GARCH Model USJapan

Jane

27/04/2021

Forcasting Exchange Rate Using GARCH Model for US Dollar and Japenese Yen

Reading Canadian and Japanes Currency into r

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
USJapCurrencyGARCH <-  read.csv ("USDJPY_Candlestick_1_D_BID_01.01.2000-31.12.2020.csv")%>%
  select('GMT.TIME', CLOSE)%>%
  rename(Date = ('GMT.TIME'), RateUSJapan = ("CLOSE"))


head(USJapCurrencyGARCH)
```

```
##         Date RateUSJapan
## 1 2000-01-03      101.48
## 2 2000-01-04      103.25
## 3 2000-01-05      104.30
## 4 2000-01-06      105.26
## 5 2000-01-07      105.31
## 6 2000-01-10      105.06
```

Conversion of Gmt time to date format

```r
library(dplyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
USJapCurrencyGARCH$Date <- lubridate::ymd(USJapCurrencyGARCH$Date)
head(USJapCurrencyGARCH)
```

```
##         Date RateUSJapan
## 1 2000-01-03      101.48
## 2 2000-01-04      103.25
## 3 2000-01-05      104.30
## 4 2000-01-06      105.26
## 5 2000-01-07      105.31
## 6 2000-01-10      105.06
```

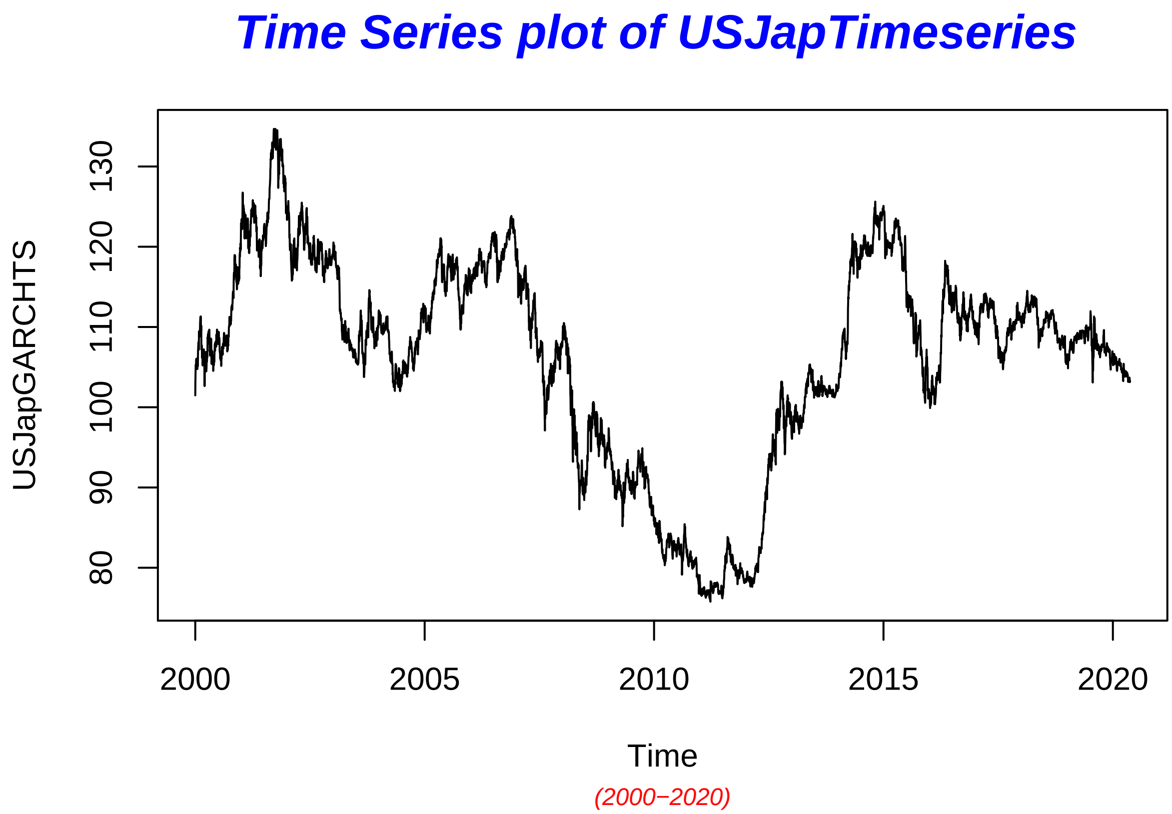
##Checking for obvious errors

```r
#Checking for obvious errors
which(is.na(USJapCurrencyGARCH))
```

```
## integer(0)
```

##Converting the data set into time series object

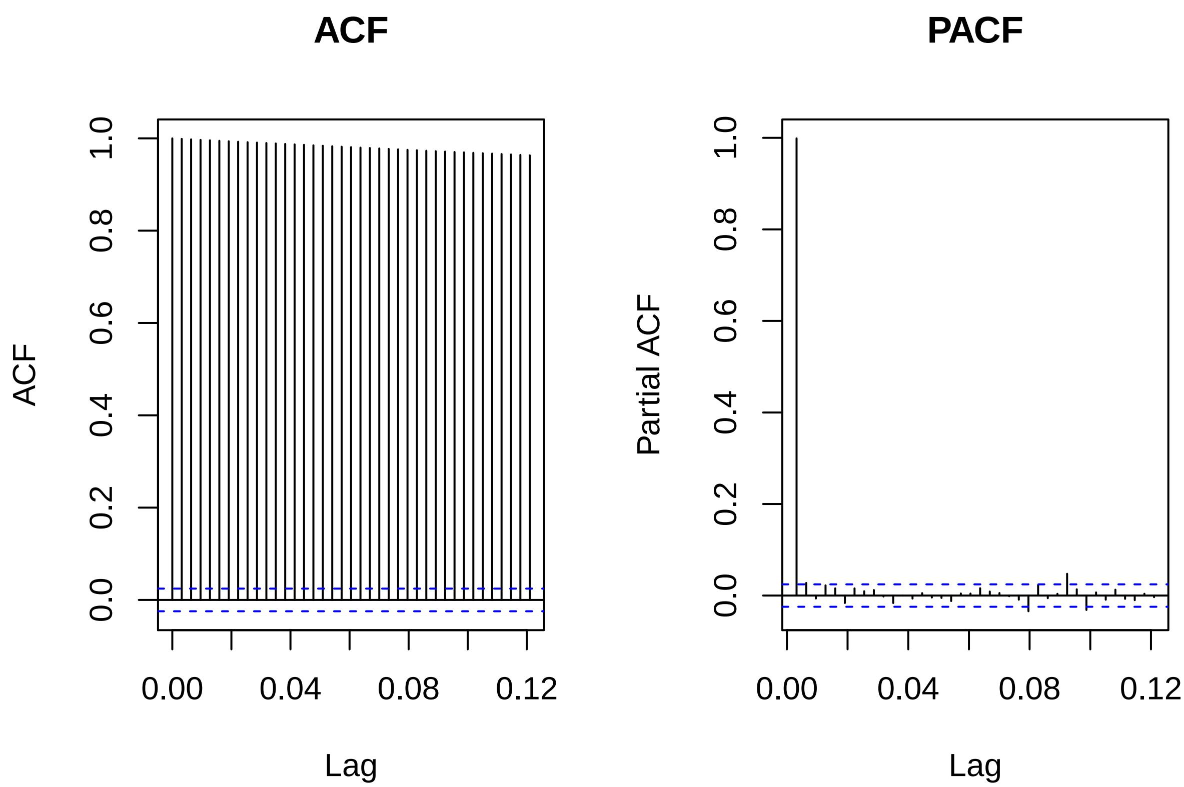
```r
#Converting the data set into time series object
USJapGARCHTS<- ts(as.vector(USJapCurrencyGARCH$Rate),  frequency = 314, start= c(2000,01,03))
plot.ts(USJapGARCHTS)
title("Time Series plot of USJapTimeseries ", sub = "(2000-2020)",
      cex.main = 1.5,   font.main= 4, col.main= "blue",
      cex.sub = 0.75, font.sub = 3, col.sub = "red")
```



Automatic Colleration and Partial Automatic Correlation

##The ACF plot clearly states the high correlation among successive points. It also shows a strong evidence of an existence of a trend as expected from the time series plot. The PACF plot shows one significant correlation on the plot. Unit root test proves the nature of non-stationarity of the series.

```
par(mfrow=c(1,2))
acf(USJapGARCHTS, main=" ACF ")
pacf(USJapGARCHTS, main=" PACF ")
```



Differening the series to ensure stationality

```
ar(diff(USJapGARCHTS))
```

```
##
## Call:
## ar(x = diff(USJapGARCHTS))
##
## Coefficients:
##      1
## -0.025
##
## Order selected 1  sigma^2 estimated as  0.3648
```

##Augmented Dickey-Fuller test ##The Augmented Dickey-Fuller test allows for higher-order autoregressive processes

```
library(tseries)
```

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

```r
adf.test(USJapGARCHTS, alternative = "stationary", k = 0)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  USJapGARCHTS
## Dickey-Fuller = -1.9869, Lag order = 0, p-value = 0.5839
## alternative hypothesis: stationary
```

##Ensuring the stationarity of the series: ##Transformation:Natural Logarithmic transformation is one of the best approach to look for stationarity of the sereis.

```r
USJapLogTranGARCH<-log(USJapGARCHTS)

ar(USJapLogTranGARCH)
```

```
##
## Call:
## ar(x = USJapLogTranGARCH)
##
## Coefficients:
##      1       2
## 0.9724  0.0266
##
## Order selected 2  sigma^2 estimated as  3.292e-05
```

Augmented Dickey-Fuller Test for log Tranformation

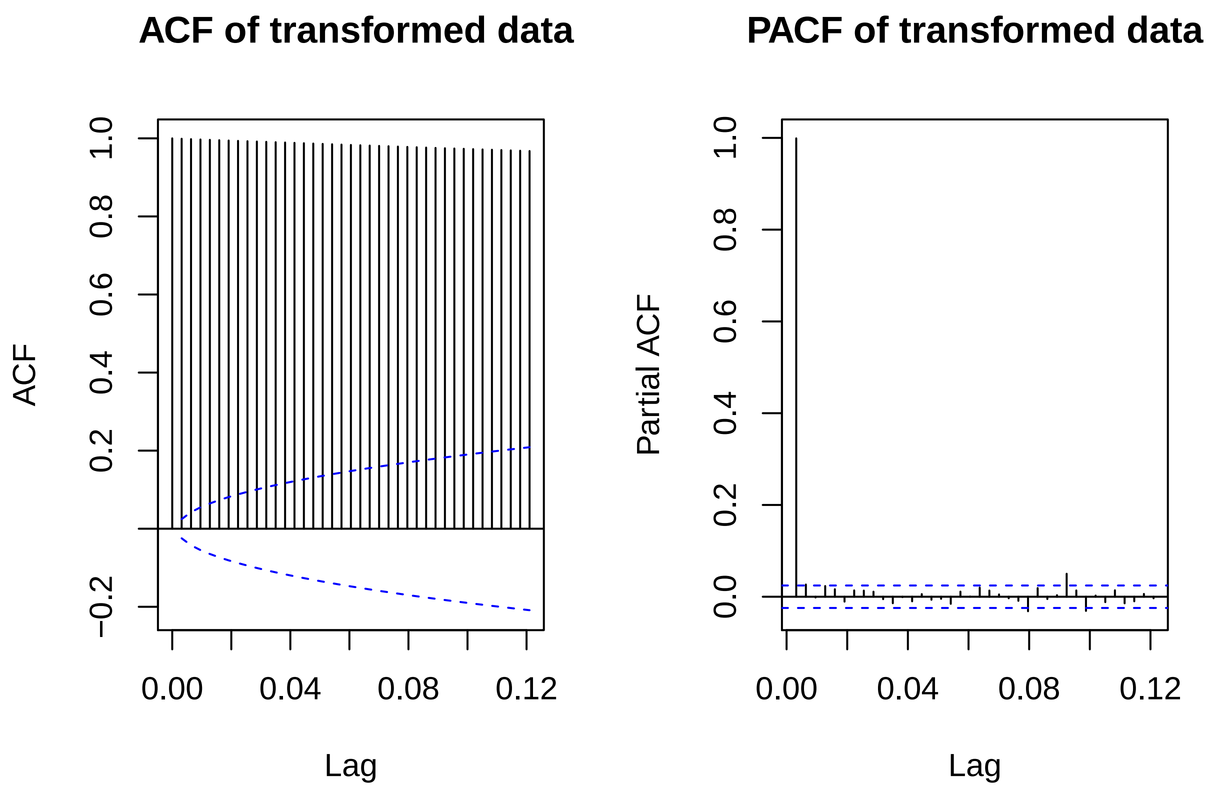
```r
adf.test(USJapLogTranGARCH, alternative = "stationary", k = 0)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  USJapLogTranGARCH
## Dickey-Fuller = -1.8632, Lag order = 0, p-value = 0.6363
## alternative hypothesis: stationary
```

ACF and PACF for log transformation

The ACF and PACF plot are similar to the original series and we can still suspect the non-stationarity of the series and this is also supported by Unit root test.

```r
par(mfrow=c(1,2))
acf(USJapLogTranGARCH, ci.type='ma', main=" ACF of transformed data")

pacf(USJapLogTranGARCH,  main="PACF of transformed data")
```



##Differencing as my next approach to achieve the stationarity of the series.

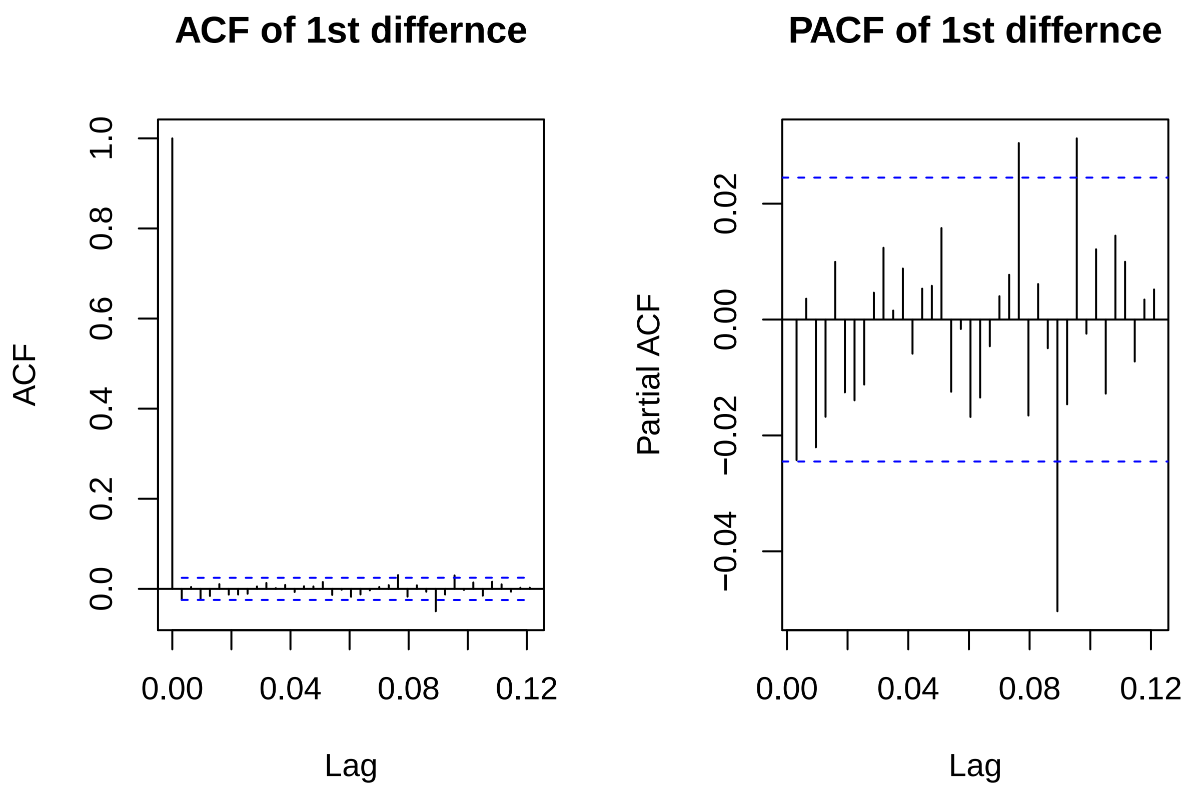
```
DiffUSJapLogTranGARCH<-diff(USJapLogTranGARCH)
ar(DiffUSJapLogTranGARCH)
```

```
##
## Call:
## ar(x = DiffUSJapLogTranGARCH)
##
## Coefficients:
##       1
## -0.0243
##
## Order selected 1  sigma^2 estimated as  3.264e-05
```

Runing ACF and PACF for the log transform Difference

##At the first difference of the transformed series, we can observe the plots of ACF and PACF shows a bit difference to the previous steep decreasing pattern. As we can ensure the assumption of stationarity with Unit-Root test.

```
par(mfrow=c(1,2))
acf(DiffUSJapLogTranGARCH, ci.type='ma', main="ACF of 1st differnce")
pacf(DiffUSJapLogTranGARCH, main="PACF of 1st differnce")
```



Augmented Dickey-Fuller Test for log Tranformation Difference

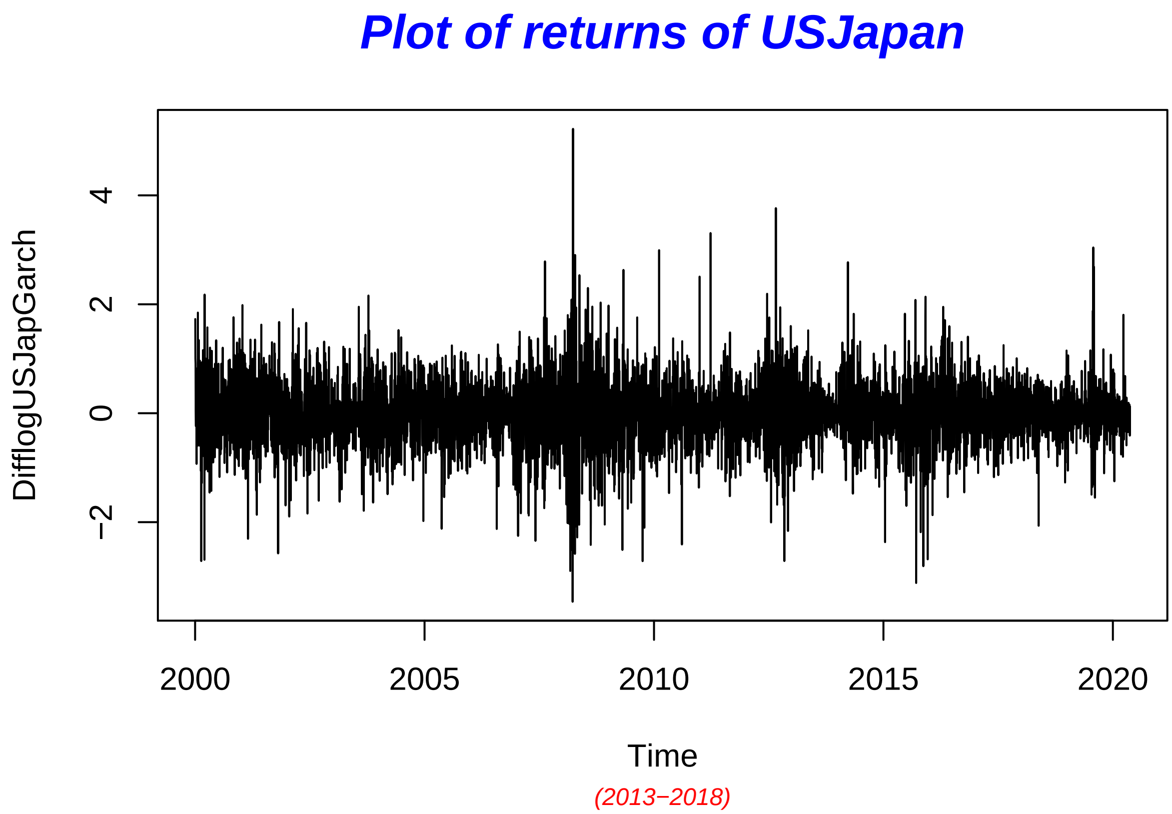
```
adf.test(DiffUSJapLogTranGARCH, alternative = "stationary", k = 0)
```

```
## Warning in adf.test(DiffUSJapLogTranGARCH, alternative = "stationary", k = 0):
## p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  DiffUSJapLogTranGARCH
## Dickey-Fuller = -81.98, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

##Dealing with Conditional Heteroscedaticity: ##To deal with volatile nature of bitcoin series, I have transformed the series as returns of the bitcoin.

```
DifflogUSJapGarch= diff(log(USJapGARCHTS))*100
plot(DifflogUSJapGarch)
title("Plot of returns of USJapan", sub = "(2013-2018)",
      cex.main = 1.5,   font.main= 4, col.main= "blue",
      cex.sub = 0.75, font.sub = 3, col.sub = "red")
```

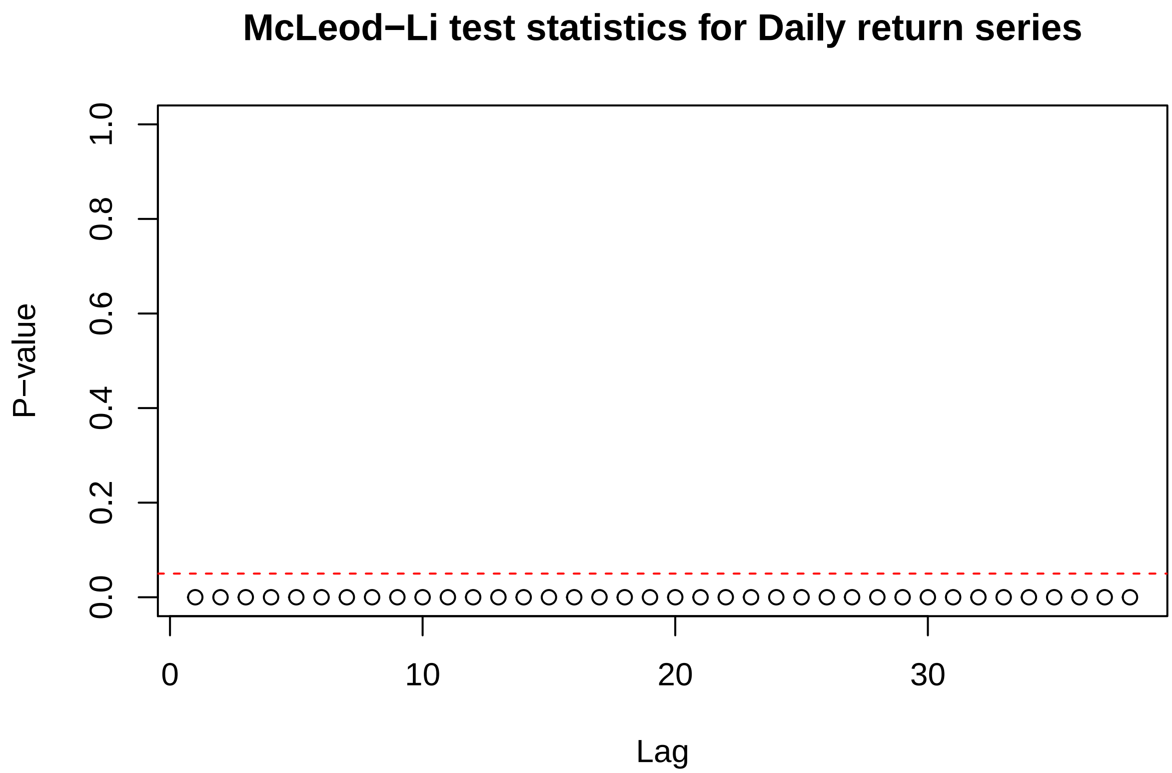


##nature as almost at all lags the p-values fall below the significance levels.

```
library(TSA)
```

```
## Warning: package 'TSA' was built under R version 4.0.5

##
## Attaching package: 'TSA'

## The following object is masked from 'package:readr':
##
##     spec

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```
McLeod.Li.test(y= DifflogUSJapGarch,main="McLeod-Li test statistics for Daily return series")
```

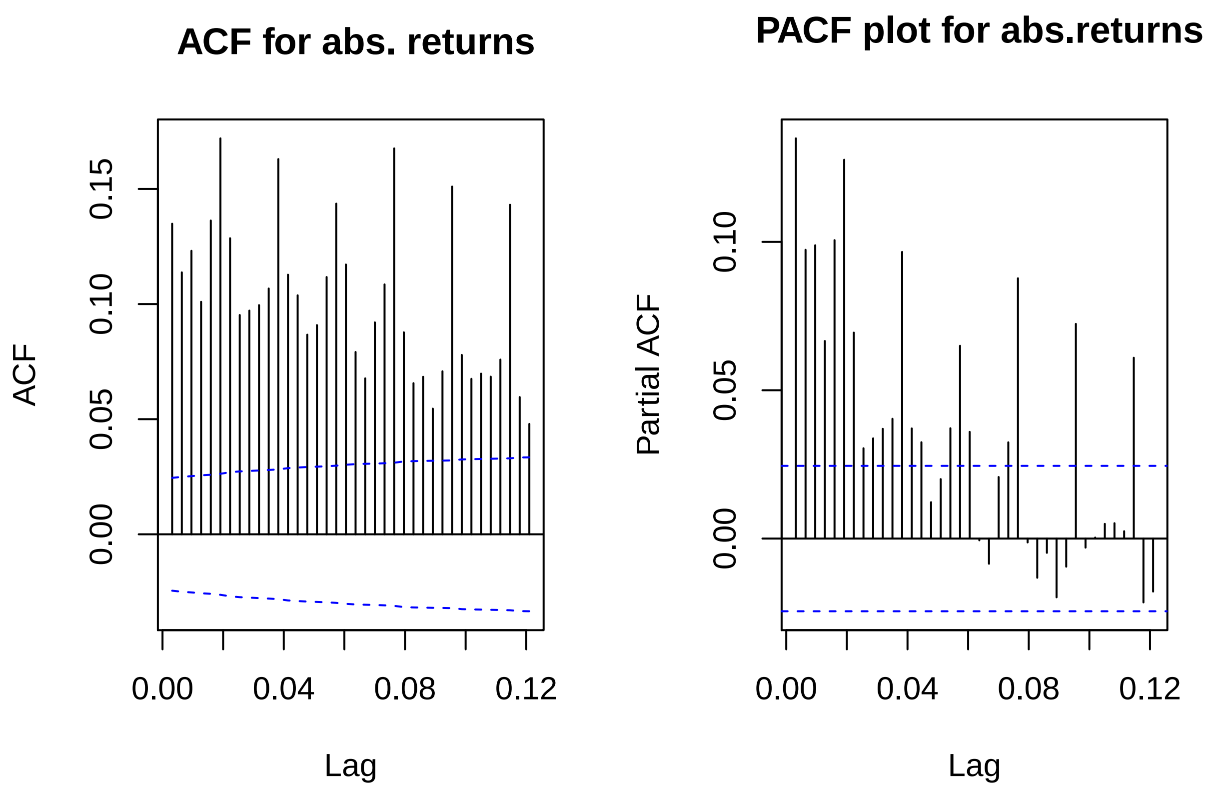


In order to get an order of GARCH , we further transform the return series into absolute values and squared return values.

```
abs = abs(DifflogUSJapGarch)
sqr = DifflogUSJapGarch^2
```

GARCH Model specification:

```
par(mfrow=c(1,2))
acf(abs, ci.type="ma",main=" ACF for abs. returns")
pacf(abs, main=" PACF plot for abs.returns")
```



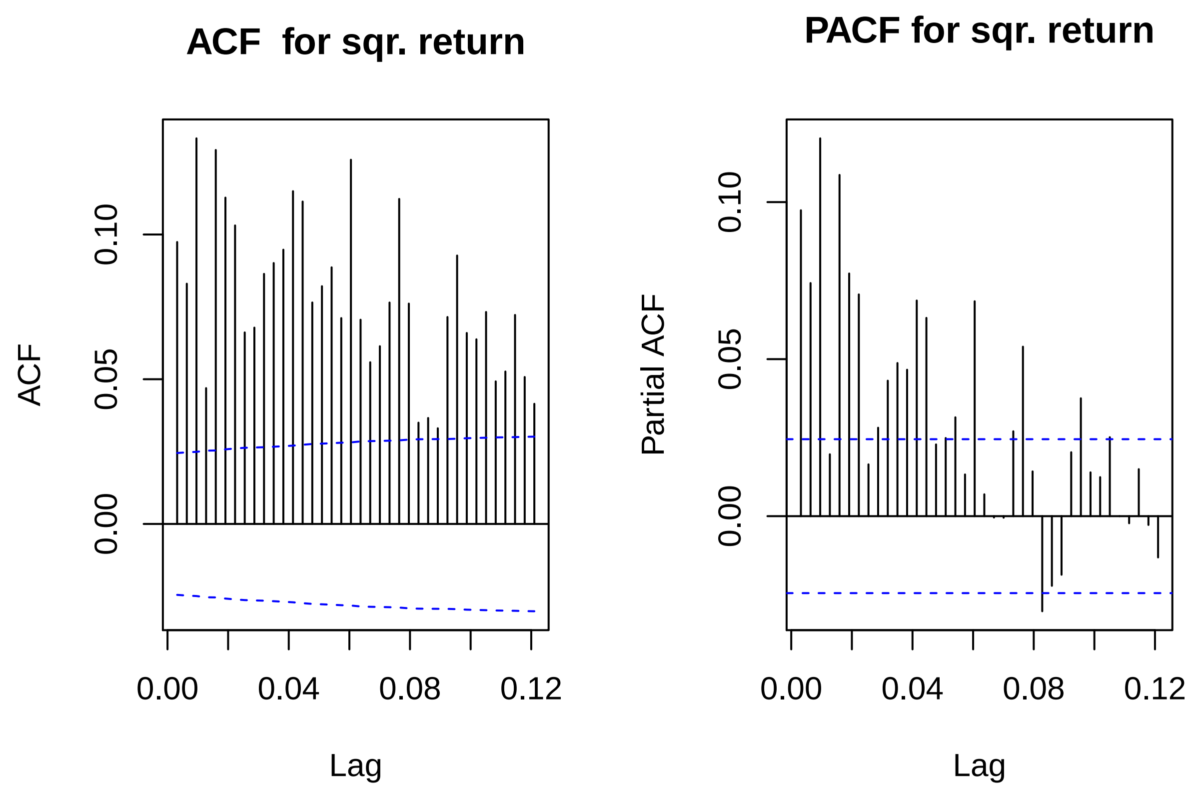
##From ACF and PACF we see many lags are significant. Hence, we plot EACF to get the candidate models

```r
eacf(abs)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x o o x o x o o o o o  x  o  o
## 2 x x o x o x o o o o o  x  o  o
## 3 x x x o o x o o o o o  x  o  o
## 4 x x x x o x o o o o o  x  o  o
## 5 x x x x x o o o o o x  x  x  o
## 6 x x x x x x o o o o o  o  o  o
## 7 x x x x o x x o o o o  o  o  x
```

##From the squared returns ACF and PACF plot, it is not that clear to derive the order of p and q. Hence, I approach EACF and the order of ARMA are ARMA (2,3), ARMA (3,3), ARMA (2,4). Thus, GARCH candidate models would be GARCH (3,2) GARCH (3,3) GARCH (4,2)

```r
par(mfrow=c(1,2))
acf(sqr, ci.type="ma",main="ACF  for sqr. return")
pacf(sqr, main="PACF for sqr. return")
```




```r
eacf(sqr)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x o x x x o o o o o o  o  o  o
## 2 x x x x x o o o x o o  o  o  x
## 3 x x x x x x x o o o o  o  o  o
## 4 x x x x x o o x o o o  o  o  x
## 5 x x x x x o o o o o o  o  o  o
## 6 x x x x x o o o o o o  o  o  o
## 7 x x x x x x x o x o o  o  o  o
```

With reference to the Dickey-Fuller Test, p-value is less than the 0.02 and we can reject the null hypothesis stating the non-stationarity. Hence , we can proceed further for model selection .

#MODEL ESTIMATION: ##GARCH (2,1): for US and Japanese Curruency Pair

```r
# GARCH(2,1)
USJapGARCHFit.21 = garch(DifflogUSJapGarch,order=c(2,1),trace =FALSE)
summary(USJapGARCHFit.21)
```

```
##
## Call:
## garch(x = DifflogUSJapGarch, order = c(2, 1), trace = FALSE)
##
## Model:
## GARCH(2,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.16155 -0.54253  0.01511  0.53900  8.86992
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 3.733e-03   7.646e-04    4.883 1.04e-06 ***
## a1 5.114e-02   9.051e-03    5.650 1.60e-08 ***
## b1 9.381e-01   1.838e-01    5.103 3.35e-07 ***
## b2 1.098e-06   1.735e-01    0.000        1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2674.3, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.11234, df = 1, p-value = 0.7375
```

GARCH (2,2):

##This model can be interpreted as an overfit model of GARCH(2,1) and p values from residual tests confirms that residuals are highly correlated. Thus this model is not consider to be a good fit.

```
USJapGARCHFit.22 = garch(DifflogUSJapGarch, order =c(2,2),trace =FALSE)
summary(USJapGARCHFit.22)
```

```
##
## Call:
## garch(x = DifflogUSJapGarch, order = c(2, 2), trace = FALSE)
##
## Model:
## GARCH(2,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.22747 -0.54280  0.01508  0.53969  8.88490
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
```

```
## a0  0.006611    0.002066    3.200  0.00138 **
## a1  0.054744    0.007533    7.267 3.67e-13 ***
## a2  0.035282    0.029611    1.192  0.23345
## b1  0.164898    0.494353    0.334  0.73871
## b2  0.725974    0.464509    1.563  0.11808
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2679.5, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.22776, df = 1, p-value = 0.6332
```

##GARCH (3,1): ##This model can be interpreted as an overfit model of GARCH(2,1) and GARCH (2,2). This model may not be consider to be a good fit.

```r
USJapGARCHFit.31 = garch(DifflogUSJapGarch,order=c(3,1),trace =FALSE)
summary(USJapGARCHFit.31)
```

```
##
## Call:
## garch(x = DifflogUSJapGarch, order = c(3, 1), trace = FALSE)
##
## Model:
## GARCH(3,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.40229 -0.54348  0.01483  0.53776  8.87886
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 5.437e-03   9.133e-04    5.953 2.64e-09 ***
## a1 7.321e-02   9.941e-03    7.364 1.78e-13 ***
## b1 4.754e-01   1.115e-01    4.262 2.02e-05 ***
## b2 4.358e-01   1.203e-01    3.623 0.000291 ***
## b3 3.430e-15   1.150e-01    0.000 1.000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2704.6, df = 2, p-value < 2.2e-16
##
```

```
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 1.2844, df = 1, p-value = 0.2571
```

##GARCH (3,2): ##This model can be interpreted as an overfitting model and p values from residual tests confirms that residuals are highly correlated. Thus this model is not consider to be a good fit.

GARCH(3,2)

```r
USJapGARCHFit.32 = garch(DifflogUSJapGarch,order=c(3,2),trace =FALSE)
summary(USJapGARCHFit.32)
```

```
##
## Call:
## garch(x = DifflogUSJapGarch, order = c(3, 2), trace = FALSE)
##
## Model:
## GARCH(3,2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0336 -0.5439  0.0150  0.5402  8.8626
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 6.378e-03   1.081e-03    5.900 3.64e-09 ***
## a1 4.622e-02   8.643e-03    5.348 8.91e-08 ***
## a2 4.313e-02   1.661e-02    2.597  0.00941 **
## b1 5.291e-01   2.632e-01    2.010  0.04439 *
## b2 9.907e-15   3.222e-01    0.000  1.00000
## b3 3.633e-01   1.351e-01    2.689  0.00718 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2636.8, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.013932, df = 1, p-value = 0.906
```

GARCH (3,3):

This model can be interpreted as an overfitting model and p values from residual tests confirms that residuals are highly correlated. Thus, this model is not consider to be a good fit.

GARCH(3,3)

```
USJapGARCHFit.33 = garch(DifflogUSJapGarch,order=c(3,3),trace =FALSE)
summary(USJapGARCHFit.33)
```

```
##
## Call:
## garch(x = DifflogUSJapGarch, order = c(3, 3), trace = FALSE)
##
## Model:
## GARCH(3,3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.21769 -0.54468  0.01501  0.54149  8.75253
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 1.000e-02   1.190e-03    8.406  < 2e-16 ***
## a1 3.433e-02   6.157e-03    5.576 2.47e-08 ***
## a2 4.667e-02   5.669e-03    8.233 2.22e-16 ***
## a3 5.982e-02   6.897e-03    8.674  < 2e-16 ***
## b1 1.459e-15   8.326e-02    0.000    1.000
## b2 9.437e-02   9.058e-02    1.042    0.297
## b3 7.364e-01   7.932e-02    9.283  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2623.1, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.15463, df = 1, p-value = 0.6941
```

##GARCH (4,2): ##This model can be interpreted as an overfitting model and p values from residual tests confirms that residuals are highly correlated. Thus, this model is not considered to be a good fit.

```
USJapGARCHFit.42 = garch(DifflogUSJapGarch,order=c(4,2),trace =FALSE)
summary(USJapGARCHFit.42)
```

```
##
## Call:
## garch(x = DifflogUSJapGarch, order = c(4, 2), trace = FALSE)
##
## Model:
## GARCH(4,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.01161 -0.54514  0.01464  0.54185  8.86427
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 6.916e-03   1.638e-03    4.223 2.41e-05 ***
## a1 5.014e-02   8.815e-03    5.689 1.28e-08 ***
## a2 4.672e-02   2.277e-02    2.052   0.0402 *
## b1 4.615e-01   3.284e-01    1.405   0.1599
## b2 5.581e-14   3.634e-01    0.000   1.0000
## b3 3.783e-01   2.198e-01    1.721   0.0852 .
## b4 4.350e-02   1.610e-01    0.270   0.7870
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2634.6, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.077325, df = 1, p-value = 0.781
```

Model Selection:

##Best possible model is selected by AIC scores of the models. From the below sort function, GARCH(3,1) would be the best model for the return series. From the p-value, 3.1 also has the lowest correlation

```r
library(dLagM)
```

```
## Warning: package 'dLagM' was built under R version 4.0.5
```

```
## Loading required package: nardl
```

```
## Warning: package 'nardl' was built under R version 4.0.5
```

```
## Loading required package: dynlm
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'

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

```
GARCHModelSelectionUSJap = AIC(USJapGARCHFit.21,USJapGARCHFit.22 ,USJapGARCHFit.31,USJapGARCHFit.32,USJa
sortScore(GARCHModelSelectionUSJap, score ="aic")
```

```
##                  df      AIC
## USJapGARCHFit.33  7 10086.34
## USJapGARCHFit.21  4 10087.77
## USJapGARCHFit.32  6 10087.98
## USJapGARCHFit.22  5 10088.96
## USJapGARCHFit.42  7 10089.54
## USJapGARCHFit.31  5 10090.43
```

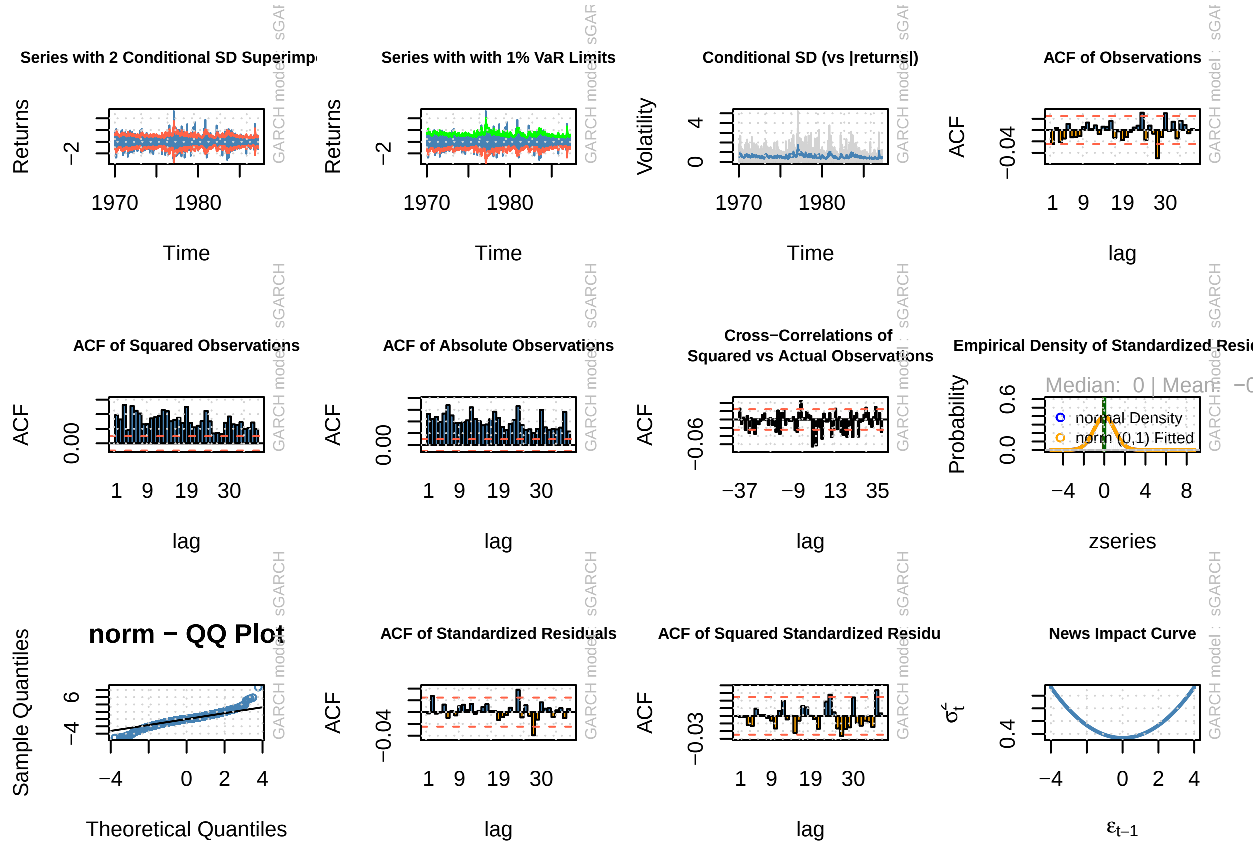
Model Fitting:

```
library(rugarch)
```

```
## Warning: package 'rugarch' was built under R version 4.0.5

## Loading required package: parallel

##
## Attaching package: 'rugarch'

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

```
USJapmodel3.1<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(3, 1)),
                  mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),
                  distribution.model = "norm")

USJapgarchMODEL3.1<-ugarchfit(spec=USJapmodel3.1,data=DifflogUSJapGarch, out.sample = 100)
plot(USJapgarchMODEL3.1,which="all")
```

```
##
## please wait...calculating quantiles...
```



##Model Diagnostics

```
USJapgarchMODEL3.1
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(3,1)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.006159    0.005763   1.068723 0.285195
## ar1     0.851688    0.159682   5.333646 0.000000
## ma1    -0.863664    0.153637  -5.621455 0.000000
## omega   0.003811    0.000896   4.252299 0.000021
## alpha1  0.049886    0.013065   3.818356 0.000134
## alpha2  0.000012    0.017723   0.000697 0.999444
## alpha3  0.000704    0.013368   0.052689 0.957980
## beta1   0.938593    0.008696 107.927733 0.000000
```

```
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.006159    0.006164  0.999099 0.317747
## ar1     0.851688    0.171652  4.961713 0.000001
## ma1    -0.863664    0.166495 -5.187313 0.000000
## omega   0.003811    0.001904  2.001798 0.045306
## alpha1  0.049886    0.018233  2.736064 0.006218
## alpha2  0.000012    0.027721  0.000446 0.999644
## alpha3  0.000704    0.024932  0.028251 0.977462
## beta1   0.938593    0.019872 47.232805 0.000000
##
## LogLikelihood : -5015.764
##
## Information Criteria
## ------------------------------------
##
## Akaike       1.5956
## Bayes        1.6042
## Shibata      1.5956
## Hannan-Quinn 1.5986
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.03346 0.85486
## Lag[2*(p+q)+(p+q)-1][5]   3.98559 0.06904
## Lag[4*(p+q)+(p+q)-1][9]   5.16693 0.41647
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                     0.01447  0.9042
## Lag[2*(p+q)+(p+q)-1][11]   2.37962  0.9322
## Lag[4*(p+q)+(p+q)-1][19]   5.64868  0.8999
## d.o.f=4
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[5]    0.5726 0.500 2.000  0.4492
## ARCH Lag[7]    0.5815 1.473 1.746  0.8750
## ARCH Lag[9]    0.7513 2.402 1.619  0.9617
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.6942
## Individual Statistics:
## mu     0.05844
## ar1    0.12246
## ma1    0.12676
## omega  0.58538
```

```
## alpha1 0.27295
## alpha2 0.25994
## alpha3 0.28349
## beta1  0.51291
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.89 2.11 2.59
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value     prob sig
## Sign Bias           0.2484 0.803818
## Negative Sign Bias  1.8963 0.057973   *
## Positive Sign Bias  1.4357 0.151125
## Joint Effect       13.0177 0.004599 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     315.6    1.250e-55
## 2    30     344.3    1.243e-55
## 3    40     366.5    7.816e-55
## 4    50     388.8    2.031e-54
##
##
## Elapsed time : 0.5397758
```

Forecasting

```
forcgarchUSJap = ugarchforecast(USJapgarchMODEL3.1, data = DiffCanJapLogTran, n.ahead = 100, n.roll =10
print(forcgarchUSJap)
```

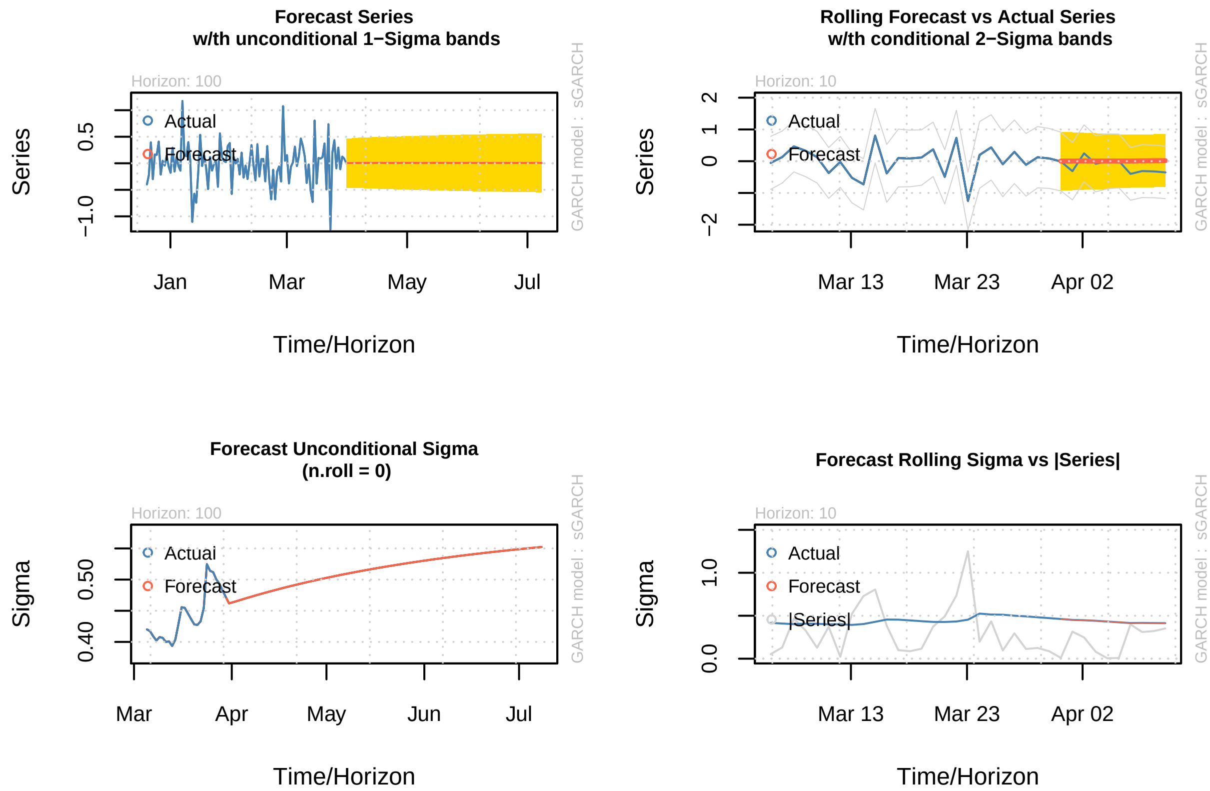
```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 100
## Roll Steps: 10
## Out of Sample: 100
##
## 0-roll forecast [T0=1987-03-30 03:00:00]:
##         Series  Sigma
## T+1   0.003425 0.4618
## T+2   0.003831 0.4633
## T+3   0.004176 0.4648
## T+4   0.004470 0.4663
## T+5   0.004720 0.4679
## T+6   0.004934 0.4694
## T+7   0.005115 0.4710
```

```
## T+8   0.005270 0.4725
## T+9   0.005402 0.4739
## T+10  0.005514 0.4754
## T+11  0.005610 0.4768
## T+12  0.005691 0.4782
## T+13  0.005761 0.4796
## T+14  0.005820 0.4810
## T+15  0.005870 0.4824
## T+16  0.005913 0.4837
## T+17  0.005949 0.4850
## T+18  0.005980 0.4863
## T+19  0.006007 0.4876
## T+20  0.006029 0.4889
## T+21  0.006048 0.4902
## T+22  0.006065 0.4914
## T+23  0.006079 0.4926
## T+24  0.006091 0.4938
## T+25  0.006101 0.4950
## T+26  0.006109 0.4962
## T+27  0.006117 0.4973
## T+28  0.006123 0.4985
## T+29  0.006128 0.4996
## T+30  0.006133 0.5007
## T+31  0.006137 0.5018
## T+32  0.006140 0.5029
## T+33  0.006143 0.5040
## T+34  0.006145 0.5050
## T+35  0.006147 0.5061
## T+36  0.006149 0.5071
## T+37  0.006150 0.5081
## T+38  0.006152 0.5091
## T+39  0.006153 0.5101
## T+40  0.006154 0.5111
## T+41  0.006154 0.5120
## T+42  0.006155 0.5130
## T+43  0.006156 0.5139
## T+44  0.006156 0.5149
## T+45  0.006156 0.5158
## T+46  0.006157 0.5167
## T+47  0.006157 0.5176
## T+48  0.006157 0.5185
## T+49  0.006158 0.5193
## T+50  0.006158 0.5202
## T+51  0.006158 0.5210
## T+52  0.006158 0.5219
## T+53  0.006158 0.5227
## T+54  0.006158 0.5235
## T+55  0.006158 0.5243
## T+56  0.006158 0.5251
## T+57  0.006158 0.5259
## T+58  0.006158 0.5267
## T+59  0.006158 0.5275
## T+60  0.006159 0.5282
## T+61  0.006159 0.5290
```

```
## T+62  0.006159 0.5297
## T+63  0.006159 0.5305
## T+64  0.006159 0.5312
## T+65  0.006159 0.5319
## T+66  0.006159 0.5326
## T+67  0.006159 0.5333
## T+68  0.006159 0.5340
## T+69  0.006159 0.5347
## T+70  0.006159 0.5354
## T+71  0.006159 0.5360
## T+72  0.006159 0.5367
## T+73  0.006159 0.5373
## T+74  0.006159 0.5380
## T+75  0.006159 0.5386
## T+76  0.006159 0.5392
## T+77  0.006159 0.5399
## T+78  0.006159 0.5405
## T+79  0.006159 0.5411
## T+80  0.006159 0.5417
## T+81  0.006159 0.5423
## T+82  0.006159 0.5429
## T+83  0.006159 0.5434
## T+84  0.006159 0.5440
## T+85  0.006159 0.5446
## T+86  0.006159 0.5451
## T+87  0.006159 0.5457
## T+88  0.006159 0.5462
## T+89  0.006159 0.5467
## T+90  0.006159 0.5473
## T+91  0.006159 0.5478
## T+92  0.006159 0.5483
## T+93  0.006159 0.5488
## T+94  0.006159 0.5493
## T+95  0.006159 0.5498
## T+96  0.006159 0.5503
## T+97  0.006159 0.5508
## T+98  0.006159 0.5513
## T+99  0.006159 0.5518
## T+100 0.006159 0.5523
```

plotting

```
plot(forcgarchUSJap, which= "all")
```



Forecasting the rate

```
RateUSJapGarch = 81.074
  RUSJGARCH <-c(0.003425, 0.003831, 0.004176, 0.004470,0.004720, 0.004934, 0.005115, 0.005270, 0.005402
0.006159, 0.006159,0.006159,  0.006159, 0.006159, 0.006159, 0.006159, 0.006159, 0.006159,
0.006159, 0.006159, 0.006159, 0.006159, 0.006159,  0.006159, 0.006159, 0.006159, 0.006159, 0.006159, 0.0

)
  USJAPgarch= 0
  for (i in 1:100){
    USJAPgarch =  RateUSJapGarch *((2.71828)^(RUSJGARCH[i]/100))
    print(USJAPgarch)
    RateUSJapGarch=USJAPgarch
  }
```

```
## [1] 81.07678
## [1] 81.07988
## [1] 81.08327
## [1] 81.08689
## [1] 81.09072
## [1] 81.09472
## [1] 81.09887
## [1] 81.10314
## [1] 81.10753
```

```
## [1] 81.112
## [1] 81.11655
## [1] 81.12116
## [1] 81.12584
## [1] 81.13056
## [1] 81.13532
## [1] 81.14012
## [1] 81.14495
## [1] 81.1498
## [1] 81.15467
## [1] 81.15957
## [1] 81.16448
## [1] 81.1694
## [1] 81.17433
## [1] 81.17928
## [1] 81.18423
## [1] 81.18919
## [1] 81.19416
## [1] 81.19913
## [1] 81.2041
## [1] 81.20909
## [1] 81.21407
## [1] 81.21906
## [1] 81.22405
## [1] 81.22904
## [1] 81.23403
## [1] 81.23903
## [1] 81.24402
## [1] 81.24902
## [1] 81.25402
## [1] 81.25902
## [1] 81.26402
## [1] 81.26902
## [1] 81.27403
## [1] 81.27903
## [1] 81.28403
## [1] 81.28904
## [1] 81.29404
## [1] 81.29905
## [1] 81.30405
## [1] 81.30906
## [1] 81.31407
## [1] 81.31908
## [1] 81.32408
## [1] 81.32909
## [1] 81.3341
## [1] 81.33911
## [1] 81.34412
## [1] 81.34913
## [1] 81.35414
## [1] 81.35915
## [1] 81.36416
## [1] 81.36917
## [1] 81.37418
```

```
## [1] 81.37919
## [1] 81.38421
## [1] 81.38922
## [1] 81.39423
## [1] 81.39924
## [1] 81.40426
## [1] 81.40927
## [1] 81.41429
## [1] 81.4193
## [1] 81.42432
## [1] 81.42933
## [1] 81.43435
## [1] 81.43936
## [1] 81.44438
## [1] 81.44939
## [1] 81.45441
## [1] 81.45943
## [1] 81.46444
## [1] 81.46946
## [1] 81.47448
## [1] 81.4795
## [1] 81.48452
## [1] 81.48954
## [1] 81.49455
## [1] 81.49957
## [1] 81.50459
## [1] 81.50961
## [1] 81.51463
## [1] 81.51965
## [1] 81.52468
## [1] 81.5297
## [1] 81.53472
## [1] 81.53974
## [1] 81.54476
## [1] 81.54979
## [1] 81.55481
## [1] 81.55983
```