GARCG Model GBPJPY

Jane

27/04/2021

Forcasting Exchange Rate Using GARCH Model for Bristish Pound And Japanese Yen

Reading GBP and JPY 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
```

```
GBPJPYGARCH <-  read.csv ("GBPJPY_Candlestick_1_D_BID_01.01.2000-31.12.2020.csv")%>%
  select('GMT.TIME', CLOSE)%>%
  rename(Date = ('GMT.TIME'), RateGBPJPY = ("CLOSE"))


head(GBPJPYGARCH)
```

```
##         Date RateGBPJPY
## 1 2000-01-03     166.01
## 2 2000-01-04     168.81
## 3 2000-01-05     171.34
## 4 2000-01-06     173.37
## 5 2000-01-07     172.56
## 6 2000-01-10     171.98
```

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
GBPJPYGARCH$Date <- lubridate::ymd(GBPJPYGARCH$Date)
head(GBPJPYGARCH)
```

```
##         Date RateGBPJPY
## 1 2000-01-03     166.01
## 2 2000-01-04     168.81
## 3 2000-01-05     171.34
## 4 2000-01-06     173.37
## 5 2000-01-07     172.56
## 6 2000-01-10     171.98
```

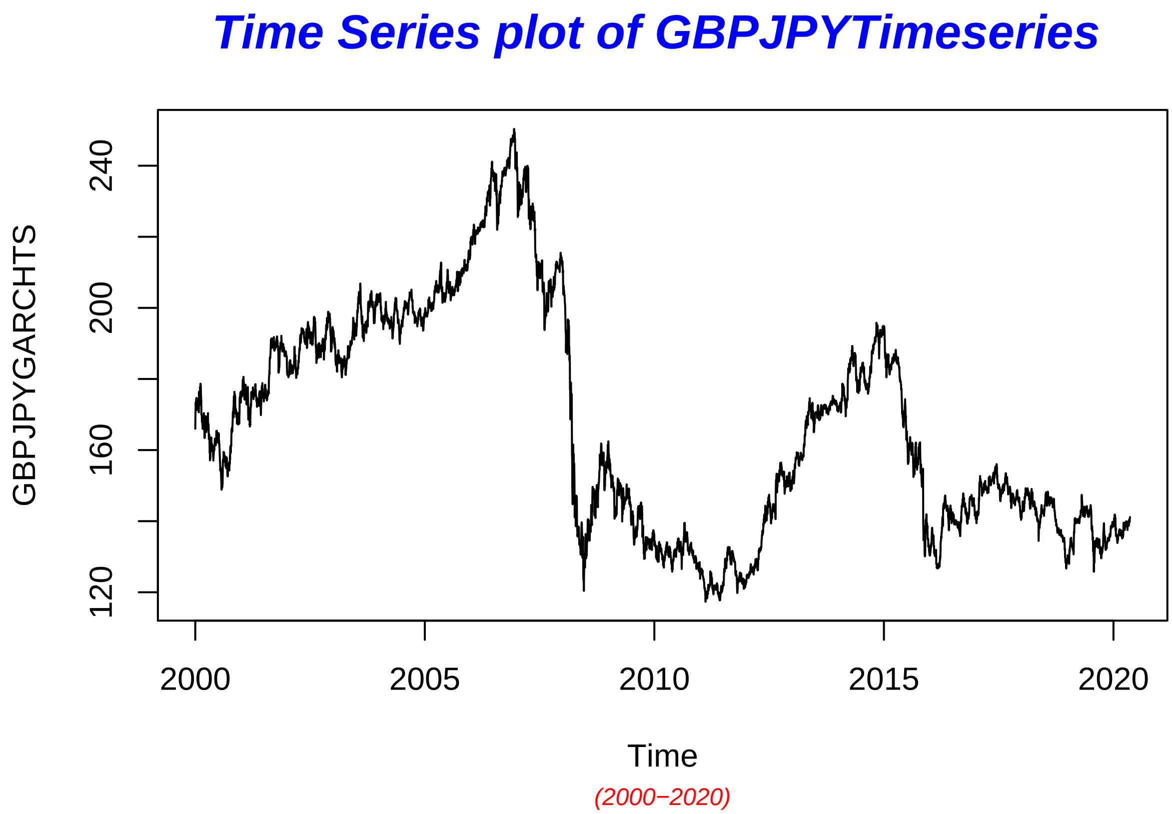
##Checking for obvious errors or missingg value

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

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

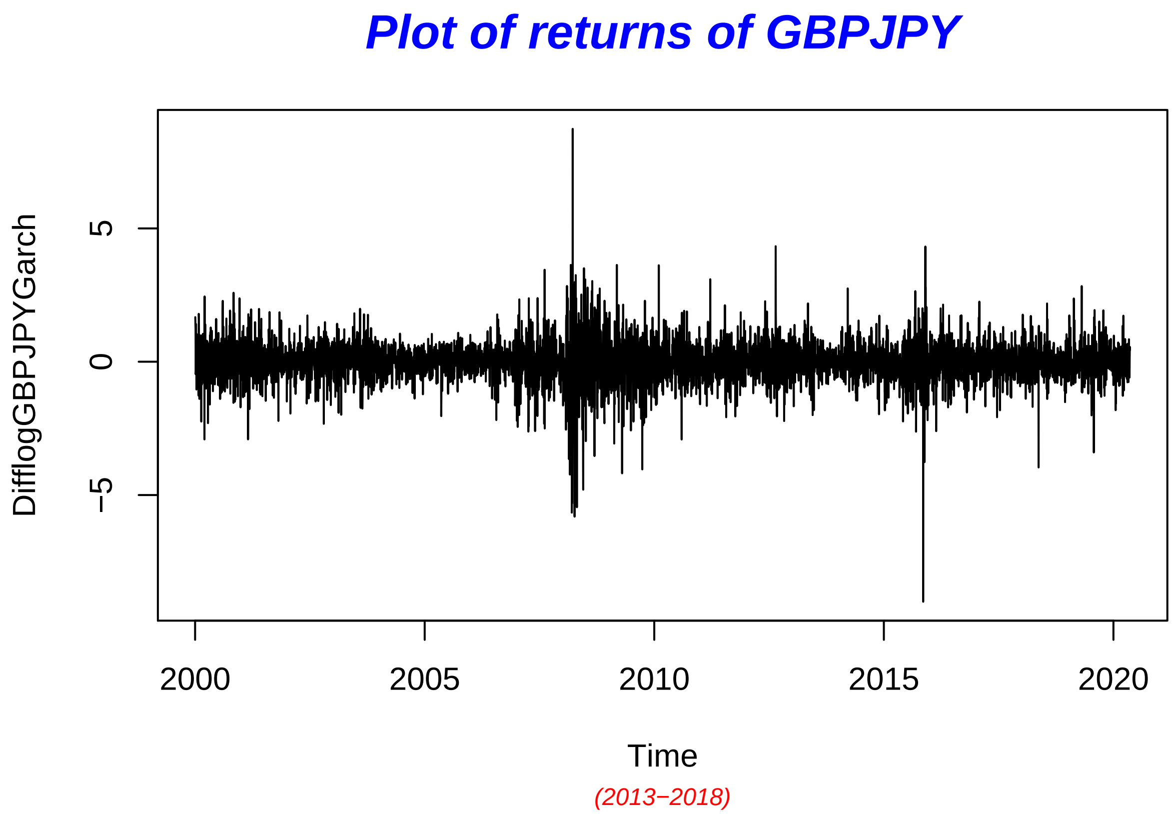
##Converting the data set into time series object

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



##Dealing with Conditional Heteroscedaticity:

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

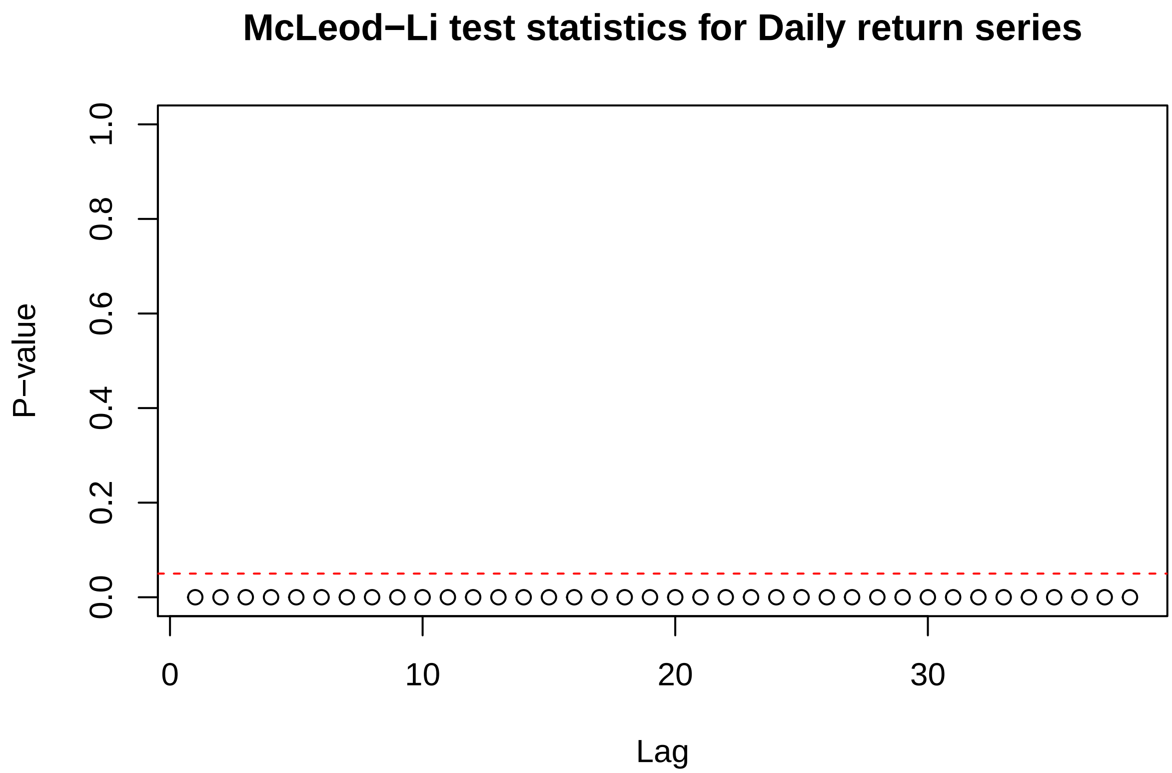


##nature as almost at all lags the p-values fall below the significance levels.

```r
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
```

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

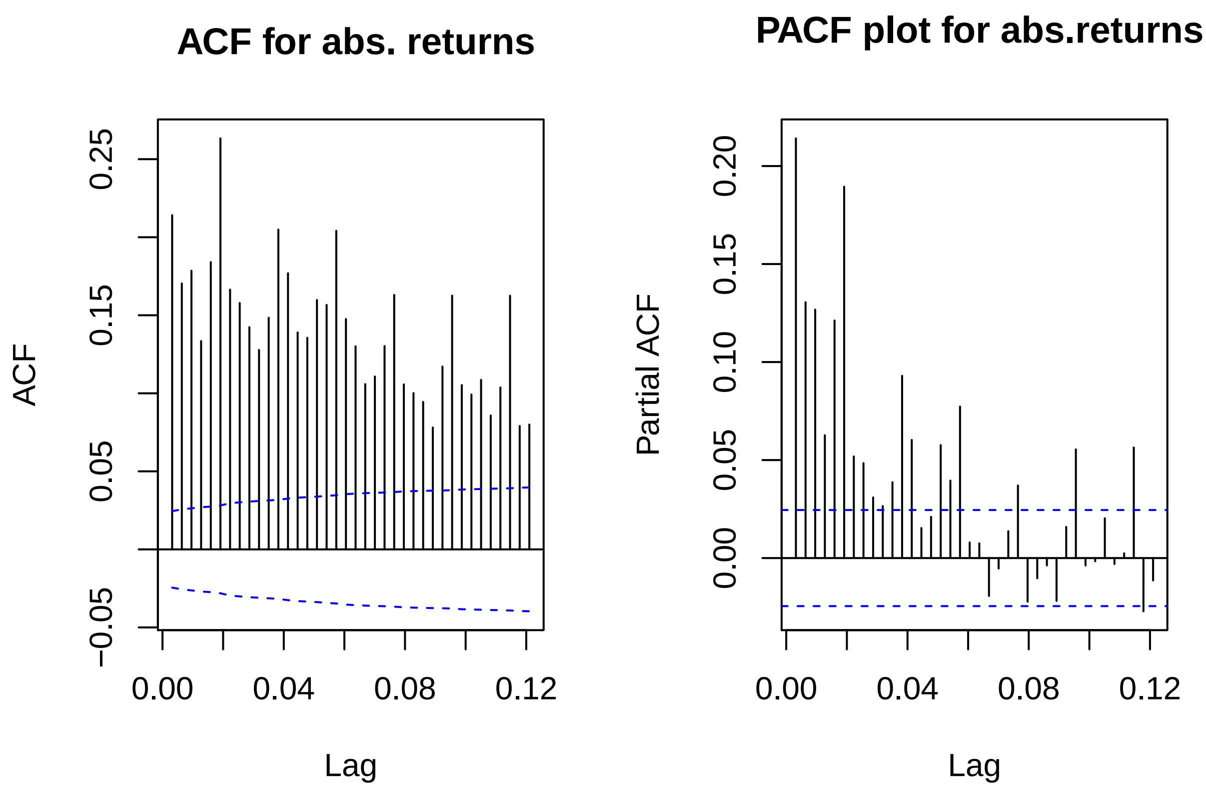


In order to get an order of GARCH , we further transform the return series into absolute values and squared return values.

```
abs = abs(DifflogGBPJPYGarch)
sqr = DifflogGBPJPYGarch^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")
```



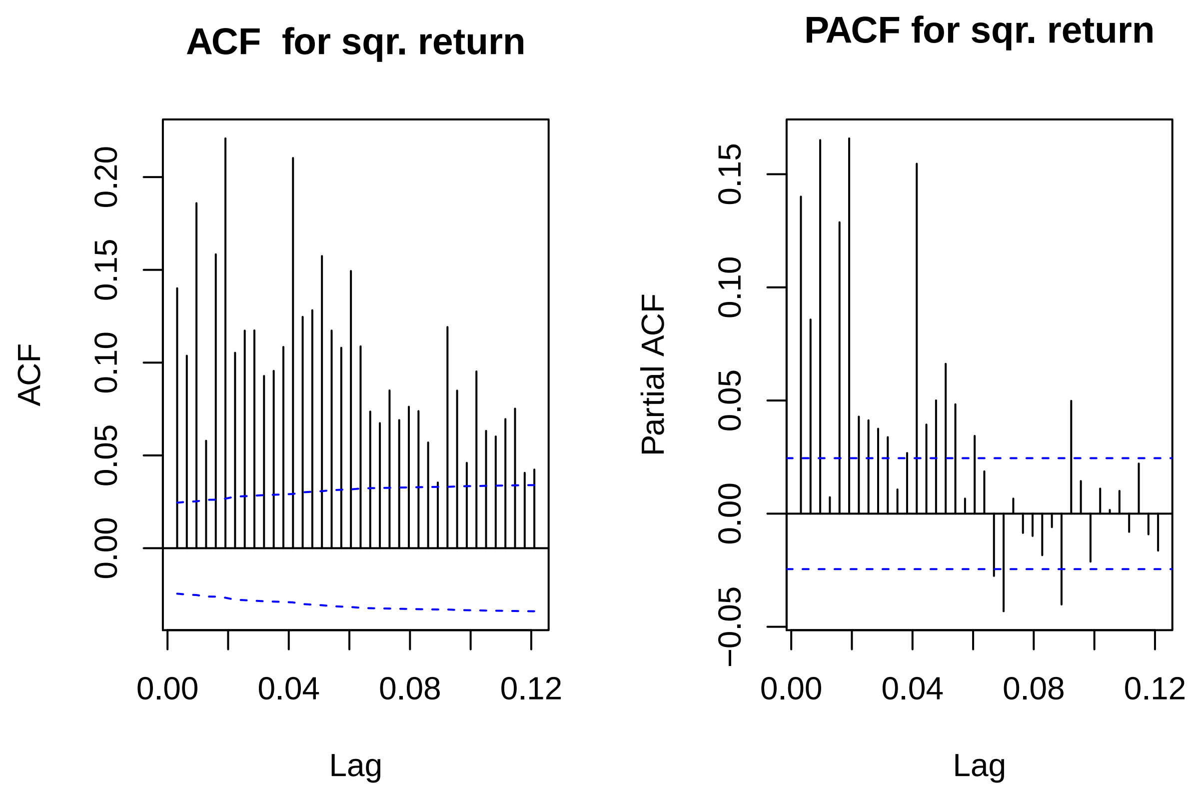
##From ACF and PACF we see many lags are significant. Hence, we plot EACF to get the candidate models

```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 x x x o x x o o o o  x  o  o
## 2 x x o x o x x o o o o  x  x  o
## 3 x x x x o x x o o o o  x  o  o
## 4 x x x o o x x o o o o  x  o  o
## 5 x x x o x x x o x x x  x  o  o
## 6 x x x x x x x x x x o  x  o  o
## 7 x x x o o x x x o x o  x  x  o
```

##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 x x x x x x o o o o  o  x  x
## 2 x x x x x x x o o o o  o  x  x
## 3 x x x x x x x x x o o  o  x  o
## 4 x x x x x x x o o o o  o  x  x
## 5 x x x x x x x o o o o  o  x  x
## 6 x x x x x x o o o x x  x  x  x
## 7 x o o x x o x x o x x  x  x  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 GBP and JPY Curruency Pair

```r
# GARCH(2,1)
library(tseries)
```

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

```
GBPJPYGARCHFit.21 = garch(DifflogGBPJPYGarch,order=c(2,1),trace =FALSE)
summary(GBPJPYGARCHFit.21)
```

```
##
## Call:
## garch(x = DifflogGBPJPYGarch, order = c(2, 1), trace = FALSE)
##
## Model:
## GARCH(2,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.54664 -0.51722  0.02074  0.54171  6.65772
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0  0.007075    0.000929    7.616 2.62e-14 ***
## a1  0.080166    0.004060   19.746  < 2e-16 ***
## b1  0.281123    0.056798    4.949 7.44e-07 ***
## b2  0.625927    0.054423   11.501  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 4956.7, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.55835, df = 1, p-value = 0.4549
```

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.

```
GBPJPYGARCHFit.22 = garch(DifflogGBPJPYGarch, order =c(2,2),trace =FALSE)
summary(GBPJPYGARCHFit.22)
```

```
##
## Call:
## garch(x = DifflogGBPJPYGarch, order = c(2, 2), trace = FALSE)
##
## Model:
```

```
## GARCH(2,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.63377 -0.51689  0.02092  0.53844  6.70066
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 7.321e-03   1.247e-03    5.872 4.31e-09 ***
## a1 7.931e-02   9.207e-03    8.614  < 2e-16 ***
## a2 1.411e-14   1.621e-02    0.000 1.000000
## b1 3.527e-01   1.624e-01    2.172 0.029889 *
## b2 5.547e-01   1.521e-01    3.646 0.000267 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 5124, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.59085, df = 1, p-value = 0.4421
```

##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.

```
GBPJPYGARCHFit.31 = garch(DifflogGBPJPYGarch,order=c(3,1),trace =FALSE)
summary(GBPJPYGARCHFit.31)
```

```
##
## Call:
## garch(x = DifflogGBPJPYGarch, order = c(3, 1), trace = FALSE)
##
## Model:
## GARCH(3,1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5534 -0.5168  0.0207  0.5406  6.6654
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0  0.007172    0.001162    6.170 6.82e-10 ***
## a1  0.081406    0.008621    9.443  < 2e-16 ***
## b1  0.270620    0.088069    3.073  0.00212 **
## b2  0.613316    0.072433    8.467  < 2e-16 ***
## b3  0.021708    0.104267    0.208  0.83507
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 4965.8, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.49673, df = 1, p-value = 0.4809
```

##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
GBPJPYGARCHFit.32 = garch(DifflogGBPJPYGarch,order=c(3,2),trace =FALSE)
summary(GBPJPYGARCHFit.32)
```

```
##
## Call:
## garch(x = DifflogGBPJPYGarch, order = c(3, 2), trace = FALSE)
##
## Model:
## GARCH(3,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.67047 -0.51514  0.02096  0.53768  6.78868
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 8.036e-03   3.049e-03    2.636  0.00839 **
## a1 9.126e-02   9.413e-03    9.695  < 2e-16 ***
## a2 3.998e-15   3.762e-02    0.000  1.00000
## b1 3.610e-01   4.297e-01    0.840  0.40085
## b2 3.322e-01   2.068e-01    1.606  0.10817
## b3 2.018e-01   2.351e-01    0.858  0.39071
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 5317.1, df = 2, p-value < 2.2e-16
##
##
```

```
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.12248, df = 1, p-value = 0.7264
```

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)

```r
GBPJPYGARCHFit.33 = garch(DifflogGBPJPYGarch,order=c(3,3),trace =FALSE)
summary(GBPJPYGARCHFit.33)
```

```
##
## Call:
## garch(x = DifflogGBPJPYGarch, order = c(3, 3), trace = FALSE)
##
## Model:
## GARCH(3,3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.71580 -0.51581  0.02086  0.53810  6.77201
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 8.673e-03   4.430e-03    1.957   0.0503 .
## a1 8.975e-02   1.022e-02    8.779   <2e-16 ***
## a2 7.855e-03   7.642e-02    0.103   0.9181
## a3 1.042e-14   3.370e-02    0.000   1.0000
## b1 2.177e-01   8.282e-01    0.263   0.7927
## b2 4.237e-01   6.187e-01    0.685   0.4935
## b3 2.457e-01   2.265e-01    1.085   0.2781
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 5301, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.16504, df = 1, p-value = 0.6846
```

##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.

```
GBPJPYGARCHFit.42 = garch(DifflogGBPJPYGarch,order=c(4,2),trace =FALSE)
summary(GBPJPYGARCHFit.42)
```

```
##
## Call:
## garch(x = DifflogGBPJPYGarch, order = c(4, 2), trace = FALSE)
##
## Model:
## GARCH(4,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.32418 -0.51109  0.02176  0.54211  6.82509
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 9.602e-03   1.980e-03    4.850 1.24e-06 ***
## a1 1.023e-01   9.541e-03   10.718  < 2e-16 ***
## a2 6.383e-03   2.347e-02    0.272   0.7857
## b1 3.731e-01   1.875e-01    1.990   0.0466 *
## b2 1.184e-01   1.141e-01    1.037   0.2997
## b3 5.157e-16   9.570e-02    0.000   1.0000
## b4 3.822e-01   8.858e-02    4.315 1.60e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 4722, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.00066967, df = 1, p-value = 0.9794
```

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

```
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
```

```r
GARCHModelSelectionGBPJPY = AIC(GBPJPYGARCHFit.21,GBPJPYGARCHFit.22 ,GBPJPYGARCHFit.31,GBPJPYGARCHFit.3
sortScore(GARCHModelSelectionGBPJPY, score ="aic")
```

```
##                   df      AIC
## GBPJPYGARCHFit.42  7 12480.94
## GBPJPYGARCHFit.31  5 12496.58
## GBPJPYGARCHFit.21  4 12497.92
## GBPJPYGARCHFit.22  5 12500.54
## GBPJPYGARCHFit.32  6 12502.07
## GBPJPYGARCHFit.33  7 12504.32
```

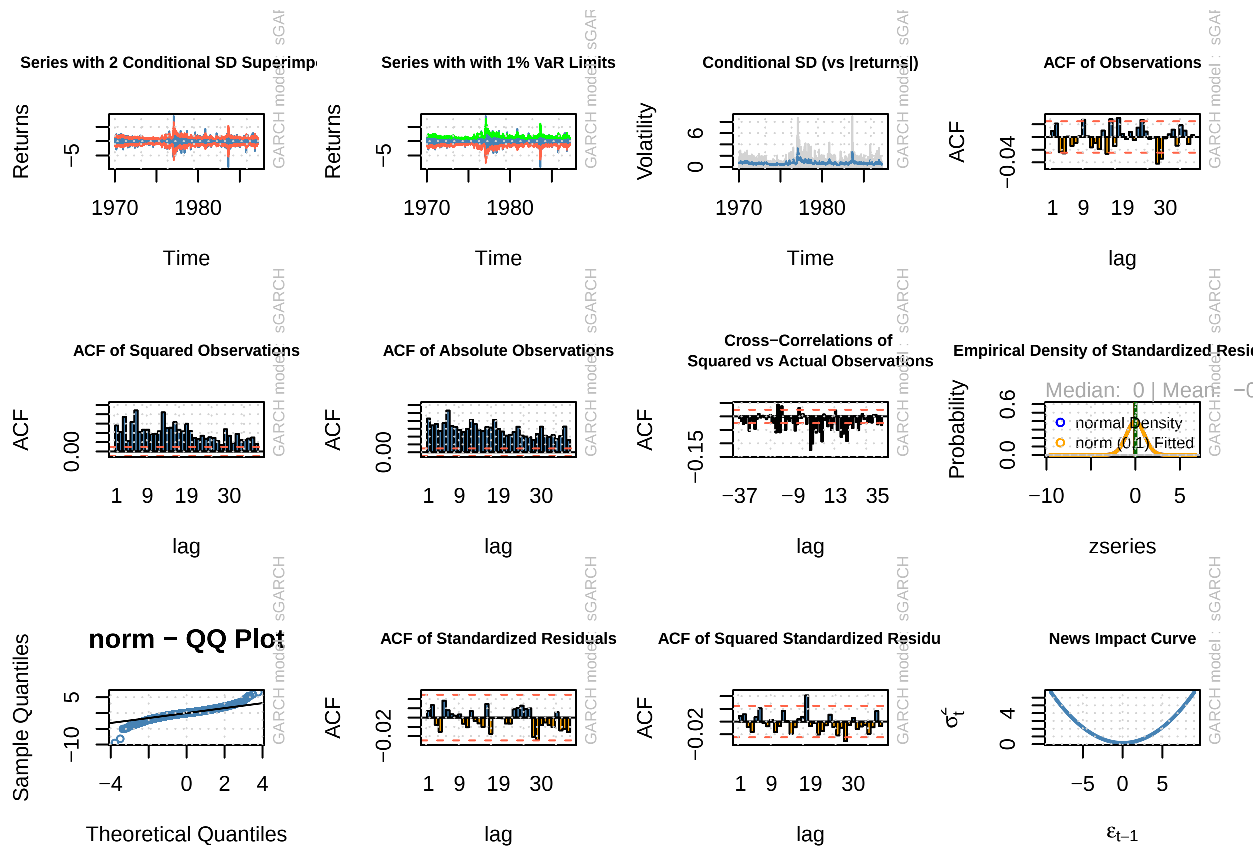
Model Fitting:

```r
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
```

```r
GBPJPYmodel4.2<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(4, 2)),
                  mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),
                  distribution.model = "norm")

GBPJPYgarchMODEL4.2<-ugarchfit(spec=GBPJPYmodel4.2,data=DifflogGBPJPYGarch, out.sample = 100)
plot(GBPJPYgarchMODEL4.2,which="all")
```

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



##Model Diagnostics

```
GBPJPYgarchMODEL4.2
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(4,2)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.009953    0.007544  1.319218 0.187096
## ar1     0.338255    0.392711  0.861333 0.389055
## ma1    -0.322162    0.395069 -0.815459 0.414810
## omega   0.006888    0.001663  4.141388 0.000035
## alpha1  0.080713    0.013121  6.151562 0.000000
## alpha2  0.000000    0.011010  0.000029 0.999977
## alpha3  0.000000    0.015454  0.000001 0.999999
## alpha4  0.000000    0.013550  0.000001 0.999999
```

```
## beta1   0.275429    0.040195  6.852302 0.000000
## beta2   0.631659    0.037299 16.935169 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.009953    0.007726  1.288258 0.197656
## ar1     0.338255    0.213924  1.581191 0.113834
## ma1    -0.322162    0.215716 -1.493457 0.135318
## omega   0.006888    0.003687  1.868006 0.061761
## alpha1  0.080713    0.016792  4.806538 0.000002
## alpha2  0.000000    0.012472  0.000025 0.999980
## alpha3  0.000000    0.022680  0.000001 0.999999
## alpha4  0.000000    0.022413  0.000000 1.000000
## beta1   0.275429    0.020521 13.422143 0.000000
## beta2   0.631659    0.014112 44.760794 0.000000
##
## LogLikelihood : -6167.916
##
## Information Criteria
## ------------------------------------
##
## Akaike       1.9634
## Bayes        1.9741
## Shibata      1.9634
## Hannan-Quinn 1.9671
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.3013  0.5831
## Lag[2*(p+q)+(p+q)-1][5]    2.4498  0.8046
## Lag[4*(p+q)+(p+q)-1][9]    3.9828  0.6950
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                      0.5323  0.4656
## Lag[2*(p+q)+(p+q)-1][17]    7.6935  0.6023
## Lag[4*(p+q)+(p+q)-1][29]   17.2591  0.2820
## d.o.f=6
##
## Weighted ARCH LM Tests
## ------------------------------------
##              Statistic Shape Scale P-Value
## ARCH Lag[7]    0.02393 0.500 2.000  0.8771
## ARCH Lag[9]    0.36343 1.485 1.796  0.9371
## ARCH Lag[11]   1.77820 2.440 1.677  0.8203
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  2.4099
## Individual Statistics:
```

```
## mu     0.09370
## ar1    0.06582
## ma1    0.06554
## omega  0.26862
## alpha1 0.10312
## alpha2 0.12405
## alpha3 0.12105
## alpha4 0.12097
## beta1  0.11150
## beta2  0.11123
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          2.29 2.54 3.05
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias            0.246 0.80566
## Negative Sign Bias   1.970 0.04891  **
## Positive Sign Bias   1.213 0.22500
## Joint Effect        10.962 0.01193  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     344.1    1.681e-61
## 2    30     364.0    1.389e-59
## 3    40     402.6    6.119e-62
## 4    50     397.9    3.803e-56
##
##
## Elapsed time : 0.8677609
```

Forecasting

```
forcgarchGBPJPY= ugarchforecast(GBPJPYgarchMODEL4.2, data = DiffGBPJPYLogTran, n.ahead = 100, n.roll =1
print(forcgarchGBPJPY)
```

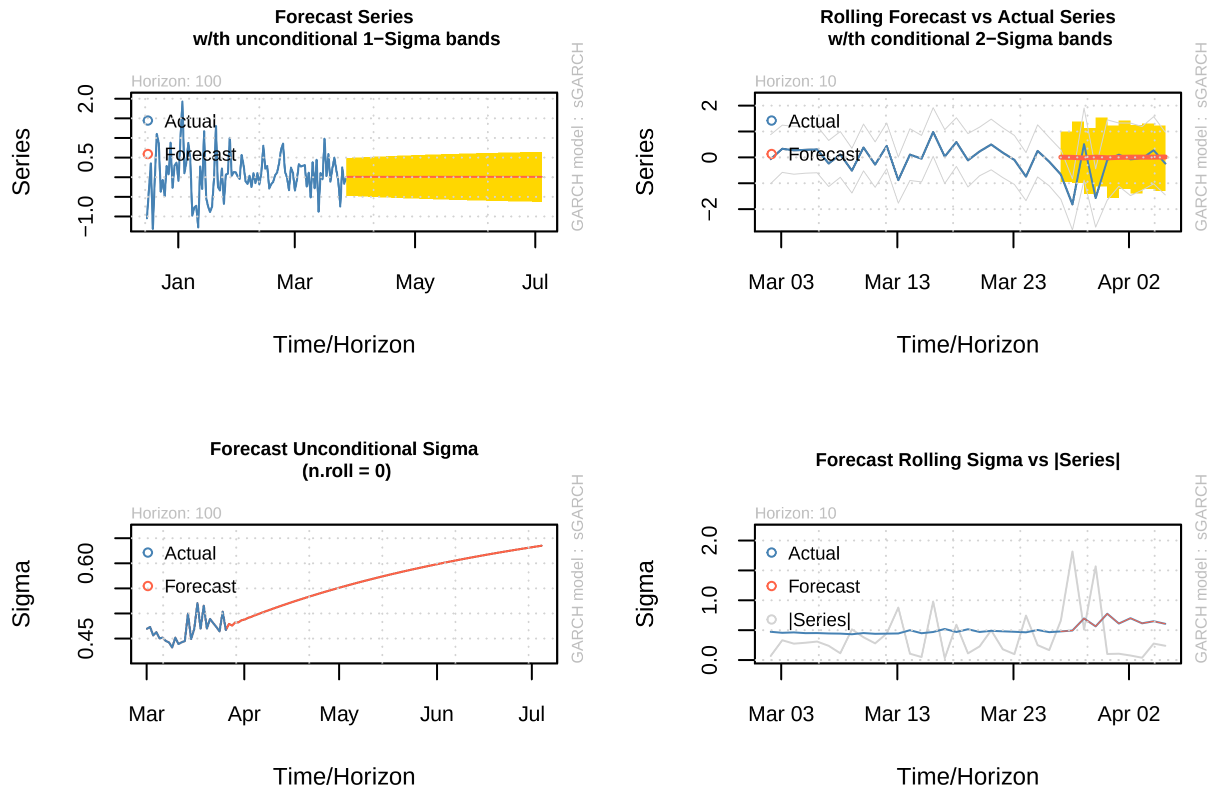
```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 100
## Roll Steps: 10
## Out of Sample: 100
##
## 0-roll forecast [T0=1987-03-26 02:00:00]:
##         Series  Sigma
## T+1   0.007108 0.4792
```

```
## T+2   0.008990 0.4758
## T+3   0.009627 0.4822
## T+4   0.009843 0.4824
## T+5   0.009916 0.4865
## T+6   0.009940 0.4880
## T+7   0.009949 0.4911
## T+8   0.009951 0.4932
## T+9   0.009952 0.4958
## T+10  0.009953 0.4981
## T+11  0.009953 0.5005
## T+12  0.009953 0.5028
## T+13  0.009953 0.5052
## T+14  0.009953 0.5074
## T+15  0.009953 0.5097
## T+16  0.009953 0.5119
## T+17  0.009953 0.5141
## T+18  0.009953 0.5163
## T+19  0.009953 0.5184
## T+20  0.009953 0.5205
## T+21  0.009953 0.5226
## T+22  0.009953 0.5247
## T+23  0.009953 0.5268
## T+24  0.009953 0.5288
## T+25  0.009953 0.5308
## T+26  0.009953 0.5328
## T+27  0.009953 0.5347
## T+28  0.009953 0.5367
## T+29  0.009953 0.5386
## T+30  0.009953 0.5405
## T+31  0.009953 0.5424
## T+32  0.009953 0.5442
## T+33  0.009953 0.5461
## T+34  0.009953 0.5479
## T+35  0.009953 0.5497
## T+36  0.009953 0.5514
## T+37  0.009953 0.5532
## T+38  0.009953 0.5549
## T+39  0.009953 0.5567
## T+40  0.009953 0.5584
## T+41  0.009953 0.5601
## T+42  0.009953 0.5617
## T+43  0.009953 0.5634
## T+44  0.009953 0.5650
## T+45  0.009953 0.5666
## T+46  0.009953 0.5682
## T+47  0.009953 0.5698
## T+48  0.009953 0.5714
## T+49  0.009953 0.5729
## T+50  0.009953 0.5745
## T+51  0.009953 0.5760
## T+52  0.009953 0.5775
## T+53  0.009953 0.5790
## T+54  0.009953 0.5804
## T+55  0.009953 0.5819
```

```
## T+56  0.009953 0.5834
## T+57  0.009953 0.5848
## T+58  0.009953 0.5862
## T+59  0.009953 0.5876
## T+60  0.009953 0.5890
## T+61  0.009953 0.5904
## T+62  0.009953 0.5917
## T+63  0.009953 0.5931
## T+64  0.009953 0.5944
## T+65  0.009953 0.5957
## T+66  0.009953 0.5971
## T+67  0.009953 0.5984
## T+68  0.009953 0.5996
## T+69  0.009953 0.6009
## T+70  0.009953 0.6022
## T+71  0.009953 0.6034
## T+72  0.009953 0.6047
## T+73  0.009953 0.6059
## T+74  0.009953 0.6071
## T+75  0.009953 0.6083
## T+76  0.009953 0.6095
## T+77  0.009953 0.6107
## T+78  0.009953 0.6118
## T+79  0.009953 0.6130
## T+80  0.009953 0.6141
## T+81  0.009953 0.6153
## T+82  0.009953 0.6164
## T+83  0.009953 0.6175
## T+84  0.009953 0.6186
## T+85  0.009953 0.6197
## T+86  0.009953 0.6208
## T+87  0.009953 0.6219
## T+88  0.009953 0.6230
## T+89  0.009953 0.6240
## T+90  0.009953 0.6251
## T+91  0.009953 0.6261
## T+92  0.009953 0.6271
## T+93  0.009953 0.6281
## T+94  0.009953 0.6291
## T+95  0.009953 0.6301
## T+96  0.009953 0.6311
## T+97  0.009953 0.6321
## T+98  0.009953 0.6331
## T+99  0.009953 0.6340
## T+100 0.009953 0.6350
```

plotting

```r
plot(forcgarchGBPJPY, which= "all")
```



Forecasting the rate

```
RateUSCanadaGarch = 1.27219
  RUSCanadaGARCH <-c(-0.0260997,0.0157132, -0.0133859, 0.0068652, -0.0072283, 0.0025799, -0.0042460, 0.(
-0.0014450, -0.0014450, -0.0014450, -0.0014450, -0.0014450, -0.0014450, -0.0014450, -0.0014450, -0.0014


)
  USCanadagarch= 0
  for (i in 1:100){
    USCanadagarch =  RateUSCanadaGarch *((2.71828)^(RUSCanadaGARCH[i]/100))
    print(USCanadagarch)
    RateUSCandaGarch=USCanadagarch
  }
```

```
## [1] 1.271858
## [1] 1.27239
## [1] 1.27202
## [1] 1.272277
## [1] 1.272098
## [1] 1.272223
## [1] 1.272136
## [1] 1.272196
## [1] 1.272154
```

```
## [1] 1.272184
## [1] 1.272163
## [1] 1.272177
## [1] 1.272168
## [1] 1.272174
## [1] 1.27217
## [1] 1.272173
## [1] 1.272171
## [1] 1.272172
## [1] 1.272171
## [1] 1.272172
## [1] 1.272171
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
```

```
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
## [1] 1.272172
```