Team Essay 6

Sahil, Ariel, Ilya, Lekha, Anthony, Trang

5/1/2021

Introduction

In this essay, our team built a GARCH model, the Generalized Autoregressive Conditional Heteroscedastic model, which is an extension of the ARCH model that incorporates a moving average component together with the autoregressive component. We used this model to predict the daily prices of Facebook stocks from May 18th 2012 to April 28th 2021 (the day our group downloaded the data set). The day the inbuilt dataset is downloaded determines the size of the dataset, so it's never a constant value. We found the number of data points to be 2251, as that's simply the number of days that have passed from May 19 2012 to April 28 this year. Our team made the prediction based on variables such as: opening price(USD), closing price(USD) , highest price(USD), lowest price(USD), the number of shares trade on given day(unit) and the adjusted closing price(USD).

Loading R packages

```
library(foreign) # Allows us to import data from common statistical packages
library(ggplot2) # Used for plotting data
library(grid) # Used to arrange plots of data
library(gridExtra) # Used to arrange plots
library(rugarch) # Used for creating Garch models

## Loading required package: parallel

##
## Attaching package: 'rugarch'

## The following object is masked from 'package:stats':
##
##     sigma

library(quantmod) # Used for plotting time series

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(fGarch) # Used to estimate variance
```

```
## Loading required package: timeDate

## Loading required package: timeSeries

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##
##     time<-

## Loading required package: fBasics

##
## Attaching package: 'fBasics'

## The following object is masked from 'package:TTR':
##
##     volatility
```

```
library(dynlm) # Used to estimate mean equation
library(tidyverse) #Used for data manipulation
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.
3.0 ──

## ✓ tibble  3.0.6     ✓ dplyr   1.0.4
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.1
## ✓ purrr   0.3.4

## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks timeSeries::filter(), stats::filter()
## x dplyr::first()   masks xts::first()
## x dplyr::lag()     masks timeSeries::lag(), stats::lag()
## x dplyr::last()    masks xts::last()
## x purrr::reduce()  masks rugarch::reduce()
```

```
library(FinTS) # Used to perform Arch test and check ARCH effects
theme_set(theme_bw())
```

Formula and basics

For auto-regressive AR(1)

$$Y_t = \phi + e_t \qquad [1]$$

$$et|Tt - 1 \sim N(0, h_t) \qquad [2]$$

$$h_t = \alpha_0 + \alpha 1 e^2 t - 1$$

,

$$\alpha_0 > 0$$

,

$$0 \leq \alpha_1 < 1 \qquad [3]$$

Equation [1] is the mean equation

Test for Arch effect

$$\hat{e}_t^2 = \gamma_0 + \gamma 1 \hat{e}^2 t - 1 + \cdots + \gamma q \hat{e}^2 t - q + \nu \qquad [4]$$

Hypothesis test:

$$H_0 : \gamma_1 = \gamma_2 = \cdots = \gamma_q = 0 \ H_A : \gamma_1 \neq 0 \ or \ldots \gamma_q \neq 0 \qquad [5]$$

The test shown in [4] may include several lag terms, in which case the null hypothesis [5] would be that all of them are jointly insignificant. In that :

$$\hat{e}t$$

: estimated residual

$$\nu$$

: random error term

If there is no ARCH effect, the test statistic is

$$(T - q)R^2 \sim \chi^2 1 - \alpha, q$$

Data description

- FB.Open: The opening price of 1 share (in USD) of facebook once the market opens
- FB.High: The highest price of 1 share (in USD) of facebook on the given day
- FB.Low: The lowest price of 1 share (in USD) of facebook on the given day
- FB.Close: The closing price of 1 share (in USD) of facebook once the market closes
- FB.Volume: The number of shares of facebook traded on the given day
- FB.adjusted: The adjusted closing price of 1 share (in USD) of facebook after accounting corporate actions

```
# FB data-set
fb <- getSymbols("FB", auto.assign = F)

## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
```

```
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
nrow(fb) # Gives you the number of data points
```

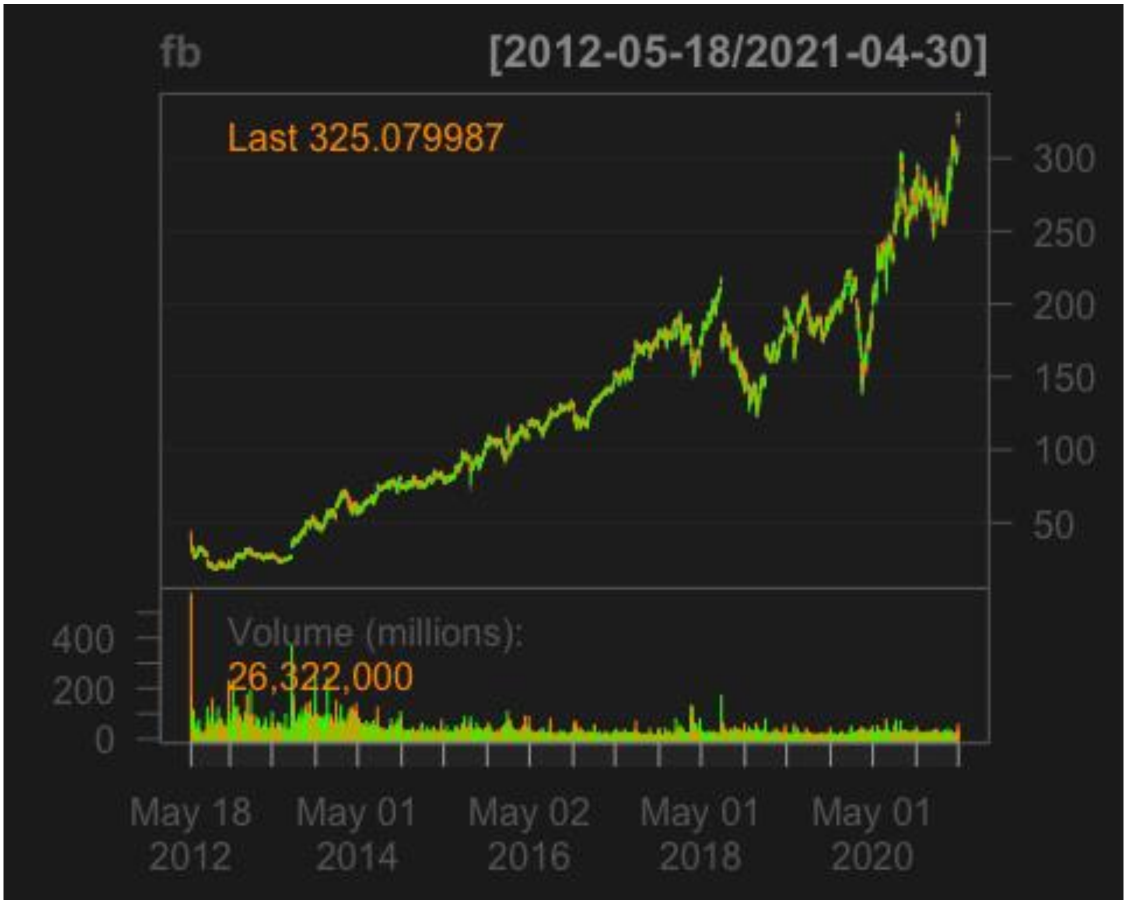
```
## [1] 2252
```

```
# FB data-set
fb.df <- as.data.frame(fb)



# Renaming the phantom "Date" column
fb.df <- cbind(Date = rownames(fb.df), fb.df)
rownames(fb.df) <- 1:nrow(fb.df)

fb_train_data = fb.df[1:2100,]
fb_test_data = fb.df[2100:2247,]
```

Visualization

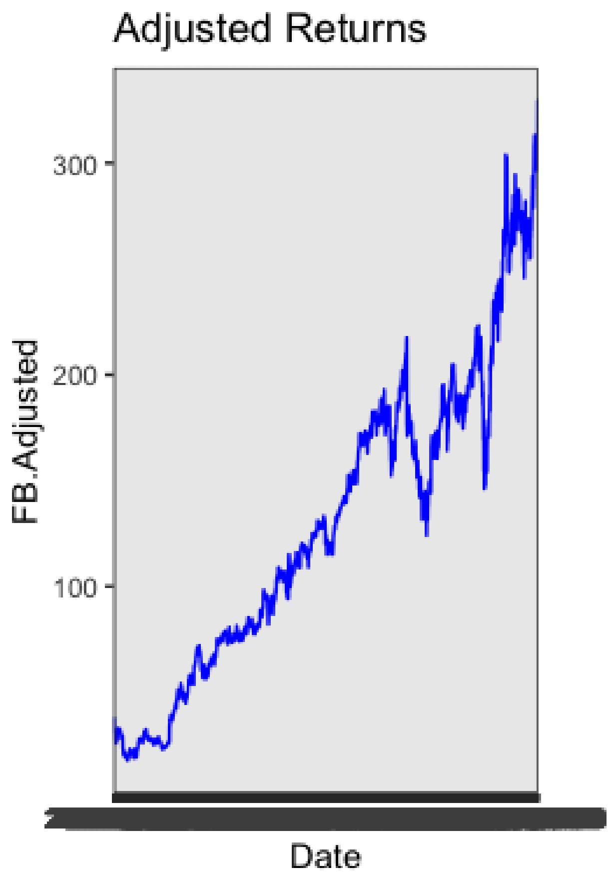
From plotting the time series graph of the Facebook stock prices, from May 18 2012 to Apr 28 2021, we see that there's an overall rise in the prices that is noticeable, and is accompanied by a lot of noise. This indicates volatility in the data. The noise is also not constant; some parts of the graph show a much lower drop in stock price than others, indicating that the data doesn't have constant variance, and hence is heteroscedastic. Hence, visualizing the data proves the need for us to build and apply the GARCH model.

```
# FB data-set
chartSeries(fb) # Plots the time series
```

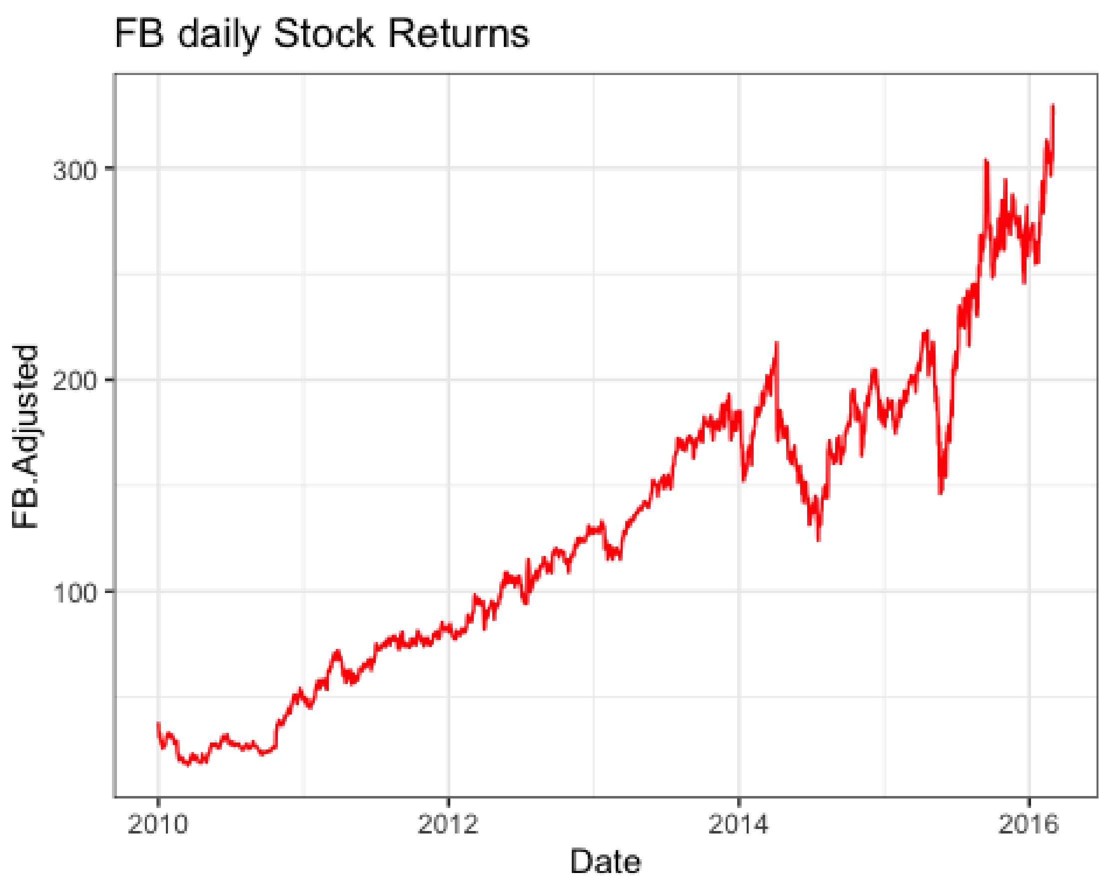


Analysis

```
# FB data-set
fb_plot_1 <- ggplot(fb.df, aes(y = FB.Adjusted, x = Date)) + geom_line(col =
'blue', group = 1) + labs(ylab = 'return', xlab = 'Time', title = 'Adjusted R
eturns')

grid.arrange(fb_plot_1, ncol = 2);
```




```r
# FB data-set
# Examine the FB's daily stock returns trend
fb.df$Date <- seq.Date(as.Date('2010-01-01'), by = 'day', length.out = length
(fb.df$FB.Adjusted))
ggplot(fb.df, aes(y = FB.Adjusted, x = Date )) + geom_line(col = 'red', group
= 1) +
  labs(title = 'FB daily Stock Returns', ylab = 'return')
```



Model Evaluation

```
# Step 1: Estimate mean equation r = beta + error
fb_data_mean <- dynlm(FB.Adjusted ~ 1, data = fb_train_data)

# Step 2: Retrieve the residuals from the former model and square them
ehatsq <- ts(resid(fb_data_mean)^2)

# Step 3: regress squared residuals on one-lagged squared residuals
fb_data_arch <- dynlm(ehatsq ~ L(ehatsq), data = ehatsq)

summary(fb_data_arch)

##
## Time series regression with "ts" data:
## Start = 2, End = 2100
##
## Call:
## dynlm(formula = ehatsq ~ L(ehatsq), data = ehatsq)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6391.2   -86.4   -15.2    87.9  8071.5
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.360700  14.314181   1.422    0.155
## L(ehatsq)    0.996340   0.002457 405.442   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 474 on 2097 degrees of freedom
## Multiple R-squared:  0.9874, Adjusted R-squared:  0.9874
## F-statistic: 1.644e+05 on 1 and 2097 DF,  p-value: < 2.2e-16
```

```
# FB data-set
fb_data_archTest <- ArchTest(fb_train_data$FB.Adjusted, lags = 1, demean = TR
UE)
fb_data_archTest
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  fb_train_data$FB.Adjusted
## Chi-squared = 2072.6, df = 1, p-value < 2.2e-16
```

```
#Reject Null Hypothesis
```

Because the p-value is < 0.05, we reject the null hypothesis and conclude the presence of ARCH(1) effects.

```
# FB data-set
# Plot the conditional variance
fb_arch_fit <- garchFit(~garch(1,0), data = fb_train_data$FB.Adjusted, trace
= F)
```

```
## Warning in sqrt(diag(fit$cvar)): NaNs produced
```

```
## Warning: Using formula(x) is deprecated when x is a character vector of le
ngth > 1.
##   Consider formula(paste(x, collapse = " ")) instead.
```

```
summary(fb_arch_fit)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 0), data = fb_train_data$FB.Adjusted,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 0)
## <environment: 0x7fa5add16c78>
##  [data = fb_train_data$FB.Adjusted]
```

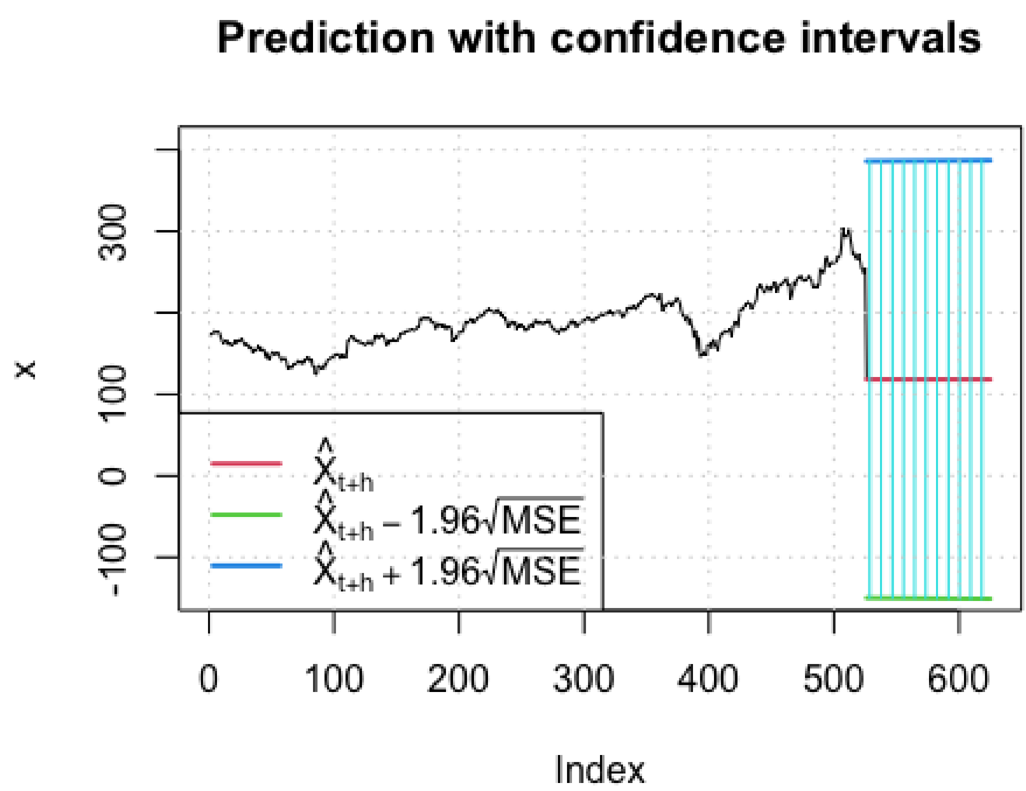
```
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##       mu     omega    alpha1
## 118.0674    1.4485    1.0000
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      118.0674      0.0391   3019.3   <2e-16 ***
## omega     1.4485          NA       NA       NA
## alpha1    1.0000      0.0215     46.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  -10752.63    normalized:  -5.120299
##
## Description:
##  Sat May  1 20:34:05 2021 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  310.1598  0
##  Shapiro-Wilk Test  R    W      0.7274295 0
##  Ljung-Box Test     R    Q(10)  18875.74  0
##  Ljung-Box Test     R    Q(15)  28112.64  0
##  Ljung-Box Test     R    Q(20)  37143.87  0
##  Ljung-Box Test     R^2  Q(10)  25.45076  0.004553517
##  Ljung-Box Test     R^2  Q(15)  38.55771  0.0007458975
##  Ljung-Box Test     R^2  Q(20)  64.525    1.388928e-06
##  LM Arch Test       R    TR^2   25.9258   0.01099633
##
## Information Criterion Statistics:
##      AIC      BIC      SIC     HQIC
## 10.24346 10.25153 10.24345 10.24641

fb_pred <- predict(fb_arch_fit, n.ahead = 100, trace = TRUE, mse = c("cond","
uncond"),
        plot=TRUE, nx=NULL, crit_val=NULL, conf=NULL)

##
## Model Parameters:
##         mu         ar1         ma1       omega      alpha1      gamma1       bet
a1
```

```
## 118.067374   0.000000   0.000000   1.448506   1.000000   0.000000   0.0000
00
##      delta       skew      shape
##   2.000000   1.000000   4.000000
##
## Forecast GARCH Variance:
##
## Forecast ARMA Mean:
##
## Call:
## arima(x = object@data, order = c(max(u, 1), 0, max(v, 1)), transform.pars
= FALSE,
##     init = c(ar, ma, mu), optim.control = list(maxit = 0))
##
## Coefficients:
```

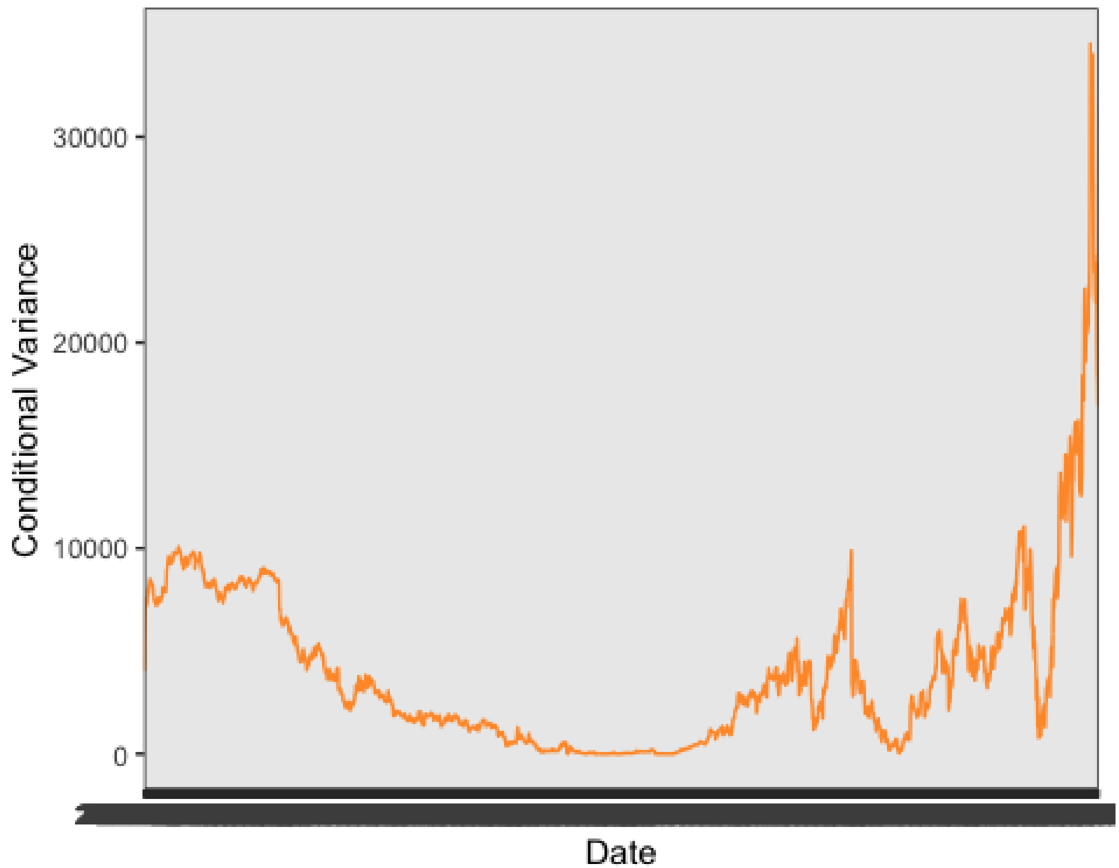
```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##       ar1     ma1  intercept
##         0  0.0000   118.0674
## s.e.  NaN  0.0218     1.3863
##
## sigma^2 estimated as 4033:  log likelihood = -11697.24,  aic = 23402.47
##
## $pred
## Time Series:
## Start = 2101
## End = 2200
## Frequency = 1
##   [1] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##   [9] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [17] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [25] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [33] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [41] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [49] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [57] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [65] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [73] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [81] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
```

```
674
##  [89] 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0674 118.0
674
##  [97] 118.0674 118.0674 118.0674 118.0674
##
## $se
## Time Series:
## Start = 2101
## End = 2200
## Frequency = 1
##   [1] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##   [9] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [17] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [25] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [33] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [41] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [49] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [57] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [65] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [73] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [81] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [89] 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50855 63.50
855
##  [97] 63.50855 63.50855 63.50855 63.50855
```




```
summary(fb_pred)

##   meanForecast     meanError     standardDeviation lowerInterval
##  Min.   :118.1   Min.   :136.7   Min.   :136.7     Min.   :-150.9
##  1st Qu.:118.1   1st Qu.:136.8   1st Qu.:136.8     1st Qu.:-150.6
##  Median :118.1   Median :136.9   Median :136.9     Median :-150.3
##  Mean   :118.1   Mean   :136.9   Mean   :136.9     Mean   :-150.3
##  3rd Qu.:118.1   3rd Qu.:137.1   3rd Qu.:137.1     3rd Qu.:-150.1
##  Max.   :118.1   Max.   :137.2   Max.   :137.2     Max.   :-149.8
##  upperInterval
##  Min.   :386.0
##  1st Qu.:386.2
##  Median :386.5
##  Mean   :386.5
##  3rd Qu.:386.7
##  Max.   :387.0

fb_train_data$ht <- fb_arch_fit@h.t
# How to rename the first column as "Date"?
ggplot(fb_train_data, aes(y = ht, x = Date)) + geom_line(col = '#ff9933', gro
up = 1) + ylab('Conditional Variance') + xlab('Date')
```



Looking at the graph, you can see the periods in which volatility was high.

Prediction and Model Accuracy

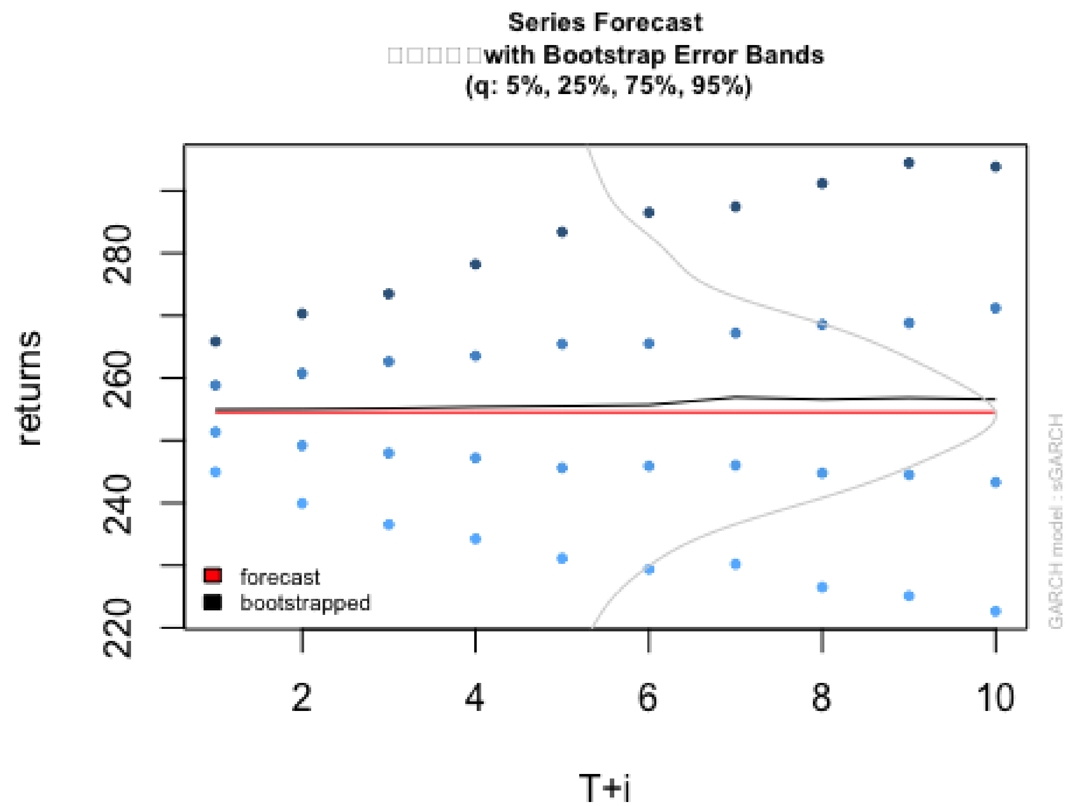
```
# FB data-set
# We can change the armaOrder to create different models and keep the model w
ith lowest AIC!

# 1st Model: AIC = 4.0681
fb1 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1
)), mean.model = list(armaOrder = c(1, 1)), distribution.model = "std")

fbGarch1 <- ugarchfit(spec = fb1, data = fb_train_data$FB.Adjusted)

# 2nd Model: AIC = 10.437
fb2 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1
)), mean.model = list(armaOrder = c(0, 0)), distribution.model = "std")

fbGarch2 <- ugarchfit(spec = fb2, data = fb.df$FB.Adjusted)

# 3rd Model: Does not converge
fb3 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1
)), mean.model = list(armaOrder = c(1, 0)), distribution.model = "std")
```

```r
fbGarch3 <- ugarchfit(spec = fb3, data = fb.df$FB.Adjusted)

## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :
## ugarchfit-->warning: solver failer to converge.

# 4th Model: AIC = 9.2589
fb4 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1
)), mean.model = list(armaOrder = c(0, 1)), distribution.model = "std")

fbGarch4 <- ugarchfit(spec = fb4, data = fb.df$FB.Adjusted)
```

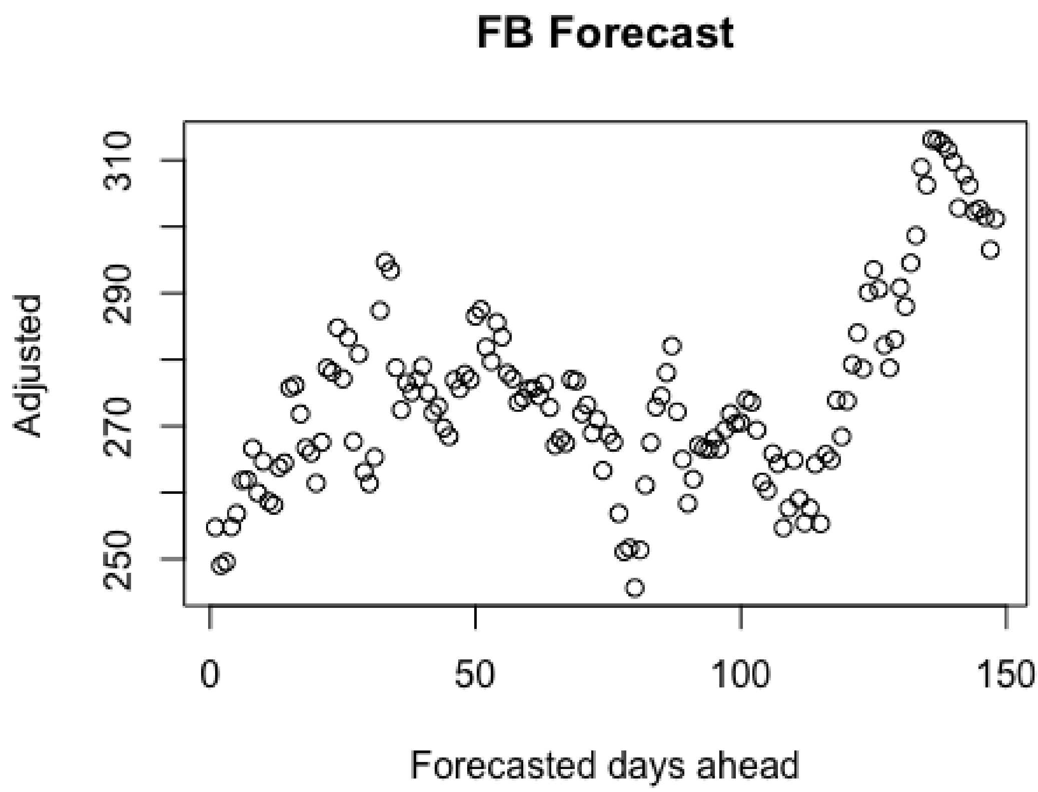
From the above computations, we say that fGarch 1 was a better model as it has an AIC = 4.0681, which is the lowest amongst the rest. We will use this model to make predictions.

```r
# FB data-set
# Finally forecast whatever you want!
# This is an example of what you want to forecast where n.ahead is the number
of predictions you want to make and fbGarch3 is an example for the model that
you want to use.
fbPredict <- ugarchboot(fbGarch1, n.ahead = 10, method = c("Partial", "Full")
[1])
fbPredict

##
## *-----------------------------------*
## *     GARCH Bootstrap Forecast      *
## *-----------------------------------*
## Model : sGARCH
## n.ahead : 10
## Bootstrap method:  partial
## Date (T[0]): 1975-10-01 20:00:00
##
## Series (summary):
##          min   q.25   mean   q.75    max forecast[analytic]
## t+1  149.319 251.36 254.96 258.83 282.88             254.52
## t+2  152.500 249.20 255.02 260.74 345.25             254.52
## t+3  130.348 247.99 255.14 262.64 346.35             254.52
## t+4   87.549 247.22 255.37 263.53 322.93             254.52
## t+5   82.720 245.63 255.51 265.44 326.85             254.52
## t+6   77.560 245.91 255.67 265.52 372.89             254.52
## t+7  115.172 246.02 256.90 267.22 387.41             254.52
## t+8  107.761 244.79 256.56 268.56 383.98             254.52
## t+9   91.403 244.53 256.81 268.81 402.22             254.52
## t+10 102.628 243.32 256.57 271.21 396.67             254.52
## .....................
##
## Sigma (summary):
##         min  q0.25   mean  q0.75     max forecast[analytic]
## t+1  7.0166 7.0166 7.0166 7.0166  7.0166             7.0166
## t+2  6.8217 6.8308 7.0210 7.0154 25.3803             7.0139
```

```
## t+3  6.6325 6.6907 7.0412 7.0445 24.6731            7.0112
## t+4  6.4500 6.5929 7.0030 7.0350 24.5324            7.0086
## t+5  6.2751 6.5102 6.9790 7.0425 25.9106            7.0059
## t+6  6.1077 6.4171 7.0260 7.1124 25.2319            7.0033
## t+7  5.9542 6.3438 7.0006 7.1020 28.0584            7.0006
## t+8  5.8156 6.2756 6.9980 7.0698 29.6611            6.9980
## t+9  5.6551 6.2085 7.0321 7.0932 29.0217            6.9953
## t+10 5.5137 6.1699 7.0039 7.0616 28.8879            6.9927
## .....................

plot(fbPredict, which = 2)
```




```
plot(fb_test_data$FB.Adjusted, xlab="Forecasted days ahead", ylab="Adjusted",
main="FB Forecast")
```




```
pred <- as.data.frame(as.table(fbGarch4@fit$fitted.values))
pred <- pred[1:10, ]
testPred <- as.data.frame(fb_test_data)
testPred <- testPred[1:10, ]
mean(abs((pred$Freq-testPred$FB.Adjusted)/pred$Freq)) # Prediction Error

## [1] 2.938957
```

Conclusion

All in all, our team has attempted to build GARCH models (GARCH - Generalized Autoregressive Conditionally Heteroscedastic), and from these select the optimal model which can be considered for performing a time series analysis on data that is volatile and heteroscedastic (not having constant variance). Our dataset encapsulates all the daily prices of Facebook stocks, dated from when the stocks initially were IPOs (initial public offerings), May 18 2012, till the day our group initially downloaded the dataset itself, that is April 28 2021. We found the number of data points to be 2251. After looking at all the variables in our dataset, we found the best variable to use for time series analysis as the adjusted price of the stock per day, as it additionally accounts for when external companies trade the stock. After installing the required R packages, we initially plotted the time series for the adjusted stock prices taken from dataset directly; this visually showed us the volatility of the data, marking the signature nature of stock prices generally in the market, and its heteroscedasticity, justifying our usage of the GARCH model. We also examined the daily stock returns by graphing the data and saw similar trends. We then did the required steps to evaluate one model first: we used the mean equation, found the residuals and regressed their squared values; these were used for carrying out the ARCH LM-test, from which the presence of ARCH effects proved to be active. After observing the conditional variance and building a GARCH model, we used this model to forecast for values and visually depicted them; we also plotted a graph of the change in conditional variance with date, and found extreme points of volatility. Following this, by changing required parameters, we built different GARCH models. Each time, they were applied to our dataset and the AIC was the calculated criterion to test the accuracy of the model; the lower the AIC, the more accurate the model. The model 1 where we don't see convergence to be the best, as it had the lowest AIC value of 4.2241; we used this model as our predictive model to carry out forecasting. This model was built from information dated in the past, so we tested its accuracy by forecasting for 10 days ahead of the current date, as well as for 150 days ahead. We graphed both the predicted values versus the actual values for ten days ahead forecasting, as well as the values versus predicted days ahead for 150 days. At a low value of 2.939, we found the prediction error of this model to be quite low, and the lines for predicted and actual values were notably almost in sync. Hence, we can affirm that the model we selected is an adequate GARCH model that can be used for forecasting and analyzing the variation of Facebook's stock prices in the future.

Reference

https://rpubs.com/cyobero/arch