Investment Term Project

Predict 412 Sec 55

Phil Franzo
Kent Kofoed
Vijay Sharma
Katie Turnbull

Northwestern University

Introduction

The efficient market hypothesis asserts that, "security prices accurately reflect available information, and respond rapidly to new information as soon as it becomes available" (Brealey & Myers, 1996). The notion that public sentiment plays a role in determining stock prices challenges the efficient market hypothesis's assumption that all market participants know all information instantaneously. Public sentiment is the cumulative response to information available about a stock and takes into account the diffusion rate of information in the market. Using social media as a proxy for real-time public sentiment, early indication of a change in sentiment could predict the direction and magnitude of price moves in a stock.

Literature Review

A quick scan of Twitter feeds reveals a large amount of noise in the data and the presence of multiple tweets or retweets of the same data, which may introduce bias. Barbosa and Feng (2010) investigated a number of methods of detecting sentiment on Twitter and specifically addressed this issue of biased and noisy data. Agarwal, et al. (2011) provided a tree based classification methodology to classify sentiment of Twitter data. Their paper addressed issues of feature selection, the treatment of non-text characters, and acronyms commonly used in microblogging. Lui's introduction to sentiment analysis and opinion mining (2012) covered topics ranging from document classification to sentence and aspect-based sentiment analysis. It also offered insights into some of the intricacies of language processing, such as sarcasm and conditional sentence structures. O'Connor, et al. (2010) connected measures of public opinion from polling data with public sentiment measured from Twitter messages. They proposed methods for opinion estimation from a Twitter corpus and investigated the time series correlation of public opinion from polling data to sentiment from Twitter.

Bollen, Mao, & Xiaojun (2011) used two different methods of sentiment analysis applied to daily Twitter feeds to develop a mood time series and validate this series against DJIA time series. The authors conducted a Granger causality analysis and used a self-organizing fuzzy neural network to investigate whether the mood data was predictive of the DJIA data. Chen and Lazer's paper (2011) also investigated the relationship between tweets and stock market movements by constructing a model using sentiment information and testing the predictive accuracy of the model, using a somewhat simpler process than Bollen.

Loughran and McDonald (2011) identified the deficiencies of using word lists developed for other disciplines to measure the sentiment of financial text. In particular, they showed that the widely used Harvard Dictionary misclassified almost 75% of negative words in a sample of 10-K reports. The alternative word lists they developed are a better reflection of the tone of financial text and were considered in the sentiment analysis portion of our project.

Cryer and Chan's book (2008) was the main source for the time series portion of the analysis. The text covers the majority of the time series concepts needed (fundamental time series concepts, models for stationary and nonstationary time series, model specification issues, parameter estimation, model diagnostics, forecasting, and multiple different types of models, such as seasonal time series models), as well as covering the R commands and packages. Additionally, Ruppert's text (2011) has a couple of chapters on time series analysis that solely focus on financial applications of time series analysis, which provided the topic-specific knowledge that was required for the time series portion of the project.

Methods – Data Collection

For the text analytics portion of the analysis, we used the R package "twitteR" to connect to the Twitter search API and gather a sample of Tweets containing either #Exxon or Exxon. The Twitter search API pulls tweets from an index containing the tweets from the previous 6 to 9 days. Data collection of a sufficient sample size proved a hurdle, as the API only allows users to pull the most recent 450 tweets every 15 minutes. This means we could collect a maximum of 43,200 tweets per day, but since each pull only contained the most recent tweets, the daily files had a number of duplicate records. The initial data set comprised some 200,000 tweets for the continuous period 29 July-26 Aug, but the data was exceptionally noisy, containing corrupt tweets and incomplete records, in addition to the duplicate records. We were not able to find a repository of tweets outside the Twitter API, though in the past there were large datasets available. The only alternate means of collecting tweets that we found was to copy and paste directly from the twitter website, which presented problems with formatting in general, but particularly with time stamps. This allowed us to analyze older data, but required a fair bit of additional data cleaning. Cleaning the data resulted in approximately 2,283 tweets.

For the portion of the assignment using more traditional methods, we used information such as commodity price data, index and sector data, and stock price data for XOM. We obtained the commodity price data from the FRED Excel Add-In (the economic data Excel add-in from the Federal Reserve Bank of St. Louis) and the index, sector and stock price data from Yahoo! Finance, dating back to 2008.

Method & Results – Time Series Model

The first step in creating the time series model for our research project was to perform an EDA using monthly data, with the goal of finding relevant predictor variables that could be used to forecast the stock price of Exxon Mobil (XOM). We discovered some potential multicollinearity when examining the correlation and scatterplot matrices[1] for these variables, especially between heatingOil/crudeOil, rbob/crudeOil, naturalGas/heatingOil, and rbob/heatingOil. The correlation between sp500 and xom is also fairly large, indicating that the sp500 variable is an important predictor.

We used OLS regression analysis for the EDA; we did not use lags, because the goal of this part of the analysis was to simply get a feel for the relationships among the variables (i.e., prediction was not the objective). We did, however, use dummy indicator variables to measure monthly seasonality, which did not appear to be much of an issue. The first model built for the EDA included each of the predictor variables[2]. Since the RBOB variable was insignificant, we removed it from the model and ran the regression once more[3], and found all of the remaining predictor variables were significant (other than the seasonal dummy variables). We computed VIFs for each variable in this model and found none of the variables appear to have issues with multicollinearity; however, three of the remaining variables are quite similar (Crude Oil, Heating Oil, and Natural Gas)[4]. The biggest take-away from the EDA was that rbob should be removed, and possibly either crudeOil or heatOil. This makes sense from an economic perspective because xom typically deals with petroleum products prior to refinement occurring. Also, the most significant variables seem to be sp500 and natGas.

The second step was to perform an actual time series analysis on the XOM price data. Figure 2.1 includes a line plot of the XOM percentage returns, a line plot of differenced XOM

[1] See Figure 1.1.
[2] See Output 1.1.
[3] See Output 1.2.
[4] See Figures 1.2 & 1.3 for additional plots from this analysis.

percentage returns, an ACF plot, and a PACF plot. Differencing returns seemed to skew returns a bit. The ACF and PACF plots are indicative of autocorrelation in the time series. Next, we performed the Augmented Dickey-Fuller (ADF) test, which indicated that the times series data is stationary. We also calculated kurtosis and skewness, which were 14.9349 and 0.1454203, respectively. These values indicate that there is excess kurtosis (fat tails) and positive skew.

We used auto.arima to select a model, using AIC and BIC as the selection criteria[5]. AIC selected an ARIMA(2,0,1) model and BIC selected an ARIMA(2,0,3) model[6]. The line plots show the large error variance that occurs towards the beginning of the time series (which was during the financial crisis) and the ACF plots show that the models still suffer from a little bit of autocorrelation. Results from the Ljung-Box test were significant[7], which indicated that both of the models do suffer from autocorrelation. The McLeod-Li test indicated that an ARCH model is likely needed (for modeling the heteroskedastic error variance)[8].

Figure 2.4 shows the predicted values from each of these models, as well as the prediction intervals. Both of these models are quite similar and most of the results are pretty much the same, including the mean squared error for each of the models (the RMSE for the AIC model and the BIC model was 0.01199 and 0.01190, respectively). Ultimately, we chose the model selected based on the AIC, because it is the more parsimonious model.

Next, we examined GARCH models, beginning with an ARMA(2,1)/GARCH(1,1) model (this model is similar to the AIC model)[9]. We also examined an ARMA(2,3)/GARCH(1,1)

[5] See Output 2.1 & 2.2, respectively.
[6] See Figures 2.1 & 2.2.
[7] See Output 2.3.
[8] See Figure 2.3.
[9] See Output 2.4.

model (similar to the BIC model) [10] and an ARMA(1,1)/GARCH(1,1) model[11]. The last model of the three performed the best, with an RMSE value of 0.01790917 and only one variable with an insignificant coefficient. The first plot for each model shows the predictions for the entire series, as well as the confidence interval for the next ten forecasted values. The second plot shows the same confidence intervals and the actual response values from the test set[12].

Figure 2.7 shows a plot of the XOM percentage returns, a plot of the absolute value of the XOM percentage returns, and a plot of the conditional variance component of the ARMA(2,1) /GARCH(1,1) model. The first plot (the XOM percentage returns plot) shows volatility cluster that occurred near the beginning of the time series (there is also another volatility cluster towards the end of the time series), while the second plot, by plotting the absolute value of these percentage returns, shows the volatility clustering a little bit more clearly. The third plot (the conditional variance plot) plots the conditional variance component of the GARCH model. By allowing the variance to be conditional on previous values (instead of requiring the variance to be known and constant), GARCH models are able to capture the increased volatility that tends to occur in short, temporary bursts. Figure 2.8 is a similar plot, except this plot shows the absolute returns of XOM (for comparison purposes) and the conditional variance plots of the two other models.

Finally, we created the ACF plots from each of these models[13]. Each model still shows some autocorrelation, but the largest autocorrelations are only around 0.15. Instead of adding on additional autoregressive, difference, moving average or GARCH terms, it would probably be best to stick with a less complex model that is less susceptible to overfitting. The best model,

[10] See Output 2.5.
[11] See Output 2.6.
[12] See Figures 2.4-2.6.
[13] See Figure 2.9.

then, would be the ARMA(1,1)/GARCH(1,1) model, which had the best performance (only by a very small margin) and was the most parsimonious model.

Method & Results – Sentiment Analysis

For the sentiment analysis, the first step after cleaning the data was to score the sentiment of each tweet using two primary word lists: a large publicly available dataset called the Hu and Liu Opinion Lexicon and a financially focused dataset from Loughran and McDonald (2012). The Loughran did not impact the sentiment analysis, likely due to the fact that tweets even regarding financial info are written quite differently than the financial reports that the Loughran dataset was developed for.

Using sentiment dictionaries for positive and negative words from the Hu and Liu Opinion Lexicon, we rated the Tweets in the corpus, aggregated the data by date, and created two daily sentiment indicators. The sentiment score works by matching words in the Tweet with words in the dictionary to provide a positive and negative word count. The first indicator is a sentiment index, which is the quotient of total positive words by total negative words. A large index value indicates positive sentiment and a small value indicates negative sentiment; a value of 1 indicates neutral sentiment. The second is a binary indicator which is +1 if the number of positive words is greater than the number of negative words (i.e., there is no weighting). We plotted both measures, as well as creating word clouds to depict the nature of the positive and negative sentiment[14]. The word clouds contained sentiment words with a minimum frequency of 12. There were 11 words in the positive cloud and 16 words in the negative cloud. Environmentally, words such as poison, destroy, defect, and stress were included, while more financial terms such as fell, weak, and miss were also included.

[14] See Figures 3.1-3.4.

Our initial analysis revealed that the Tweets retrieved based on #Exxon are overwhelming negative in nature, especially the periods of July 29 through Aug 5 and Aug 14 through Aug 17. These tweets appear to relate largely to public protest surrounding Exxon's environmental stewardship and general backlash against carbon-based energy.

Having established a sentiment time series, the next step in the investigation was to assess whether we can use the sentiment to enhance the predictive accuracy of the stock price time series models. We aggregated the sentiment scores into a daily average, which we utilized as an index in the next piece of the analysis.

To determine whether we can use sentiment analysis to predict the price, direction or magnitude of Exxon stock, we studied the sentiment time series and stock price time series of the same period[15]. To conduct a Granger Causality, we used the Toda-Yamamoto (1995) procedure test, as it reduces the occurrence of spurious causality when two time series are non-stationary. In addition, with this procedure there is no requirement to test for co-integration, which eliminates pretest bias. The stock price time series failed the Augmented Dickey-Fuller (ADF) and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests for stationarity. The sentiment time series failed only the ADF test[16]. The stock price time series was made stationary with a differencing of two and the sentiment time series was made stationary with a differencing of one[17].

We determined optimal lag length using the VARSelect function from the vars package. Based on AIC, HQ, SC and FPE criteria, we found this to be either 3 or 4, and conducted a

[15] See Figure 3.5.
[16] See Table 3.1.
[17] See Table 3.2.

test for serial autocorrelation. At 3 lags, chi-sq=28.7201, p=0.9964, df=52 and at 4 lags, chi-sq=31.1257, p=0.9719, df=48. We chose 3 lags since they are less likely to be serially auto-correlated versus 4 lags. Based on the Toda-Yamamoto procedure, we created a model with an additional lag and performed the Chi-sq test against a reference model for Granger Causality. The test concluded at the 95% significance level that stock price does not cause sentiment, nor does sentiment cause stock price[18].

Method & Results: Combined Model

For this analysis, we used the sentiment index data, the sentiment polarity data, and the percentage change in the price of XOM. Additionally, we transformed the sentiment polarity variable into a 0/1 dummy variable and lagged both of the sentiment attributes in order to avoid "look-ahead bias."

Due to the small number of observations, we used an OLS regression model (i.e., the lm() function in R), where the response variable was the percentage change in the price of XOM (percent_response) and the predictor variables included a lagged value of the percentage change in the price of XOM (percent_lag), the sentiment index lag (sent_index_lag), and the sentiment polarity dummy variable lag (sent_dummy_lag). The base model did not perform very well[19]. None of the predictor variables were significant, the F-statistic p-value was 0.8596, and the adjusted R^2 value was -0.1342. We tried additional models where we removed one or both of the sentiment variables, but the results from those regressions were just as poor[20].

[18] See Table 3.3.
[19] See Output 4.1.
[20] See Output 4.2 & 4.3.

We examined the closing price of XOM and the S&P 500 index for each day in the period analyzed[21] and realized that the returns during the period were overwhelmingly negative for both values, indicating that the time period used for our sample was not representative of a typical period. Only 14.2% of the XOM observations were days with positive returns and 85.7% of the observations were days with negative returns. The pullback that occurred for the broad market over our entire observation period biased the results of the regression.

Conclusion

Overall, we found that the limited time frame for this analysis presented a number of hurdles. Additionally, the tweets related to the public protest of Exxon's environmental policies seem to not be related to the sentiment of Exxon's financial performance. Based on the data we have, it seems that traditional methods produce better predictions of future stock price than sentiment alone or sentiment included as part of the time series analysis, as confirmed by the Granger Causality test. It may, however, be worthwhile to reexamine this question with a broader time period of sentiment data so that the sentiment index could be used as a variable in a traditional time series model instead of just the OLS regression we used.

[21] See Figure 4.1.

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of Twitter data. *Proceedings of the Workshop on Language in Social Media (LSM 2011),* 30-38. Portland, OR: Association for Computational Linguistics.

Barbosa, L. & Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. *Coling 2010: Poster Volume*, *August 2010,* 36-44.

Bollen, J., Mao, H., & Xiaojun, Z. (2011). Twitter mood predicts the stock market. *Journal of Computational Science, 2* (2011), 1 – 8.

Brealey, R. and Myers, S. (1996). *Principles of Corporate Finance* (5^{th} Ed.), New York, NY: McGraw-Hill.

Chang, W. (2013). *R Graphics Cookbook.* CA: O'Reilly Media Inc.

Chen, R., & Lazer, M. (2011). Sentiment analysis of Twitter feeds for the prediction of stock market movement. Retrieved from: http://cs229.stanford.edu/proj2011/ChenLazer-SentimentAnalysisOfTwitterFeedsForThePredictionOfStockMarketMovement.pdf

Cryer, J.D., & Chan, K. (2008). *Time series analysis: With applications in R.* New York, NY: Springer.

Fox, J., & Weisberg, S. (2011). *An R companion to applied regression.* Thousand Oaks, CA: Sage.

Granger, C.W.J. (1969). Investigating causal relations by Econometric Models and Cross-spectral Methods. *Econometrica, 37* (3), 424-438.

Kirchgässner, G. & Wolters, J. (2008). *Introduction to Modern Time Series Analysis.* NY: Springer.

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance, 66* (1), 35-65.

Loughran, T. & McDonald,W. (2012). *Loughran and McDonald Financial Sentiment Dictionaries* [Data file]. Retrieved from http://www3.nd.edu/~mcdonald/Word_Lists.html.

Lui, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, *5* (1), 1-167.

O'Connor, B., Balasubramanyan, R., Routledge, B.R., & Smith, N.A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*.

Ruppert, D. (2011). *Statistics and data analysis for financial engineering.* New York, NY: Springer.

Toda, H.Y. and Yamamoto, T. (1995). Statistical inference in vector autoregressions with possibly integrated processes. *Journal of Econometrics*, *66* (1-2), 225-250.

Tsay, R.S. (2001). *Analysis of Financial Time Series: Financial Econometrics.* NY: Wiley-InterScience.

R Packages

Bouchet-Valat, M. (2013). SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library (Version 0.5) [Software]. Available from http://cran.r-project.org/.

Chan, K. & Ripley, B. (2012). TSA: Time Series Analysis (Version 1.01) [Software]. Available from http://cran.r-project.org/.

Feinerer, I. & Hornik, K. (2013). tm: Text Mining Package (Version 0.5-9.1) [Software]. Available from http://cran.r-project.org/.

Fellows, I. (2013). wordcloud: Word Clouds (Version 2.4) [Software]. Available from http://cran.r-project.org/.

Fox, J., et al. (2013). car: Companion to Applied Regression (Version 2.0-18) [Software]. Available from http://cran.r-project.org/.

Hyndman,R., et al. (2013). forecast: Forecasting functions for time series and linear models (Version 4.0.6) [Software]. Available from http://cran.r-project.org/.

James, D., Hornik, K., & Grothendieck, G. (2012). chron: Chronological objects which can handle dates and times (Version 2.3-43) [Software]. Available from http://cran.r-project.org/.

Lesnoff, M. & Lancelot, R. (2012). aod: Analysis of Overdispersed Data (Version 1.3) [Software]. Available from http://cran.r-project.org/.

Pfaff, B. & Stigler, M. (2013). urca: Unit root and cointegration tests for time series data (Version 1.2-8) [Software]. Available from http://cran.r-project.org/.

Pfaff, B. & Stigler, M. (2013). vars: VAR Modelling (Version 1.5-2) [Software]. Available from http://cran.r-project.org/.

Trapletti, A., Hornik, K. & LeBaron, B. (2013). tseries: Time series analysis and computational finance (Version 0.10-32) [Software]. Available from http://cran.r-project.org/.

Turlach, B. (2013). quadprog: Functions to solve Quadratic Programming Problems (Version 1.5-5) [Software]. Available from http://cran.r-project.org/.

Wickham, H. & Chang, W. (2013). ggplot2: An implementation of the grammar of graphics (Version 0.9.3.1) [Software]. Available from http://cran.r-project.org/.

Wickham, H. (2012). plyr: Tools for splitting, applying and combining data (Version 1.8) [Software]. Available from http://cran.r-project.org/.

Wickham, H. (2012). scales: Scale functions for graphics (Version 0.2.3) [Software]. Available from http://cran.r-project.org/.

Wickham, H. (2012). stringr: Make it easier to work with strings (Version 0.6.2) [Software]. Available from http://cran.r-project.org/.

Wright, K. (2012). corrgram: Plot a correlogram (Version 1.4) [Software]. Available from http://cran.r-project.org/.

Wuertz, D., et al. (2013). fBasics: Rmetrics - Markets and Basic Statistics (Version 3010.86) [Software]. Available from http://cran.r-project.org/.

Wuertz, D., et al. (2013). fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling (Version 3010.82) [Software]. Available from http://cran.r-project.org/.

Wuertz, D., et al. (2013). fUnitRoots: Trends and Unit Roots (Version 3010.78) [Software]. Available from http://cran.r-project.org/.

Zeileis, A., Grothendieck, G., Ryan, J., & Andrews, F. (2013). zoo: S3 Infrastructure for Regular and Irregular Time Series (Z's ordered observations) (Version 1.7-10) [Software]. Available from http://cran.r-project.org/.

Appendix

Time Series Model

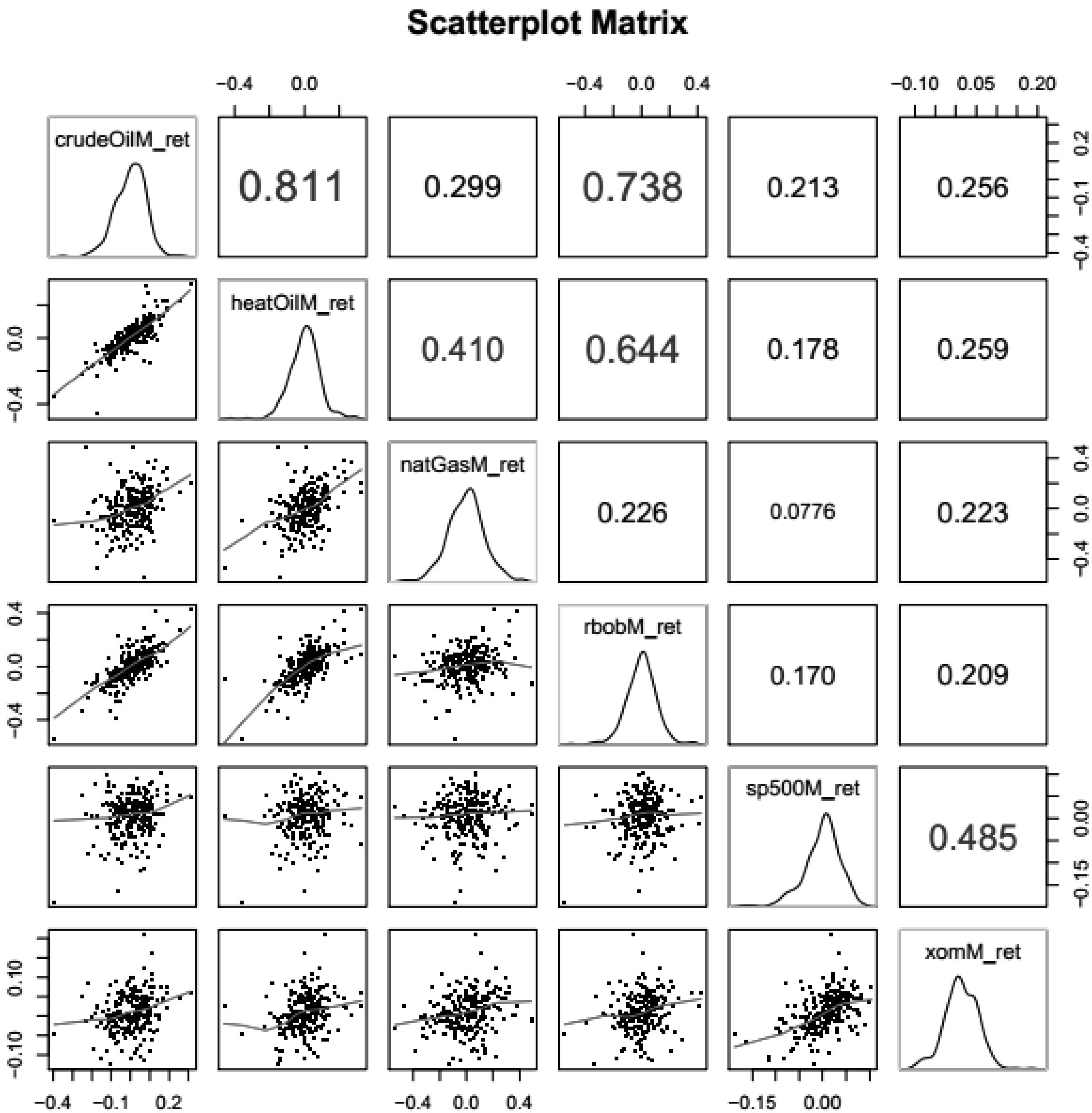


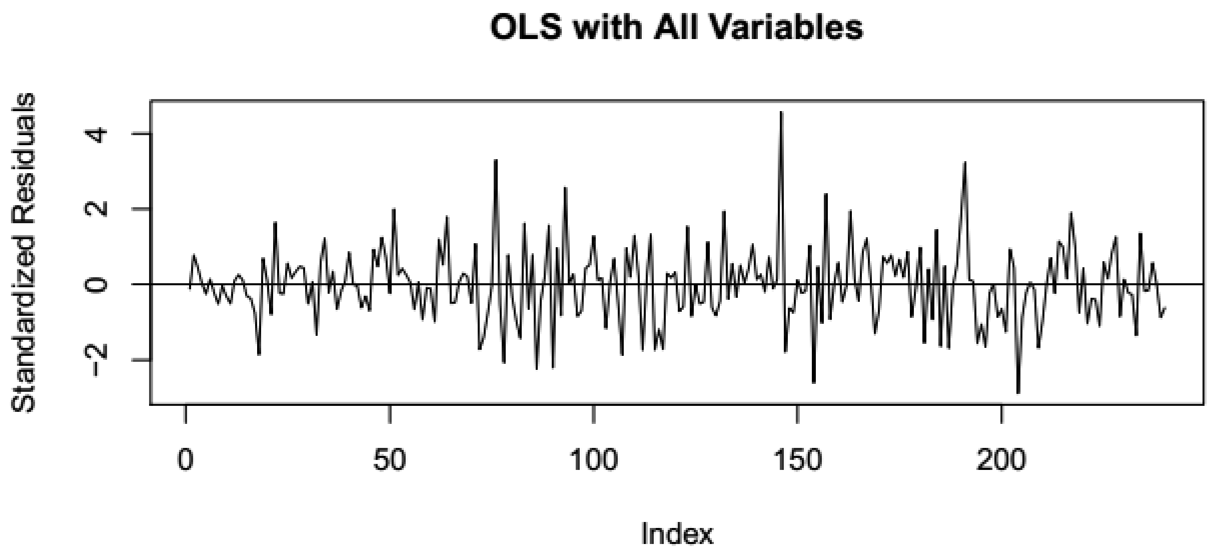
Figure 1.1: Scatterplot Matrix

```
Call:
lm(formula = xom_ret ~ crude_ret + heatoil_ret + natgas_ret +
    rbob_ret + sp500_ret, data = percent)

Residuals:
      Min        1Q    Median        3Q       Max
-0.086665 -0.013288  0.000129  0.013426  0.092952

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0011710  0.0006827   1.715  0.08658 .
crude_ret   -0.0475312  0.0221774  -2.143  0.03230 *
heatoil_ret  0.0638993  0.0230474   2.773  0.00565 **
natgas_ret   0.0260232  0.0098463   2.643  0.00833 **
rbob_ret    -0.0084854  0.0193649  -0.438  0.66133
sp500_ret    0.7180758  0.0288529  24.887  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02337 on 1171 degrees of freedom
Multiple R-squared:  0.3523,    Adjusted R-squared:  0.3495
F-statistic: 127.4 on 5 and 1171 DF,  p-value: < 2.2e-16

> vif(reg_model)
  crude_ret heatoil_ret  natgas_ret    rbob_ret   sp500_ret
   2.659139    2.691943    1.113179    2.357476    1.001256
```

Output 1.1: OLS with All Variables

```
Call:
lm(formula = xom_ret ~ crude_ret + heatoil_ret + natgas_ret +
    sp500_ret, data = percent)

Residuals:
      Min        1Q    Median        3Q       Max
-0.086812 -0.013257  0.000188  0.013436  0.093151

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0011690  0.0006825   1.713  0.08701 .
crude_ret   -0.0514339  0.0203034  -2.533  0.01143 *
heatoil_ret  0.0602839  0.0215125   2.802  0.00516 **
natgas_ret   0.0259605  0.0098418   2.638  0.00846 **
sp500_ret    0.7183008  0.0288384  24.908  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02336 on 1172 degrees of freedom
Multiple R-squared:  0.3522,    Adjusted R-squared:  0.3499
F-statistic: 159.3 on 4 and 1172 DF,  p-value: < 2.2e-16

> vif(reg_model_1)
  crude_ret heatoil_ret  natgas_ret   sp500_ret
   2.230253    2.346939    1.112944    1.000938
```

Output 1.2: OLS without RBOB



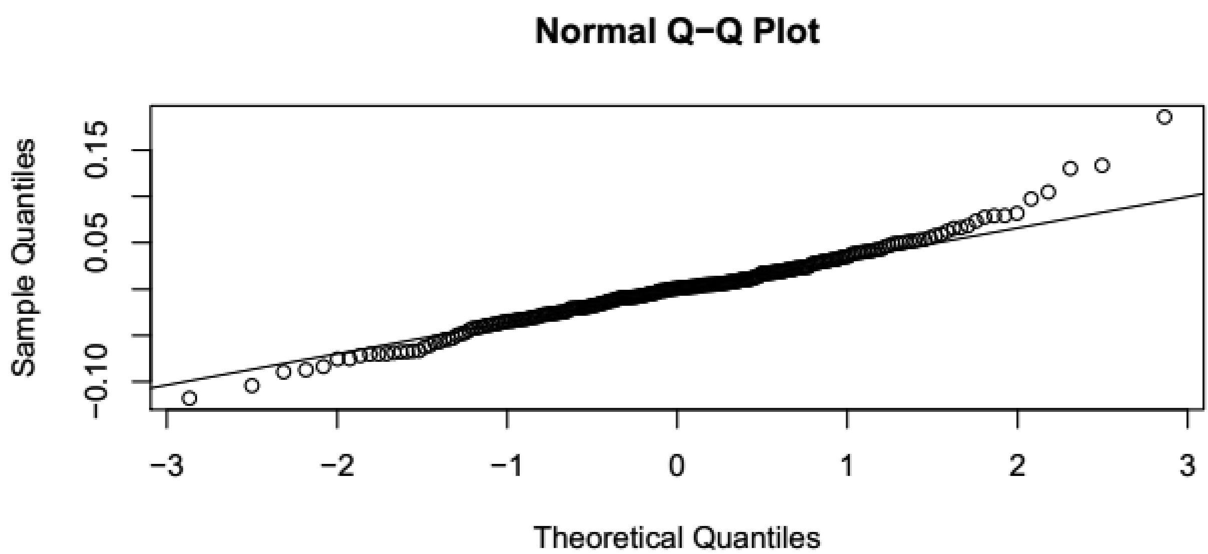
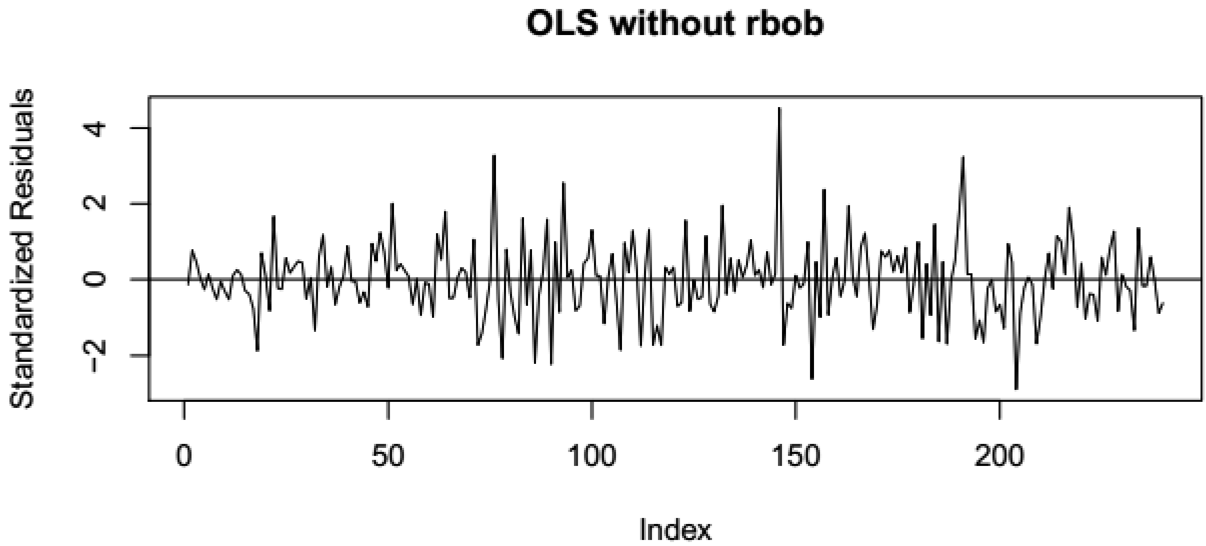


Figure 1.2: Diagnostics for OLS with All Variables



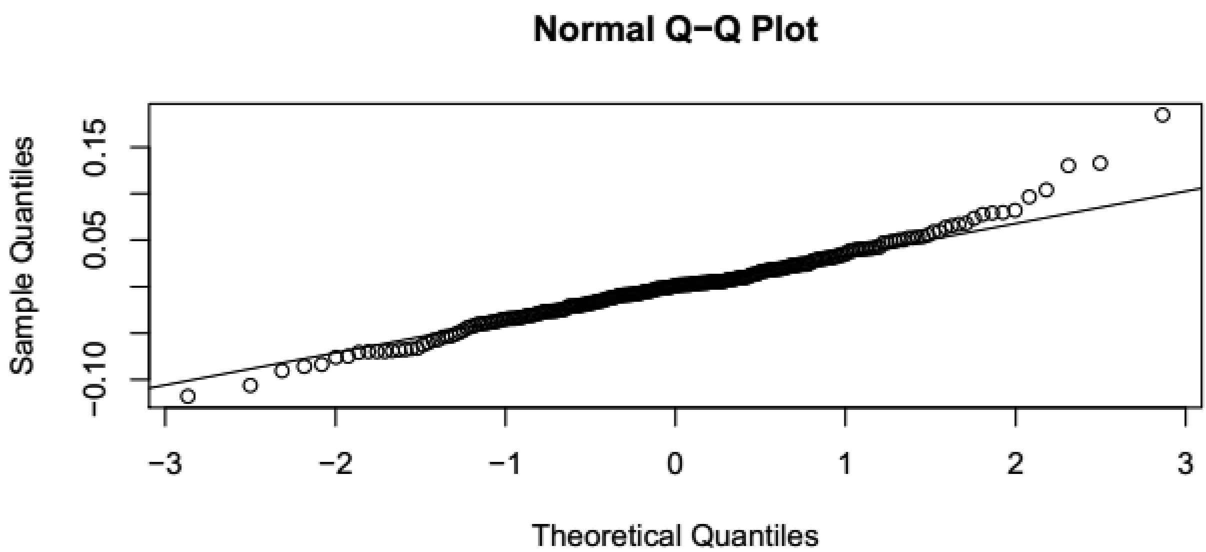


Figure 1.3: Diagnostics for OLS without rbob

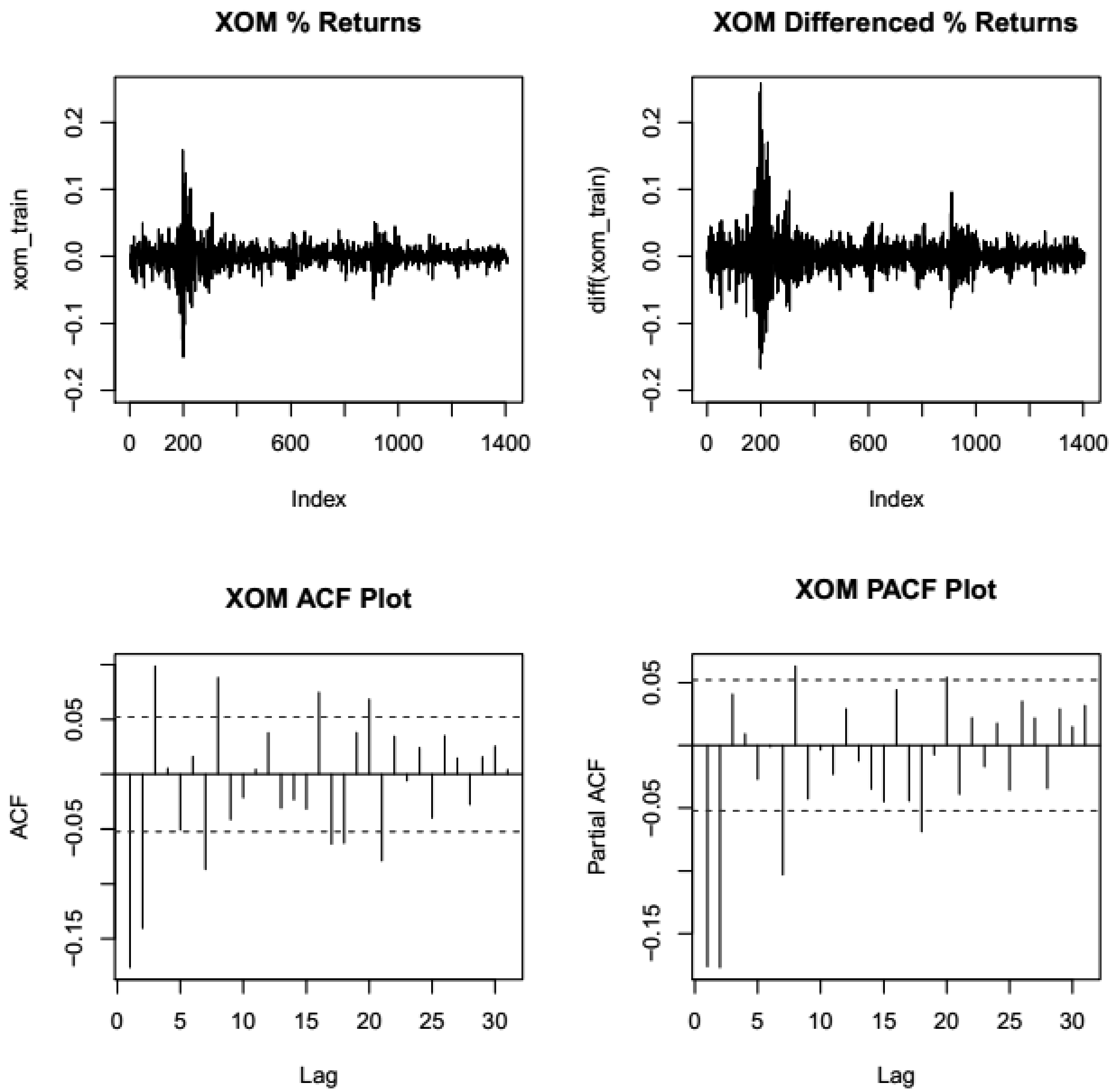


Figure 2.1: XOM Time Series Plots

```
> # auto.arima using aic
> #### arima(2,0,1) with non-zero mean
> auto_aic <- auto.arima(xom_train, max.P=0,max.Q=0,ic="aic"); auto_aic
Series: xom_train
ARIMA(2,0,1) with zero mean

Coefficients:
          ar1      ar2     ma1
      -0.3828  -0.2086  0.1811
s.e.   0.1197   0.0310  0.1215

sigma^2 estimated as 0.0003051:  log likelihood=3700.92
AIC=-7393.84   AICc=-7393.81   BIC=-7372.84
```

Output 2.1: ARIMA Model Using AIC Selection

```
> # auto.arima using bic
> #### arima(2,0,3) with zero mean
> auto_bic <- auto.arima(xom_train, max.P=0,max.Q=0,ic="bic"); auto_bic
Series: xom_train
ARIMA(2,0,3) with zero mean

Coefficients:
          ar1      ar2     ma1     ma2      ma3
      -1.4790  -0.6678  1.2932  0.2506  -0.2634
s.e.   0.0897   0.0840  0.0874  0.0859   0.0262

sigma^2 estimated as 0.0003023:  log likelihood=3707.29
AIC=-7402.57   AICc=-7402.51   BIC=-7371.07
```

Output 2.2: ARIMA Model Using BIC Selection

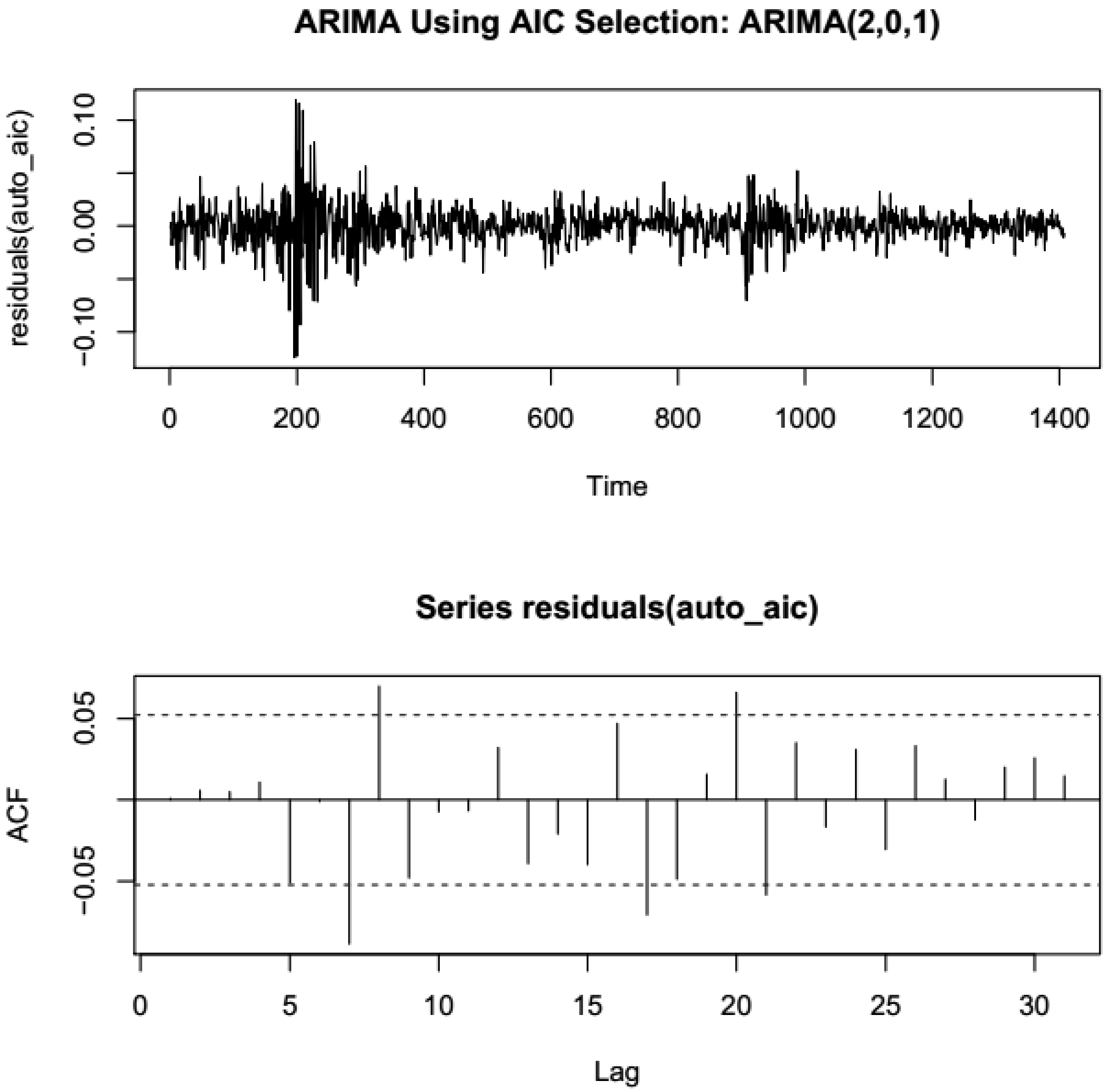
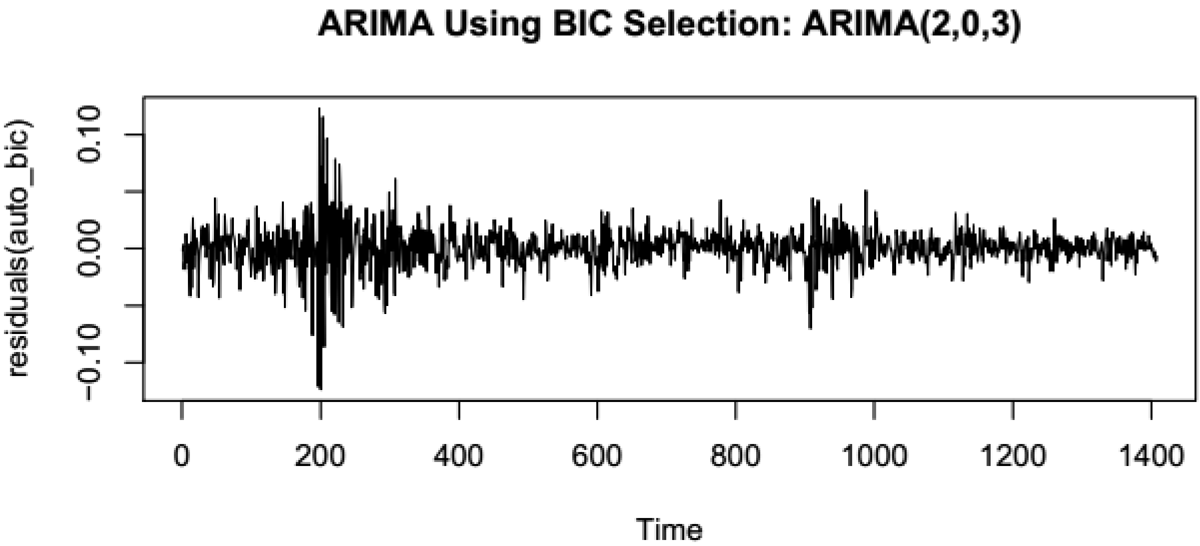


Figure 2.1: Diagnostics From Model Using AIC Selection



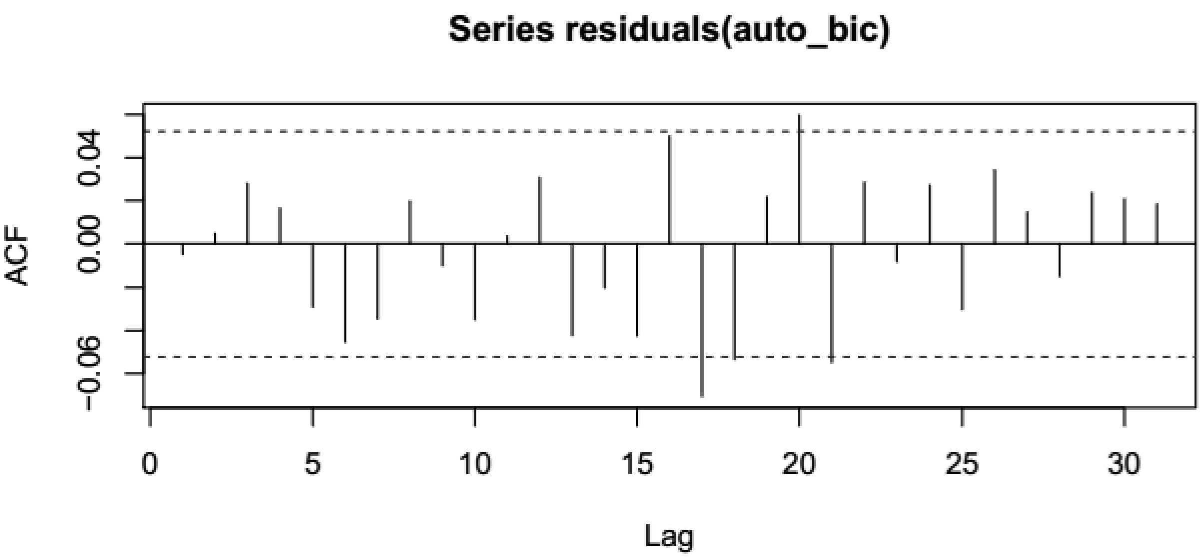


Figure 2.2: Diagnostics From Model Using BIC Selection

```
> # Ljung-Box test for volatility clustering by looking for autocorrelation
> #### results are indicative of autocorrelation
> resid_aic <- residuals(auto_aic)^2
> Box.test(resid_aic, lag=10, type="Ljung")

	Box-Ljung test

data:  resid_aic
X-squared = 1383.61, df = 10, p-value < 2.2e-16

> resid_bic <- residuals(auto_bic)^2
> Box.test(resid_bic, lag=10, type="Ljung")

	Box-Ljung test

data:  resid_bic
X-squared = 1410.45, df = 10, p-value < 2.2e-16
```

Output 2.3: Box-Ljung Test

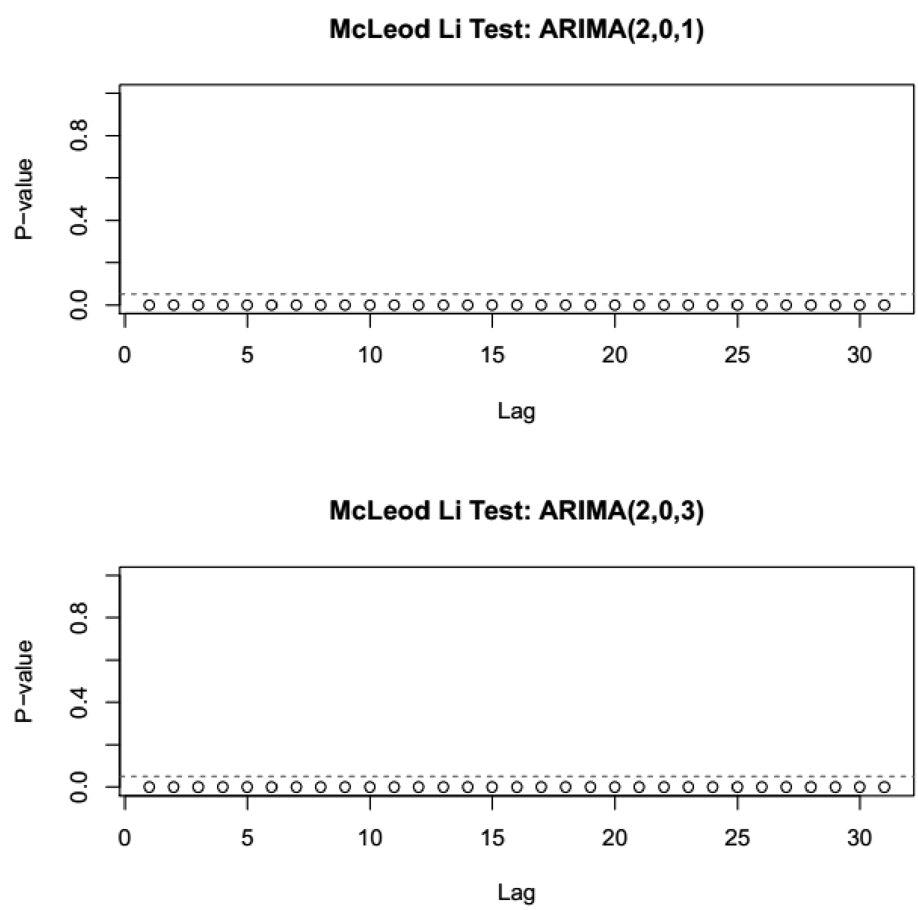


Figure 2.3: McLeod-Li Test

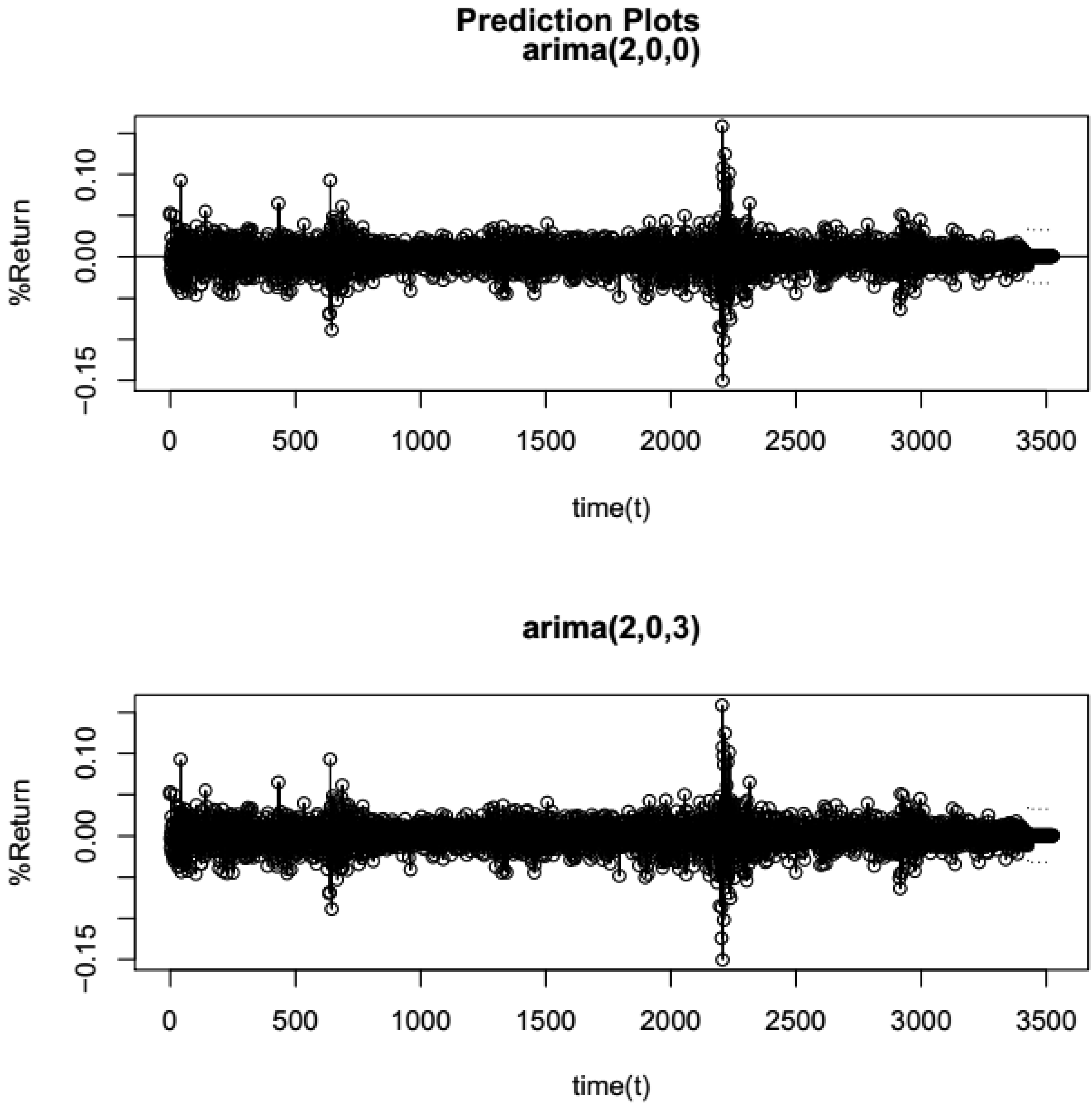


Figure 2.4: Predicted Values by Model

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(2, 1) + garch(1, 1), data = xom_train,
    trace = F)

Mean and Variance Equation:
 data ~ arma(2, 1) + garch(1, 1)
<environment: 0x00000000176339d8>
 [data = xom_train]

Conditional Distribution:
 norm

Coefficient(s):
         mu          ar1          ar2          ma1        omega       alpha1
 7.6863e-04  -9.1215e-01  -1.2696e-02   8.7216e-01   3.4623e-06   1.0273e-01
      beta1
 8.8386e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      7.686e-04   5.766e-04    1.333  0.18252
ar1    -9.122e-01   9.499e-02   -9.603  < 2e-16 ***
ar2    -1.270e-02   3.096e-02   -0.410  0.68176
ma1     8.722e-01   9.019e-02    9.670  < 2e-16 ***
omega   3.462e-06   1.081e-06    3.203  0.00136 **
alpha1  1.027e-01   1.624e-02    6.325 2.53e-10 ***
beta1   8.839e-01   1.685e-02   52.443  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 4066.141    normalized:  2.887885

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  111.4014  0
 Shapiro-Wilk Test  R    W      0.9872048 8.46521e-10
 Ljung-Box Test     R    Q(10)  5.17707   0.8790399
 Ljung-Box Test     R    Q(15)  7.631742  0.937697
 Ljung-Box Test     R    Q(20)  13.70545  0.845111
 Ljung-Box Test     R^2  Q(10)  14.47862  0.1522579
 Ljung-Box Test     R^2  Q(15)  18.07372  0.2588099
 Ljung-Box Test     R^2  Q(20)  22.94667  0.2914207
 LM Arch Test       R    TR^2   14.95446  0.2439365

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.765826 -5.739725 -5.765875 -5.756072
```

Output 2.4: AMRA(2,1)/GARCH(1,1)

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(2, 3) + garch(1, 1), data = xom_train,
    trace = F)

Mean and Variance Equation:
 data ~ arma(2, 3) + garch(1, 1)
<environment: 0x0000000017c2cd08>
 [data = xom_train]

Conditional Distribution:
 norm

Coefficient(s):
         mu          ar1          ar2          ma1          ma2          ma3
 1.1064e-04   6.8361e-02   7.0410e-01  -1.1287e-01  -6.9572e-01  -1.9722e-02
      omega       alpha1        beta1
 3.4891e-06   1.0372e-01   8.8292e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      1.106e-04   7.607e-05    1.454  0.14583
ar1     6.836e-02   1.522e-01    0.449  0.65327
ar2     7.041e-01   1.431e-01    4.921 8.60e-07 ***
ma1    -1.129e-01   1.542e-01   -0.732  0.46419
ma2    -6.957e-01   1.493e-01   -4.660 3.16e-06 ***
ma3    -1.972e-02   3.258e-02   -0.605  0.54500
omega   3.489e-06   1.088e-06    3.207  0.00134 **
alpha1  1.037e-01   1.640e-02    6.325 2.53e-10 ***
beta1   8.829e-01   1.701e-02   51.918  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 4069.313    normalized:  2.890137

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  121.5516  0
 Shapiro-Wilk Test  R    W      0.9863639 3.112782e-10
 Ljung-Box Test     R    Q(10)  5.366769  0.8653706
 Ljung-Box Test     R    Q(15)  6.95979   0.9587559
 Ljung-Box Test     R    Q(20)  13.26295  0.8658188
 Ljung-Box Test     R^2  Q(10)  14.59889  0.147384
 Ljung-Box Test     R^2  Q(15)  18.90324  0.2181523
 Ljung-Box Test     R^2  Q(20)  23.24884  0.2767419
 LM Arch Test       R    TR^2   15.02019  0.2403343

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.767489 -5.733932 -5.767571 -5.754948
```

Output 2.5: AMRA(2,3)/GARCH(1,1)

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~arma(1, 1) + garch(1, 1), data = xom_train,
    trace = F)

Mean and Variance Equation:
 data ~ arma(1, 1) + garch(1, 1)
<environment: 0x0000000017789c78>
 [data = xom_train]

Conditional Distribution:
 norm

Coefficient(s):
         mu          ar1          ma1        omega       alpha1        beta1
 6.5460e-05   8.6057e-01  -8.9229e-01   3.4167e-06   1.0172e-01   8.8513e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      6.546e-05   4.890e-05    1.339  0.18070
ar1     8.606e-01   7.566e-02   11.375  < 2e-16 ***
ma1    -8.923e-01   6.527e-02  -13.671  < 2e-16 ***
omega   3.417e-06   1.060e-06    3.223  0.00127 **
alpha1  1.017e-01   1.587e-02    6.411 1.45e-10 ***
beta1   8.851e-01   1.642e-02   53.892  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 4066.194    normalized:  2.887922

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  130.5344  0
 Shapiro-Wilk Test  R    W      0.9854392 1.081442e-10
 Ljung-Box Test     R    Q(10)  9.030745  0.5291882
 Ljung-Box Test     R    Q(15)  10.93382  0.7572717
 Ljung-Box Test     R    Q(20)  17.01234  0.652172
 Ljung-Box Test     R^2  Q(10)  13.35712  0.2043843
 Ljung-Box Test     R^2  Q(15)  17.38755  0.2962292
 Ljung-Box Test     R^2  Q(20)  21.60478  0.3623353
 LM Arch Test       R    TR^2   14.08145  0.2955371

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-5.767321 -5.744949 -5.767357 -5.758960
```

Output 2.6: AMRA(1,1)/GARCH(1,1)

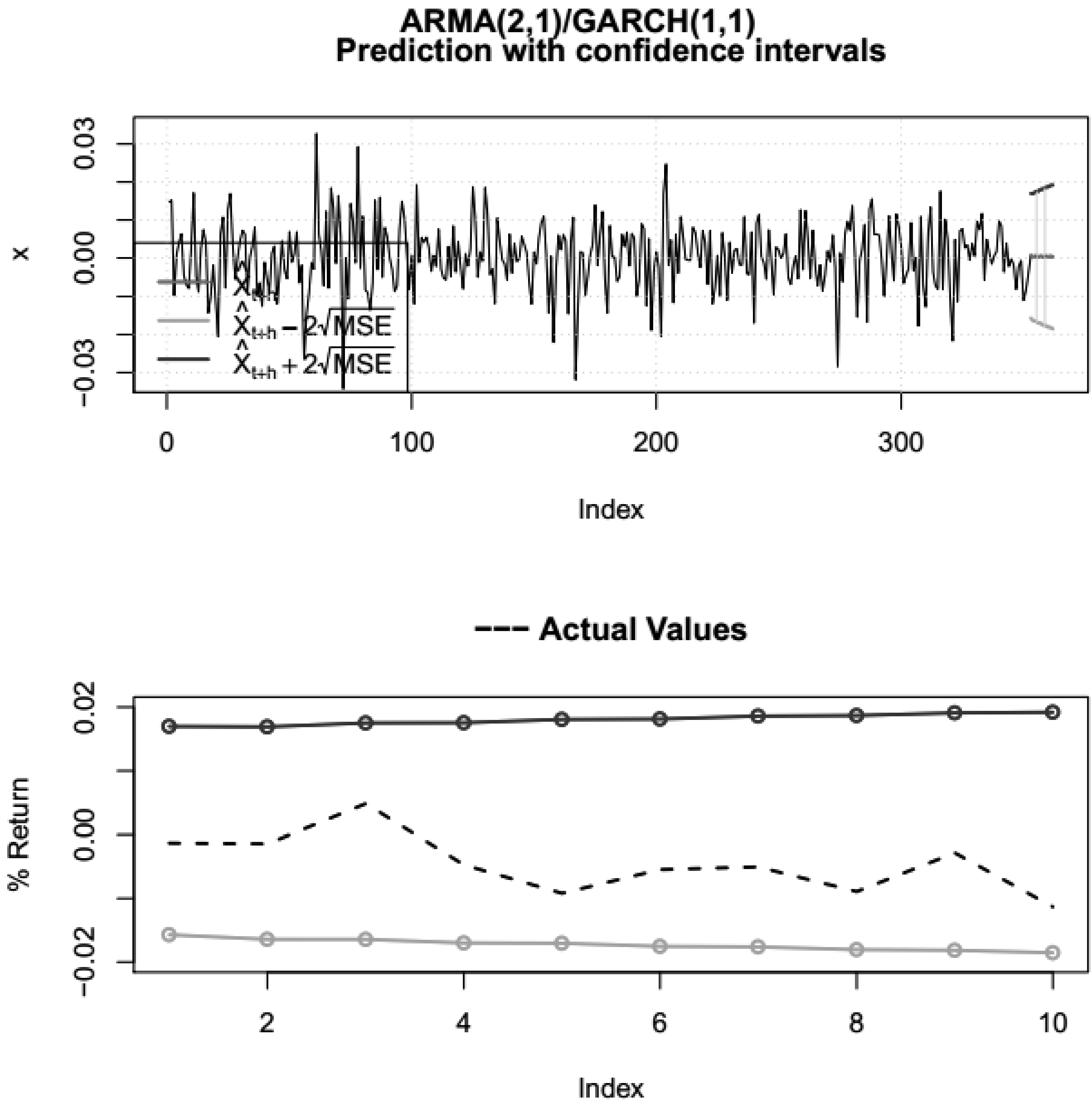


Figure 2.4: AMRA(2,1)/GARCH(1,1)

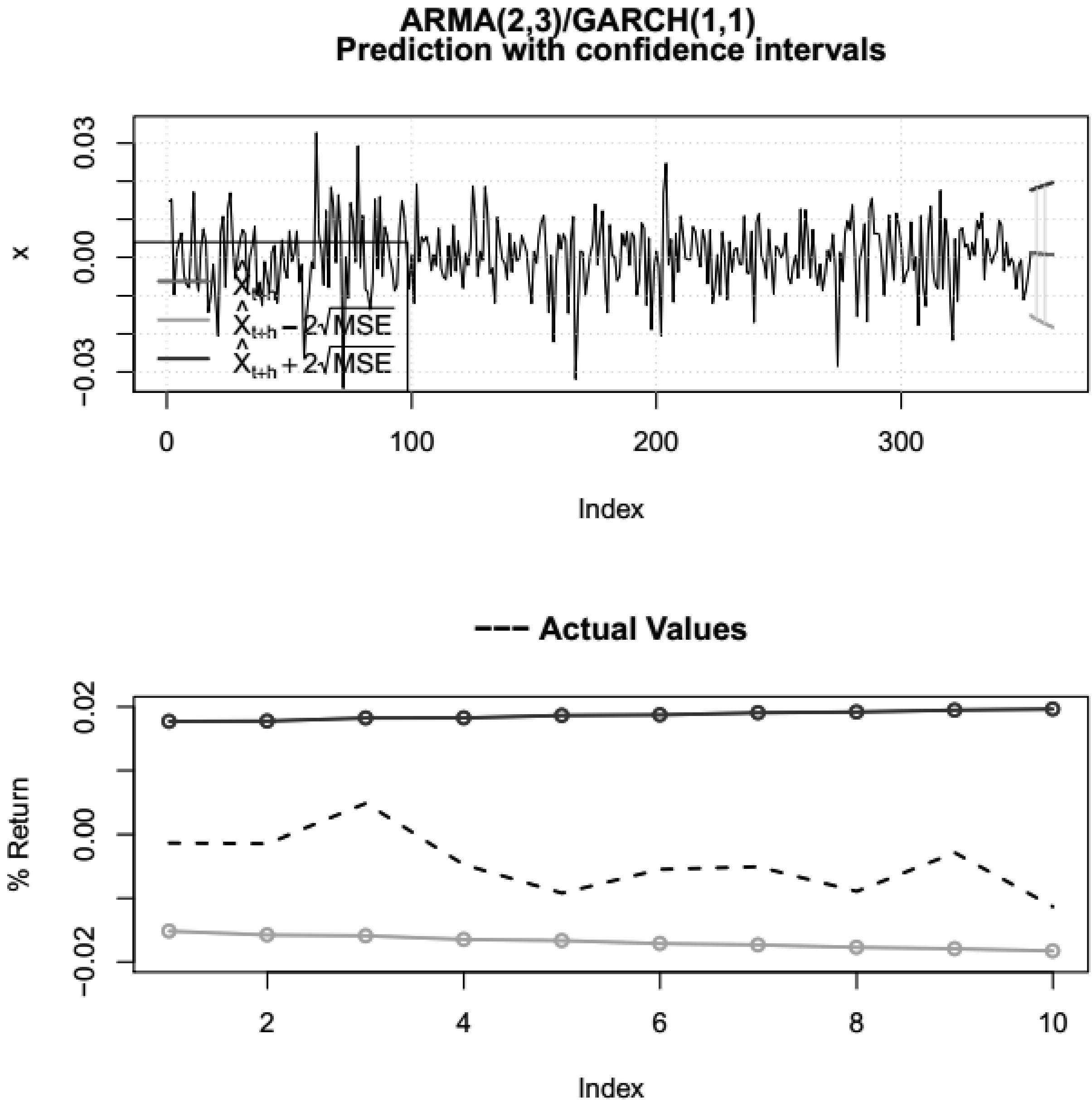


Figure 2.5: AMRA(2,3)/GARCH(1,1)

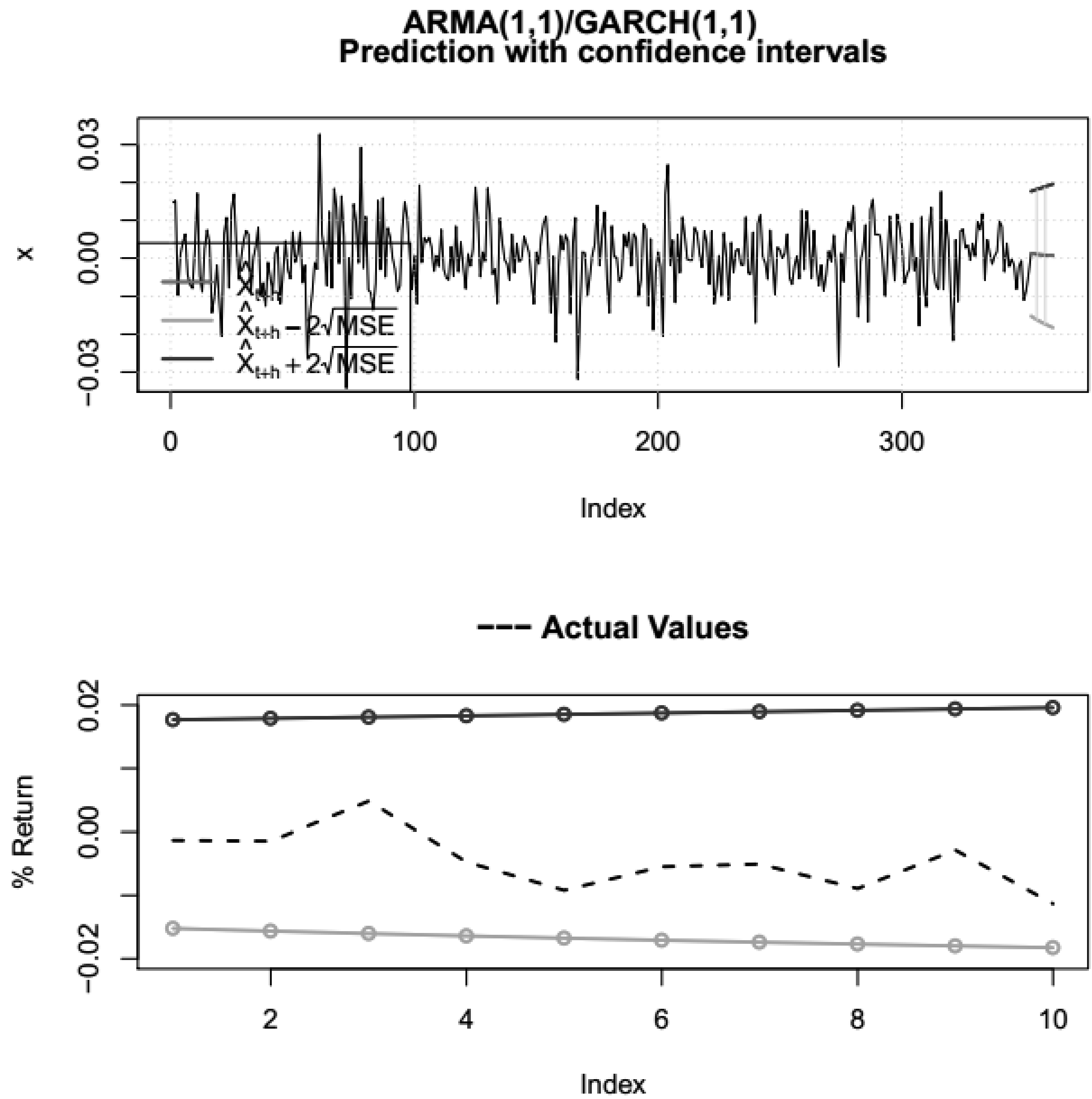


Figure 2.6: AMRA(1,1)/GARCH(1,1)

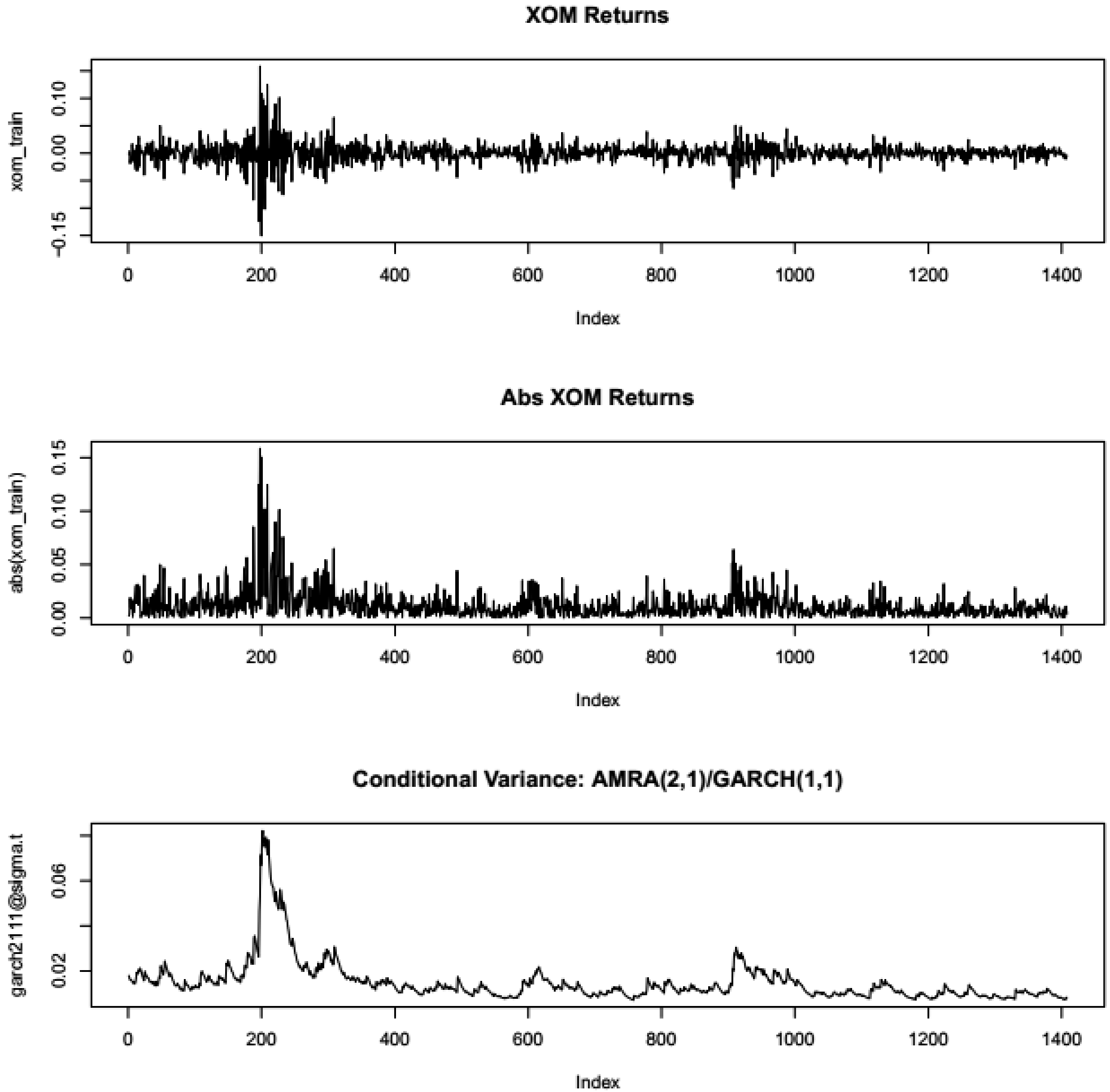


Figure 2.7: Returns & Conditional Variance AMRA(2,1)/GARCH(1,1)

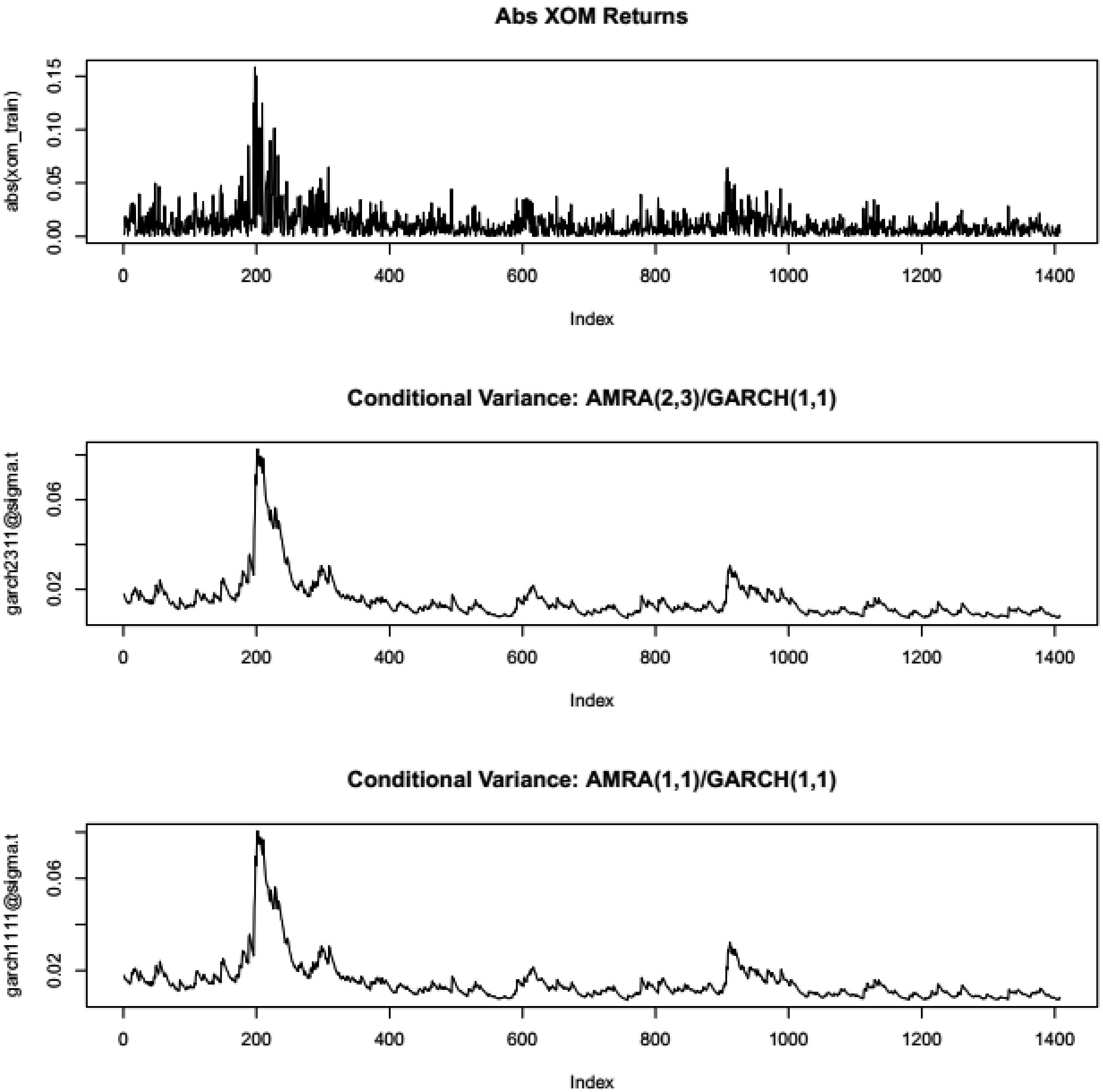


Figure 2.8: Conditional Variance for AMRA(2,3)/GARCH(1,1) and AMRA(1,1)/ GARCH(1,1)

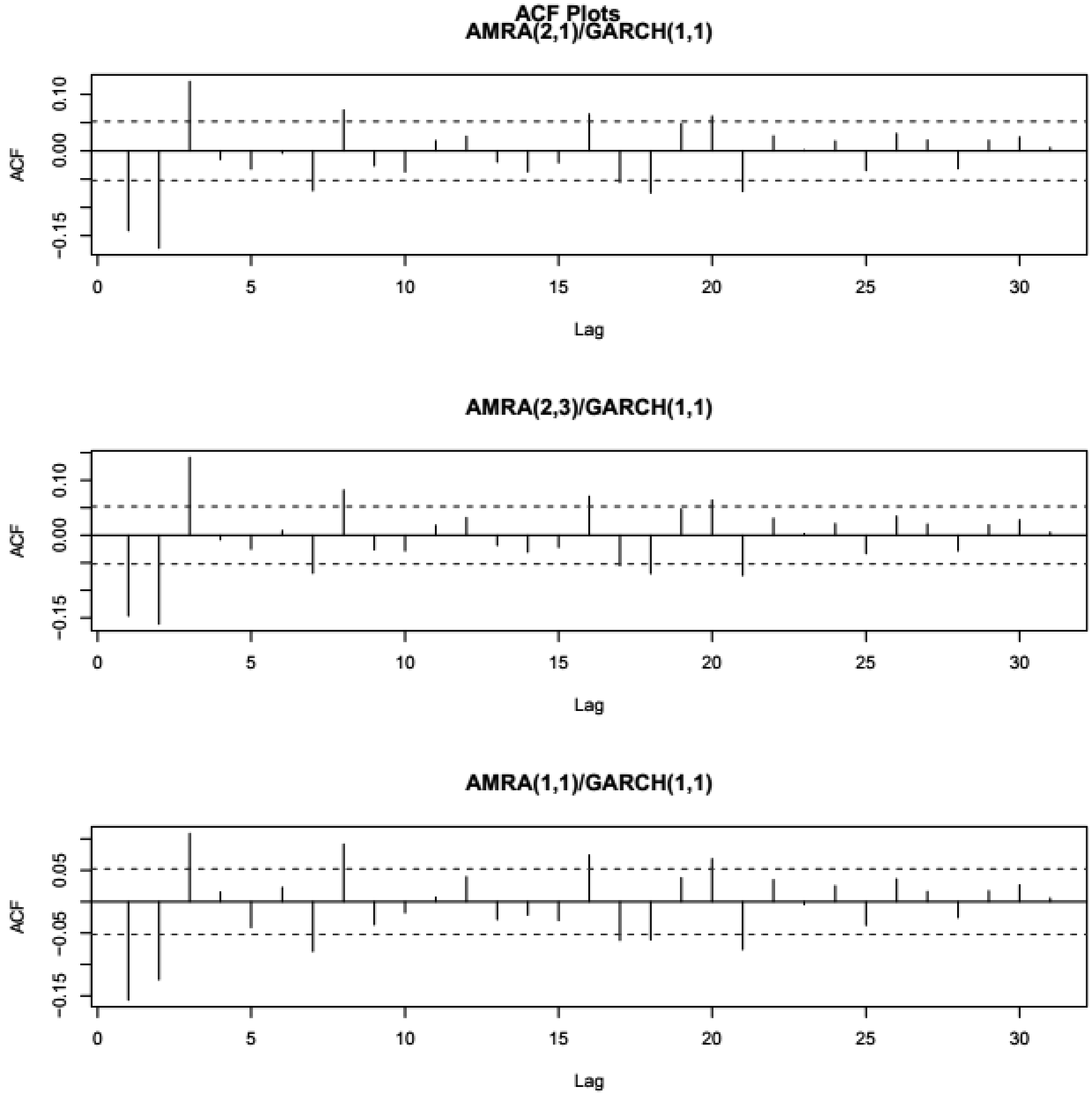


Figure 2.9: ACF Plots for All GARCH Models

Sentiment Analysis Model

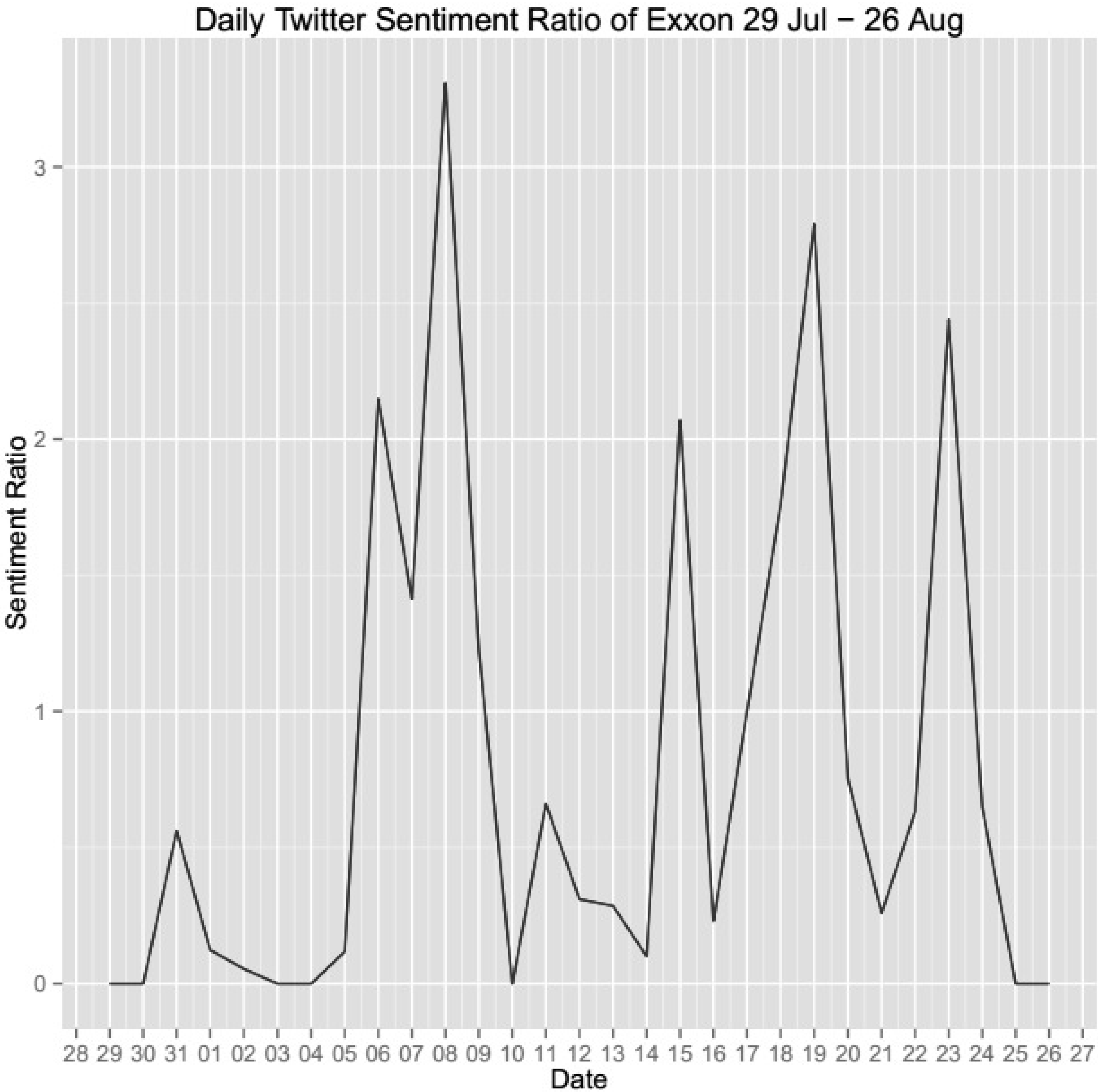


Figure 3.1: Daily Twitter Sentiment Index

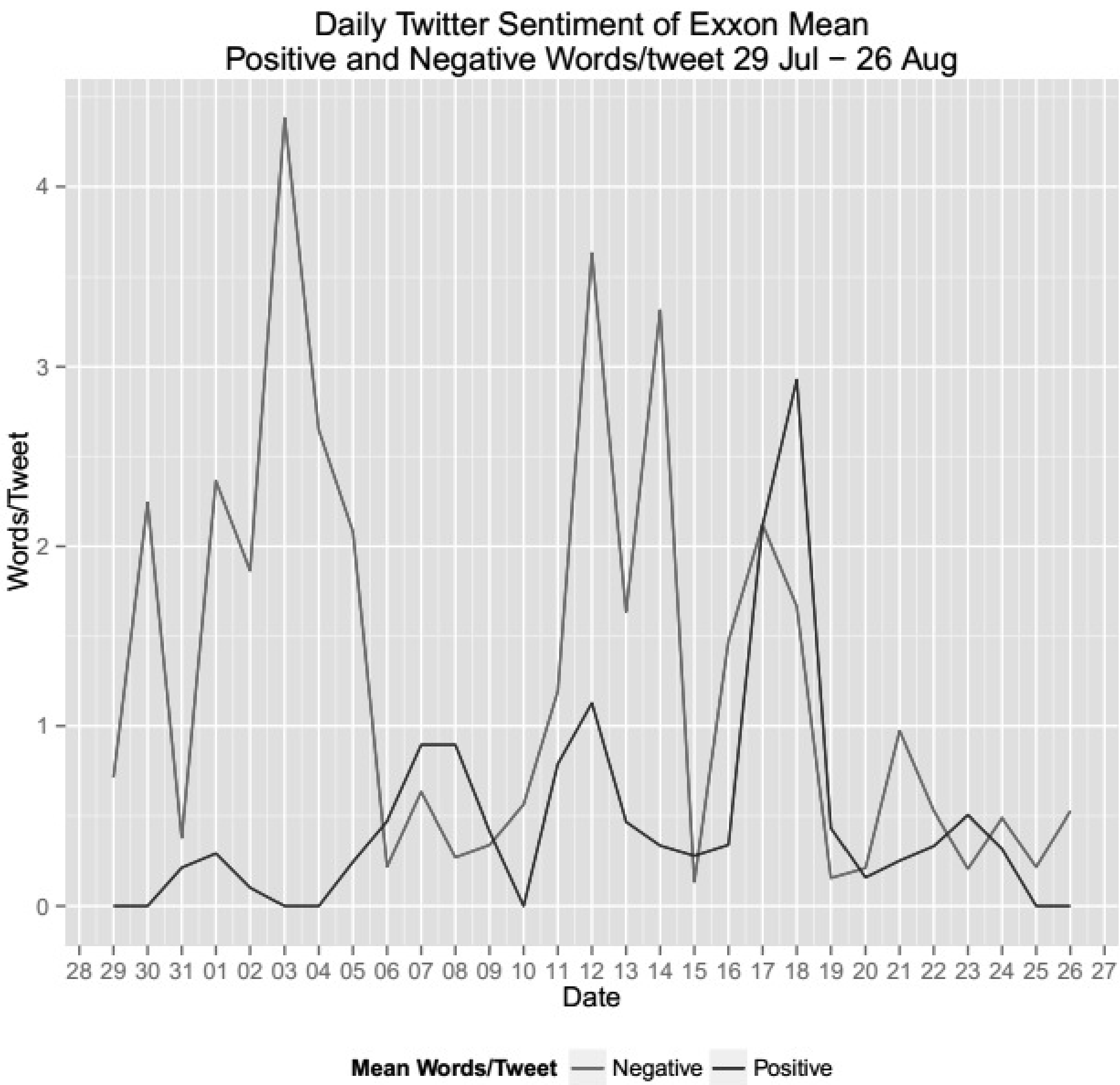


Figure 3.2: Mean Positive/Negative Word per Tweet

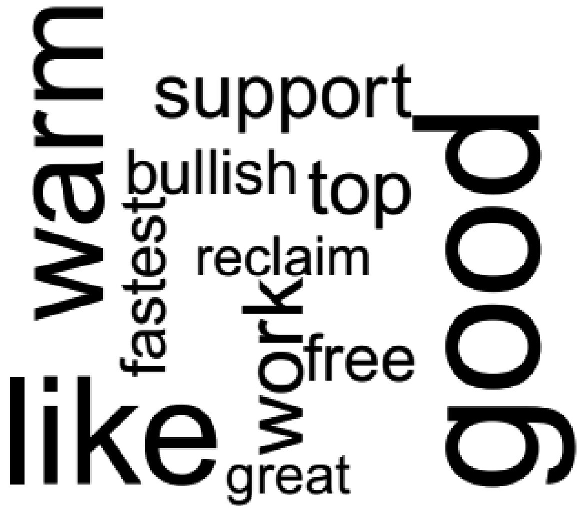


Figure 3.3: Positive Word Cloud

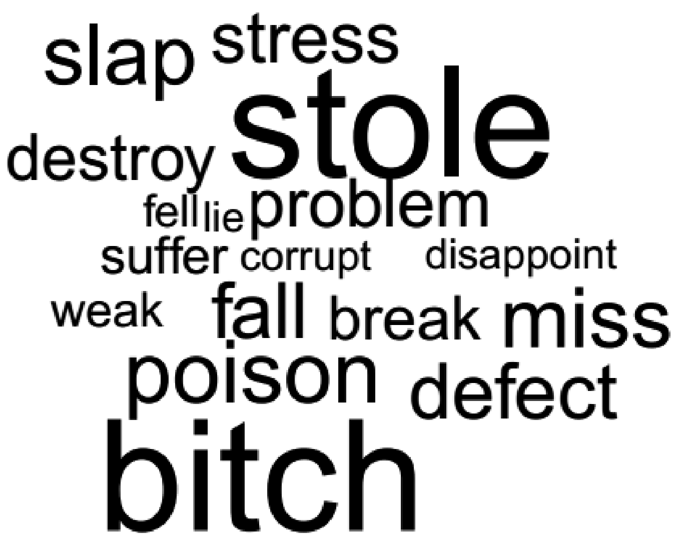


Figure 3.4: Negative Word Cloud

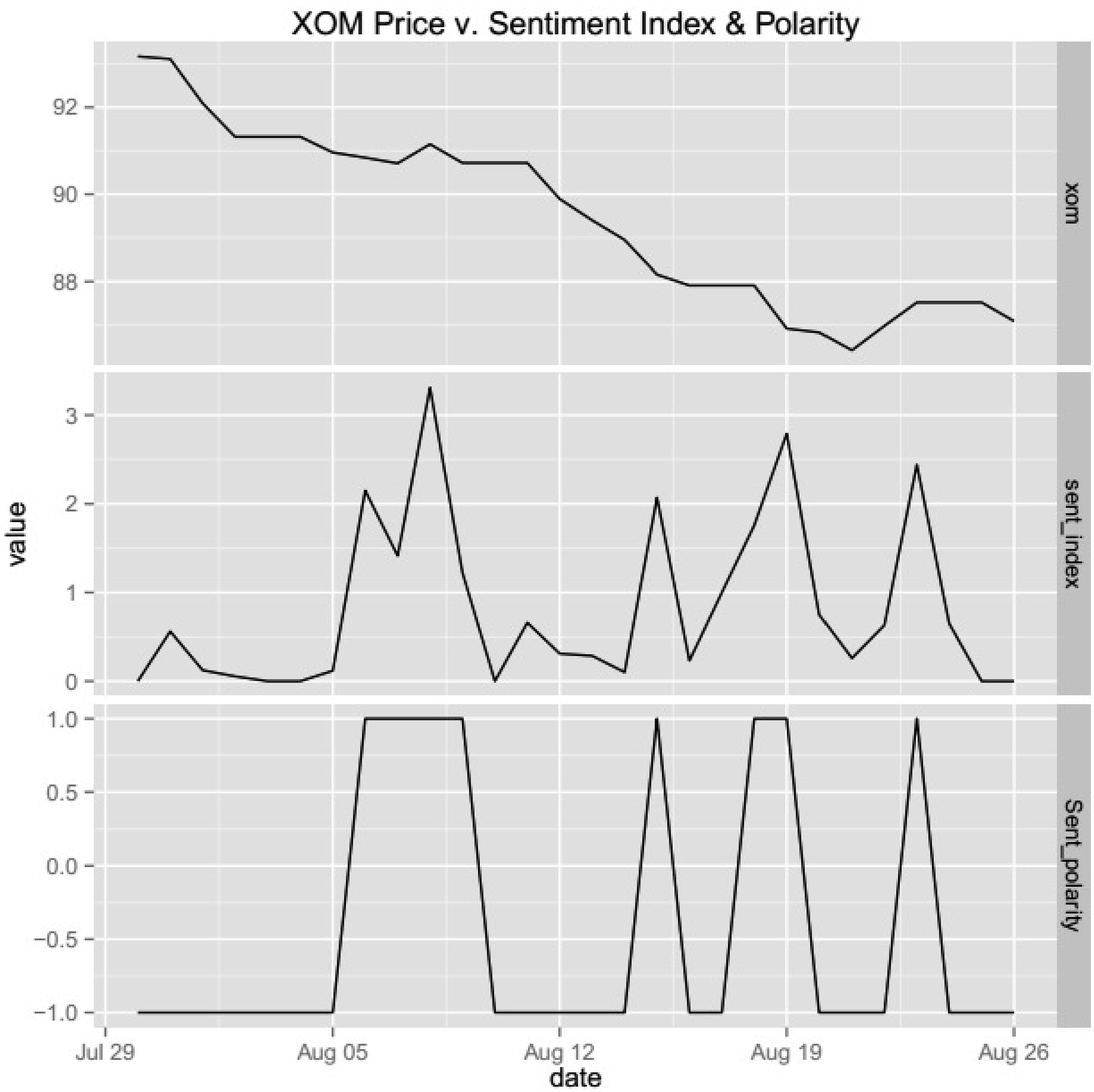


Figure 3.5: Stock Price and Sentiment 29 Jul - 26 Aug

	Stock Price	Sentiment
ADF Test Ho: Series is not stationary	-2.2226 (0.4874)	-2.0779 (0.5427)
KPSS Test Ho: Series is level or trend stationary	1.4017 (<0.01)	0.1386 (0.1)

Table 3.1: ADF & KPSS Tests

	Stock Price, diff=2	Sentiment, diff=1
ADF Test Ho: Series is not stationary	-3.6246 (0.04824)	-4.5592 (0.01)
KPSS Test Ho: Series is level or trend stationary	0.0437 (>0.1)	0.067 (>0.1)

Table 3.2: ADF & KPSS Tests (Differenced)

Granger Causality Test	Chi-Sq	df	P-value
Ho: Sentiment does not Granger Cause Stock Price	2.6	3	0.46
Ho: Stock price does not Granger Cause Sentiment	1.6	3	0.67

Table 3.3: Granger Causality Test

```
Call:
lm(formula = percent_response ~ percent_lag + sent_index_lag +
    sent_dummy_lag)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0077609 -0.0034204 -0.0000303  0.0024334  0.0104356

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.0027988  0.0021871  -1.280    0.219
percent_lag     0.1954310  0.2606750   0.750    0.464
sent_index_lag -0.0008395  0.0029438  -0.285    0.779
sent_dummy_lag  0.0023651  0.0061060   0.387    0.704

Residual standard error: 0.005479 on 16 degrees of freedom
Multiple R-squared:  0.0449,    Adjusted R-squared:  -0.1342
F-statistic: 0.2508 on 3 and 16 DF,  p-value: 0.8596
```

Output 4.1: Combined Model with Stock & Sentiment Data

```
Call:
lm(formula = percent_response ~ percent_lag + sent_dummy_lag)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0077657 -0.0034565 -0.0000785  0.0025999  0.0104279

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.0031328  0.0017967  -1.744   0.0993 .
percent_lag     0.1691804  0.2371999   0.713   0.4854
sent_dummy_lag  0.0007911  0.0025399   0.311   0.7592
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.005328 on 17 degrees of freedom
Multiple R-squared:  0.04005,   Adjusted R-squared:  -0.07288
F-statistic: 0.3546 on 2 and 17 DF,  p-value: 0.7065
```

Output 4.2: Combined Model with Sentiment Index Removed

```
Call:
lm(formula = percent_response ~ percent_lag)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0080831 -0.0033144 -0.0002863  0.0023213  0.0101653

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.002807   0.001425  -1.971   0.0643 .
percent_lag  0.182540   0.227363   0.803   0.4325
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.005193 on 18 degrees of freedom
Multiple R-squared:  0.03457,   Adjusted R-squared:  -0.01906
F-statistic: 0.6446 on 1 and 18 DF,  p-value: 0.4325
```

Output 4.3: Combined Model with Sentiment Index & Polarity Removed

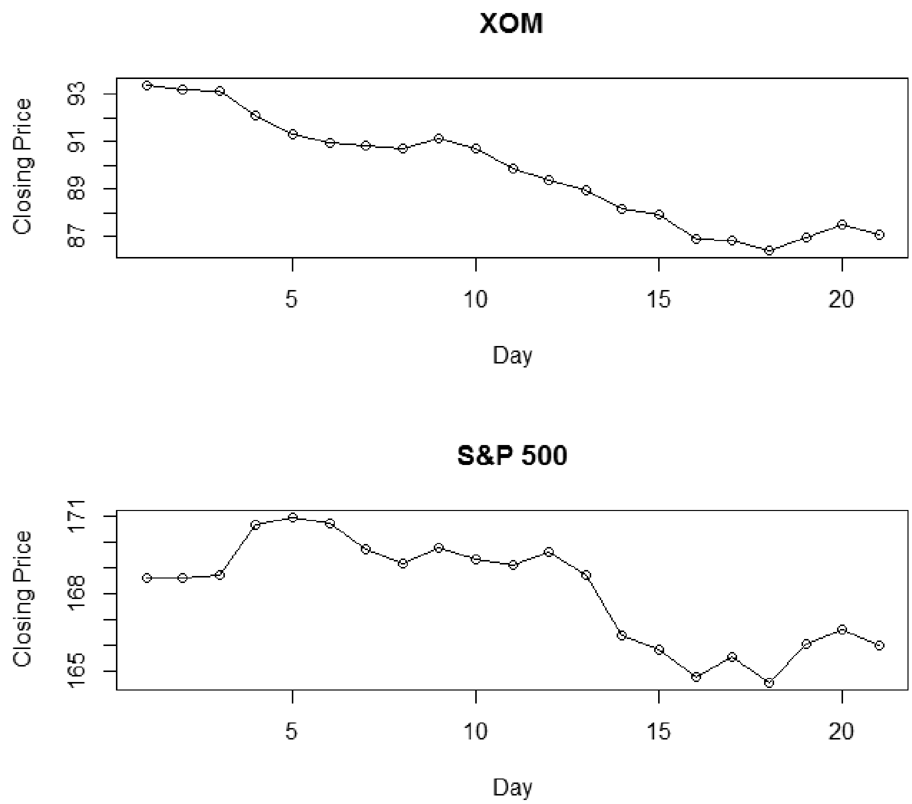


Figure 4.1: Price History of XOM and S&P500 Over Analysis Timer Period