GARCH Model EUR And GBP

Jane

29/04/2021

Forcasting Exchange Rate Using GARCH Model for EUR And GBP

Reading EUR and GBP 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
```

```
EURGBPGARCH<-  read.csv ("EURGBP_Candlestick_1_D_BID_01.01.2000-31.12.2020.csv")%>%
  select('GMT.TIME', CLOSE)%>%
  rename(Date = ('GMT.TIME'), RateEURGBP = ("CLOSE"))


head(EURGBPGARCH)
```

```
##         Date RateEURGBP
## 1 2000-01-03     0.6261
## 2 2000-01-04     0.6293
## 3 2000-01-05     0.6281
## 4 2000-01-06     0.6263
## 5 2000-01-07     0.6277
## 6 2000-01-10     0.6264
```

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

```
##         Date RateEURGBP
## 1 2000-01-03     0.6261
## 2 2000-01-04     0.6293
## 3 2000-01-05     0.6281
## 4 2000-01-06     0.6263
## 5 2000-01-07     0.6277
## 6 2000-01-10     0.6264
```

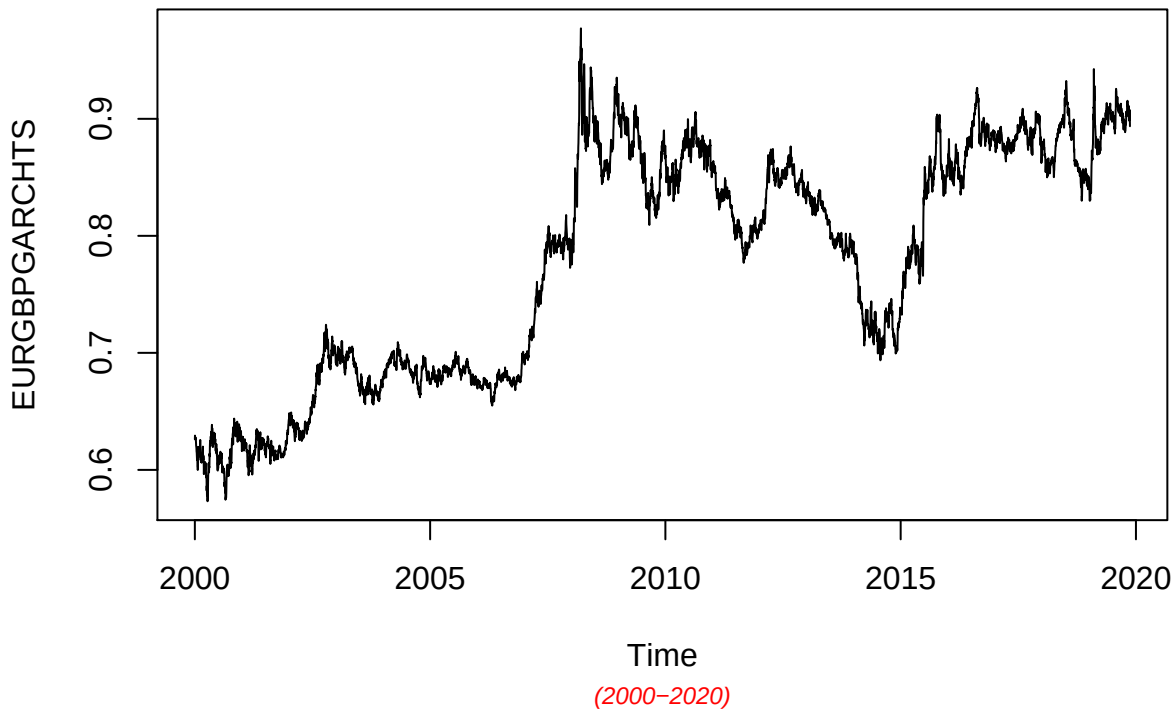
##Checking for obvious errors or missingg value

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

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

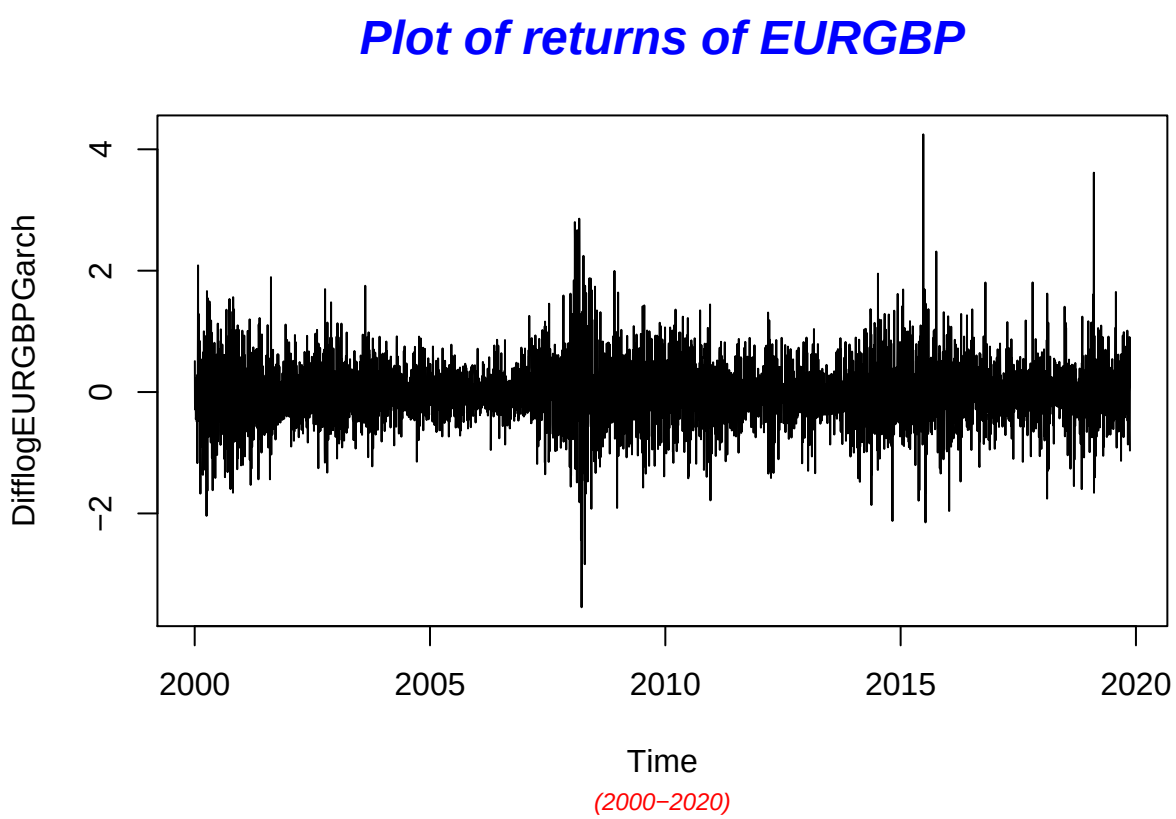
##Converting the data set into time series object

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

```
DifflogEURGBPGarch= diff(log(EURGBPGARCHTS))*100
plot(DifflogEURGBPGarch)
title("Plot of returns of EURGBP", 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.

```
library(TSA)
```

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

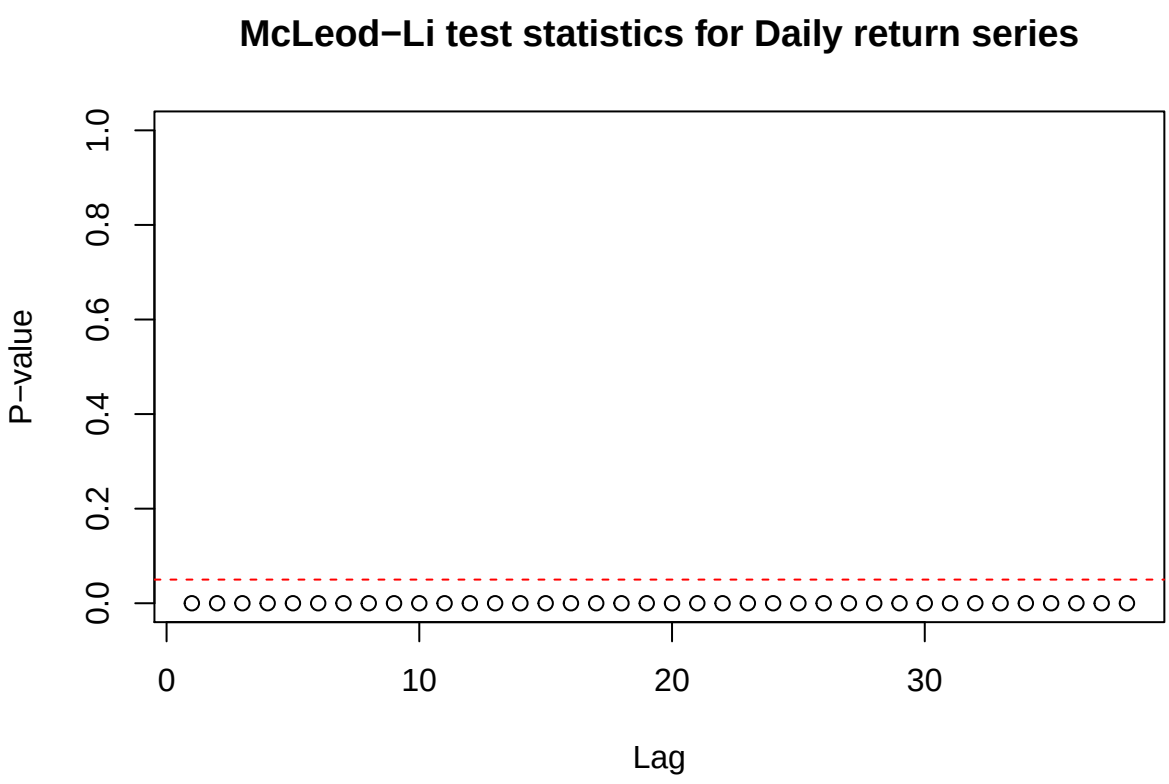
```
##
## Attaching package: 'TSA'
```

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

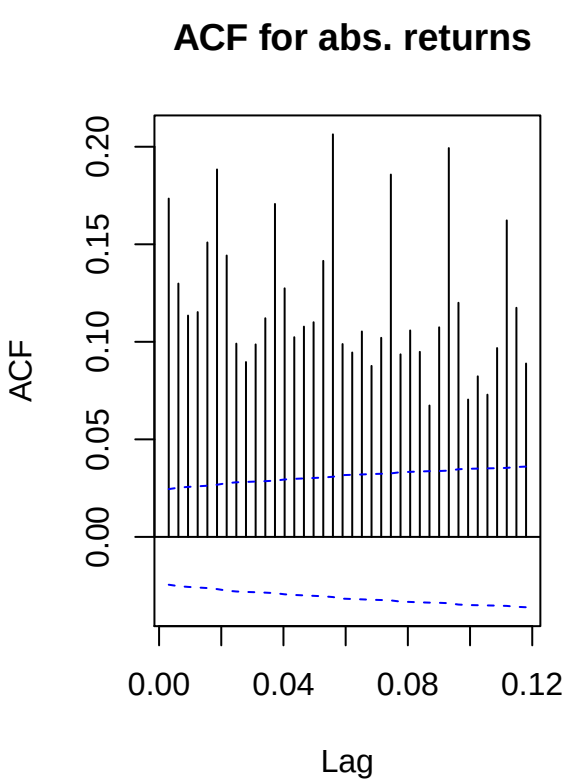


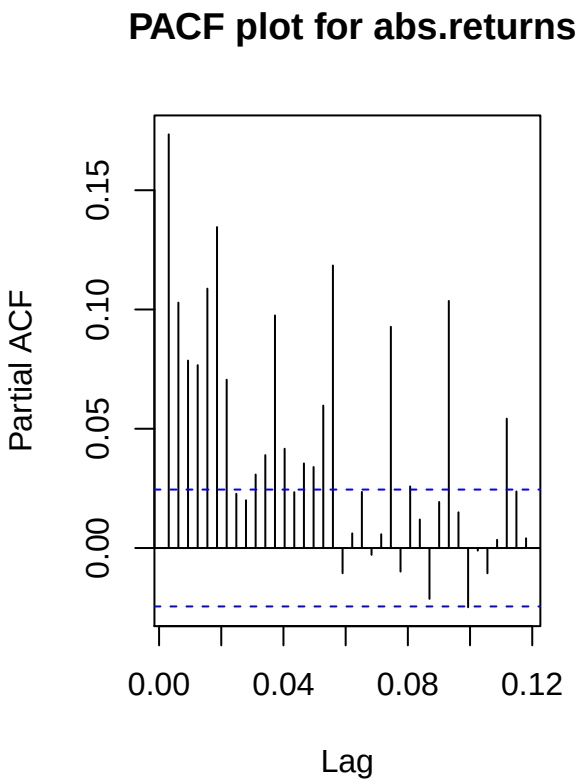
In order to get an order of GARCH , we further transform the return series into absolute values and squared return values.

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





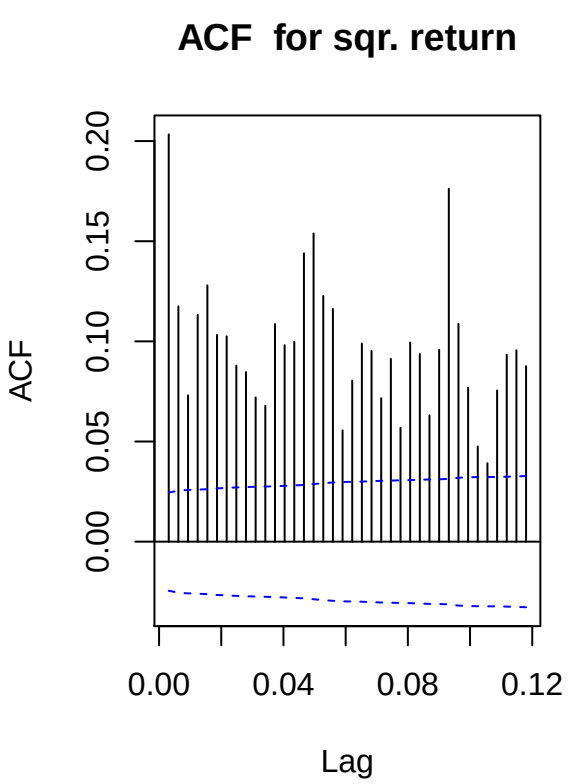
##From ACF and PACF we see many lags are significant. Hence, we plot EACF to get the candidate models

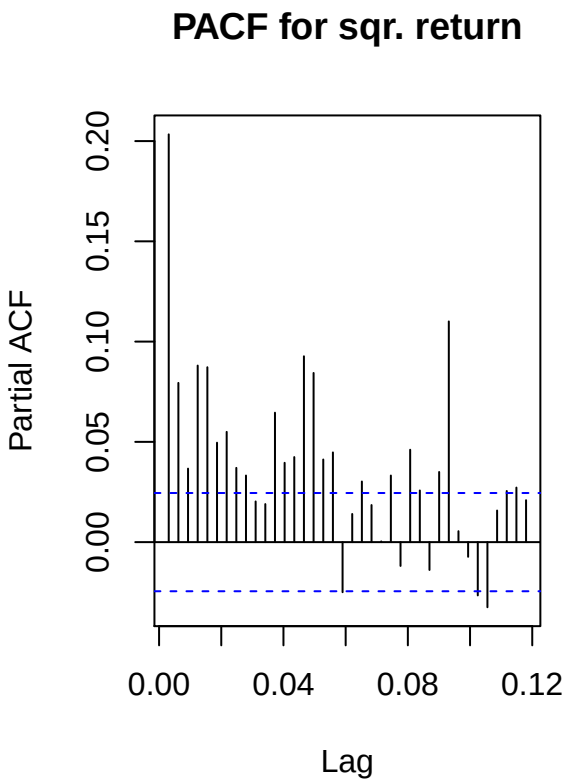
```
eacf(abs)
```

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

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






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

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

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

```
EURGBPGARCHFit.21 = garch(DifflogEURGBPGarch,order=c(2,1),trace =FALSE)
summary(EURGBPGARCHFit.21)
```

```
##
## Call:
## garch(x = DifflogEURGBPGarch, order = c(2, 1), trace = FALSE)
##
## Model:
## GARCH(2,1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -4.421254 -0.553172  0.007501  0.566513  6.018373
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 0.0019796   0.0003765    5.258 1.46e-07 ***
## a1 0.0608949   0.0062239    9.784  < 2e-16 ***
## b1 0.5789107   0.1267521    4.567 4.94e-06 ***
## b2 0.3523961   0.1211942    2.908  0.00364 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 759.06, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 3.2331, df = 1, p-value = 0.07217
```

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.

```
EURGBPGARCHFit.22 = garch(DifflogEURGBPGarch, order =c(2,2),trace =FALSE)
summary(EURGBPGARCHFit.22)
```

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

```
## GARCH(2,2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -4.425124 -0.553644  0.007464  0.566711  5.943074
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 2.086e-03   6.056e-04    3.444 0.000573 ***
## a1 6.562e-02   7.899e-03    8.307  < 2e-16 ***
## a2 2.912e-15   2.071e-02    0.000 1.000000
## b1 4.558e-01   3.164e-01    1.441 0.149675
## b2 4.705e-01   2.990e-01    1.574 0.115516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 767.63, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 2.2484, df = 1, p-value = 0.1338
```

##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.

```
EURGBPGARCHFit.31 = garch(DifflogEURGBPGarch,order=c(3,1),trace =FALSE)
summary(EURGBPGARCHFit.31)
```

```
##
## Call:
## garch(x = DifflogEURGBPGarch, order = c(3, 1), trace = FALSE)
##
## Model:
## GARCH(3,1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -4.418602 -0.551795  0.007475  0.563893  5.676349
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 2.302e-03   4.466e-04    5.154 2.55e-07 ***
## a1 8.375e-02   6.903e-03   12.133  < 2e-16 ***
## b1 5.316e-01   1.045e-01    5.087 3.65e-07 ***
## b2 1.005e-15   1.454e-01    0.000 1.000000
## b3 3.773e-01   1.018e-01    3.706 0.000211 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 702.52, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.22046, df = 1, p-value = 0.6387
```

##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)

```
EURGBPGARCHFit.32 = garch(DifflogEURGBPGarch,order=c(3,2),trace =FALSE)
summary(EURGBPGARCHFit.32)
```

```
##
## Call:
## garch(x = DifflogEURGBPGarch, order = c(3, 2), trace = FALSE)
##
## Model:
## GARCH(3,2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -4.381323 -0.551725  0.007592  0.565241  5.867779
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 2.905e-03   6.395e-04    4.542 5.56e-06 ***
## a1 7.159e-02   7.530e-03    9.507  < 2e-16 ***
## a2 1.882e-02   1.650e-02    1.140   0.2541
## b1 3.859e-01   1.931e-01    1.998   0.0457 *
## b2 2.195e-06   1.553e-01    0.000   1.0000
## b3 5.122e-01   1.099e-01    4.660 3.16e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 697.29, df = 2, p-value < 2.2e-16
##
##
```

```
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 1.2862, df = 1, p-value = 0.2568
```

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)

```
EURGBPGARCHFit.33 = garch(DifflogEURGBPGarch,order=c(3,3),trace =FALSE)
summary(EURGBPGARCHFit.33)
```

```
##
## Call:
## garch(x = DifflogEURGBPGarch, order = c(3, 3), trace = FALSE)
##
## Model:
## GARCH(3,3)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -4.418160 -0.553909  0.007512  0.566955  5.837280
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 2.450e-03   1.085e-03    2.257    0.024 *
## a1 7.110e-02   8.529e-03    8.337   <2e-16 ***
## a2 1.997e-02   5.385e-02    0.371    0.711
## a3 1.223e-14   2.888e-02    0.000    1.000
## b1 2.787e-01   7.455e-01    0.374    0.709
## b2 3.121e-01   7.491e-01    0.417    0.677
## b3 3.101e-01   2.826e-01    1.097    0.273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 730.26, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 1.3356, df = 1, p-value = 0.2478
```

##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.

```
EURGBPGARCHFit.42 = garch(DifflogEURGBPGarch,order=c(4,2),trace =FALSE)
summary(EURGBPGARCHFit.42)
```

```
##
## Call:
## garch(x = DifflogEURGBPGarch, order = c(4, 2), trace = FALSE)
##
## Model:
## GARCH(4,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.38463 -0.55559  0.00748  0.57100  5.99505
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 3.167e-03   1.865e-03    1.698   0.0895 .
## a1 6.933e-02   7.677e-03    9.031   <2e-16 ***
## a2 3.466e-02   6.236e-02    0.556   0.5783
## b1 3.951e-01   9.032e-01    0.437   0.6618
## b2 4.719e-15   8.972e-01    0.000   1.0000
## b3 8.563e-02   5.597e-01    0.153   0.8784
## b4 4.030e-01   4.961e-01    0.812   0.4166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 742.33, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 1.5132, df = 1, p-value = 0.2186
```

##

```
EURGBPGARCHFit.41 = garch(DifflogEURGBPGarch,order=c(4,1),trace =FALSE)
summary(EURGBPGARCHFit.41)
```

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

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -4.376240 -0.555976  0.007487  0.569502  5.959987
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 2.407e-03   4.625e-04    5.204 1.95e-07 ***
## a1 7.913e-02   7.302e-03   10.837  < 2e-16 ***
## b1 5.945e-01   1.130e-01    5.262 1.43e-07 ***
## b2 1.705e-14   1.807e-01    0.000 1.000000
## b3 2.943e-02   1.589e-01    0.185 0.853098
## b4 2.873e-01   7.975e-02    3.602 0.000315 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 745.05, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.49716, df = 1, p-value = 0.4807
```

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
```

```
GARCHModelSelectionEURGBP = AIC(EURGBPGARCHFit.21,EURGBPGARCHFit.22 ,EURGBPGARCHFit.31,EURGBPGARCHFit.3
sortScore(GARCHModelSelectionEURGBP, score ="aic")
```

```
##                    df      AIC
## EURGBPGARCHFit.41   6 7729.134
## EURGBPGARCHFit.42   7 7733.284
## EURGBPGARCHFit.31   5 7733.784
## EURGBPGARCHFit.32   6 7735.384
## EURGBPGARCHFit.21   4 7740.737
## EURGBPGARCHFit.33   7 7741.609
## EURGBPGARCHFit.22   5 7743.652
```

Model Fitting:

```
library(rugarch)
```

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

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

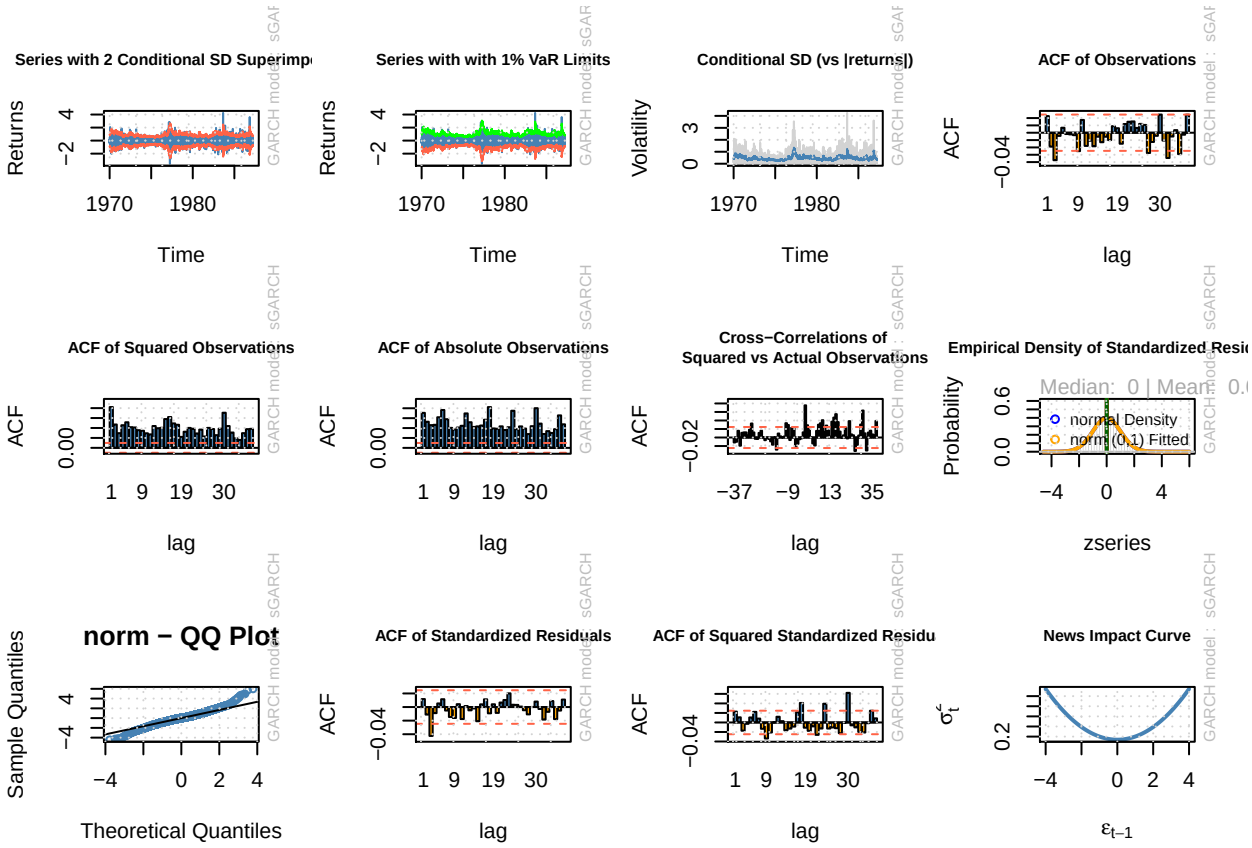
```
##
## Attaching package: 'rugarch'
```

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

```
EURGBPmodel2.2<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2,2)),
                  mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),
                  distribution.model = "norm")

EURGBPgarchMODEL2.2<-ugarchfit(spec=EURGBPmodel2.2,data=DifflogEURGBPGarch, out.sample = 100)
plot(EURGBPgarchMODEL2.2,which="all")
```

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



##Model Diagnostics

```
EURGBPgarchMODEL2.2
```

```
##
## *----------------------------------*
## *          GARCH Model Fit         *
## *----------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(2,2)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.001502    0.005093  0.294982 0.768007
## ar1    -0.167686    0.403172 -0.415917 0.677471
## ma1     0.178676    0.402171  0.444279 0.656841
## omega   0.002008    0.000524  3.832350 0.000127
## alpha1  0.062747    0.011039  5.684121 0.000000
## alpha2  0.000000    0.011553  0.000011 0.999991
## beta1   0.547023    0.008282 66.049033 0.000000
## beta2   0.382185    0.007774 49.159071 0.000000
```

```
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.001502    0.004796  0.313278 0.754069
## ar1    -0.167686    0.127878 -1.311300 0.189756
## ma1     0.178676    0.128109  1.394721 0.163100
## omega   0.002008    0.000835  2.404554 0.016192
## alpha1  0.062747    0.015684  4.000750 0.000063
## alpha2  0.000000    0.017596  0.000007 0.999994
## beta1   0.547023    0.008584 63.726192 0.000000
## beta2   0.382185    0.007903 48.358066 0.000000
##
## LogLikelihood : -3795.86
##
## Information Criteria
## ------------------------------------
##
## Akaike       1.2078
## Bayes        1.2163
## Shibata      1.2078
## Hannan-Quinn 1.2107
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic   p-value
## Lag[1]                     0.8725 3.503e-01
## Lag[2*(p+q)+(p+q)-1][5]    8.5870 5.205e-10
## Lag[4*(p+q)+(p+q)-1][9]   11.8016 1.439e-03
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                       3.278 0.07022
## Lag[2*(p+q)+(p+q)-1][11]    10.769 0.06715
## Lag[4*(p+q)+(p+q)-1][19]    18.165 0.02816
## d.o.f=4
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[5]    0.7013 0.500 2.000 0.40236
## ARCH Lag[7]    3.6910 1.473 1.746 0.23079
## ARCH Lag[9]    8.5447 2.402 1.619 0.05428
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  2.641
## Individual Statistics:
## mu     0.09213
## ar1    0.13647
## ma1    0.13877
## omega  0.38242
```

```
## alpha1 0.12947
## alpha2 0.16895
## beta1  0.13937
## beta2  0.13623
##
## 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           1.0999 0.2714
## Negative Sign Bias  0.7683 0.4423
## Positive Sign Bias  1.2242 0.2209
## Joint Effect        5.5561 0.1353
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     240.5    2.613e-40
## 2    30     254.6    6.584e-38
## 3    40     291.7    2.009e-40
## 4    50     285.2    4.608e-35
##
##
## Elapsed time : 0.7869942
```

Forecasting

```
forcgarchEURGBP= ugarchforecast(EURGBPgarchMODEL2.2, data = DifflogEURGBPGarch, n.ahead = 100, n.roll =
print(forcgarchEURGBP)
```

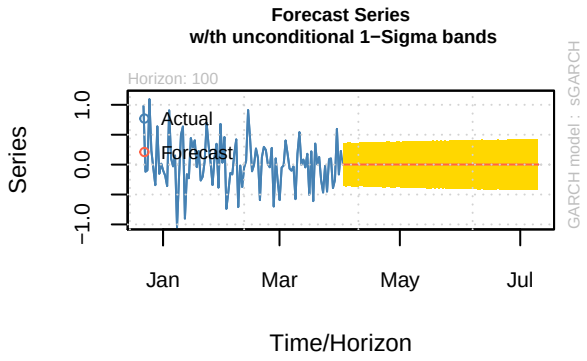
```
##
## *----------------------------------*
## *       GARCH Model Forecast       *
## *----------------------------------*
## Model: sGARCH
## Horizon: 100
## Roll Steps: 10
## Out of Sample: 100
##
## 0-roll forecast [T0=1987-04-01 03:00:00]:
##          Series  Sigma
## T+1   0.0045458 0.3583
## T+2   0.0009921 0.3590
## T+3   0.0015880 0.3601
## T+4   0.0014880 0.3610
## T+5   0.0015048 0.3620
## T+6   0.0015020 0.3629
## T+7   0.0015025 0.3639
```

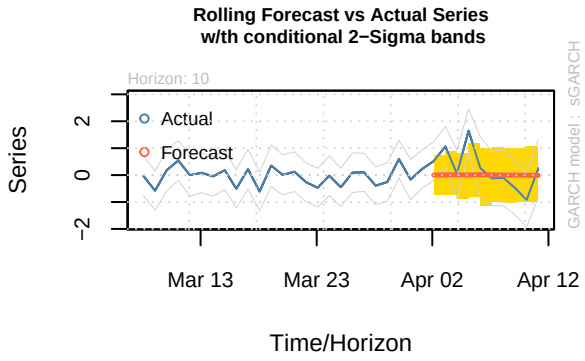
```
## T+8   0.0015024 0.3648
## T+9   0.0015024 0.3657
## T+10  0.0015024 0.3666
## T+11  0.0015024 0.3676
## T+12  0.0015024 0.3685
## T+13  0.0015024 0.3694
## T+14  0.0015024 0.3703
## T+15  0.0015024 0.3711
## T+16  0.0015024 0.3720
## T+17  0.0015024 0.3729
## T+18  0.0015024 0.3737
## T+19  0.0015024 0.3746
## T+20  0.0015024 0.3754
## T+21  0.0015024 0.3763
## T+22  0.0015024 0.3771
## T+23  0.0015024 0.3779
## T+24  0.0015024 0.3788
## T+25  0.0015024 0.3796
## T+26  0.0015024 0.3804
## T+27  0.0015024 0.3812
## T+28  0.0015024 0.3820
## T+29  0.0015024 0.3828
## T+30  0.0015024 0.3835
## T+31  0.0015024 0.3843
## T+32  0.0015024 0.3851
## T+33  0.0015024 0.3859
## T+34  0.0015024 0.3866
## T+35  0.0015024 0.3874
## T+36  0.0015024 0.3881
## T+37  0.0015024 0.3889
## T+38  0.0015024 0.3896
## T+39  0.0015024 0.3903
## T+40  0.0015024 0.3910
## T+41  0.0015024 0.3918
## T+42  0.0015024 0.3925
## T+43  0.0015024 0.3932
## T+44  0.0015024 0.3939
## T+45  0.0015024 0.3946
## T+46  0.0015024 0.3953
## T+47  0.0015024 0.3960
## T+48  0.0015024 0.3966
## T+49  0.0015024 0.3973
## T+50  0.0015024 0.3980
## T+51  0.0015024 0.3986
## T+52  0.0015024 0.3993
## T+53  0.0015024 0.4000
## T+54  0.0015024 0.4006
## T+55  0.0015024 0.4013
## T+56  0.0015024 0.4019
## T+57  0.0015024 0.4025
## T+58  0.0015024 0.4032
## T+59  0.0015024 0.4038
## T+60  0.0015024 0.4044
## T+61  0.0015024 0.4050
```

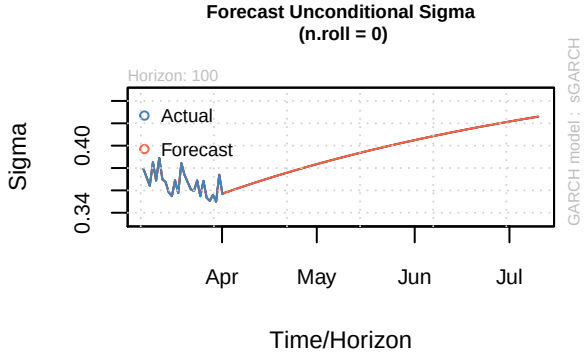
```
## T+62  0.0015024 0.4057
## T+63  0.0015024 0.4063
## T+64  0.0015024 0.4069
## T+65  0.0015024 0.4075
## T+66  0.0015024 0.4081
## T+67  0.0015024 0.4087
## T+68  0.0015024 0.4092
## T+69  0.0015024 0.4098
## T+70  0.0015024 0.4104
## T+71  0.0015024 0.4110
## T+72  0.0015024 0.4115
## T+73  0.0015024 0.4121
## T+74  0.0015024 0.4127
## T+75  0.0015024 0.4132
## T+76  0.0015024 0.4138
## T+77  0.0015024 0.4143
## T+78  0.0015024 0.4149
## T+79  0.0015024 0.4154
## T+80  0.0015024 0.4160
## T+81  0.0015024 0.4165
## T+82  0.0015024 0.4170
## T+83  0.0015024 0.4176
## T+84  0.0015024 0.4181
## T+85  0.0015024 0.4186
## T+86  0.0015024 0.4191
## T+87  0.0015024 0.4196
## T+88  0.0015024 0.4201
## T+89  0.0015024 0.4206
## T+90  0.0015024 0.4211
## T+91  0.0015024 0.4216
## T+92  0.0015024 0.4221
## T+93  0.0015024 0.4226
## T+94  0.0015024 0.4231
## T+95  0.0015024 0.4236
## T+96  0.0015024 0.4241
## T+97  0.0015024 0.4246
## T+98  0.0015024 0.4250
## T+99  0.0015024 0.4255
## T+100 0.0015024 0.4260
```

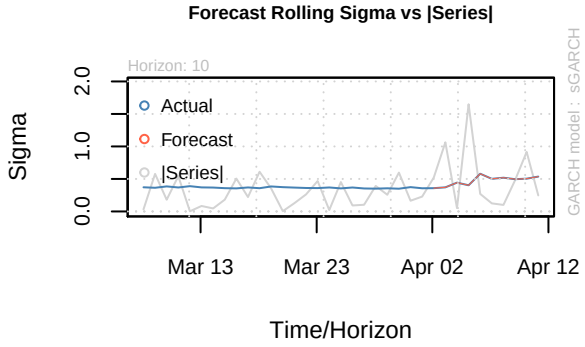
plotting

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









Forecasting the rate

```
p.t_1 = 0.89365
  R_t <- c( 0.0045458, 0.0009921, 0.0015880, 0.0014880, 0.0015048, 0.0015020, 0.0015025, 0.0015024, 0.0
0.0015024, 0.0015024, 0.0015024, 0.0015024, 0.0015024, 0.0015024, 0.0015024, 0.0015024, 0.0015024, 0.00
)
  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] 0.8936906
## [1] 0.8936995
## [1] 0.8937137
## [1] 0.893727
## [1] 0.8937404
## [1] 0.8937539
```

```
## [1] 0.8937673
## [1] 0.8937807
## [1] 0.8937941
## [1] 0.8938076
## [1] 0.893821
## [1] 0.8938344
## [1] 0.8938479
## [1] 0.8938613
## [1] 0.8938747
## [1] 0.8938881
## [1] 0.8939016
## [1] 0.893915
## [1] 0.8939284
## [1] 0.8939419
## [1] 0.8939553
## [1] 0.8939687
## [1] 0.8939822
## [1] 0.8939956
## [1] 0.894009
## [1] 0.8940225
## [1] 0.8940359
## [1] 0.8940493
## [1] 0.8940627
## [1] 0.8940762
## [1] 0.8940896
## [1] 0.894103
## [1] 0.8941165
## [1] 0.8941299
## [1] 0.8941433
## [1] 0.8941568
## [1] 0.8941702
## [1] 0.8941836
## [1] 0.8941971
## [1] 0.8942105
## [1] 0.894224
## [1] 0.8942374
## [1] 0.8942508
## [1] 0.8942643
## [1] 0.8942777
## [1] 0.8942911
## [1] 0.8943046
## [1] 0.894318
## [1] 0.8943314
## [1] 0.8943449
## [1] 0.8943583
## [1] 0.8943717
## [1] 0.8943852
## [1] 0.8943986
## [1] 0.8944121
## [1] 0.8944255
## [1] 0.8944389
## [1] 0.8944524
## [1] 0.8944658
## [1] 0.8944792
```

```
## [1] 0.8944927
## [1] 0.8945061
## [1] 0.8945196
## [1] 0.894533
## [1] 0.8945464
## [1] 0.8945599
## [1] 0.8945733
## [1] 0.8945868
## [1] 0.8946002
## [1] 0.8946136
## [1] 0.8946271
## [1] 0.8946405
## [1] 0.894654
## [1] 0.8946674
## [1] 0.8946809
## [1] 0.8946943
## [1] 0.8947077
## [1] 0.8947212
## [1] 0.8947346
## [1] 0.8947481
## [1] 0.8947615
## [1] 0.8947749
## [1] 0.8947884
## [1] 0.8948018
## [1] 0.8948153
## [1] 0.8948287
## [1] 0.8948422
## [1] 0.8948556
## [1] 0.8948691
## [1] 0.8948825
## [1] 0.8948959
## [1] 0.8949094
## [1] 0.8949228
## [1] 0.8949363
## [1] 0.8949497
## [1] 0.8949632
## [1] 0.8949766
## [1] 0.8949901
## [1] 0.8950035
## [1] 0.895017
```