GARCH Model EUR And JPY

Jane

29/04/2021

Forcasting Exchange Rate Using GARCH Model for EUR And JPY

Reading EUR 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
```

```
EURJPYGARCH<-  read.csv ("EURJPY_Candlestick_1_D_BID_01.01.2000-31.12.2020.csv")%>%
  select('GMT.TIME', CLOSE)%>%
  rename(Date = ('GMT.TIME'), RateEURJPY = ("CLOSE"))


head(EURJPYGARCH)
```

```
##         Date RateEURJPY
## 1 2000-01-03     104.00
## 2 2000-01-04     106.28
## 3 2000-01-05     107.57
## 4 2000-01-06     108.68
## 5 2000-01-07     108.42
## 6 2000-01-10     107.78
```

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

```
##         Date RateEURJPY
## 1 2000-01-03     104.00
## 2 2000-01-04     106.28
## 3 2000-01-05     107.57
## 4 2000-01-06     108.68
## 5 2000-01-07     108.42
## 6 2000-01-10     107.78
```

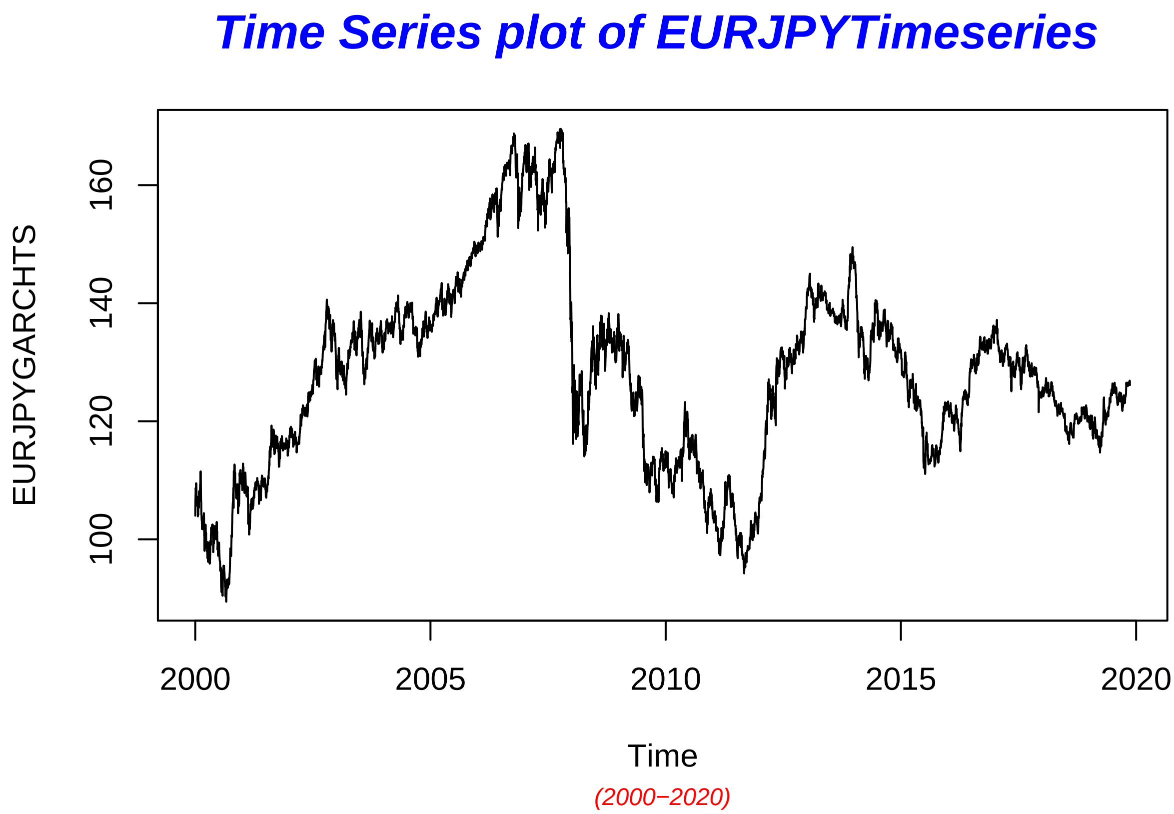
##Checking for obvious errors or missingg value

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

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

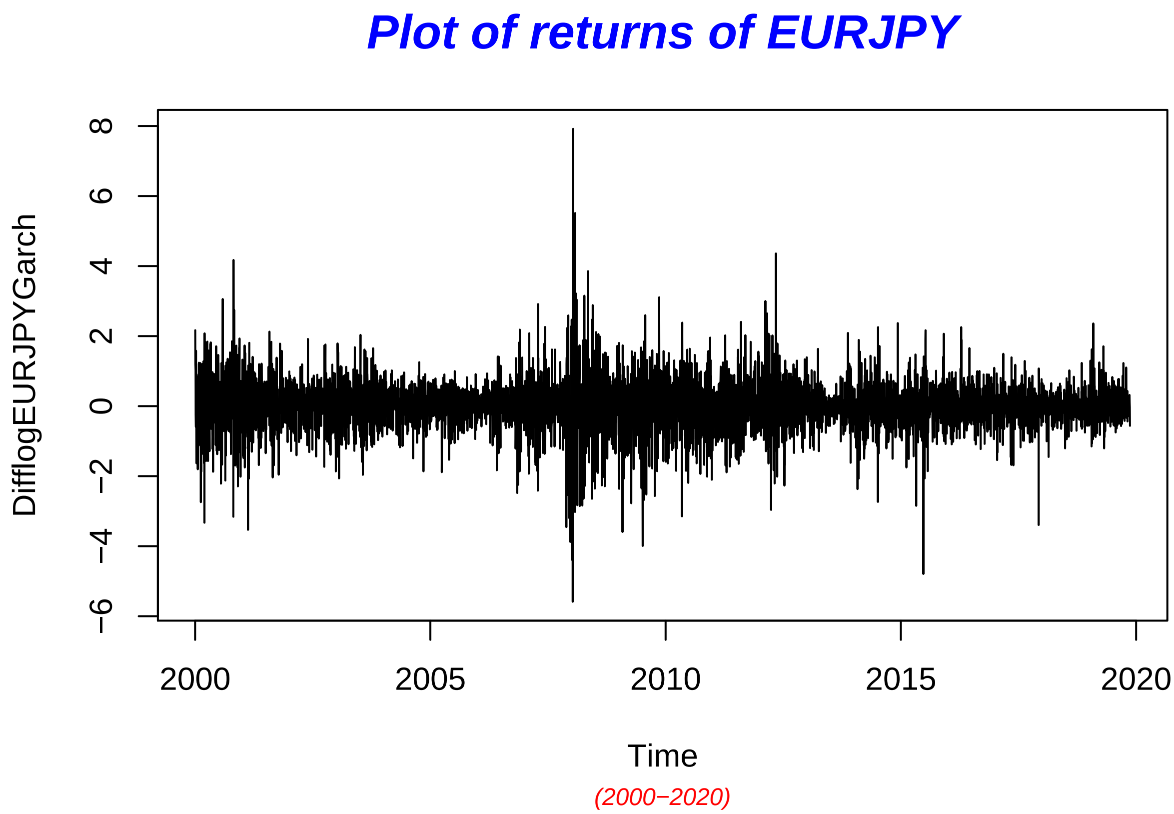
##Converting the data set into time series object

```r
#Converting the data set into time series object
EURJPYGARCHTS<- ts(as.vector(EURJPYGARCH$Rate),  frequency = 322, start= c(2000,01,03))
plot.ts(EURJPYGARCHTS)
title("Time Series plot of EURJPYTimeseries ", 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:

```
DifflogEURJPYGarch= diff(log(EURJPYGARCHTS))*100
plot(DifflogEURJPYGarch)
title("Plot of returns of EURJPY", sub = "(2000-2020)",
      cex.main = 1.5,   font.main= 4, col.main= "blue",
      cex.sub = 0.75, font.sub = 3, col.sub = "red")
```

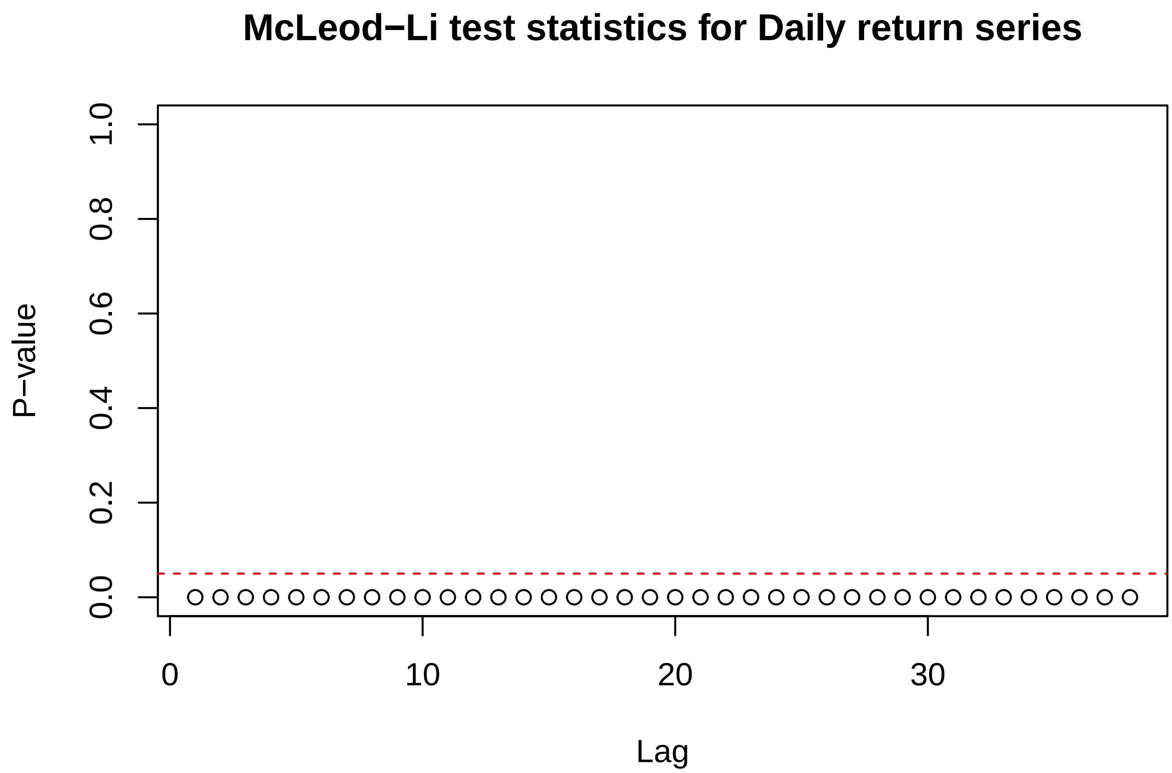


##nature as almost at all lags the p-values fall below the significance levels.

```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= DifflogEURJPYGarch,main="McLeod-Li test statistics for Daily return series")
```

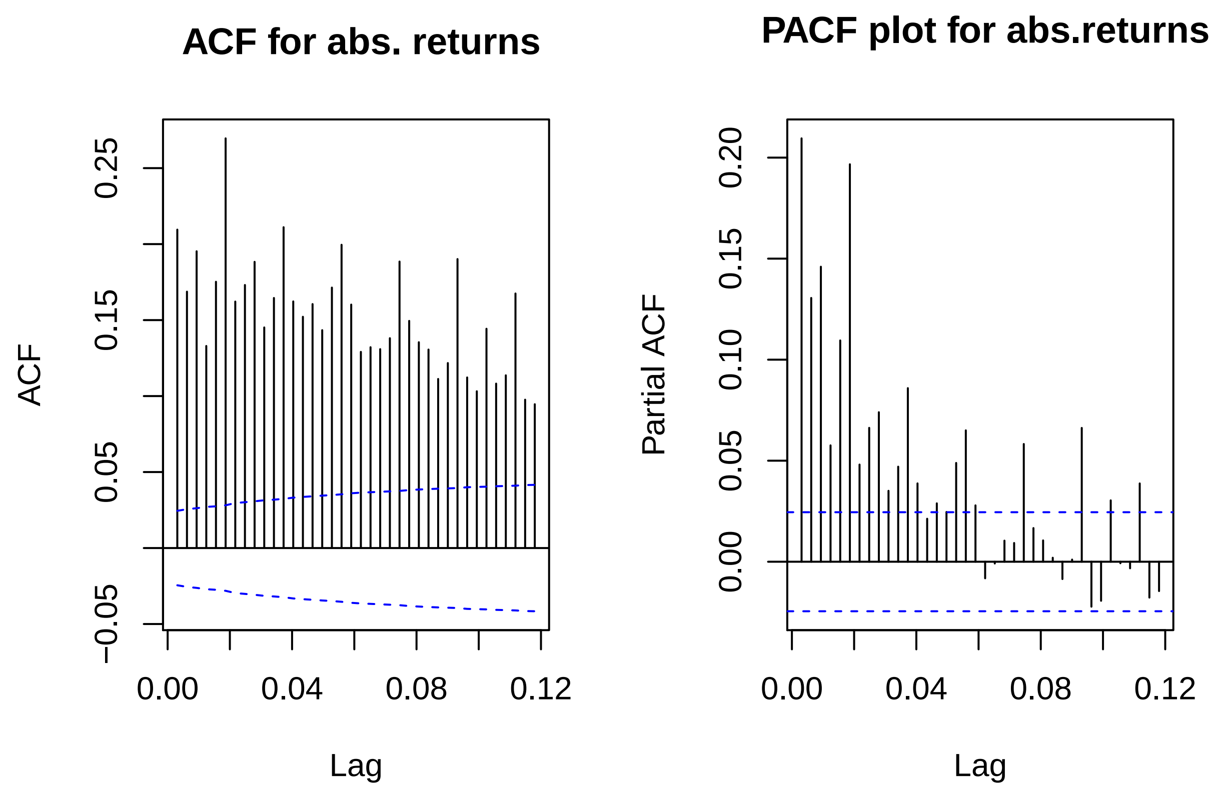


In order to get an order of GARCH , we further transform the return series into absolute values and squared return values.

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



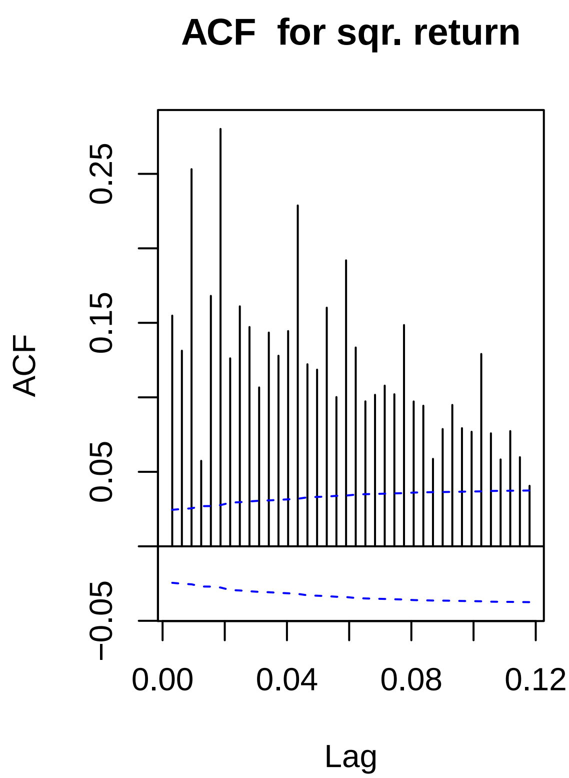
##From ACF and PACF we see many lags are significant. Hence, we plot EACF to get the candidate models

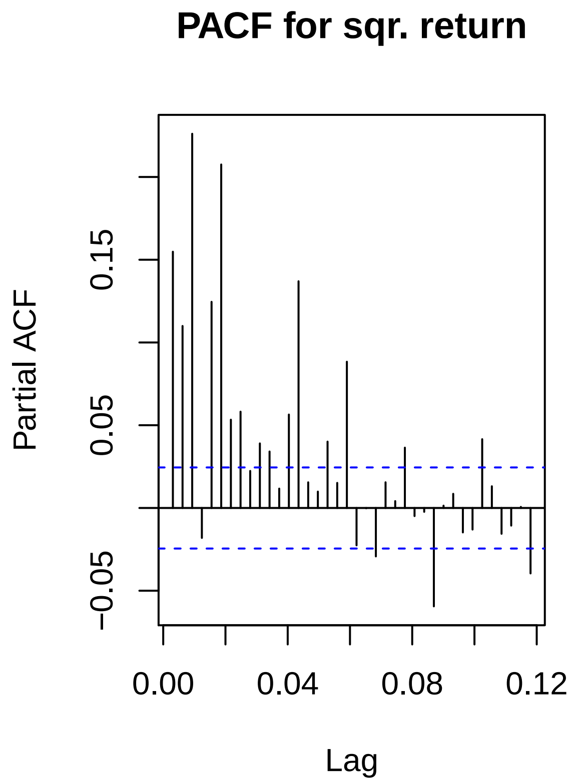
```r
eacf(abs)
```

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

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 USD Curruency Pair

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

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

```
EURJPYGARCHFit.21 = garch(DifflogEURJPYGarch,order=c(2,1),trace =FALSE)
summary(EURJPYGARCHFit.21)
```

```
##
## Call:
## garch(x = DifflogEURJPYGarch, order = c(2, 1), trace = FALSE)
##
## Model:
## GARCH(2,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.30658 -0.51855  0.03215  0.57293  5.73568
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 0.0052255   0.0007295    7.163 7.88e-13 ***
## a1 0.0822957   0.0058118   14.160  < 2e-16 ***
## b1 0.3861694   0.0842114    4.586 4.52e-06 ***
## b2 0.5213671   0.0801527    6.505 7.79e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2836.6, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.055, df = 1, p-value = 0.8146
```

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.

```
EURJPYGARCHFit.22 = garch(DifflogEURJPYGarch, order =c(2,2),trace =FALSE)
summary(EURJPYGARCHFit.22)
```

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

```
## GARCH(2,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.33520 -0.51490  0.03206  0.57338  5.73444
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 0.0060970   0.0008782    6.943 3.85e-12 ***
## a1 0.0762235   0.0084142    9.059  < 2e-16 ***
## a2 0.0188737   0.0131528    1.435   0.1513
## b1 0.1920932   0.1132182    1.697   0.0898 .
## b2 0.7009783   0.1059567    6.616 3.70e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2784.6, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.24048, df = 1, p-value = 0.6239
```

##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.

```
EURJPYGARCHFit.31 = garch(DifflogEURJPYGarch,order=c(3,1),trace =FALSE)
summary(EURJPYGARCHFit.31)
```

```
##
## Call:
## garch(x = DifflogEURJPYGarch, order = c(3, 1), trace = FALSE)
##
## Model:
## GARCH(3,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.30213 -0.51864  0.03214  0.57298  5.73610
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 5.206e-03   8.834e-04    5.893 3.79e-09 ***
## a1 8.197e-02   9.870e-03    8.305  < 2e-16 ***
## b1 3.945e-01   1.027e-01    3.839 0.000123 ***
## b2 5.134e-01   1.005e-01    5.107 3.27e-07 ***
## b3 1.894e-06   1.048e-01    0.000 0.999986
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2834.8, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.061834, df = 1, p-value = 0.8036
```

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

```
##
## Call:
## garch(x = DifflogEURJPYGarch, order = c(3, 2), trace = FALSE)
##
## Model:
## GARCH(3,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.33094 -0.51581  0.03203  0.57296  5.73045
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 7.120e-03   4.757e-03    1.497  0.13446
## a1 7.779e-02   9.690e-03    8.028 8.88e-16 ***
## a2 3.394e-02   6.791e-02    0.500  0.61726
## b1 4.590e-13   8.477e-01    0.000  1.00000
## b2 7.436e-01   2.699e-01    2.755  0.00588 **
## b3 1.308e-01   5.078e-01    0.258  0.79670
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2792.2, df = 2, p-value < 2.2e-16
##
##
```

```
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.17994, df = 1, p-value = 0.6714
```

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
EURJPYGARCHFit.33 = garch(DifflogEURJPYGarch,order=c(3,3),trace =FALSE)
summary(EURJPYGARCHFit.33)
```

```
##
## Call:
## garch(x = DifflogEURJPYGarch, order = c(3, 3), trace = FALSE)
##
## Model:
## GARCH(3,3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.10625 -0.50737  0.03222  0.56497  6.11852
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 1.095e-02   6.763e-03    1.619    0.105
## a1 1.026e-01   1.221e-02    8.407   <2e-16 ***
## a2 6.351e-02   6.387e-02    0.994    0.320
## a3 7.478e-16   5.420e-02    0.000    1.000
## b1 3.567e-01   6.666e-01    0.535    0.593
## b2 3.325e-01   6.085e-01    0.546    0.585
## b3 1.337e-01   4.679e-01    0.286    0.775
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2674.6, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.15051, df = 1, p-value = 0.6981
```

##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.

```
EURJPYGARCHFit.42 = garch(DifflogEURJPYGarch,order=c(4,2),trace =FALSE)
summary(EURJPYGARCHFit.42)
```

```
##
## Call:
## garch(x = DifflogEURJPYGarch, order = c(4, 2), trace = FALSE)
##
## Model:
## GARCH(4,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.35322 -0.51592  0.03244  0.57729  5.80432
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0  0.008259    0.001342    6.156 7.47e-10 ***
## a1  0.098688    0.009567   10.315  < 2e-16 ***
## a2  0.048451    0.019223    2.520   0.0117 *
## b1  0.114321    0.128137    0.892   0.3723
## b2  0.118365    0.090312    1.311   0.1900
## b3  0.030067    0.060986    0.493   0.6220
## b4  0.574598    0.075240    7.637 2.22e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2838.5, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.16757, df = 1, p-value = 0.6823
```

##

```
EURJPYGARCHFit.41 = garch(DifflogEURJPYGarch,order=c(4,1),trace =FALSE)
summary(EURJPYGARCHFit.41)
```

```
##
## Call:
## garch(x = DifflogEURJPYGarch, order = c(4, 1), trace = FALSE)
##
## Model:
## GARCH(4,1)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.34244 -0.51531  0.03226  0.57469  5.70129
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 6.198e-03   9.046e-04    6.852 7.29e-12 ***
## a1 1.102e-01   7.616e-03   14.472  < 2e-16 ***
## b1 4.518e-01   7.198e-02    6.276 3.46e-10 ***
## b2 3.411e-02   8.069e-02    0.423    0.672
## b3 3.836e-14   8.142e-02    0.000    1.000
## b4 3.924e-01   6.608e-02    5.938 2.89e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 2906.3, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.59531, df = 1, p-value = 0.4404
```

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
```

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

```
##                     df      AIC
## EURJPYGARCHFit.42  7 11772.03
## EURJPYGARCHFit.41  6 11774.43
## EURJPYGARCHFit.31  5 11789.68
## EURJPYGARCHFit.21  4 11790.30
## EURJPYGARCHFit.32  6 11790.82
## EURJPYGARCHFit.22  5 11791.68
## EURJPYGARCHFit.33  7 11834.51
```

Model Fitting:

```r
library(rugarch)
```

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

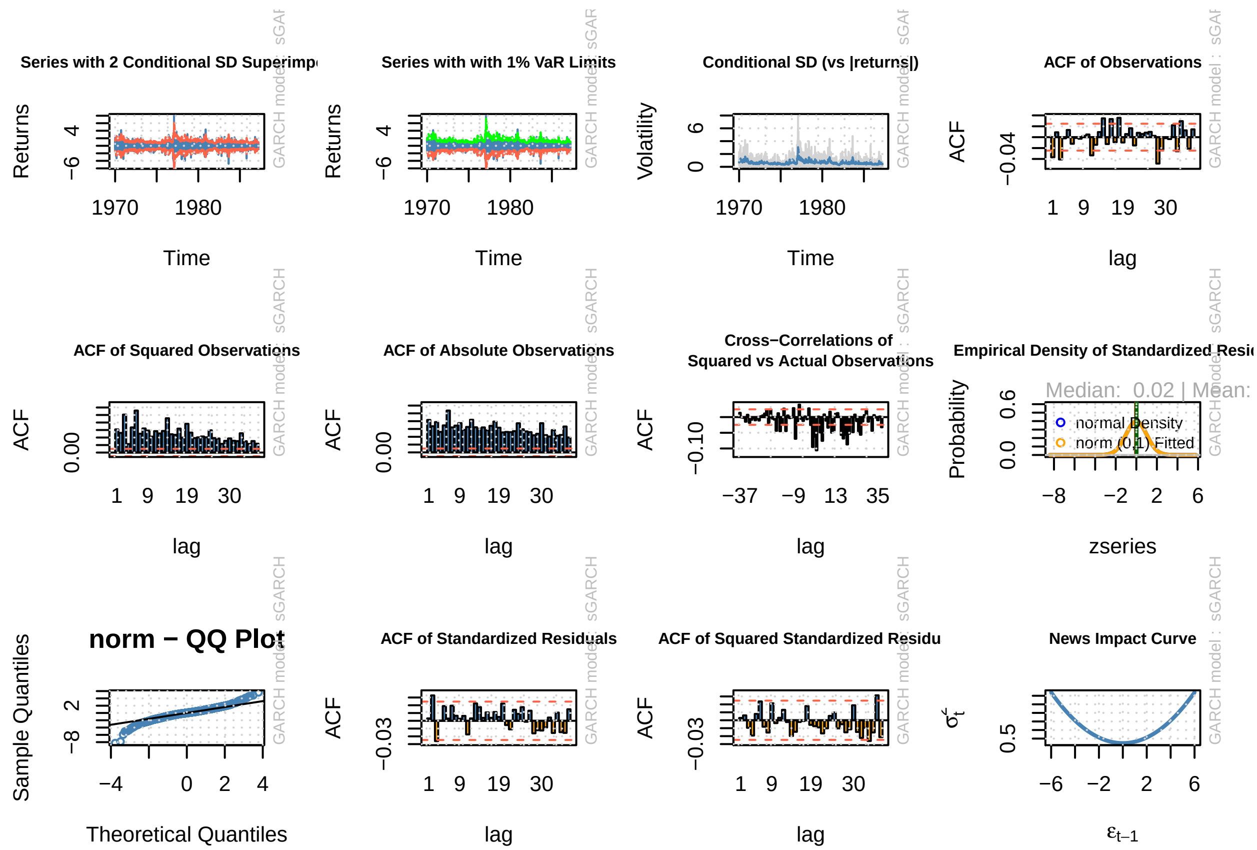
```
## Loading required package: parallel
```

```
##
## Attaching package: 'rugarch'
```

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

```r
EURJPYmodel4.2<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(4,2)),
                   mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),
                   distribution.model = "norm")

EURJPYgarchMODEL4.2<-ugarchfit(spec=EURJPYmodel4.2,data=DifflogEURJPYGarch, out.sample = 100)
plot(EURJPYgarchMODEL4.2,which="all")
```

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



##Model Diagnostics

```
EURJPYgarchMODEL4.2
```

```
##
## *---------------------------------*
## *          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      0.010003    0.005654    1.769135 0.076871
## ar1     0.919737    0.003508  262.210999 0.000000
## ma1    -0.933829    0.005696 -163.952647 0.000000
## omega   0.005448    0.001374    3.964455 0.000074
## alpha1  0.082383    0.013944    5.908098 0.000000
## alpha2  0.000008    0.013969    0.000563 0.999551
## alpha3  0.000000    0.015474    0.000004 0.999997
## alpha4  0.000000    0.015018    0.000003 0.999997
```

```
## beta1   0.434076    0.041986   10.338643 0.000000
## beta2   0.473167    0.039437   11.998081 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error     t value Pr(>|t|)
## mu      0.010003    0.005893    1.697552 0.089592
## ar1     0.919737    0.005766  159.506893 0.000000
## ma1    -0.933829    0.002217 -421.192129 0.000000
## omega   0.005448    0.002912    1.871121 0.061328
## alpha1  0.082383    0.019209    4.288848 0.000018
## alpha2  0.000008    0.018063    0.000435 0.999653
## alpha3  0.000000    0.026366    0.000002 0.999998
## alpha4  0.000000    0.024679    0.000002 0.999998
## beta1   0.434076    0.017449   24.876235 0.000000
## beta2   0.473167    0.015166   31.199711 0.000000
##
## LogLikelihood : -5851.913
##
## Information Criteria
## ------------------------------------
##
## Akaike       1.8615
## Bayes        1.8722
## Shibata      1.8615
## Hannan-Quinn 1.8652
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic   p-value
## Lag[1]                     0.0603 8.060e-01
## Lag[2*(p+q)+(p+q)-1][5]    8.4943 8.304e-10
## Lag[4*(p+q)+(p+q)-1][9]   11.5541 1.875e-03
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                     0.05589  0.8131
## Lag[2*(p+q)+(p+q)-1][17]   9.50908  0.3927
## Lag[4*(p+q)+(p+q)-1][29]  14.87403  0.4711
## d.o.f=6
##
## Weighted ARCH LM Tests
## ------------------------------------
##              Statistic Shape Scale P-Value
## ARCH Lag[7]     0.4205 0.500 2.000  0.5167
## ARCH Lag[9]     4.0047 1.485 1.796  0.2122
## ARCH Lag[11]    4.2505 2.440 1.677  0.3918
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  2.0208
## Individual Statistics:
```

```
## mu     0.39093
## ar1    0.12940
## ma1    0.14145
## omega  0.05824
## alpha1 0.14050
## alpha2 0.10848
## alpha3 0.11776
## alpha4 0.09096
## beta1  0.09970
## beta2  0.10073
##
## 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           0.1635 0.87016
## Negative Sign Bias  1.7136 0.08664   *
## Positive Sign Bias  1.6469 0.09963   *
## Joint Effect        9.7164 0.02114  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     287.9    6.065e-50
## 2    30     299.6    9.803e-47
## 3    40     317.9    1.993e-45
## 4    50     347.1    1.676e-46
##
##
## Elapsed time : 0.95205
```

Forecasting

```
forcgarchEURJPY= ugarchforecast(EURJPYgarchMODEL4.2, data = DiffEURJPYLogTran, n.ahead = 100, n.roll =1
print(forcgarchEURJPY)
```

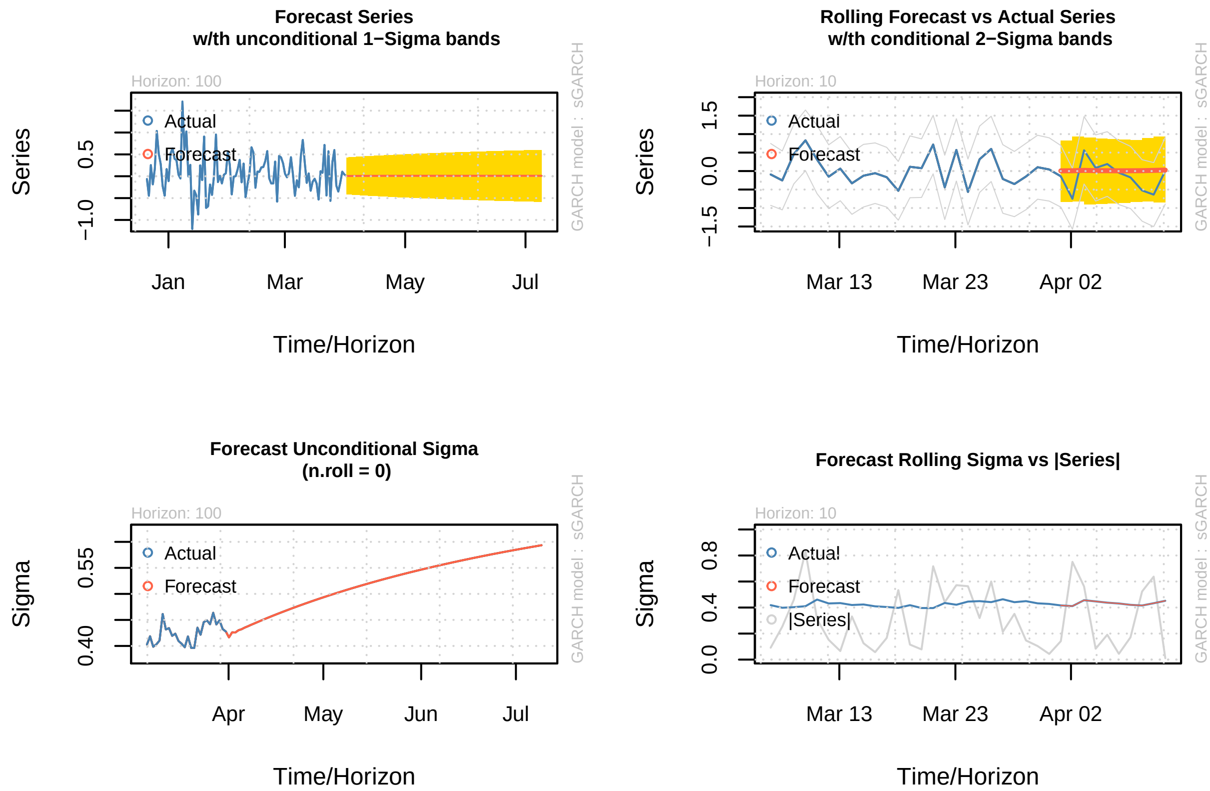
```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 100
## Roll Steps: 10
## Out of Sample: 100
##
## 0-roll forecast [T0=1987-03-31 03:00:00]:
##         Series  Sigma
## T+1   0.005658 0.4165
```

```
## T+2   0.006007 0.4261
## T+3   0.006328 0.4258
## T+4   0.006623 0.4301
## T+5   0.006894 0.4322
## T+6   0.007143 0.4353
## T+7   0.007373 0.4378
## T+8   0.007584 0.4406
## T+9   0.007778 0.4432
## T+10  0.007957 0.4458
## T+11  0.008121 0.4483
## T+12  0.008272 0.4509
## T+13  0.008411 0.4534
## T+14  0.008539 0.4558
## T+15  0.008656 0.4583
## T+16  0.008764 0.4607
## T+17  0.008864 0.4631
## T+18  0.008955 0.4654
## T+19  0.009039 0.4678
## T+20  0.009117 0.4700
## T+21  0.009188 0.4723
## T+22  0.009253 0.4746
## T+23  0.009313 0.4768
## T+24  0.009369 0.4790
## T+25  0.009420 0.4811
## T+26  0.009466 0.4833
## T+27  0.009510 0.4854
## T+28  0.009549 0.4875
## T+29  0.009586 0.4896
## T+30  0.009619 0.4916
## T+31  0.009650 0.4936
## T+32  0.009678 0.4956
## T+33  0.009704 0.4976
## T+34  0.009728 0.4996
## T+35  0.009750 0.5015
## T+36  0.009771 0.5034
## T+37  0.009789 0.5053
## T+38  0.009806 0.5072
## T+39  0.009822 0.5090
## T+40  0.009837 0.5109
## T+41  0.009850 0.5127
## T+42  0.009862 0.5145
## T+43  0.009874 0.5163
## T+44  0.009884 0.5180
## T+45  0.009894 0.5198
## T+46  0.009902 0.5215
## T+47  0.009910 0.5232
## T+48  0.009918 0.5249
## T+49  0.009925 0.5265
## T+50  0.009931 0.5282
## T+51  0.009937 0.5298
## T+52  0.009942 0.5314
## T+53  0.009947 0.5331
## T+54  0.009951 0.5346
## T+55  0.009956 0.5362
```

```
## T+56  0.009959 0.5378
## T+57  0.009963 0.5393
## T+58  0.009966 0.5408
## T+59  0.009969 0.5423
## T+60  0.009972 0.5438
## T+61  0.009974 0.5453
## T+62  0.009977 0.5468
## T+63  0.009979 0.5482
## T+64  0.009981 0.5497
## T+65  0.009982 0.5511
## T+66  0.009984 0.5525
## T+67  0.009986 0.5539
## T+68  0.009987 0.5553
## T+69  0.009988 0.5567
## T+70  0.009989 0.5580
## T+71  0.009991 0.5594
## T+72  0.009992 0.5607
## T+73  0.009992 0.5620
## T+74  0.009993 0.5633
## T+75  0.009994 0.5646
## T+76  0.009995 0.5659
## T+77  0.009995 0.5672
## T+78  0.009996 0.5685
## T+79  0.009997 0.5697
## T+80  0.009997 0.5709
## T+81  0.009998 0.5722
## T+82  0.009998 0.5734
## T+83  0.009998 0.5746
## T+84  0.009999 0.5758
## T+85  0.009999 0.5770
## T+86  0.009999 0.5781
## T+87  0.010000 0.5793
## T+88  0.010000 0.5804
## T+89  0.010000 0.5816
## T+90  0.010000 0.5827
## T+91  0.010001 0.5838
## T+92  0.010001 0.5849
## T+93  0.010001 0.5860
## T+94  0.010001 0.5871
## T+95  0.010001 0.5882
## T+96  0.010001 0.5893
## T+97  0.010002 0.5903
## T+98  0.010002 0.5914
## T+99  0.010002 0.5924
## T+100 0.010002 0.5935
```

plotting

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



Forecasting the rate

```
p.t_1 = 126.121
  R_t <- c(  0.005658, 0.006007, 0.006328, 0.006623, 0.006894, 0.007143, 0.007373, 0.007584, 0.007778, 0
0.008121, 0.008272, 0.008411, 0.008539, 0.008656, 0.008764, 0.008864, 0.008955, 0.009039, 0.009117, 0.00
0.009956, 0.009959, 0.009963, 0.009966, 0.009969, 0.009972, 0.009974, 0.009977, 0.009979, 0.009981, 0.00
0.009984, 0.009986, 0.009987, 0.009988, 0.009989, 0.009991, 0.009992, 0.009992, 0.009993, 0.009994, 0.00
0.010002, 0.010002

)
  p_t= 0
  for (i in 1:100){
    p_t =  p.t_1  *((2.71828)^(R_t[i]/100))
   print(p_t)
    p.t_1=p_t
  }
```

```
## [1] 126.1281
## [1] 126.1357
## [1] 126.1437
## [1] 126.152
## [1] 126.1607
## [1] 126.1698
## [1] 126.1791
```

```
## [1] 126.1886
## [1] 126.1984
## [1] 126.2085
## [1] 126.2187
## [1] 126.2292
## [1] 126.2398
## [1] 126.2506
## [1] 126.2615
## [1] 126.2726
## [1] 126.2838
## [1] 126.2951
## [1] 126.3065
## [1] 126.318
## [1] 126.3296
## [1] 126.3413
## [1] 126.3531
## [1] 126.3649
## [1] 126.3768
## [1] 126.3888
## [1] 126.4008
## [1] 126.4129
## [1] 126.425
## [1] 126.4371
## [1] 126.4493
## [1] 126.4616
## [1] 126.4739
## [1] 126.4862
## [1] 126.4985
## [1] 126.5109
## [1] 126.5232
## [1] 126.5356
## [1] 126.5481
## [1] 126.5605
## [1] 126.573
## [1] 126.5855
## [1] 126.598
## [1] 126.6105
## [1] 126.623
## [1] 126.6356
## [1] 126.6481
## [1] 126.6607
## [1] 126.6732
## [1] 126.6858
## [1] 126.6984
## [1] 126.711
## [1] 126.7236
## [1] 126.7362
## [1] 126.7488
## [1] 126.7615
## [1] 126.7741
## [1] 126.7867
## [1] 126.7994
## [1] 126.812
## [1] 126.8247
```

```
## [1] 126.8373
## [1] 126.85
## [1] 126.8626
## [1] 126.8753
## [1] 126.888
## [1] 126.9006
## [1] 126.9133
## [1] 126.926
## [1] 126.9387
## [1] 126.9514
## [1] 126.964
## [1] 126.9767
## [1] 126.9894
## [1] 127.0021
## [1] 127.0148
## [1] 127.0275
## [1] 127.0402
## [1] 127.0529
## [1] 127.0656
## [1] 127.0783
## [1] 127.091
## [1] 127.1037
## [1] 127.1164
## [1] 127.1291
## [1] 127.1418
## [1] 127.1546
## [1] 127.1673
## [1] 127.18
## [1] 127.1927
## [1] 127.2054
## [1] 127.2182
## [1] 127.2309
## [1] 127.2436
## [1] 127.2563
## [1] 127.2691
## [1] 127.2818
## [1] 127.2945
## [1] 127.3073
## [1] 127.32
```