P— title: “GARCH Model” author: “Jane” date: “19/04/2021” output: html_document —

Forcasting Exchange Rate Using GARCH Model

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
```

```
CanJapCurrency <- readxl::read_xlsx ("CADJPY_Candlestick_1_D_BID_01.01.2000-31.12.2020.xlsx")%>%
  select('Gmt time', Close)%>%
  rename(Date = ('Gmt time'), RateCADJPY = ("Close"))

head(CanJapCurrency)
```

```
## # A tibble: 6 x 2
##   Date                RateCADJPY
##   <dttm>                   <dbl>
## 1 2000-01-03 00:00:00       70.1
## 2 2000-01-04 00:00:00       71.0
## 3 2000-01-05 00:00:00       71.9
## 4 2000-01-06 00:00:00       72.1
## 5 2000-01-07 00:00:00       72.3
## 6 2000-01-10 00:00:00       72.2
```

Conversion of Gmt time to date format

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

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

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

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

```
## # A tibble: 6 x 2
##   Date       RateCADJPY
##   <date>          <dbl>
## 1 2000-01-03       70.1
## 2 2000-01-04       71.0
## 3 2000-01-05       71.9
## 4 2000-01-06       72.1
## 5 2000-01-07       72.3
## 6 2000-01-10       72.2
```

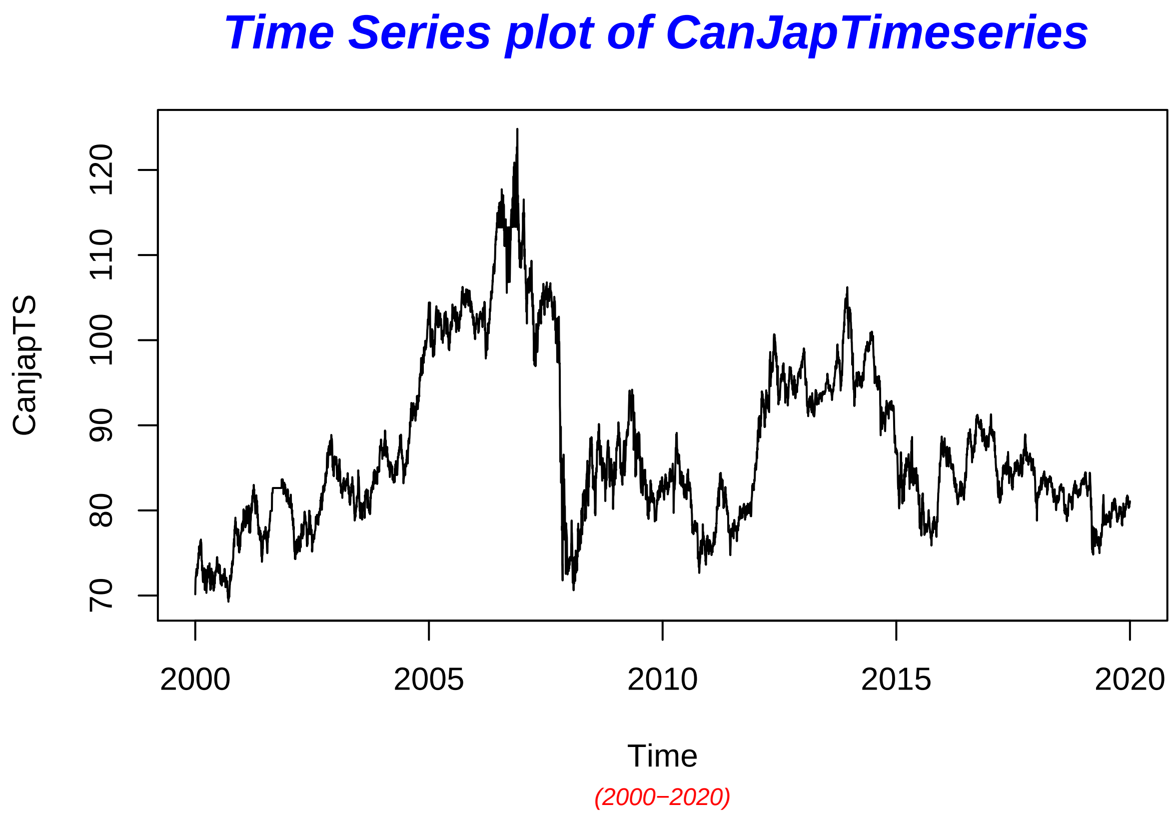
##Checking for obvious errors

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

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

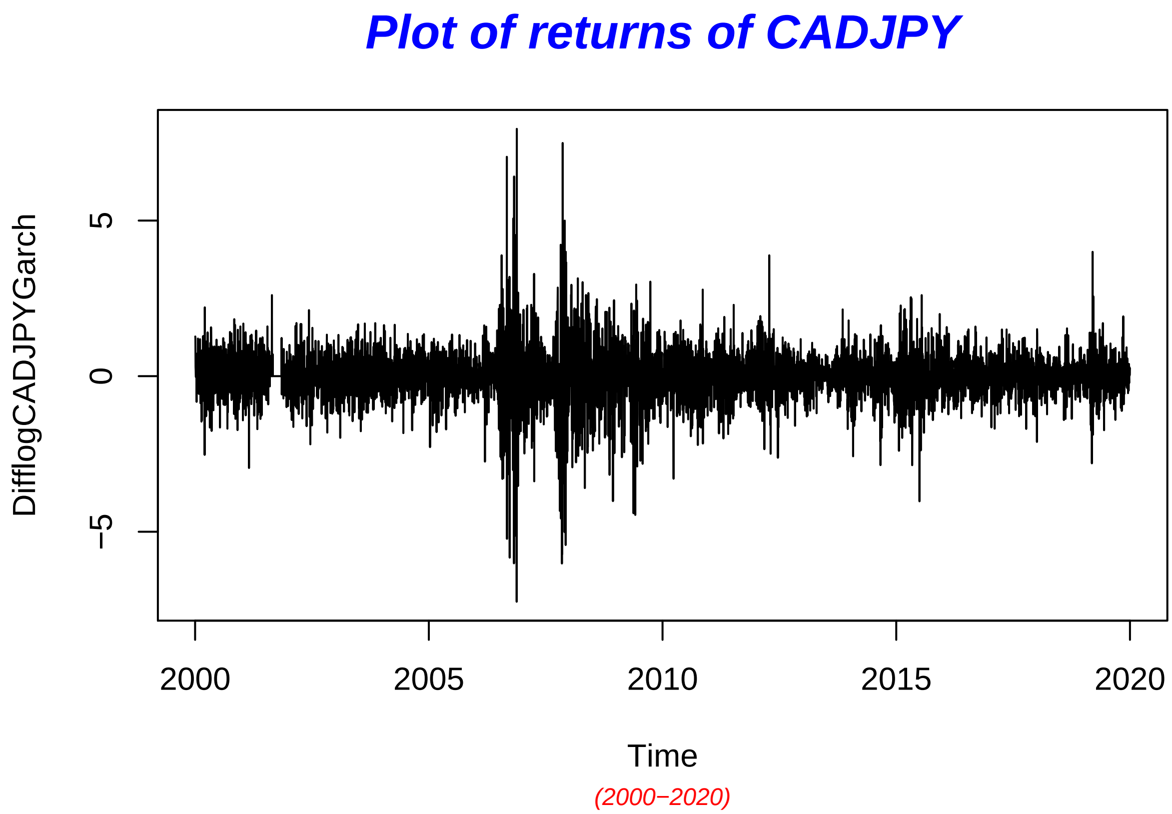
##Converting the data set into time series object

```r
#Converting the data set into time series object
CanjapTS<- ts(as.vector(CanJapCurrency$RateCADJPY),  frequency = 314, start= c(2000,01,03))
plot.ts(CanjapTS)
title("Time Series plot of CanJapTimeseries ", 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:

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

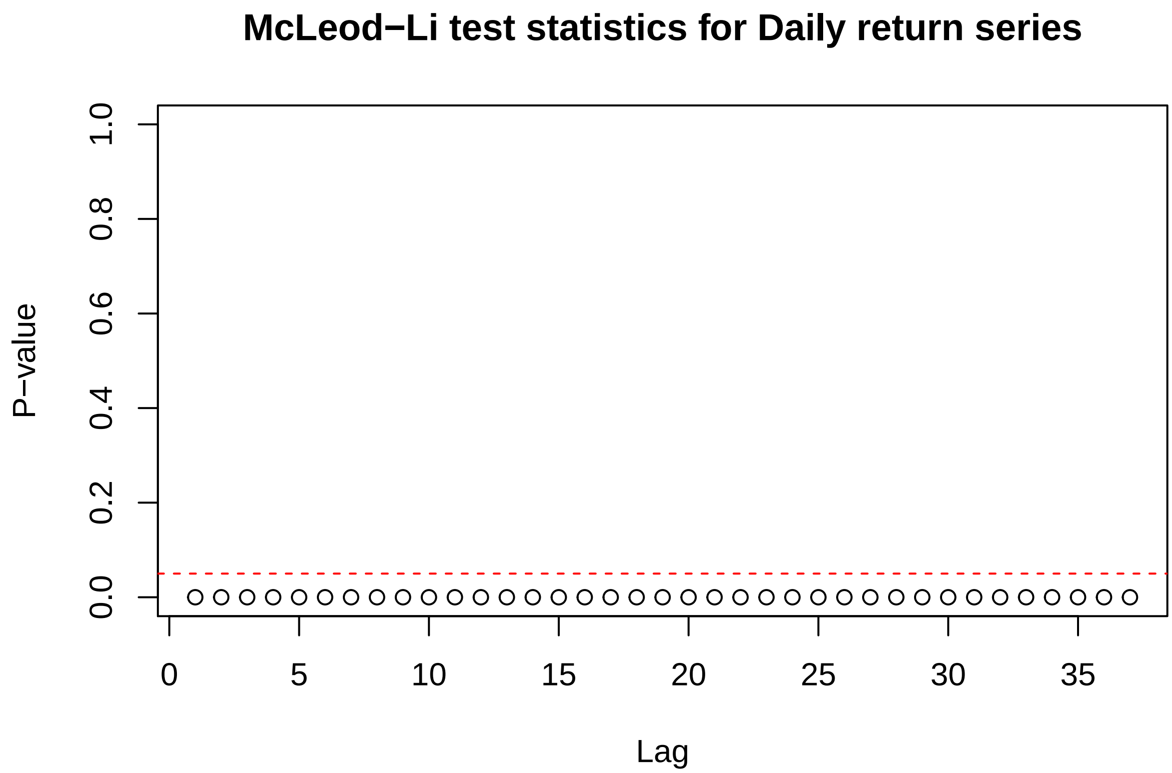


##nature as almost at all lags the p-values fall below the significance levels.

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

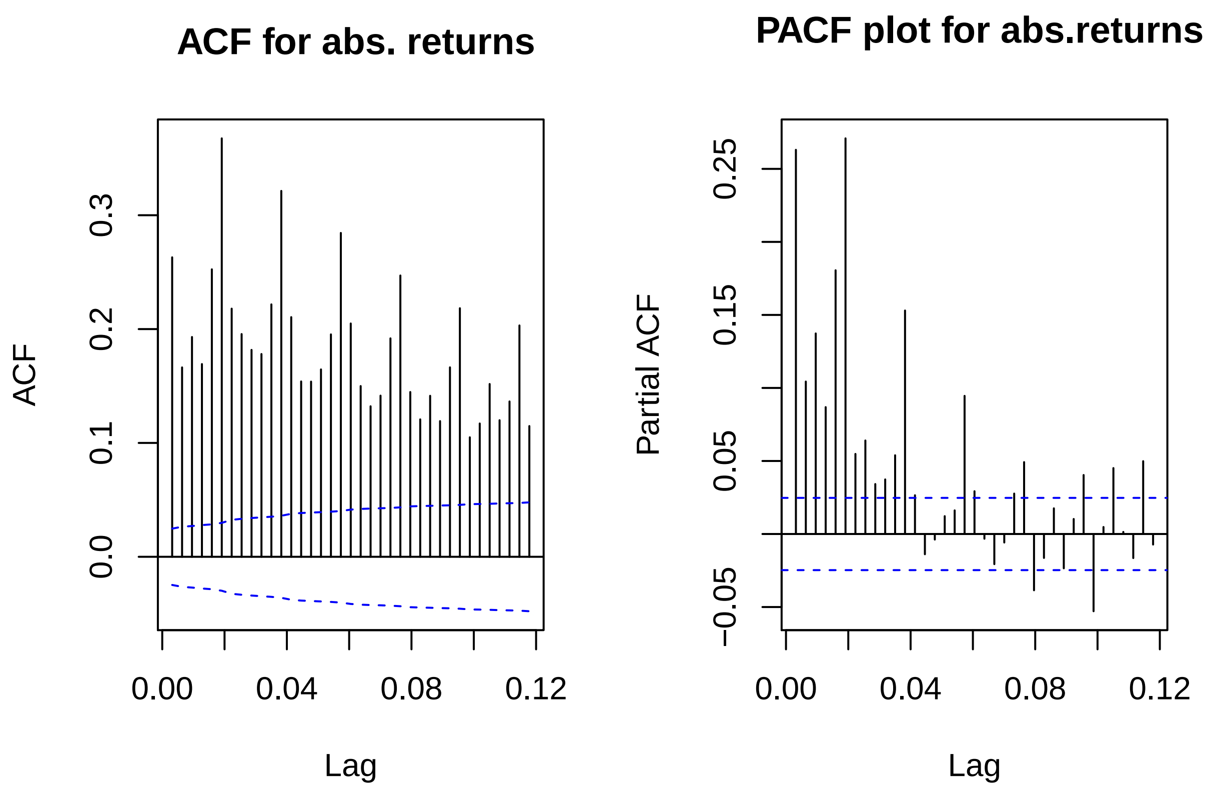


In order to get an order of GARCH , we further transform the return series into absolute values and squared return values.

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



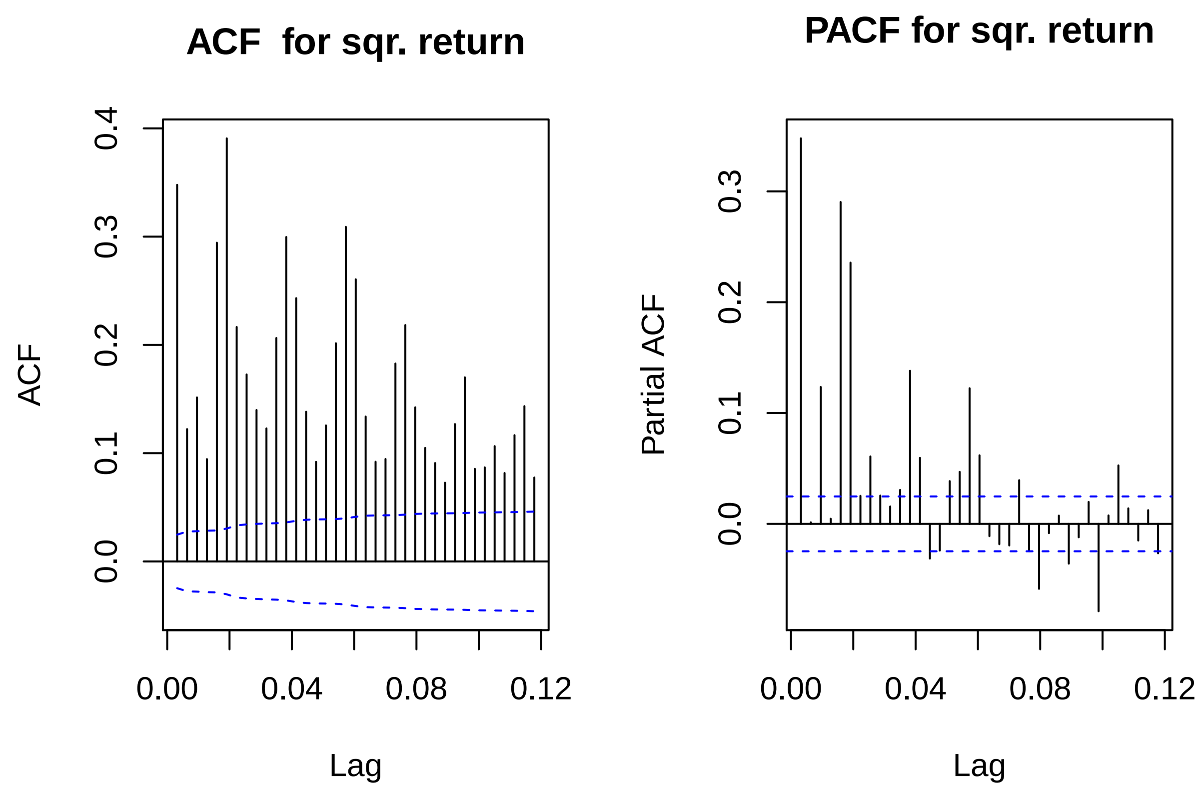
##From ACF and PACF we see many lags are significant. Hence, we plot EACF to get the candidate models

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

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

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

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

```
CADJPYGARCHFit.21 = garch(DifflogCADJPYGarch,order=c(2,1),trace =FALSE)
summary(CADJPYGARCHFit.21)
```

```
##
## Call:
## garch(x = DifflogCADJPYGarch, order = c(2, 1), trace = FALSE)
##
## Model:
## GARCH(2,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.82344 -0.53717  0.01714  0.57392  6.00782
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 0.0070264   0.0009442    7.442 9.95e-14 ***
## a1 0.0853361   0.0075587   11.290  < 2e-16 ***
## b1 0.5062561   0.1156088    4.379 1.19e-05 ***
## b2 0.3968365   0.1085996    3.654 0.000258 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 806.9, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.10875, df = 1, p-value = 0.7416
```

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.

```
CADJPYGARCHFit.22 = garch(DifflogCADJPYGarch, order =c(2,2),trace =FALSE)
summary(CADJPYGARCHFit.22)
```

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

```
## GARCH(2,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.90214 -0.53979  0.01714  0.57157  6.14195
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 7.423e-03   2.529e-03    2.935  0.00334 **
## a1 8.817e-02   1.134e-02    7.775 7.55e-15 ***
## a2 1.455e-15   3.346e-02    0.000  1.00000
## b1 5.893e-01   3.964e-01    1.487  0.13712
## b2 3.109e-01   3.649e-01    0.852  0.39423
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 830.12, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.28066, df = 1, p-value = 0.5963
```

##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
CADJPYGARCHFit.31 = garch(DifflogCADJPYGarch,order=c(3,1),trace =FALSE)
summary(CADJPYGARCHFit.31)
```

```
##
## Call:
## garch(x = DifflogCADJPYGarch, order = c(3, 1), trace = FALSE)
##
## Model:
## GARCH(3,1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.82001 -0.53699  0.01706  0.57282  6.01311
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 7.249e-03   1.118e-03    6.483 9.00e-11 ***
## a1 8.792e-02   1.093e-02    8.042 8.88e-16 ***
## b1 4.702e-01   1.122e-01    4.191 2.78e-05 ***
## b2 4.300e-01   1.308e-01    3.286  0.00102 **
## b3 6.385e-15   1.161e-01    0.000  1.00000
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 804.73, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.20947, df = 1, p-value = 0.6472
```

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

```
##
## Call:
## garch(x = DifflogCADJPYGarch, order = c(3, 2), trace = FALSE)
##
## Model:
## GARCH(3,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.82011 -0.53796  0.01704  0.57281  6.02567
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 7.594e-03   1.896e-02    0.400    0.689
## a1 8.191e-02   1.098e-02    7.460 8.64e-14 ***
## a2 1.018e-02   2.269e-01    0.045    0.964
## b1 4.195e-01   2.719e+00    0.154    0.877
## b2 4.760e-01   1.446e+00    0.329    0.742
## b3 3.961e-15   1.024e+00    0.000    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 805.12, df = 2, p-value < 2.2e-16
##
##
```

```
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.037538, df = 1, p-value = 0.8464
```

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
CADJPYGARCHFit.33 = garch(DifflogCADJPYGarch,order=c(3,3),trace =FALSE)
summary(CADJPYGARCHFit.33)
```

```
##
## Call:
## garch(x = DifflogCADJPYGarch, order = c(3, 3), trace = FALSE)
##
## Model:
## GARCH(3,3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.86955 -0.48886  0.01529  0.51797  5.01778
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 2.205e-01   1.832e-02   12.033  < 2e-16 ***
## a1 1.577e-01   1.405e-02   11.224  < 2e-16 ***
## a2 1.223e-01   1.867e-02    6.551 5.72e-11 ***
## a3 1.700e-01   1.798e-02    9.453  < 2e-16 ***
## b1 2.376e-16   7.347e-02    0.000 1.000000
## b2 8.573e-02   4.586e-02    1.870 0.061543 .
## b3 1.494e-01   4.296e-02    3.477 0.000506 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 1530, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 0.044216, df = 1, p-value = 0.8335
```

##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.

```
CADJPYGARCHFit.42 = garch(DifflogCADJPYGarch,order=c(4,2),trace =FALSE)
summary(CADJPYGARCHFit.42)
```

```
##
## Call:
## garch(x = DifflogCADJPYGarch, order = c(4, 2), trace = FALSE)
##
## Model:
## GARCH(4,2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.90742 -0.53569  0.01732  0.56932  5.90202
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0  0.011271    0.002774    4.063 4.85e-05 ***
## a1  0.101942    0.010803    9.436  < 2e-16 ***
## a2  0.036263    0.035837    1.012   0.3116
## b1  0.335247    0.322905    1.038   0.2992
## b2  0.017996    0.275916    0.065   0.9480
## b3  0.089137    0.167707    0.532   0.5951
## b4  0.400596    0.169745    2.360   0.0183 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 779.03, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 1.038, df = 1, p-value = 0.3083
```

##

```
CADJPYGARCHFit.41 = garch(DifflogCADJPYGarch,order=c(4,1),trace =FALSE)
summary(CADJPYGARCHFit.41)
```

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

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.80509 -0.53584  0.01724  0.57223  5.77193
##
## Coefficient(s):
##     Estimate  Std. Error  t value Pr(>|t|)
## a0 9.337e-03   1.204e-03    7.756 8.66e-15 ***
## a1 1.161e-01   7.677e-03   15.129  < 2e-16 ***
## b1 4.207e-01   7.668e-02    5.485 4.12e-08 ***
## b2 6.795e-16   9.335e-02    0.000    1.000
## b3 7.789e-02   9.239e-02    0.843    0.399
## b4 3.692e-01   6.773e-02    5.451 5.00e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Diagnostic Tests:
##  Jarque Bera Test
##
## data:  Residuals
## X-squared = 758.11, df = 2, p-value < 2.2e-16
##
##
##  Box-Ljung test
##
## data:  Squared.Residuals
## X-squared = 2.2038, df = 1, p-value = 0.1377
```

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
GARCHModelSelectionCADJPY = AIC(CADJPYGARCHFit.21,CADJPYGARCHFit.22 ,CADJPYGARCHFit.31,CADJPYGARCHFit.3
sortScore(GARCHModelSelectionCADJPY, score ="aic")
```

```
##                    df      AIC
## CADJPYGARCHFit.42  7 12962.87
## CADJPYGARCHFit.41  6 12964.79
## CADJPYGARCHFit.21  4 12965.75
## CADJPYGARCHFit.31  5 12965.80
## CADJPYGARCHFit.32  6 12967.39
## CADJPYGARCHFit.22  5 12969.19
## CADJPYGARCHFit.33  7 13484.61
```

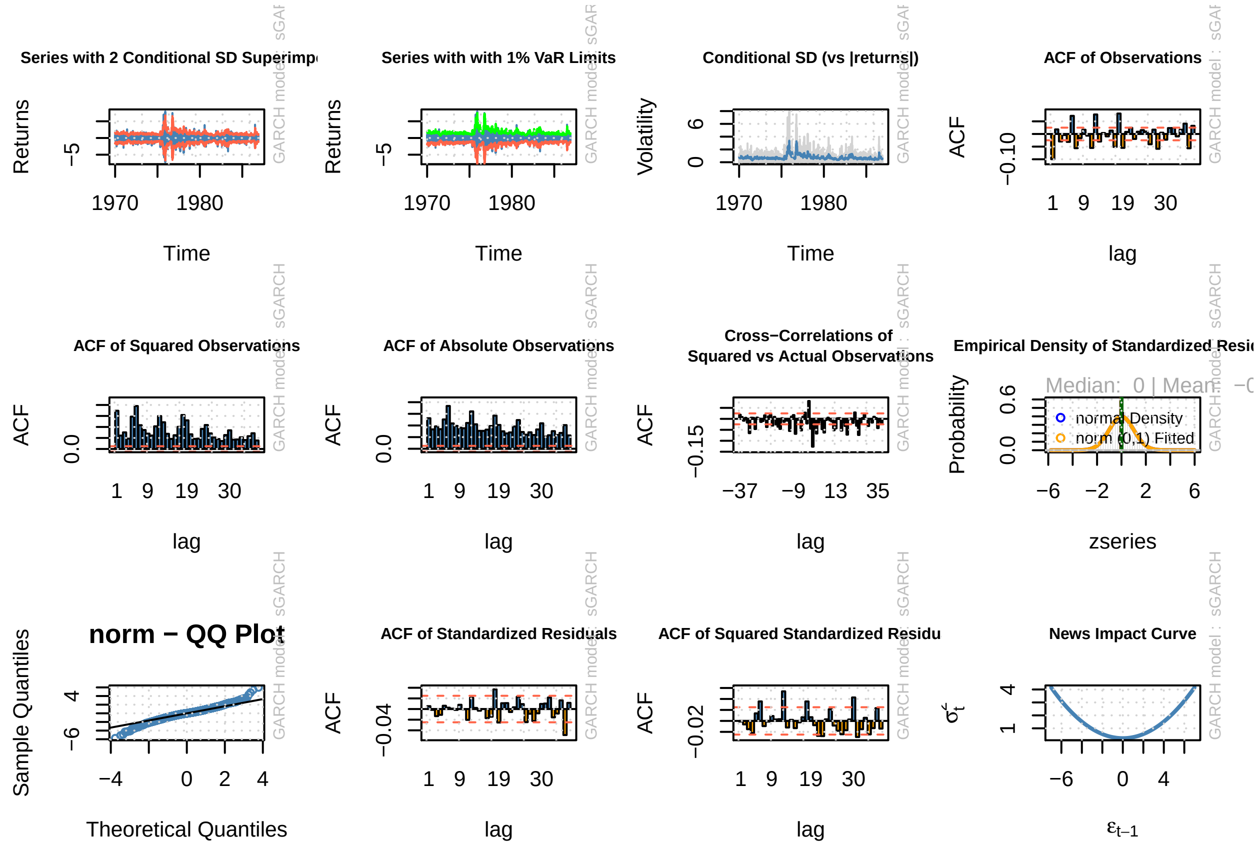
Model Fitting:

```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
CADJPYmodel2.2<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2,2)),
                  mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),
                  distribution.model = "norm")

CADJPYgarchMODEL2.2<-ugarchfit(spec=CADJPYmodel2.2,data=DifflogCADJPYGarch, out.sample = 100)
plot(CADJPYgarchMODEL2.2,which="all")
```

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



##Model Diagnostics

```
CADJPYgarchMODEL2.2
```

```
##
## *---------------------------------*
## *          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      0.013101    0.007487  1.74990 0.080136
## ar1    -0.502546    0.192131 -2.61564 0.008906
## ma1     0.462433    0.197091  2.34630 0.018961
## omega   0.007932    0.001634  4.85602 0.000001
## alpha1  0.079991    0.012028  6.65030 0.000000
## alpha2  0.015035    0.016647  0.90313 0.366458
## beta1   0.369882    0.134506  2.74993 0.005961
## beta2   0.522280    0.124587  4.19210 0.000028
```

```
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.013101    0.007591  1.72596 0.084355
## ar1    -0.502546    0.161189 -3.11774 0.001822
## ma1     0.462433    0.165304  2.79747 0.005150
## omega   0.007932    0.002494  3.18075 0.001469
## alpha1  0.079991    0.017944  4.45787 0.000008
## alpha2  0.015035    0.020133  0.74676 0.455211
## beta1   0.369882    0.060255  6.13863 0.000000
## beta2   0.522280    0.054792  9.53210 0.000000
##
## LogLikelihood : -6408.085
##
## Information Criteria
## ------------------------------------
##
## Akaike       2.0764
## Bayes        2.0851
## Shibata      2.0764
## Hannan-Quinn 2.0794
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.1835  0.6683
## Lag[2*(p+q)+(p+q)-1][5]    1.2003  0.9999
## Lag[4*(p+q)+(p+q)-1][9]    1.7485  0.9927
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                    0.007894  0.9292
## Lag[2*(p+q)+(p+q)-1][11]  8.996059  0.1453
## Lag[4*(p+q)+(p+q)-1][19] 20.544808  0.0100
## d.o.f=4
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[5]     1.176 0.500 2.000 0.27822
## ARCH Lag[7]     8.277 1.473 1.746 0.02245
## ARCH Lag[9]     8.842 2.402 1.619 0.04701
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.384
## Individual Statistics:
## mu     0.1700
## ar1    0.1353
## ma1    0.1295
## omega  0.1775
```

```
## alpha1 0.2769
## alpha2 0.2060
## beta1  0.2676
## beta2  0.2739
##
## 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           0.8982 3.691e-01
## Negative Sign Bias  2.5515 1.075e-02  **
## Positive Sign Bias  1.4429 1.491e-01
## Joint Effect       22.4274 5.315e-05 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     259.0    4.724e-44
## 2    30     286.7    3.409e-44
## 3    40     311.7    3.083e-44
## 4    50     319.3    2.534e-41
##
##
## Elapsed time : 0.7489619
```

Forecasting

```
forcgarchCADJPY= ugarchforecast(CADJPYgarchMODEL2.2, data = DifflogCADJPYGarch, n.ahead = 100, n.roll =
print(forcgarchCADJPY)
```

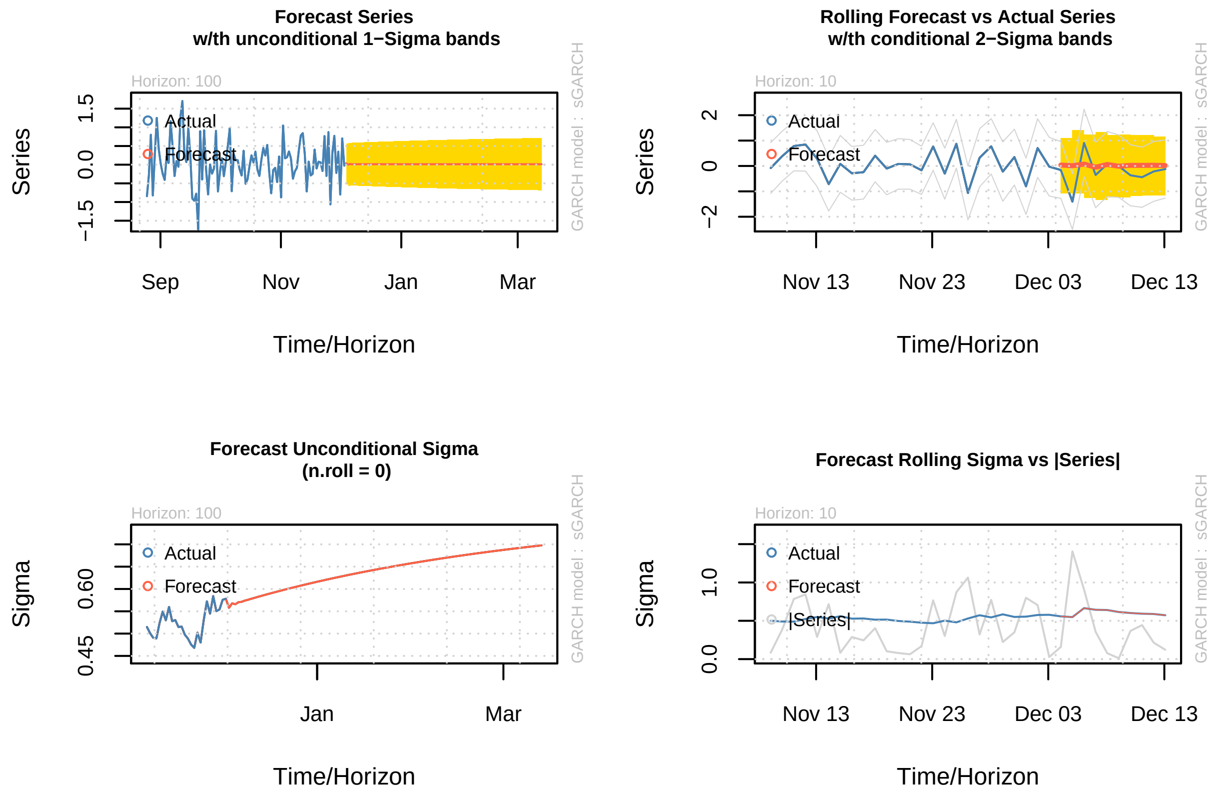
```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 100
## Roll Steps: 10
## Out of Sample: 100
##
## 0-roll forecast [T0=1986-12-03 02:00:00]:
##         Series  Sigma
## T+1   0.036517 0.5578
## T+2   0.001334 0.5678
## T+3   0.019015 0.5658
## T+4   0.010129 0.5703
## T+5   0.014595 0.5712
## T+6   0.012351 0.5740
## T+7   0.013478 0.5757
```

```
## T+8   0.012912 0.5780
## T+9   0.013196 0.5799
## T+10  0.013053 0.5820
## T+11  0.013125 0.5840
## T+12  0.013089 0.5859
## T+13  0.013107 0.5879
## T+14  0.013098 0.5898
## T+15  0.013103 0.5917
## T+16  0.013100 0.5936
## T+17  0.013102 0.5955
## T+18  0.013101 0.5973
## T+19  0.013101 0.5991
## T+20  0.013101 0.6009
## T+21  0.013101 0.6027
## T+22  0.013101 0.6045
## T+23  0.013101 0.6062
## T+24  0.013101 0.6080
## T+25  0.013101 0.6097
## T+26  0.013101 0.6113
## T+27  0.013101 0.6130
## T+28  0.013101 0.6147
## T+29  0.013101 0.6163
## T+30  0.013101 0.6179
## T+31  0.013101 0.6195
## T+32  0.013101 0.6211
## T+33  0.013101 0.6226
## T+34  0.013101 0.6242
## T+35  0.013101 0.6257
## T+36  0.013101 0.6272
## T+37  0.013101 0.6287
## T+38  0.013101 0.6302
## T+39  0.013101 0.6316
## T+40  0.013101 0.6331
## T+41  0.013101 0.6345
## T+42  0.013101 0.6359
## T+43  0.013101 0.6373
## T+44  0.013101 0.6387
## T+45  0.013101 0.6401
## T+46  0.013101 0.6414
## T+47  0.013101 0.6428
## T+48  0.013101 0.6441
## T+49  0.013101 0.6454
## T+50  0.013101 0.6467
## T+51  0.013101 0.6480
## T+52  0.013101 0.6493
## T+53  0.013101 0.6506
## T+54  0.013101 0.6518
## T+55  0.013101 0.6530
## T+56  0.013101 0.6543
## T+57  0.013101 0.6555
## T+58  0.013101 0.6567
## T+59  0.013101 0.6579
## T+60  0.013101 0.6590
## T+61  0.013101 0.6602
```

```
## T+62  0.013101 0.6614
## T+63  0.013101 0.6625
## T+64  0.013101 0.6636
## T+65  0.013101 0.6648
## T+66  0.013101 0.6659
## T+67  0.013101 0.6670
## T+68  0.013101 0.6680
## T+69  0.013101 0.6691
## T+70  0.013101 0.6702
## T+71  0.013101 0.6712
## T+72  0.013101 0.6723
## T+73  0.013101 0.6733
## T+74  0.013101 0.6743
## T+75  0.013101 0.6753
## T+76  0.013101 0.6763
## T+77  0.013101 0.6773
## T+78  0.013101 0.6783
## T+79  0.013101 0.6793
## T+80  0.013101 0.6803
## T+81  0.013101 0.6812
## T+82  0.013101 0.6822
## T+83  0.013101 0.6831
## T+84  0.013101 0.6840
## T+85  0.013101 0.6849
## T+86  0.013101 0.6858
## T+87  0.013101 0.6867
## T+88  0.013101 0.6876
## T+89  0.013101 0.6885
## T+90  0.013101 0.6894
## T+91  0.013101 0.6903
## T+92  0.013101 0.6911
## T+93  0.013101 0.6920
## T+94  0.013101 0.6928
## T+95  0.013101 0.6936
## T+96  0.013101 0.6945
## T+97  0.013101 0.6953
## T+98  0.013101 0.6961
## T+99  0.013101 0.6969
## T+100 0.013101 0.6977
```

plotting

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



Forecasting the rate

```
p.t_1 =81.074
  R_t <- c(  0.036517, 0.001334, 0.019015, 0.010129, 0.014595, 0.012351, 0.013478, 0.012912, 0.013196, 





)
  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] 81.10361
## [1] 81.10469
## [1] 81.12012
## [1] 81.12833
## [1] 81.14018
## [1] 81.1502
## [1] 81.16114
```

```
## [1] 81.17162
## [1] 81.18233
## [1] 81.19293
## [1] 81.20358
## [1] 81.21421
## [1] 81.22486
## [1] 81.2355
## [1] 81.24614
## [1] 81.25679
## [1] 81.26743
## [1] 81.27808
## [1] 81.28873
## [1] 81.29938
## [1] 81.31003
## [1] 81.32068
## [1] 81.33134
## [1] 81.34199
## [1] 81.35265
## [1] 81.36331
## [1] 81.37397
## [1] 81.38463
## [1] 81.3953
## [1] 81.40596
## [1] 81.41663
## [1] 81.42729
## [1] 81.43796
## [1] 81.44863
## [1] 81.4593
## [1] 81.46997
## [1] 81.48065
## [1] 81.49132
## [1] 81.502
## [1] 81.51268
## [1] 81.52336
## [1] 81.53404
## [1] 81.54472
## [1] 81.55541
## [1] 81.56609
## [1] 81.57678
## [1] 81.58747
## [1] 81.59816
## [1] 81.60885
## [1] 81.61954
## [1] 81.63023
## [1] 81.64093
## [1] 81.65162
## [1] 81.66232
## [1] 81.67302
## [1] 81.68372
## [1] 81.69442
## [1] 81.70513
## [1] 81.71583
## [1] 81.72654
## [1] 81.73725
```

```
## [1] 81.74796
## [1] 81.75867
## [1] 81.76938
## [1] 81.78009
## [1] 81.79081
## [1] 81.80152
## [1] 81.81224
## [1] 81.82296
## [1] 81.83368
## [1] 81.8444
## [1] 81.85512
## [1] 81.86585
## [1] 81.87657
## [1] 81.8873
## [1] 81.89803
## [1] 81.90876
## [1] 81.91949
## [1] 81.93023
## [1] 81.94096
## [1] 81.9517
## [1] 81.96243
## [1] 81.97317
## [1] 81.98391
## [1] 81.99465
## [1] 82.0054
## [1] 82.01614
## [1] 82.02689
## [1] 82.03763
## [1] 82.04838
## [1] 82.05913
## [1] 82.06988
## [1] 82.08063
## [1] 82.09139
## [1] 82.10214
## [1] 82.1129
## [1] 82.12366
## [1] 82.13442
## [1] 82.14518
## [1] 82.15594
```