DSC425 Final Project AAPL

YunTzu

Introduction

Stock refers to purchasing or selling fractional ownership of equity in an organization, people could invest in a company by purchasing or selling its stock in the stock market which allows for seamless exchange. People's economic status could be significantly affected by stock price, as a result, effectively predicting stock trends could aim to increase the profit and reduce the risk.

We'll apply time-series analysis on stock price which allows us to find predictive information in data collected over time.

Dataset Introduction

To conduct this study, stock daily prices for Apple was extracted from Yahoo's finance website.

Exploratory Data Analysis

Load the data

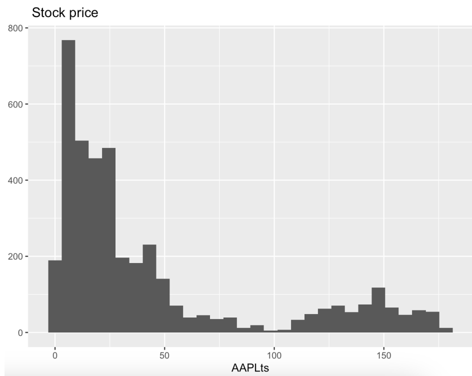
The IPO of AAPL was in the year 1980, we'll just focus on the data from 2007 to now since it is the year that the iPhone was released, which is the key factor that affects the fluctuation of the Apple stock price.

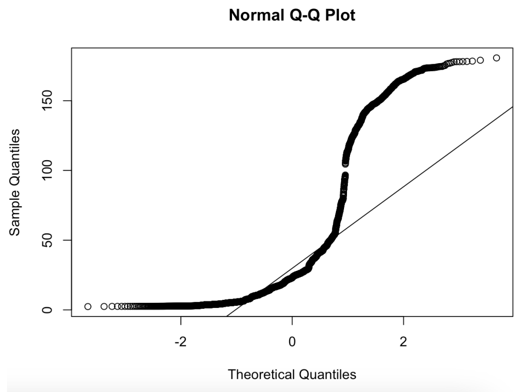
```
> head(AAPL)
# A tibble: 6 × 7
  Date        Open  High   Low Close `Adj Close`     Volume
  <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>      <dbl>
1 2007-01-03  3.08  3.09  2.92  2.99        2.55 1238319600
2 2007-01-04  3.00  3.07  2.99  3.06        2.60  847260400
3 2007-01-05  3.06  3.08  3.01  3.04        2.59  834741600
4 2007-01-08  3.07  3.09  3.05  3.05        2.60  797106800
5 2007-01-09  3.09  3.32  3.04  3.31        2.81 3349298400
6 2007-01-10  3.38  3.49  3.34  3.46        2.95 2952880000
```

Data distribution

From the histogram, we could tell it is not a normal distribution and it is heavily skewed to the left side.

From the qq-plot, we could see that it has a heavy tail.

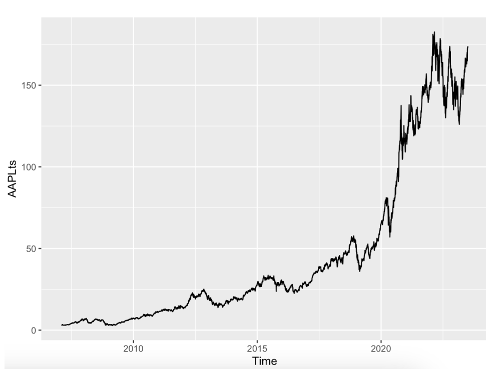


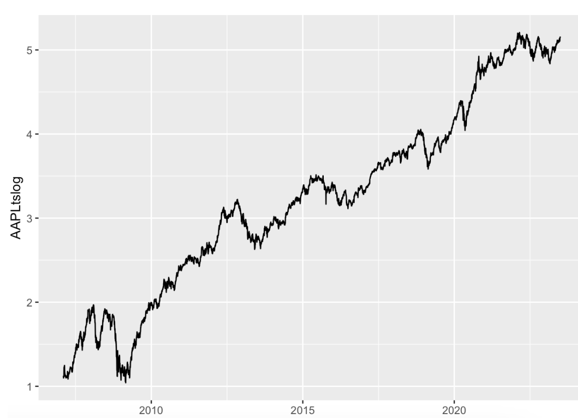


Time trend displayed by the plot

From the graph before logging, we can see upward trends, which means the stock price
emotionally going up.
This is a multiplicative series, because the bigger value is, the bigger the swing are gonna be, so we need to take the log to help to stabilize the variance of a time series.
After logging the data, we can see a plot likely to be a random walk.

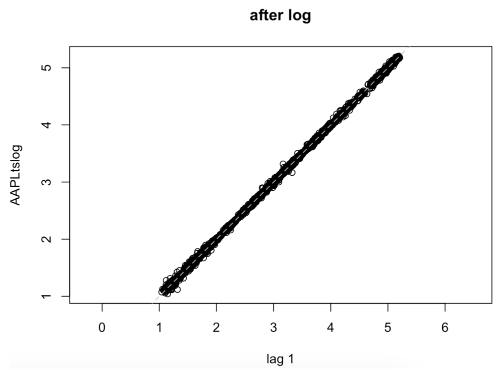
Before log After log





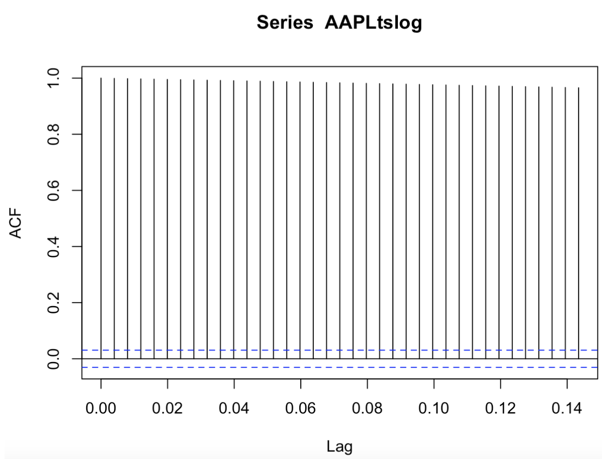
Lag plot

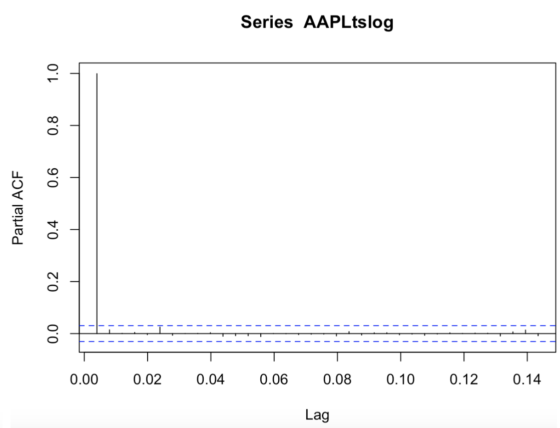
We can see a strong correlation between the series of time t and the series of time t-1, so it demonstrates there is a serial dependence.



Plot the ACF and PACF

We can see that the ACF decay slowly and it should be a non-stationary time series, its PACF has a strong correlation at lag 1,






```
> adfTest(AAPLtslog,type="ct")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -2.7463
  P VALUE:
    0.2623

Description:
 Mon May 29 17:05:48 2023 by user:
```

```
> kpss.test(AAPLtslog)

	KPSS Test for Level Stationarity

data:  AAPLtslog
KPSS Level = 35.323, Truncation lag parameter = 10, p-value = 0.01
```

Differencing

Differencing can help stabilize the mean of a time series by removing changes in the level of a time series, and therefore eliminating or reducing trend and seasonality.

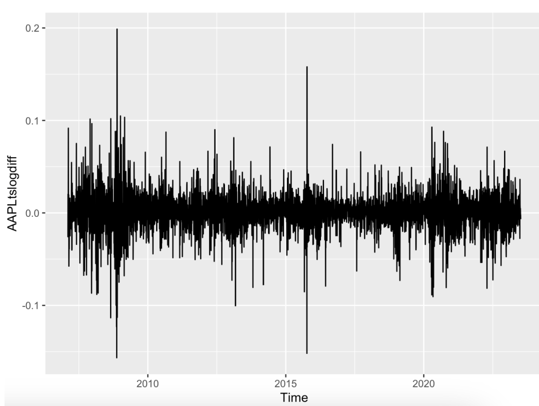
After differencing we could see that the autoplot is stable although there are something left at two points.

Then we can check the Acf and Pacf of the series, we can see that there are some autocorrelations but it is close to white noise. A lot of stock time series will have a little autocorrelation because of the efficient market hypothesis and because of trading.

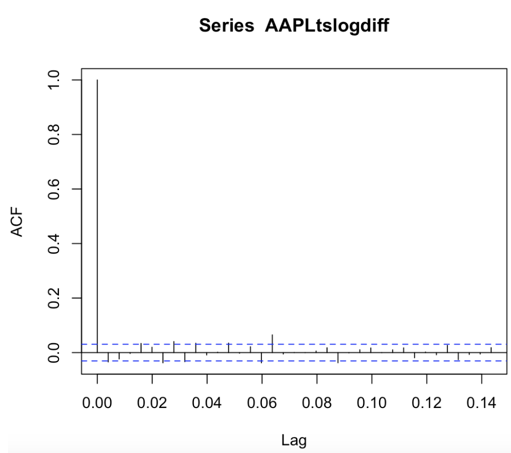
From ACF we can see that maybe it is an AR(2) series, from the PACF we can see there is an observed correlation at lag4, lag 7 and lag 17 , so maybe we could try MA(4), MA(7) to our arima series.

Next, we can check adfTest that it indicates that we can reject the null hypothesis with unit root, at the same time, we can check kpss.test which also indicates that we fail to reject the null hypothesis with stationery.

Autoplot

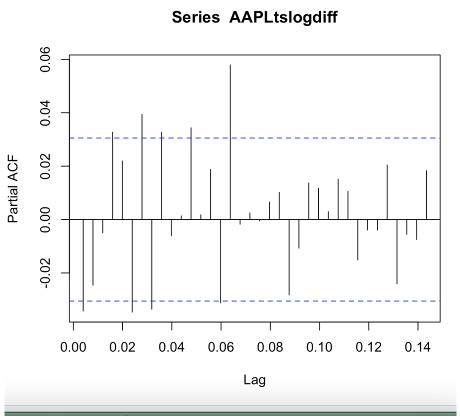


ACF



PACF

EACF




```
> eacf(AAPLtslogdiff)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x o o x o x x x x o o  x  o  o
1 x o o o o o o o o o o  x  o  o
2 x x o o o o o o o o o  o  o  o
3 x x o o o o o o o o o  x  o  o
4 x x o x o o o o o o o  o  o  o
5 x o x x x o o o o o o  o  o  o
6 x x x x x x o o o o o  o  o  o
7 x x x o x x o o o o o  o  o  o
```

```
> adfTest(AAPLtslogdiff,type="nc")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -47.0962
  P VALUE:
    0.01

Description:
 Wed May 24 14:15:34 2023 by user:
```

```
> adfTest(AAPLtslogdiff,type="ct")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -47.2752
  P VALUE:
    0.01
```

```
> adfTest(AAPLtslogdiff,type="c")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -47.2799
  P VALUE:
    0.01
```

```
> kpss.test(AAPLtslogdiff)

        KPSS Test for Level Stationarity

data:  AAPLtslogdiff
KPSS Level = 0.040289, Truncation lag parameter = 10, p-value = 0.1
```

Data Analysis

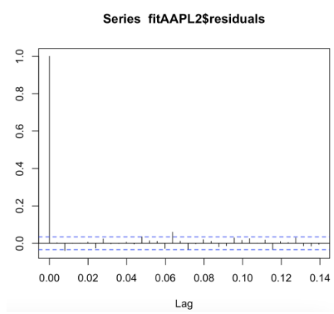
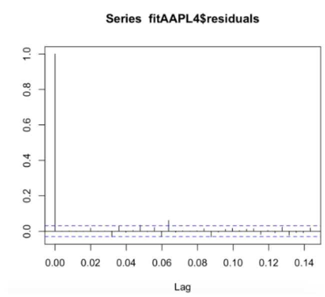
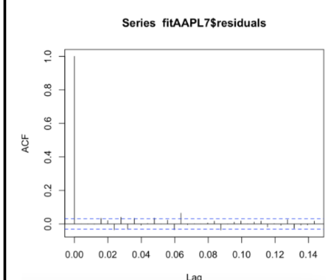
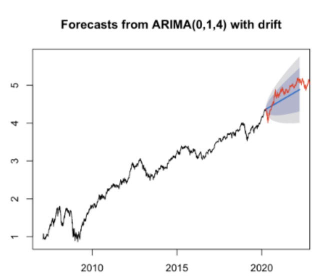
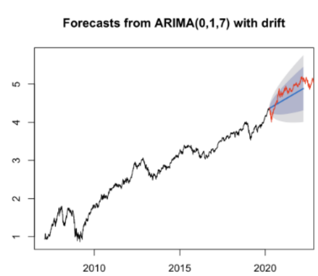
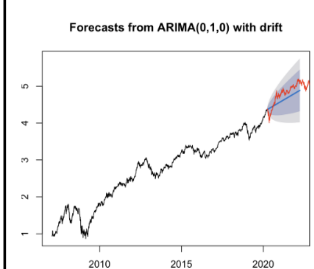
Methodology -ARIMA

ARIMA stands for Autoregressive Integrated Moving Average that is capable of capturing a suite of different standard temporal structures in time-series data; the model is good fit if the residuals show white noise behavior.
From ACF and PACF we can try ARIMA(0,1,4) with drift, ARIMA(0,1,7) with drift and auto.arima.
We can check performance of the model based on these criterias :

1. Lowest sum square error
2. Lowest AIC , BIC
3. Maximum likelihood
4. Box.test
5. BackTest

The model ARIMA(0,1,7) performs best among these three models.

fitAAPL4<-Arima(AAPLtslog, order = c(0,1,7),include.drift = TRUE,fixed = c(NA,NA,0,NA,0,NA,NA,NA))

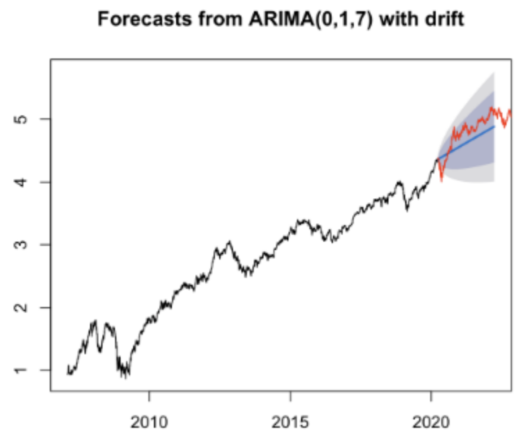
Model	ARIMA(0,1,4) with drift	ARIMA(0,1,7) with drift	ARIMA(0,1,2) with drift
Sigma^2	0.0003866	0.0004103	0.0004116
AIC	-16537.47	-20430.04	-20419.89
BIC	-16519.17	-20385.78	-20394.59
log likelihood	8271.7	10222.02	10213.94
Box.test p-value	0.5005	0.5192	0.03028
ACF Residual			
MAPE	0.0387	0.0386	0.0384
Forecast Plot			

We separate 80% of the observation from the dataset as training data and 20% of the observation as test data to see the accuracy of the model and to see the performance of the backtest, we can see that the result of MAPE from backtest of three model are around 3.8%, there is no much different on the result.

Forecasting

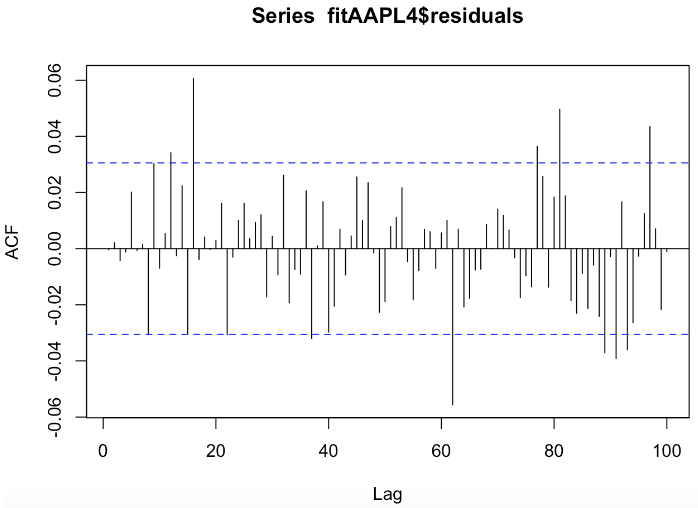
```
fitAAPL4<-Arima(AAPLTrain, order = c(0,1,7),include.drift = TRUE,fixed = c(NA,NA,0,NA,0,NA,NA,NA))
forcastm2 = forecast::forecast(fitAAPL4)
plot(forcastm2)
lines(AAPLTest, col="red")
```

We build the forecasting model based on a test dataset.
As we can see, we have a blue line that represents the mean of our prediction and the red line represent the real value.With the blue line explained we can see darker and lighter darker areas, representing 80% and 95% confidence intervals respectively in lower and upper scenarios.



The residuals seem like they're no more significantly correlated but they're not completely a white noise process.

```
> Box.test(fitAAPL4$residuals,lag=10,type = "Ljung")

	Box-Ljung test

data:  fitAAPL4$residuals
X-squared = 9.6536, df = 10, p-value = 0.4714
```



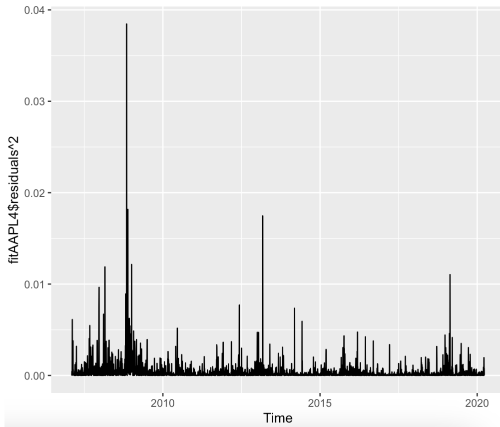
Volatility modeling comes in when we achieved our final model and the residual series is serially uncorrelated (or has only lower order serial correlation), but "something isn't quite right"

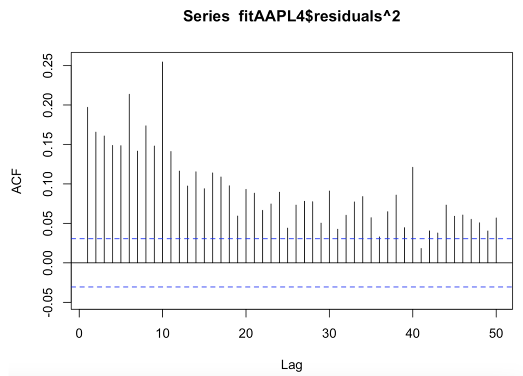
Some effects are not explained by the model (residuals are not white noise), so the residuals are still dependent.

Methodology -Garch

Garch stands for Generalized Autoregressive Conditional Heteroscedasticity that has capability to make the volatility clustering.

The result of the previous ARIMA model is pretty good ,but we got some spike from the residual square, by check Acf of residual square it has a lot of lower lag ,which could be a cause of the volatile observations of the financial volatile observation.





So here we could try to apply the Garch model to minimize the volatility effect.

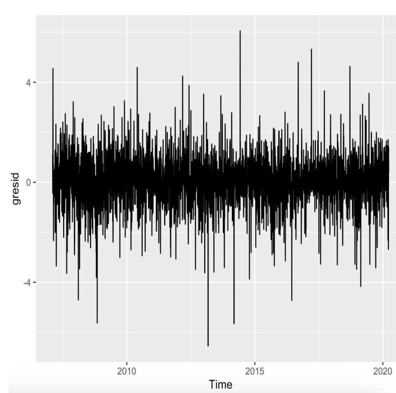
We try to apply a standard GARCH (1, 1) model over ARIMA(0,1,7), looking if we have improved our accuracy and model parameters.

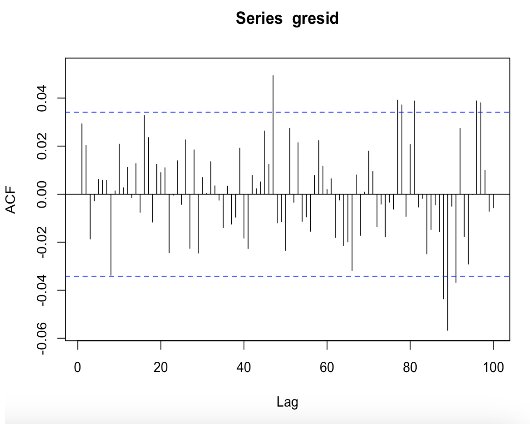
We can see the coefficients of the model are all significant, and we can see the residual is close to white noise from the acf plot.

From the box test, it indicates that we can't reject null hypothesis which means it is white noies.

```
res = fitAAPL4$residuals
gFit = garch(res, order =c(1,1))
```

```
> coeftest(gFit)

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
a0 1.2299e-05 1.3145e-06   9.356 < 2.2e-16 ***
a1 9.4219e-02 8.5995e-03  10.956 < 2.2e-16 ***
b1 8.7282e-01 1.0698e-02  81.584 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



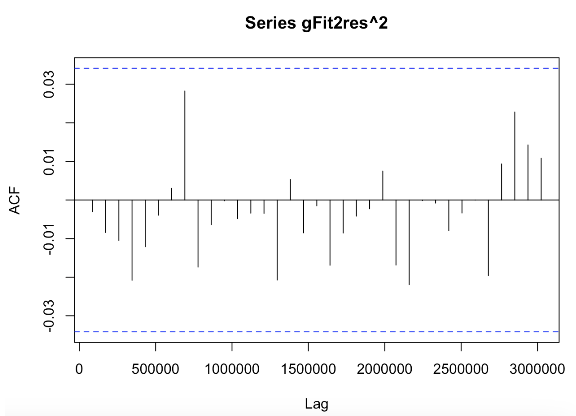



```
> Box.test(gresid^2,type = "Ljung")

        Box-Ljung test

data:  gresid^2
X-squared = 0.019766, df = 1, p-value = 0.8882
```

Though the model perform pretty well after we include the GARCH (1,1) in to our model, we still visually could detect som autocorrelation at lag42 and at some higer lag, maybe here we can try to integrate the GARCH(1,1) model and ARMA(0,7) model together, and we can see the the plot from Acf indicate that is the white noise, by checking the box.test we can see that we can't reject the null hypothesis with high confidence.

ugarch.spec = ugarchspec(variance.model=list(garchOrder=c(1, 1)),
mean.model=list(armaOrder=c(0, 7)))

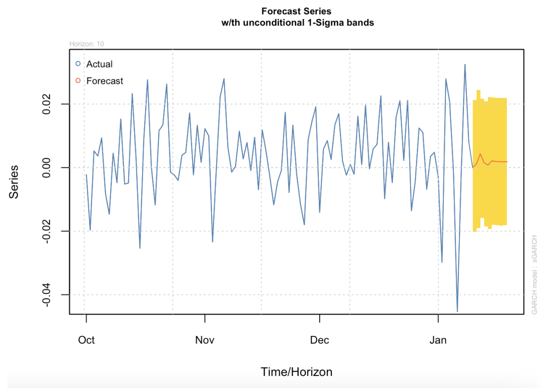
gFit2 <- ugarchfit(ugarch.spec, diff(AAPLTrain))




```
> Box.test(gFit2res^2, lag = 15, type = 'Ljung')

        Box-Ljung test

data:  gFit2res^2
X-squared = 7.9799, df = 15, p-value = 0.9246
```

Checking the performance of the model, we got MSE is 0.0002489 and MAE is 0.0111900 which is pretty good.

```
> report(gFit2.rolltest, type = 'fpm')

GARCH Roll Mean Forecast Performance Measures
---------------------------------------------
Model          : sGARCH
No.Refits      : 3
No.Forecasts: 1295

        Stats
MSE 0.0002489
MAE 0.0111900
DAC 0.5189000
```



Analysis of results

After conducting a rigorous exploratory data analysis and attempting many different models, the best-performing model for each time series was kept. Despite performing well, both ARIMA models still showed a considerable amount of volatility for their residuals, which was addressed by integrating the ARIMA models with GARCH models. The ARMA(0, 7) GARCH(1, 1) for Apple, ended up with conforming residuals and very low MSE and MAE.