Modeling S&P 500 data

Raymond N Saitoti

March 2nd 2020

```
knitr::opts_chunk$set(echo = TRUE)

library(gridExtra)

library("knitr")

library("tidyr")

library("tidyverse")

library("leaflet")

library("dplyr")

library("forecast")

library("tseries")

library("rugarch")

library("TSA")

library("ggplot2")

library(mosaic)
```

```
all_stocks <- read_csv("all_stocks_5yr.csv")

#Data chosen for analysis

CHK <- all_stocks %>%

  filter(Name == "CHK")
```

Abstract

This project is a basic exploration of methods modelling the changing variance in time series model. More specifically it is an introduction to Autoregressive Conditional Heteroskedasticity and Generalized Autoregressive Conditional Heteroskedasticity models `ARCH` and `GARCH`. It is an extension of the exploration and application of the Auto Regressive Integrated Moving Average methods `ARIMA` used to model future air quality measures conducted as part of the major's STAT 495 final project. This study is meant to show the application of these novel methods in modelling the changing variance of time of a time series. The data used in this study is obtained from the S&P 500 index, which is a measure that estimates the performance of 500 companies listed in the United States Stock exchange market. The data used includes Chesapeake Energy Corporation's daily stock data spanning the years 2013 - 2018.

Introduction

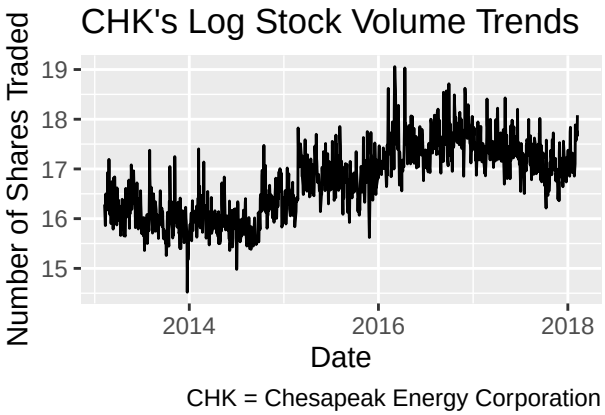
ARIMA Models

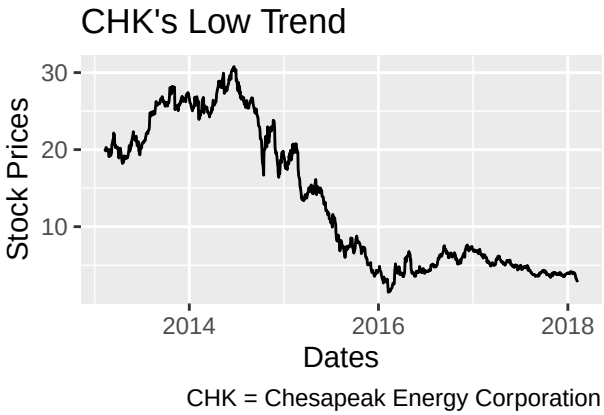
ARIMA stands for Auto Regressive Moving Average models. They consist of two components; the Autoregressive Component and the Moving Average Component. Is denoted by `ARIMA(p,d,q)`, with p representing the number of autoregressive terms, d the number of differencing and q the number of Moving Average terms.

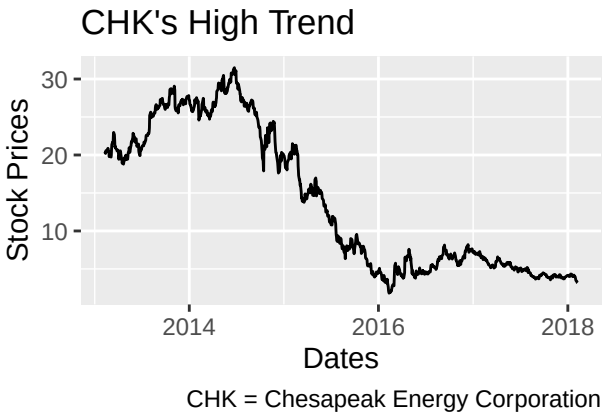
Before we begin any model fitting, we make the line graphs shown below. This step is meant as an initial exploration aimed at showing the trends in volume and price of Chesapeake Energy Corporation's stocks. All the plots below are similar because they show that the volume, opening, high and low values exhibited high volatility. Volatility in a time series refers to the phenomenon where the conditional variance of the time series varies over time (Cryer and Chan, 2008). Stock volume seems more volatile than low, high and close prices, as can be seen from the more sudden irregular shifts in trends with time.

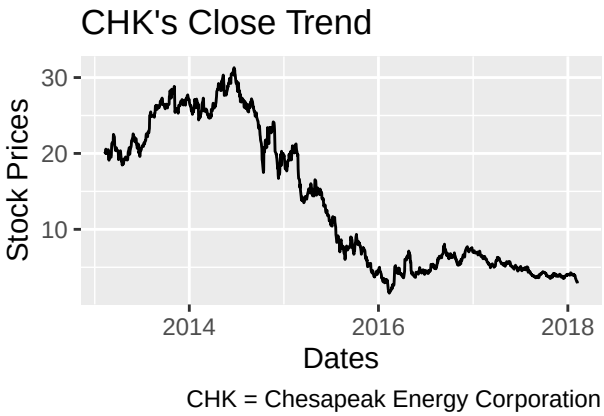
The data wrangling required before fitting the model/ checking conditions is minimal. It begins by decomposing Chesapeake monthly data time series into seasonal, trend and irregular components, then moves on to the removal of seasonal components to create a seasonally adjusted component.

```
Volume_plot <- ggplot(data = CHK, aes(x = date, y = log(volume))) + geom_line() +
  ylab("Number of Shares Traded") +
  xlab("Date") + labs(title = "CHK's Log Stock Volume Trends",
                                              caption = "CHK = Chesapeak Energy Corporation" )
Low_plot <- ggplot()+geom_line(data = CHK, aes(x = date, y = low), color = "black") +
  ylab("Stock Prices") + xlab("Dates") + labs(title = "CHK's Low Trend",
                                              caption = "CHK = Chesapeak Energy Corporation" )
High_plot <- ggplot()+geom_line(data = CHK, aes(x = date, y = high), color = "black")+
  ylab("Stock Prices") + xlab("Dates") + labs(title = "CHK's High Trend",
                                              caption = "CHK = Chesapeak Energy Corporation" )
Close_plot <- ggplot()+geom_line(data = CHK, aes(x = date, y = close), color = "black") +
  ylab("Stock Prices") + xlab("Dates") + labs(title = "CHK's Close Trend",
                                              caption = "CHK = Chesapeak Energy Corporation" )
grid.arrange(Volume_plot, Low_plot, High_plot, Close_plot, ncol = 2)
```









Model Conditons

Before fitting an ARIMA time series model, we need to make sure that it is free of trends and non seasonal behavior. We also check to make sure that the time series has a constant mean and variance. If variation in trends is present, we difference in order to get rid of those trends and prepare the data for model fitting. After all these checks are performed, we run the Augmented Dickey Fuller test to make sure that stationarity is satisfied :

The hypothesis test to check whether our data is stationary is as follows :

H_0:Chesapeake time series is not stationary.

H_A:Chesapeake time series is stationary.

Our test yields a statistic of -3.5423 and a p value of 0.03823. We therefore reject the null as there is strong evidence of stationarity in the data, and proceed to the model fitting phase.

```
#adf test checks whether ts is stationary or not (data condition)
adf.test(CHK_deseasonal_value, alternative = "stationary")
```

```
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  CHK_deseasonal_value
## Dickey-Fuller = -3.5423, Lag order = 10, p-value = 0.03823
## alternative hypothesis: stationary
```

Model Fitting

Here, we use the `Auto.arima()` function to help obtain model parameters using a stepwise model fitting procedure. This model selection procedure selects the model with the lowest AIC value. The p, d, q parameters of the model will be selected from the model with the lowest score. We start with a maximum order of 6 for all parameters and iterate through different combinations to find the one the produces the model with the lowest AIC score :

```
#auto fits arima model
CHK_fit = auto.arima(CHK_deseasonal_value, max.order = 6)
CHK_fit
```

```
## Series: CHK_deseasonal_value
## ARIMA(4,1,3)
##
## Coefficients:
##          ar1     ar2     ar3      ar4     ma1      ma2      ma3
##       0.2097  0.3550  0.6572  -0.4885  0.2163  -0.3094  -0.7638
## s.e.  0.0449  0.0418  0.0359   0.0263  0.0468   0.0452   0.0373
##
## sigma^2 estimated as 1.864e+12:  log likelihood=-19460.3
## AIC=38936.59   AICc=38936.71   BIC=38977.65
```

The model chosen from our selection procedure has 4 Autoregressive Terms i.e **AR(4)**, a differencing of degree 1 and 3 moving average terms i.e **MA(3)**. The fitted model from the parameters obtained above can be expressed as :

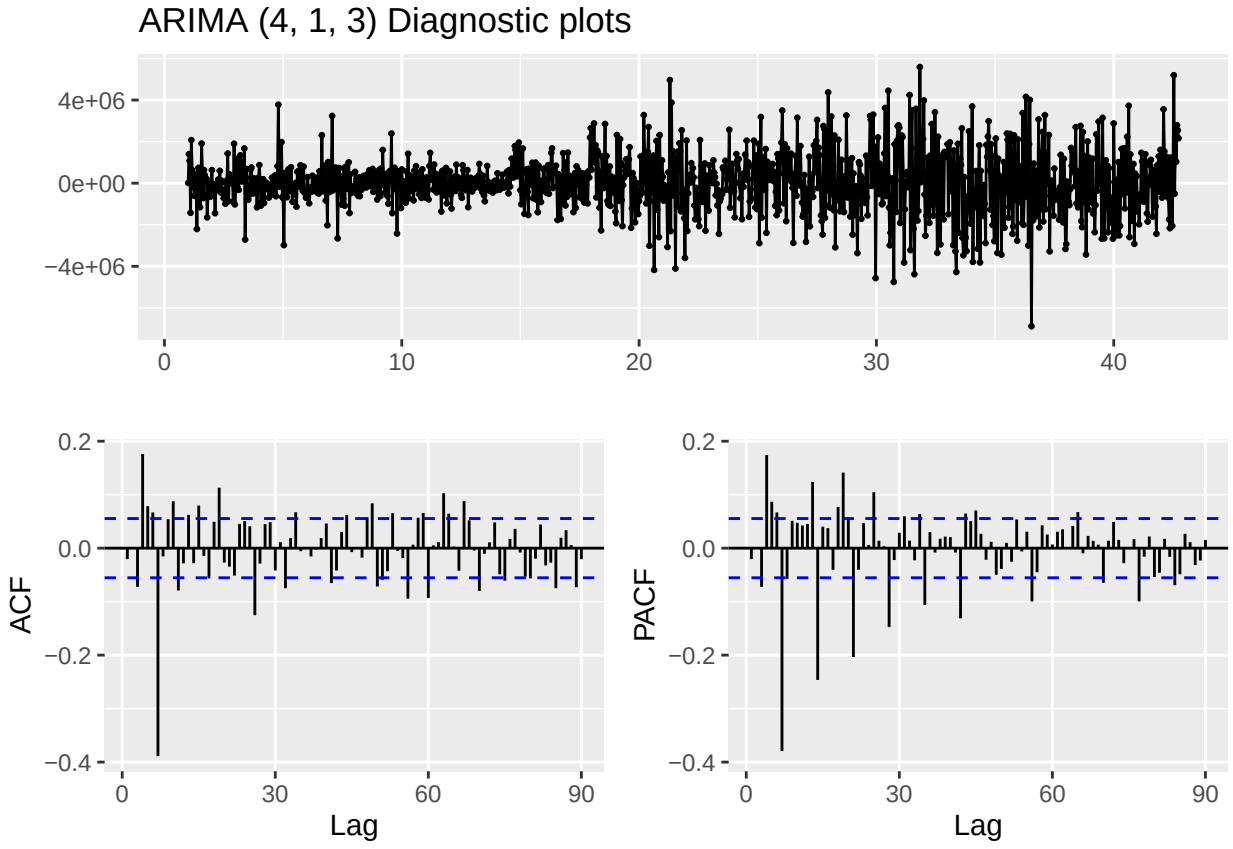
$$\hat{Y}_t = 0.2097Y_{t-1} + 0.3550Y_{t-2} + 0.6572Y_{t-3} - 0.4885Y_{t-4} + 0.2163e_{t-1} - 0.3094e_{t-2} - 0.7638e_{t-3} + \epsilon$$

The equation above is a linear combination of terms. The Y's correspond to recent stock volume values up until the $(t-4)^{th}$ time step while the e's correspond to the errors of the lags at the denoted, corresponding time steps.

In the next section we shall examine sample partial autocorrelation (PACF) and sample autocorrelation plots (ACF) to validate our choices of the p, d and q orders chosen for our ARIMA model by the stepwise model selection procedure.

Model Diagnostics

Before settling down on the model obtained from the previous section, we examine the auto correlation of residuals from the fitted model in order to make sure the model is free of auto correlations. This is necessary because it helps us establish whether the noise terms in the model are independent from each other :



ARCH/ GARCH Models

Introduction.

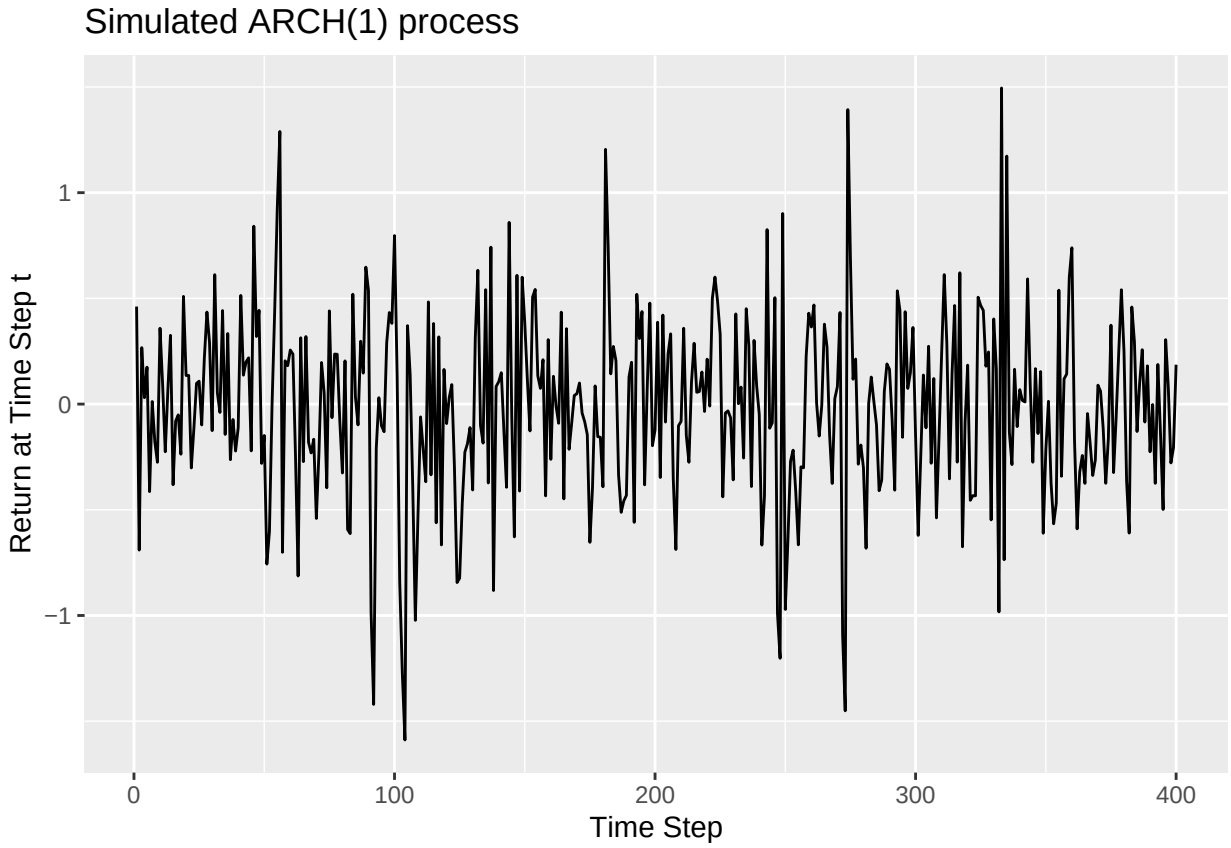
In the previous chapter we tried fitting and assessing the feasibility of an ARIMA model on Chesapeake Energy Corporation's stock data. The fitted model wasn't appropriate because the model's error terms were not independent and identically distributed. In addition, cluster volatility seemed to be a huge issue as observed from the heteroschedastic nature of modeled stock volume returns.

In this section we will use autoregressive conditional heteroschedastic models in an attempt to adequately capture and account for the heteroscedasticity observed in the ARIMA model. ARCH/GARCH are time series models used to model processes where volatility is high in provided data (Cryer and Chan, 2008). Cases involving stock market data are usually prone to unpredictable changes, and are best modeled using methods that model the variability of future values based on present and past provided trend in observed returns.

ARCH models.

ARCH models are denoted `ARCH(p)`, where p represents the order of the model. According to Cryer and Chan (2008) an `ARCH(1)` process modelling the return of a time series r takes the form $r_t = \sigma_{t|t-1}\varepsilon_t$ where ε_t is a series of independent and identically distributed random variables with a mean of zero and standard deviation of 1. The quantity $\sigma_{t|t-1}$ models the conditional variance, $\sigma^2_{t|t-1}$, of the return r_t, and is given by $\sigma^2_{t|t-1} = \omega + \alpha r^2_{t-1}$. The variance of the current return is based on conditioning upon returns until the $(t-1)^{th}$ time step. The quantities ω and α represent the ARCH model intercept and coefficient respectively.

The diagram below represents a sample `ARCH(1)` process simulated with $\omega = 0.1$ and $\alpha = 0.5$ as the chosen model parameters.



GARCH models.

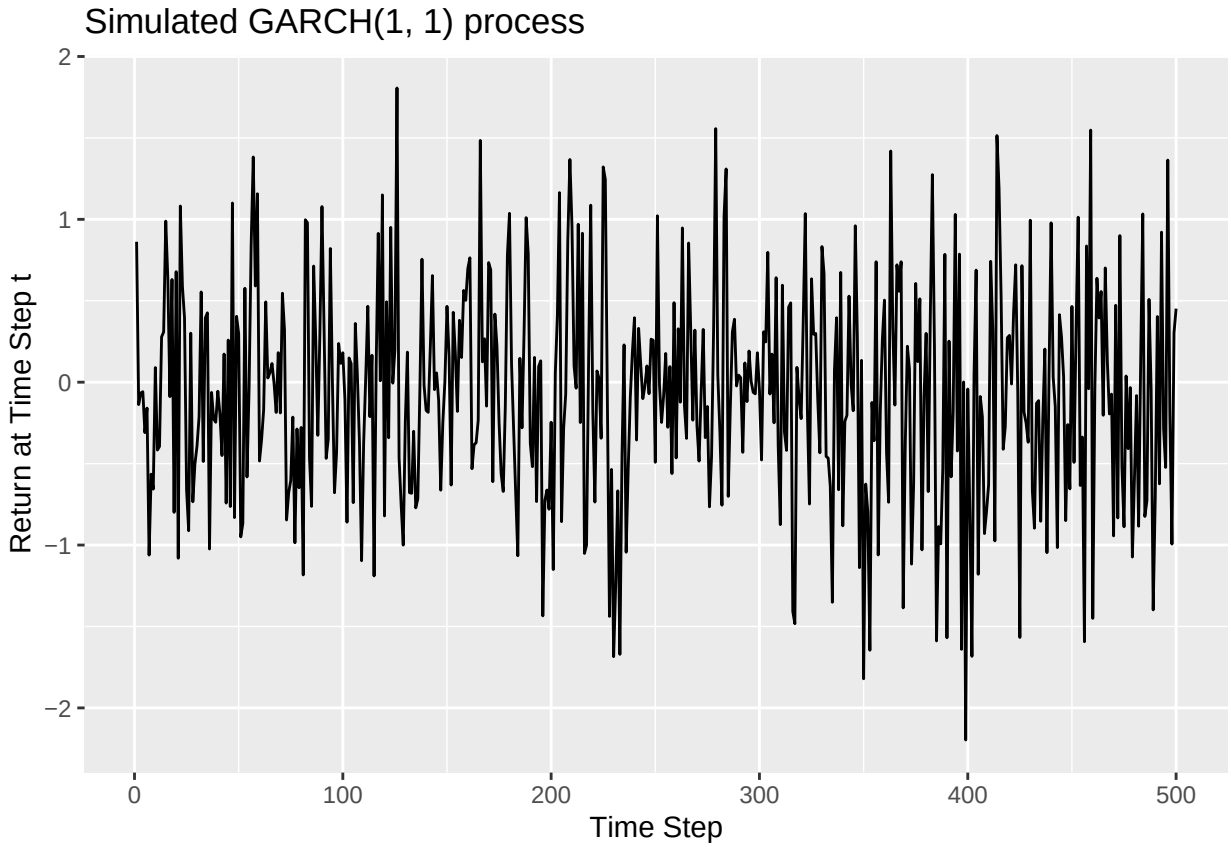
The ARCH model introduced in the previous section models future returns by conditioning the value of the variance at time t to the previous time step alone i.e $\sigma^2_{t|t-1} = \omega + \alpha r^2_{t-1}$. Bollerslev's (1986) approach encourages the backward extension of this conditioning process up until the q^{th} time step as well as the introduction of p lags to the conditional variance (Cryer and Chan, 2008). This resulting model becomes a Generalized Autoregressive Conditional Heteroscedasticity (GARCH) process, and is denoted as `GARCH(p,q)`. The return from this new proposed model takes the same form as ARCH's $r_t = \sigma_{t|t-1}\varepsilon_t$. However, the conditional variance $\sigma^2_{t|t-1}$ modeled by the quantity $\sigma_{t|t-1}$ now becomes :

$$\sigma^2_{t|t-1} = \omega + \beta_1\sigma^2_{t-1|t-2} + \cdots + \beta_p\sigma^2_{t-p|t-p-1} + \alpha_1 r^2_{t-1} + \alpha_2 r^2_{t-2} + \cdots + \alpha_q r^2_{t-q}$$

The β coefficients in the model are used to assign weights to the lags of the conditioned variance values.

The plot below (Cryer and Chan, 2008) illustrates an example of a simulated GARCH(1,1) process with parameter values

$$\omega = 0.02, \alpha = 0.05, \text{ and } \beta = 0.9$$



The parameters ω, β, α in GARCH and ω, α in ARCH are constrained to > 0, since the conditional variances have to be positive.

Estimation of GARCH model coefficents.

GARCH model coefficients are fit using the Maximum Likelihood Approach. The estimation process used to obtain the likelihood estimates for ω, β, α is based on recursively iterating through the log likelihood function modelling the GARCH coefficient estimates. The log likelihood function we aim to maximize is defined as (Cryer and Chan, 2008):

$$L(\omega, \alpha, \beta) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\left\{\log\left(\sigma^2_{t-1|t-2}\right) + r_t^2/\sigma^2_{t|t-1}\right\}$$

Model Fitting

Below, we try fitting an appropriate GARCH model using the `ugarchfit()` function from the rugarch package. The package computes the model estimates using the maximum likelihood function specified in the previous section.

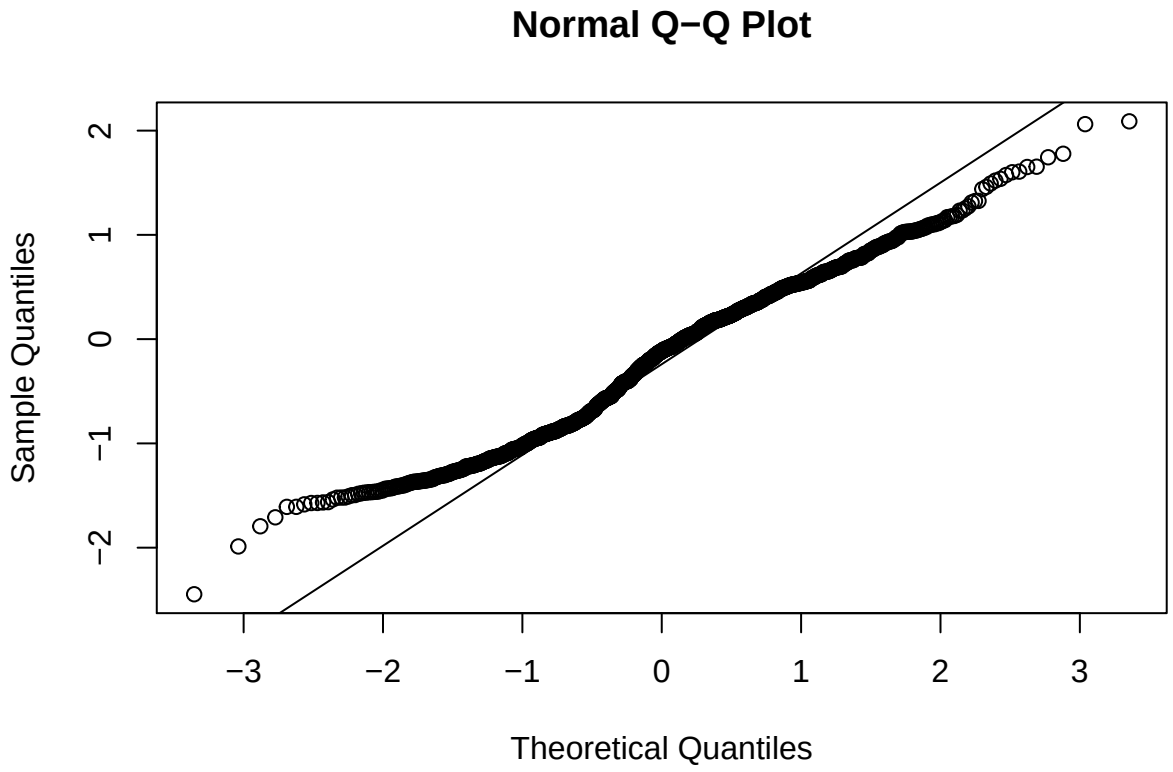
We will explore different GARCH orders, with the aim of finding the model that best fits the data. The orders we will try are arbitrarily chosen as `GARCH(1,1)`, `GARCH(2,2)`and `GARCH(3,3)`:

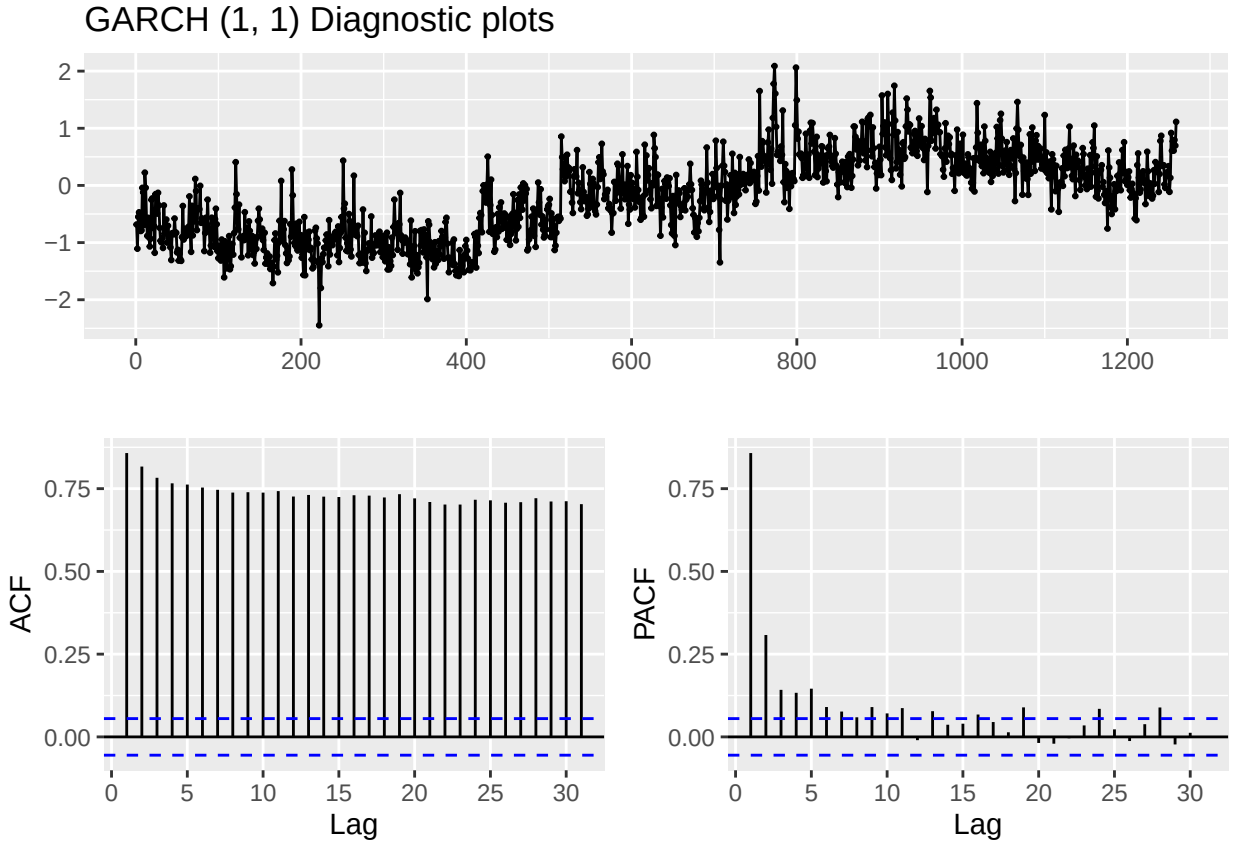
```
#from rurgarch library.
#Define GARCH params for the 3 models
spec_mod1 = ugarchspec(mean.model = list(armaOrder = c(0,0)),
                       variance.model = list(model="sGARCH",garchOrder = c(1, 1)))
spec_mod2 = ugarchspec(mean.model = list(armaOrder = c(0,0)),
                       variance.model = list(model="sGARCH",garchOrder = c(2, 2)))
spec_mod3 = ugarchspec(mean.model = list(armaOrder = c(0,0)),
                       variance.model = list(model="sGARCH",garchOrder = c(3, 3)))
#fit models from params specified above. We use log
#of returns because the code cannot handle huge numbers
#when solving for estimates
fit_mod1 = ugarchfit(spec = spec_mod1, data = log(CHK_ts))
fit_mod2 = ugarchfit(spec = spec_mod2, data = log(CHK_ts))
fit_mod3 = ugarchfit(spec = spec_mod3, data = log(CHK_ts))
#Inspect model coefficients
#print(fit_mod1)
#print(fit_mod2)
#print(fit_mod3)
```

Model Diagnostics

The `GARCH(1,1)` model seems like the most appropriate here since all but one of it's model coefficients are significant. Furthermore, it has the lowest AIC of all the models explored.

Before we accept the model we found in the previous section we need to make sure that assumptions have been met for the `ARCH(1,1)` model. The squared residuals need to be serially uncorrelated and the error terms should be normally distributed :





From the diagnostic plots made above, we see a major problem with normality, as most of the points on the qqplot veer off the line. In addition, there seems to be major issues with independence, as can be seen from

the significant lags in residuals from the autocorrelation plots.

Results

From the model diagnostics we ran in the previous section, we discovered major issues with normality and independence in error terms. We might want to exploring more model parameters to see whether we can find a model that improves upon what we currently have. For now, we will proceed with extreme caution and use the `GARCH(1,1)` model to make some predictions.

Below, we first extract the coefficients from the chosen model.

```
#Extract coefficients
fit_mod1@fit$matcoef
```

```
##           Estimate  Std. Error    t value     Pr(>|t|)
## mu     16.96828513  0.03046360 557.001998 0.000000e+00
## omega   0.04592897  0.01059138   4.336448 1.448039e-05
## alpha1  0.40407479  0.05760516   7.014559 2.306821e-12
## beta1   0.51414297  0.06573577   7.821358 5.329071e-15
```

The return at time t as given by our model is going to be given by (Boudt, 2020) :

$$R_t = 16.97 + e_t$$

where the e_t is a normally distributed random variable with a mean of 0 and variance of $\widehat{\sigma}_t^2$ i.e $e_t \sim N\left(0, \widehat{\sigma}_t^2\right)$. The variance modelling return volatility at the t^{th} time step in our fitted model is going to be $\widehat{\sigma}_t^2 = 0.05 + 0.40e_{t-1}^2 + 0.51\widehat{\sigma}_{t-1}^2$

The `ugarchroll()` function is used to obtain estimates for the last four dates in the data set. The test data that is used is the most recent week's returns in stock volume. The log volume residuals obtained after this process are printed down below :

```
## [1]  0.50445 -0.09980  0.09982 -0.50385
```

Conclusion

In this study, we have shown how ARCH and GARCH models can be used to model the changing variance in time series with observed periods of volatility. We have also seen why ARIMA models are insufficient when it comes to predicting time series with irregular trends. Inasmuch as the GARCH model chosen accurately captures the volatility in the stock data analysis conducted in this study, it doesn't effectively forecast the volume in stock returns. Exploration of further GARCH parameters would be an advisable next step, and would hopefully remediate the issues with model fit encountered in this project.

References

1. Cryer, J. D., & Chan, K.-sik. (2008). Time series analysis with applications in R. New York: Springer.
2. Nugent, C. (n.d.). S&P 500 stock data. Retrieved from https://www.kaggle.com/camnugent/sandp500.
3. Boudt, K. (n.d.). GARCH Models in R. Retrieved from https://www.datacamp.com/courses/garch-models-in-r.
4. Trapletti, A., & Hornik, K. (n.d.). Package 'tseries.' Retrieved from https://cran.r-project.org/web/packages/tseries/tseries.pdf
5. Ghalanos, A., & Kley, T. (n.d.). Package 'rugarch.' Retrieved from https://cran.r-project.org/web/packages/rugarch/rugarch.pdf
6. Glen, S. (2018, September 11). Ljung Box Test: Definition. Retrieved from https://www.statisticshowto.datasciencecentral.com/ljung-box-test/
7. Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3), 307–327. doi: 10.1016/0304-4076(86)90063-1