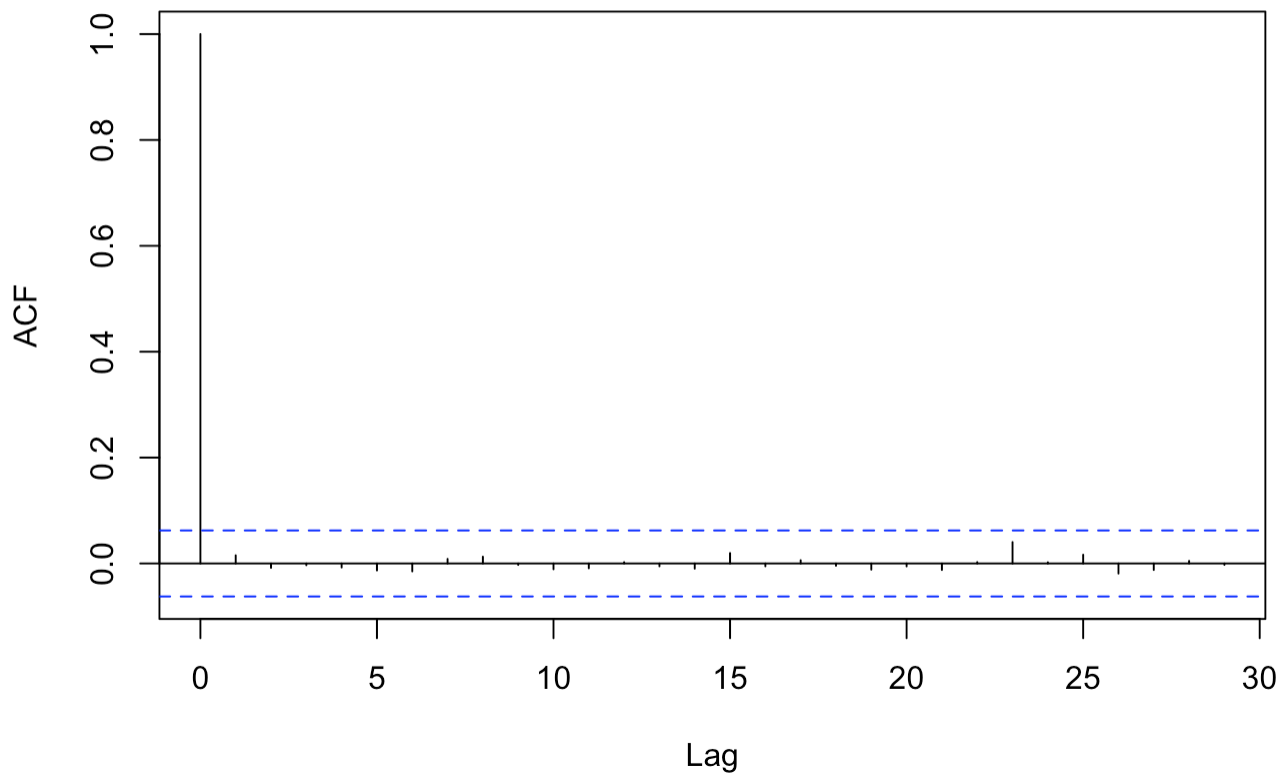
##	AIC	BIC	SIC	HQIC
##	-5.772054	-5.752234	-5.772087	-5.764517

##	AIC
----	-----

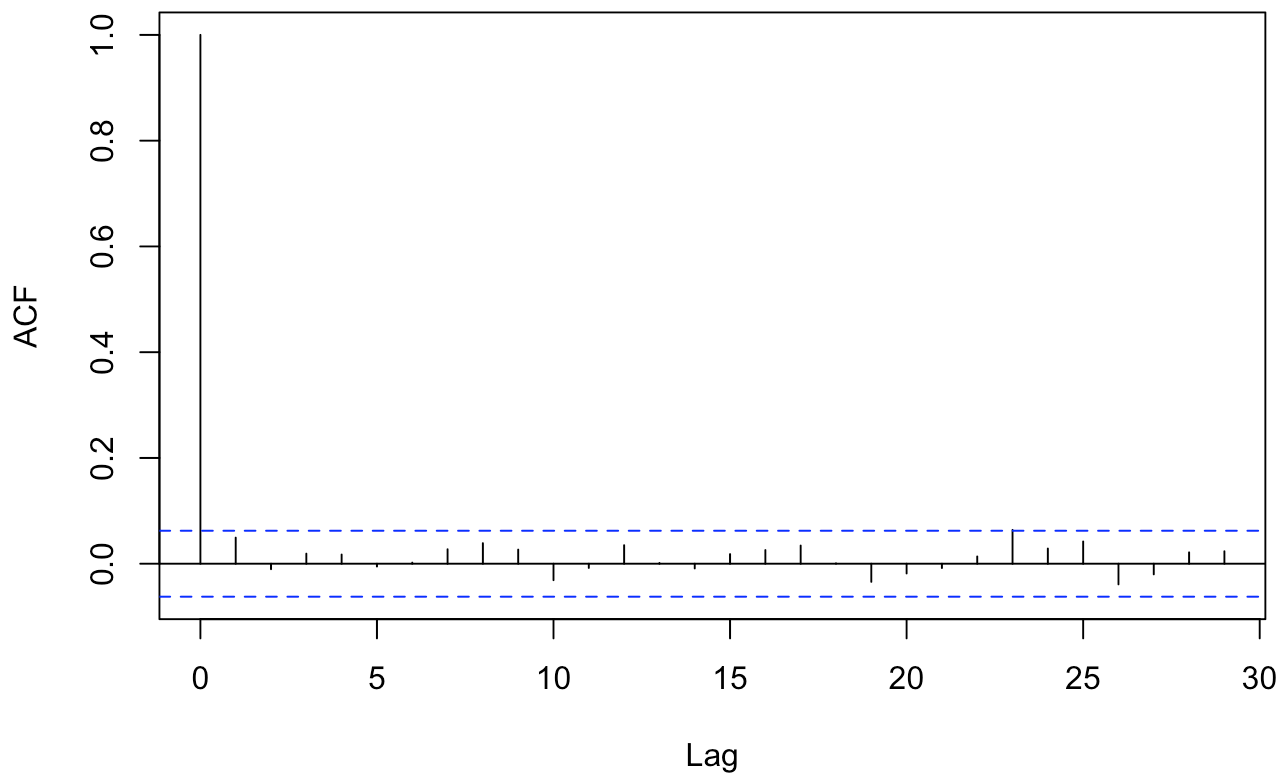
##	-5.772179
----	-----------

##	BIC
----	-----

##	-5.747404
----	-----------

ACF Plot: Residuals (GARCH(1,1))**ACF Plot: Resids. Squared (GARCH(1,1))**

ACF Plot: Abs. Val. Resids. (GARCH(1,1))

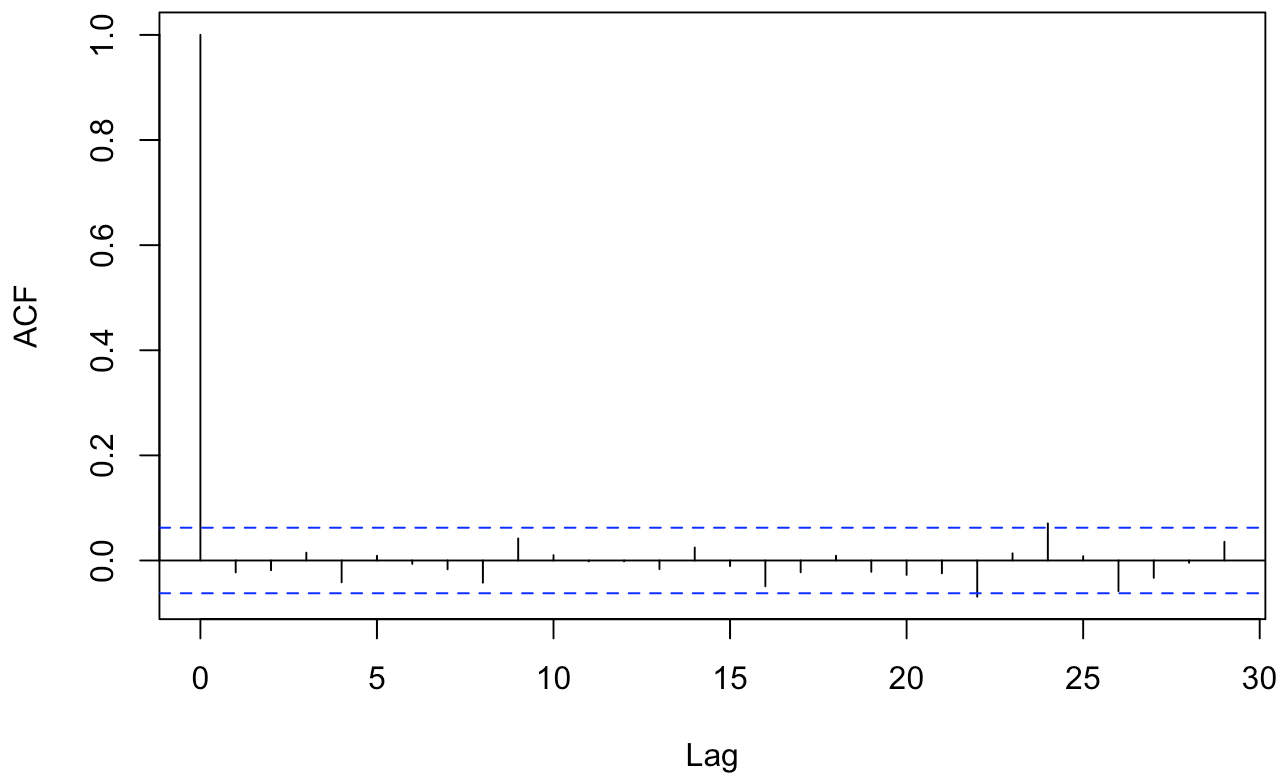
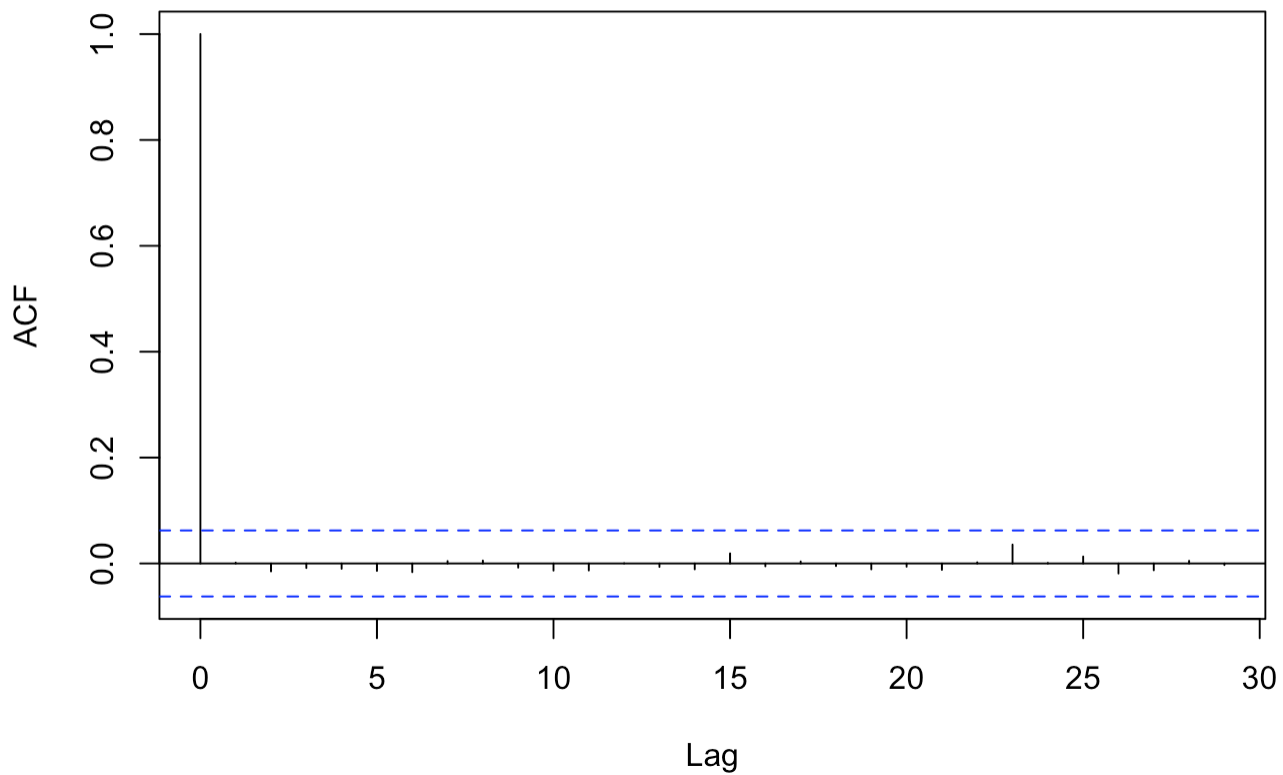


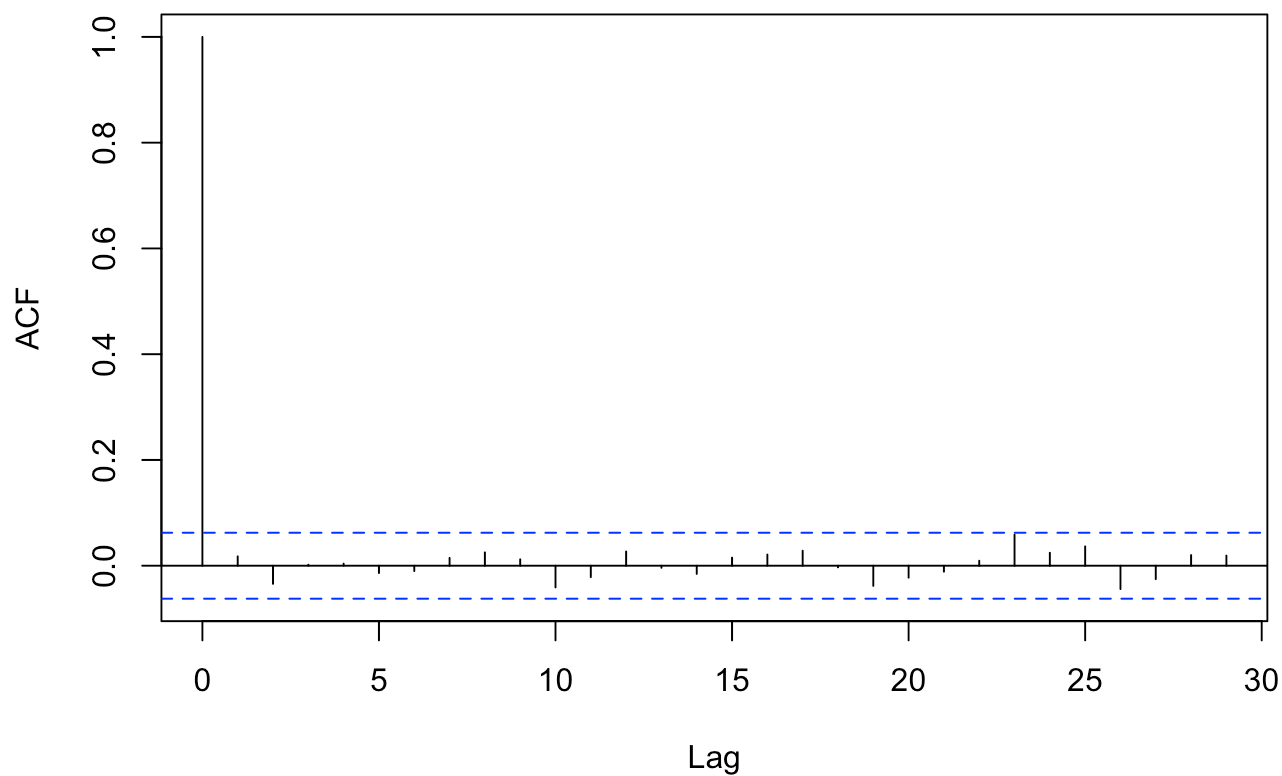
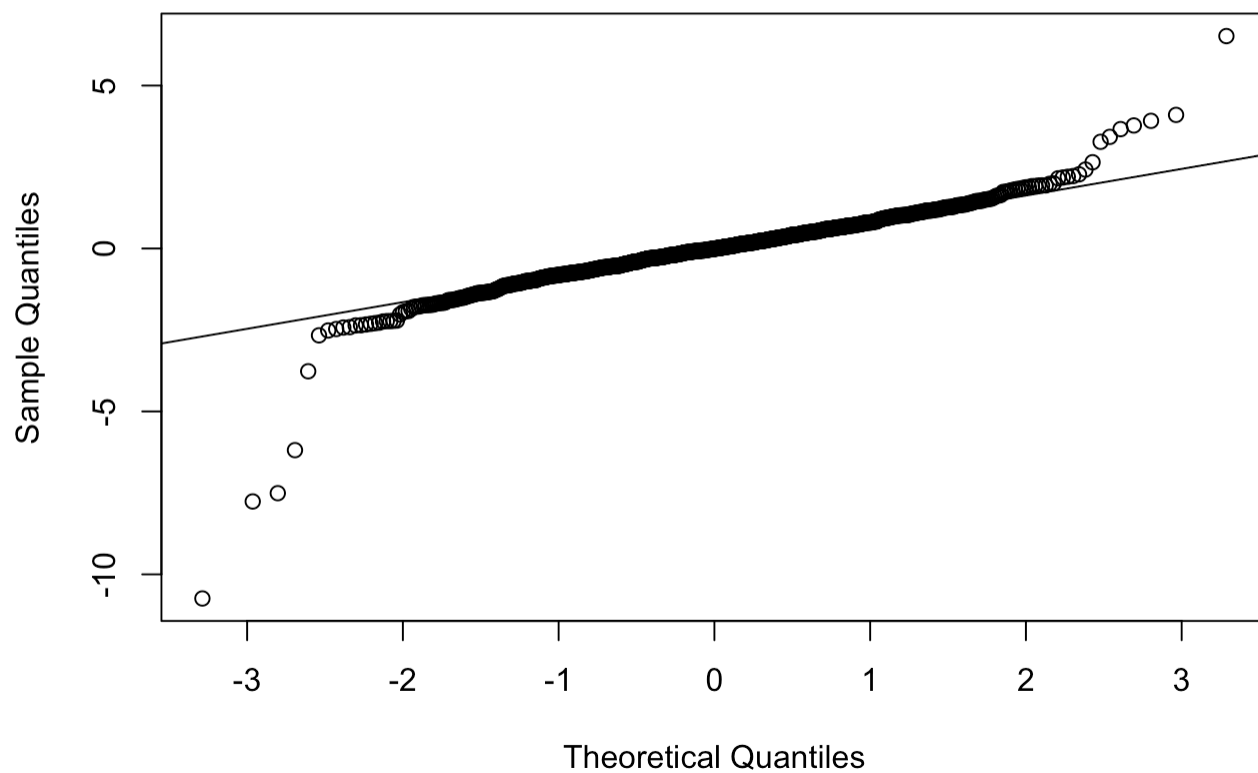
```
##  
## Asymptotic one-sample Kolmogorov-Smirnov test  
##  
## data:  res  
## D = 0.073152, p-value = 5.115e-05  
## alternative hypothesis: two-sided
```

Appendix 8

```
##
## Title:
##   GARCH Modelling
##
## Call:
##   garchFit(formula = ~garch(1, 1), data = diff.log.return, cond.dist = "std",
##     trace = FALSE)
##
## Mean and Variance Equation:
##   data ~ garch(1, 1)
## <environment: 0x1209fdac8>
##   [data = diff.log.return]
##
## Conditional Distribution:
##   std
##
## Coefficient(s):
##           mu           omega          alpha1          beta1          shape
## 4.7597e-04  1.6354e-05  1.3325e-01  7.8155e-01  4.0942e+00
##
## Std. Errors:
##   based on Hessian
##
## Error Analysis:
##           Estimate Std. Error  t value Pr(>|t|)
## mu       4.760e-04  3.198e-04   1.489  0.13662
## omega    1.635e-05  6.616e-06   2.472  0.01344 *
## alpha1   1.332e-01  4.542e-02   2.933  0.00335 **
## beta1    7.815e-01  6.624e-02  11.799 < 2e-16 ***
## shape    4.094e+00  5.079e-01   8.062 6.66e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##   2989.876    normalized:  3.02619
##
## Description:
##   Tue Dec 17 18:37:45 2024 by user:
##
##
## Standardised Residuals Tests:
##
##           Statistic    p-Value
## Jarque-Bera Test    R    Chi^2  1.371557e+04 0.0000000
## Shapiro-Wilk Test   R    W      8.703194e-01 0.0000000
## Ljung-Box Test      R    Q(10)  6.791180e+00 0.7450008
## Ljung-Box Test      R    Q(15)  7.784066e+00 0.9321476
## Ljung-Box Test      R    Q(20)  1.199748e+01 0.9161628
## Ljung-Box Test      R^2  Q(10)  1.137885e+00 0.9996898
## Ljung-Box Test      R^2  Q(15)  1.836107e+00 0.9999832
## Ljung-Box Test      R^2  Q(20)  2.061244e+00 0.9999999
```

```
## LM Arch Test      R      TR^2    1.351128e+00 0.9999257
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## -6.042259 -6.017484 -6.042310 -6.032837
```

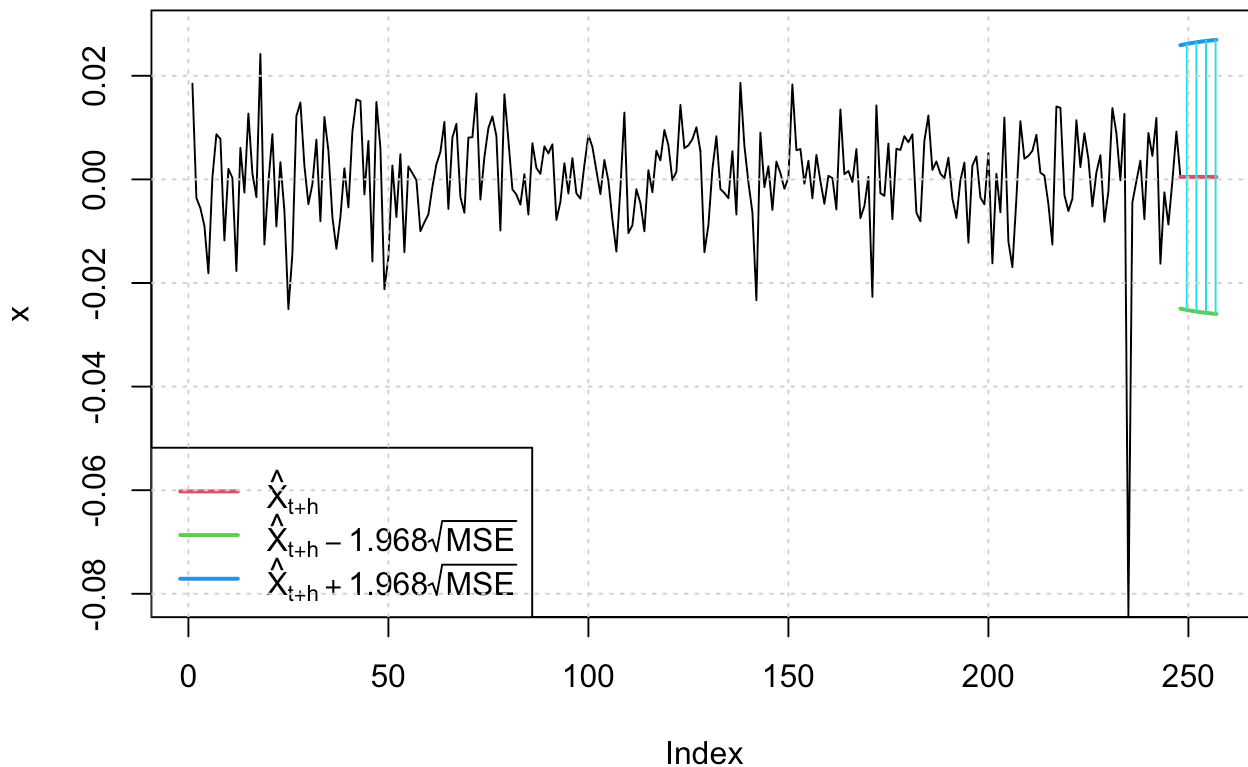
ACF: Residuals GARCH(1,1) + cond.dist**ACF: Squared Residuals GARCH(1,1) + cond.dist**

ACF: Abs. Val. Residuals GARCH(1,1) + cond.dist**QQ-PLOT GARCH(1,1) + cond.dist**


```
##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  res
## D = 0.055062, p-value = 0.005003
## alternative hypothesis: two-sided
```

Appendix 9

Prediction with confidence intervals



##	meanForecast	meanError	standardDeviation	lowerInterval	upperInterval
## 1	0.0004759673	0.01292186	0.01292186	-0.02495179	0.02590372
## 2	0.0004759673	0.01300391	0.01300391	-0.02511325	0.02606518
## 3	0.0004759673	0.01307851	0.01307851	-0.02526006	0.02621199
## 4	0.0004759673	0.01314639	0.01314639	-0.02539363	0.02634556
## 5	0.0004759673	0.01320818	0.01320818	-0.02551522	0.02646715
## 6	0.0004759673	0.01326446	0.01326446	-0.02562595	0.02657789
## 7	0.0004759673	0.01331573	0.01331573	-0.02572684	0.02667878
## 8	0.0004759673	0.01336245	0.01336245	-0.02581880	0.02677073
## 9	0.0004759673	0.01340506	0.01340506	-0.02590264	0.02685457
## 10	0.0004759673	0.01344392	0.01344392	-0.02597910	0.02693103

Appendix 10

ALL CODE COMMENTED OUT

```
#load data you getSymbols() function
#getSymbols("WMT", from = "2020-01-01", to = "2023-12-06")
#adj.close = Ad(WMT) # Adjusted Close Prices
```

```
#visualize original series
#chart_Series(adj.close, name = "Walmart (WMT): Adjusted Close Prices (01/01/2020 - 12/06-2023)")
```

```
#library(kableExtra)
#stationary?
# adf_test_pval = adf.test(adj.close)$p.value # p-value < 0.05 => stationary ||| p-value = 0.07851 > 0.05 => NOT stationary
# kpss_test_pval = kpss.test(adj.close)$p.value # p-value > 0.05 => stationary ||| p-value = 0.01 < 0.05 => NOT stationary
#
# stationary_tests = c("ADF Test", "KPSS Test")
# p_vals = c(adf_test_pval, kpss_test_pval)
#
# stationary_test_results = data.frame(
#   Test = stationary_tests,
#   'P-Value' = p_vals
# )
#
# x = kable(stationary_test_results) %>%
#   kable_styling(bootstrap_options = "condensed", font_size = 15)
# add_header_above(x, c("Stationary Test Results" = 2), color = "red")

#####NOT STATIONARY DATA##### --> use log to account for the non-constant variation, use difference to account for non-constant mean (increasing time trend)
```

```
#Use log to convert multiplicative data (like stock price) to additive data (summation)
#diff.log.return = diff(log(adj.close))[-1]
```

```
#visualize log transformed series
#chart_Series(diff.log.return, name = "WMT: Log Return, ( d=1 ) ") #looks somewhat better, now centered around 0, some variation spikes yet
```

```

#stationary?
#adf.test(diff.log.return) # p-value < 0.05 => stationary ||| p-value = 0.01 < 0.05 => s
tationary
#kpss.test(diff.log.return) # p-value > 0.05 => stationary ||| p-value = 0.1 > 0.05 => s
tationary

#adf_test_pval = adf.test(diff.log.return)$p.value # p-value < 0.05 => stationary ||| p-
value = 0.07851 > 0.05 => NOT stationary
#kpss_test_pval = kpss.test(diff.log.return)$p.value # p-value > 0.05 => stationary |||
p-value = 0.01 < 0.05 => NOT stationary

#stationary_tests = c("ADF Test", "KPSS Test")
#p_vals = c(adf_test_pval, kpss_test_pval)

#stationary_test_results = data.frame(
# Test = stationary_tests,
# 'P-Value' = p_vals
#)

#x = kable(stationary_test_results) %>%
# kable_styling(bootstrap_options = "condensed", font_size = 15)
#add_header_above(x, c("Stationary Test Results Differenced Log Returns (d=1)" = 2),colo
r = "red")

```

```

#####ORDER OF DATA#####
# par(mfrow = c(1, 1))
# par(mfrow = c(1, 2))
# acf(diff.log.return, main = "ACF Plot, Differenced Log Returns")
# pacf(diff.log.return, main = "PACF Plot, Differenced Log Returns")
# par(mfrow = c(1, 1))
#eacf(diff.log.return)
#auto.arima(diff.log.return) #ARIMA(0,0,1)

```

```

#(auto.arima(diff.log.return) )#ARIMA(0,0,1)

```

```

#model = auto.arima(diff.log.return)
#model

#checking residuals
#plot(model) # unit root test, omega = 1/x strictly inside unit circle, not x

# res =residuals(model, standardize = T)
#
# Box.test(res, type = "Ljung") # white noise; p-value = 0.95
#
# res_sq = (res)^2
#
#
# res_abs = abs(res)

#aic = -5499.27
# Coef: = -0.0682; s.e. 0.0309
# Box-Ljung test = p-value = 0.95

# par(mfrow = c(2, 3))
# plot(res, main = "Time Series Plot: MA(1) Resid.") # NOT equal var, i.e., heteroscedasticity
# acf(res, main = " MA(1) Resid. ACF Plot") # NOT white noise
# qqnorm(res, main = "QQ Plot; MA(1) Resid."); qqline(res)
# acf(res_sq, main = "MA(1) Sq. Resid. ACF Plot") # NOT white noise
# acf(res_abs, main = "MA(1) Abs.Value Resid. ACF Plot") # NOT white noise
# par(mfrow = c(1, 1))

# model_arma01_garch11 = garchFit(formula = ~ arma(0,1) + garch(1, 1), data = diff.log.return, trace = FALSE)
# model_arma01_garch11.sum = summary(model_arma01_garch11);
#
# model_arma01_garch11.sum$ics["AIC"] # AIC
# model_arma01_garch11.sum$ics["BIC"]
#
#
# res = residuals(model_arma01_garch11, standardize = T)
# par(mfrow = c(2, 2))
# acf(res)
# acf(res^2)
# acf(abs(res)) # all look good => strong white noise
# qqnorm(res); qqline(res) # normality test: not normal => maybe a different cond.dist is better
# ks.test(res, "pnorm") # normality test by Kolmogorov-Smirnov (KS) test: p-value < 0.05
=> not normal

```

```
# model_garch11 = garchFit(formula = ~ garch(1, 1), data = diff.log.return, trace = FALSE)
# model_garch11.sum = summary(model_garch11);
#
# model_arma01_garch11.sum$ics["AIC"] # AIC
# model_arma01_garch11.sum$ics["BIC"]
#
#
# res = residuals(model_garch11, standardize = T)
# acf(res)
# acf(res^2)
# acf(abs(res)) # all look good => strong white noise
# qqnorm(res); qqline(res) # normality test: not normal => maybe a different cond.dist is better
# ks.test(res, "pnorm") # normality test by Kolmogorov-Smirnov (KS) test: p-value < 0.05
# => not normal
```

```
# model_garch11.dist = garchFit(formula = ~ garch(1, 1), data = diff.log.return, trace = FALSE, cond.dist = "std") # standardized student's t dist = std
# model_garch11.dist.sum = summary(model_garch11.dist);
# #model_garch11.dist.sum$ics["AIC"] # AIC
# #model_garch11.dist.sum$ics["BIC"]
#
# res = residuals(model_garch11.dist, standardize = T)
# acf(res)
# acf(res^2)
# acf(abs(res)) # all look good => strong white noise
# #qqnorm(res); qqline(res) # normality test: not normal => maybe a different cond.dist is better
# #ks.test(res, "pnorm") # normality test by Kolmogorov-Smirnov (KS) test: p-value < 0.05 => not normal
#
#
# # standardized residual distribution test
# set.seed(10)
# x = rstd(1000, mean = 0, sd = 1, nu = 3.482) # make a standardized student t distribution for comparison
# ks.test(res, x) # compare residual's distribution with x's by Kolmogorov-Smirnov (KS) test: p-value > 0.05 => the same
# # the result matches the original cond.dist setup
```

```
# forecast
#predict(model_garch11.dist, n.ahead = 10, plot = TRUE) # predict 10 more values with pt forecast and se
```