Time Series Analysis of Infoysis Stock Returns

Name:Devendra Sanjay Patil - 2234

Introduction:The financial market is a complex system where stock prices are influenced by a myriad of factors. Understanding these dynamics and predicting future prices is a challenging yet crucial task for investors, financial analysts, and economists. This project aims to analyze and forecast the returns of Infoysis stock using time series analysis.

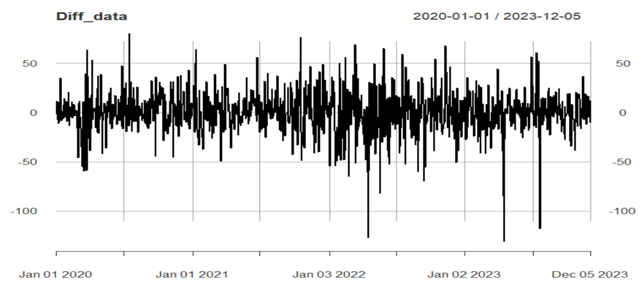
The project utilizes the **quantmod library in R,** a powerful tool for quantitative financial modeling and trading. **The getSymbols function** is used to retrieve the closing prices of Infoysis stock from January 1, 2020, to the present day. This data forms the basis of our time series analysis.The objective is to understand the trends, patterns, and seasonality in the Infoysis stock returns and to build a model that can accurately forecast future returns. This analysis can provide valuable insights into the stock's performance and aid in making informed investment decisions.

Data Description: Let's plot the data to see the behaviour of time series after that we can check the stationarity of data if the data not stationary then we make differencing to make it stationary,

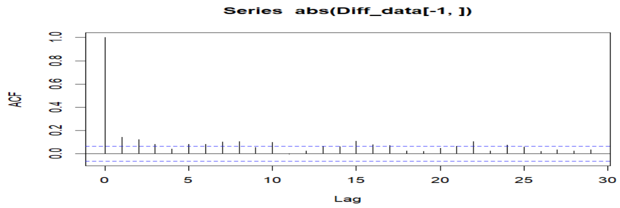

```
> adf.test(D1)

        Augmented Dickey-Fuller Test

data:  D1
Dickey-Fuller = -1.3261, Lag order = 9, p-va
lue = 0.8636
alternative hypothesis: stationary

> adf.test(Diff_data[-1,])

        Augmented Dickey-Fuller Test

data:  Diff_data[-1, ]
Dickey-Fuller = -10.277, Lag order = 9, p-va
lue = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(Diff_data[-1, ]) : p-value small
er than printed p-value
```

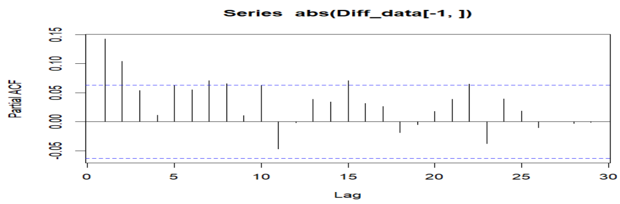
After the differencing the p-value get smaller than 0.05 ,data is now stationary,

Plot of differencing data:



ACF/PACF plot:





The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the absolute returns in data are showing signs of conditional heteroscedasticity, also known as the Autoregressive Conditional Heteroskedasticity (ARCH) effect. This phenomenon is common in financial and economic time series, where the variance of the current error term or innovation is a function of the actual sizes of the previous time periods' error terms. It's characterized by the clustering of large or small residuals.

The LB test also indicate conditional heteroscedasticity.

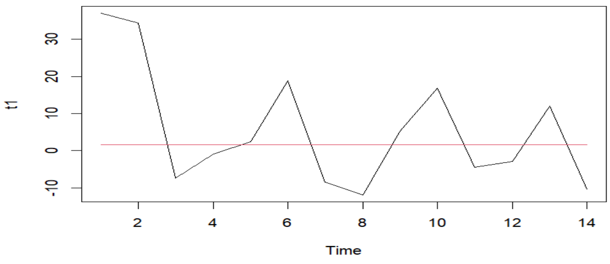
```
> Box.test(abs(Diff_data))

        Box-Pierce test

data:  abs(Diff_data)
X-squared = 19.77, df = 1, p-value = 8.735e-
06
```

ARCH Model:

```
> S=summary(model)

Title:
 GARCH Modelling
Call:
 garchFit(formula = ~1 + garch(3, 0), data = Diff_data[-1, ],
    trace = F)
Mean and Variance Equation:
 data ~ 1 + garch(3, 0)
<environment: 0x000001bddae4a968>
 [data = Diff_data[-1, ]]
Conditional Distribution:
 norm
Coefficient(s):
       mu      omega     alpha1     alpha2     alpha3
  1.66941  291.46979    0.17496    0.11239    0.16366

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu       1.66941     0.64460    2.590 0.009602 **
omega  291.46979    26.72576   10.906  < 2e-16 ***
alpha1   0.17496     0.04576    3.823 0.000132 ***
alpha2   0.11239     0.05109    2.200 0.027822 *
alpha3   0.16366     0.04897    3.342 0.000832 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 -4370.628    normalized:  -4.487298
Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
8.984863 9.009922 8.984811 8.994399
```

From the above output we can see that all the coefficients are significant from the * , "*" represent significany of coefficients and AIC& BIC also appeared to be less so GARCH(3,0) which implies ARCH(3) is fitted well all the return data of infoysis.

After that we divided the data into train & test to check model prediction power we can gone fit the model on train and forecast the values after that plot both values test data value and forecasted values in same graph to check the accuracy of forecast data the graph shown below:



So from above graph we can say that mean forecasted values are passing through the test_data which quit good **red line** is forecasted line and **black is test data line.**

Conclusion:

Overall conclusion is that the ARCH(3)/GARCH(3,0) fitted well on our return data of infoysis

Code:

```
rm(list=ls(all=T))
# Install and import the necessary libraries
install.packages("quantmod")
install.packages("fGarch")
library(quantmod)
library(fGarch)
library(tseries)
# Get the stock market data
getSymbols("INFY.NS", src = "yahoo", from = "2020-01-01", to =
Sys.Date())
data=INFY.NS
D1=data$INFY.NS.Close
plot(D1)
# Calculate the returns of the closing prices
adf.test(D1)
Diff_data=diff(D1)
adf.test(Diff_data[-1,])

plot(Diff_data)
Diff_data
acf(abs(Diff_data[-1,]))
pacf(abs(Diff_data[-1,]))
Box.test(abs(Diff_data))

model = garchFit(~1+garch(3,0),trace = F,data =Diff_data[-1,])
S=summary(model)


#####
length(Diff_data[-1,])
train_data=head(Diff_data[-1,],960)
test_data=tail(Diff_data[-1,],14)
P1=(predict(model,newdata=train_data,n.ahead=14))

plot(t1,type="l")
lines(P1[,1],col=2)
t1=as.ts(test_data)
```