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'), Rate = ("Close"))  
  
head(CanJapCurrency)

## # A tibble: 6 x 2  
## Date Rate  
## <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

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

## # A tibble: 6 x 2  
## Date Rate  
## <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

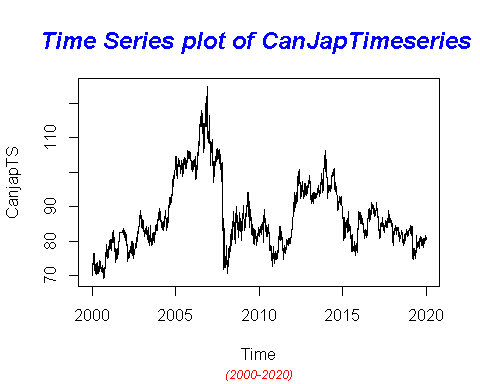
##Checking for obvious errors

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

## integer(0)

##Converting the data set into time series object

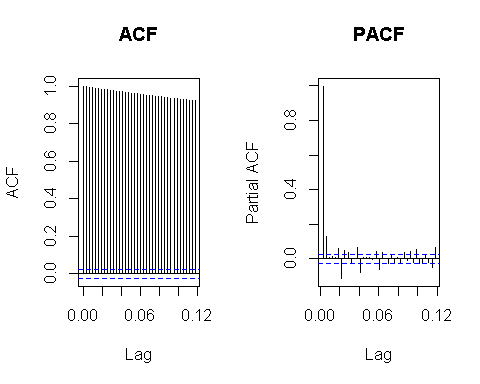
#Converting the data set into time series object  
CanjapTS<- ts(as.vector(CanJapCurrency$Rate), frequency = 314, start= c(2000,01,03))  
plot.ts(CanjapTS)  
title("Time Series plot of CanJapTimeseries ", sub = "(2000-2020)",  
 cex.main = 1.5, font.main= 4, col.main= "blue",  
 cex.sub = 0.75, font.sub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



## Automatic Colleration and Partial Automatic Correlation

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

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



## Differening the series to ensure stationality

ar(diff(CanjapTS))

##   
## Call:  
## ar(x = diff(CanjapTS))  
##   
## Coefficients:  
## 1 2 3 4 5 6 7 8   
## -0.1050 -0.0022 -0.0262 -0.0164 -0.0254 0.0935 -0.0483 -0.0362   
## 9 10 11 12 13 14 15 16   
## 0.0160 -0.0126 -0.0598 0.0786 -0.0043 -0.0108 -0.0065 0.0052   
## 17 18 19 20 21 22 23 24   
## -0.0495 0.0742 -0.0490 0.0074 0.0255 -0.0205 0.0139 0.0095   
## 25 26 27 28 29 30 31 32   
## 0.0105 -0.0417 0.0080 -0.0493 0.0026 -0.0502 0.0298 0.0324   
## 33 34 35 36 37   
## -0.0177 0.0289 0.0483 -0.0736 0.0208   
##   
## Order selected 37 sigma^2 estimated as 0.5129

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

library(tseries)

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

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

##   
## Augmented Dickey-Fuller Test  
##   
## data: CanjapTS  
## Dickey-Fuller = -3.2006, Lag order = 0, p-value = 0.08772  
## alternative hypothesis: stationary

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

CanJapLogTran<-log(CanjapTS)  
  
ar(CanJapLogTran)

##   
## Call:  
## ar(x = CanJapLogTran)  
##   
## Coefficients:  
## 1 2 3 4 5 6 7 8   
## 0.9195 0.0786 -0.0314 0.0194 -0.0011 0.0573 -0.0822 0.0122   
## 9 10 11 12 13 14 15 16   
## 0.0393 -0.0182 -0.0447 0.1031 -0.0630 0.0027 -0.0046 0.0176   
## 17 18 19 20 21 22 23 24   
## -0.0417 0.0921 -0.0989 0.0449 0.0175 -0.0418 0.0334 0.0028   
## 25 26 27 28 29 30 31 32   
## -0.0060 -0.0493 0.0484 -0.0538 0.0451 -0.0262 0.0505 0.0030   
## 33 34 35 36 37   
## -0.0459 0.0421 0.0167 -0.0874 0.0478   
##   
## Order selected 37 sigma^2 estimated as 7.202e-05

## Augmented Dickey-Fuller Test for log Tranformation

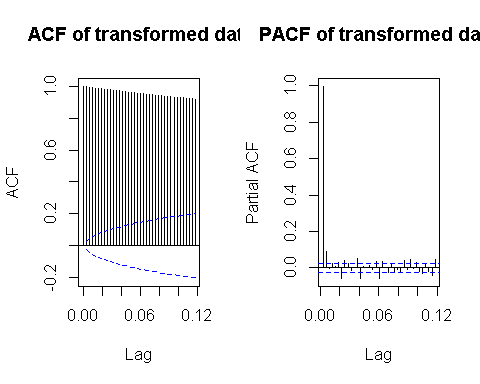
adf.test(CanJapLogTran, alternative = "stationary", k = 0)

##   
## Augmented Dickey-Fuller Test  
##   
## data: CanJapLogTran  
## Dickey-Fuller = -3.2236, Lag order = 0, p-value = 0.08377  
## alternative hypothesis: stationary

## ACF and PACF for log transformation

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

par(mfrow=c(1,2))  
acf(CanJapLogTran, ci.type='ma', main=" ACF of transformed data")  
  
pacf(CanJapLogTran, main="PACF of transformed data")



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

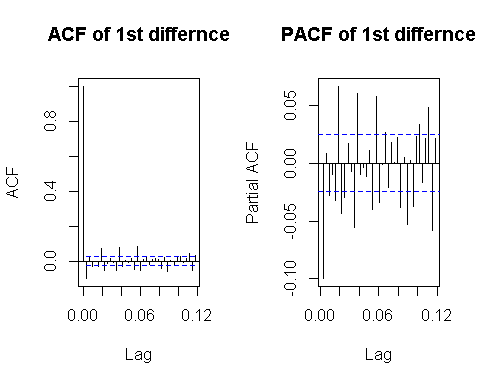
DiffCanJapLogTran<-diff(CanJapLogTran)   
ar(DiffCanJapLogTran)

##   
## Call:  
## ar(x = DiffCanJapLogTran)  
##   
## Coefficients:  
## 1 2 3 4 5 6 7 8   
## -0.0801 0.0044 -0.0338 -0.0129 -0.0138 0.0519 -0.0412 -0.0282   
## 9 10 11 12 13 14 15 16   
## 0.0150 -0.0104 -0.0506 0.0572 -0.0089 -0.0067 -0.0089 0.0098   
## 17 18 19 20 21 22 23 24   
## -0.0412 0.0625 -0.0435 0.0039 0.0246 -0.0187 0.0120 0.0121   
## 25 26 27 28 29 30 31 32   
## 0.0125 -0.0393 0.0042 -0.0523 -0.0043 -0.0307 0.0254 0.0331   
## 33 34 35 36 37   
## -0.0166 0.0262 0.0436 -0.0562 0.0215   
##   
## Order selected 37 sigma^2 estimated as 6.393e-05

## Runing ACF and PACF for the log transform Difference

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

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



## Augmented Dickey-Fuller Test for log Tranformation Difference

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

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

##   
## Augmented Dickey-Fuller Test  
##   
## data: DiffCanJapLogTran  
## Dickey-Fuller = -87.573, Lag order = 0, p-value = 0.01  
## alternative hypothesis: stationary

### 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 Canadian and Japanese Curruency Pair

# GARCH(2,1)  
CanJapGARCHFit.21 = garch(DiffCanJapLogTran,order=c(2,1),trace =FALSE)

## Warning in garch(DiffCanJapLogTran, order = c(2, 1), trace = FALSE): singular  
## information

summary(CanJapGARCHFit.21)

##   
## Call:  
## garch(x = DiffCanJapLogTran, order = c(2, 1), trace = FALSE)  
##   
## Model:  
## GARCH(2,1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.82350 -0.53716 0.01713 0.57391 6.00789   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)  
## a0 7.028e-07 NA NA NA  
## a1 8.534e-02 NA NA NA  
## b1 5.063e-01 NA NA NA  
## b2 3.968e-01 NA NA NA  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 806.92, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 0.10902, df = 1, p-value = 0.7413

## 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.

CanJapGARCHFit.22 = garch(DiffCanJapLogTran, order =c(2,2),trace =FALSE)

## Warning in garch(DiffCanJapLogTran, order = c(2, 2), trace = FALSE): singular  
## information

summary(CanJapGARCHFit.22)

##   
## Call:  
## garch(x = DiffCanJapLogTran, order = c(2, 2), trace = FALSE)  
##   
## Model:  
## GARCH(2,2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.02302 -0.51052 0.01635 0.54297 7.20419   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)  
## a0 1.753e-06 NA NA NA  
## a1 2.094e-01 NA NA NA  
## a2 3.491e-14 NA NA NA  
## b1 3.809e-01 NA NA NA  
## b2 4.093e-01 NA NA NA  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 1266.7, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 14.799, df = 1, p-value = 0.0001196

##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.

CanJapGARCHFit.31 = garch(DiffCanJapLogTran,order=c(3,1),trace =FALSE)

## Warning in garch(DiffCanJapLogTran, order = c(3, 1), trace = FALSE): singular  
## information

summary(CanJapGARCHFit.31)

##   
## Call:  
## garch(x = DiffCanJapLogTran, order = c(3, 1), trace = FALSE)  
##   
## Model:  
## GARCH(3,1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.80603 -0.53899 0.01701 0.57338 5.98266   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)  
## a0 7.562e-07 NA NA NA  
## a1 8.936e-02 NA NA NA  
## b1 4.135e-01 NA NA NA  
## b2 4.843e-01 NA NA NA  
## b3 1.937e-08 NA NA NA  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 795.22, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 0.27024, df = 1, p-value = 0.6032

##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)

CanJapGARCHFit.32 = garch(DiffCanJapLogTran,order=c(3,2),trace =FALSE)

## Warning in garch(DiffCanJapLogTran, order = c(3, 2), trace = FALSE): singular  
## information

summary(CanJapGARCHFit.32)

##   
## Call:  
## garch(x = DiffCanJapLogTran, order = c(3, 2), trace = FALSE)  
##   
## Model:  
## GARCH(3,2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.32413 -0.52578 0.01685 0.55821 6.57841   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)  
## a0 2.992e-06 NA NA NA  
## a1 1.971e-01 NA NA NA  
## a2 5.013e-02 NA NA NA  
## b1 8.747e-07 NA NA NA  
## b2 3.449e-01 NA NA NA  
## b3 3.713e-01 NA NA NA  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 1099.7, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 14.666, df = 1, p-value = 0.0001284

## 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)

CanJapGARCHFit.33 = garch(DiffCanJapLogTran,order=c(3,3),trace =FALSE)

## Warning in garch(DiffCanJapLogTran, order = c(3, 3), trace = FALSE): singular  
## information

summary(CanJapGARCHFit.33)

##   
## Call:  
## garch(x = DiffCanJapLogTran, order = c(3, 3), trace = FALSE)  
##   
## Model:  
## GARCH(3,3)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.82696 -0.52750 0.01667 0.56015 6.91365   
##   
## Coefficient(s):  
## Estimate Std. Error t value Pr(>|t|)  
## a0 3.235e-06 NA NA NA  
## a1 4.831e-02 NA NA NA  
## a2 1.112e-01 NA NA NA  
## a3 1.469e-01 NA NA NA  
## b1 1.219e-02 NA NA NA  
## b2 2.550e-01 NA NA NA  
## b3 3.950e-01 NA NA NA  
##   
## Diagnostic Tests:  
## Jarque Bera Test  
##   
## data: Residuals  
## X-squared = 995.93, df = 2, p-value < 2.2e-16  
##   
##   
## Box-Ljung test  
##   
## data: Squared.Residuals  
## X-squared = 1.4164, df = 1, p-value = 0.234

##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.

CanJapGARCHFit.42 = garch(DiffCanJapLogTran,order=c(4,2),trace =FALSE)

## Warning in garch(DiffCanJapLogTran, order = c(4, 2), trace = FALSE): singular  
## information

summary(CanJapGARCHFit.42)

##   
## Call:  
## garch(x = DiffCanJapLogTran, 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 1.127e-06 NA NA NA  
## a1 1.019e-01 NA NA NA  
## a2 3.626e-02 NA NA NA  
## b1 3.352e-01 NA NA NA  
## b2 1.800e-02 NA NA NA  
## b3 8.914e-02 NA NA NA  
## b4 4.006e-01 NA NA NA  
##   
## 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

#Model Selection:

##Best possible model is selected by AIC scores of the models. From the below sort function, GARCH(2,2) would be the best model for the return series.

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

GARCHModelSelectionCanJap = AIC(CanJapGARCHFit.21,CanJapGARCHFit.22 ,CanJapGARCHFit.31,CanJapGARCHFit.32,CanJapGARCHFit.33, CanJapGARCHFit.42)  
sortScore(GARCHModelSelectionCanJap, score ="aic")

## df AIC  
## CanJapGARCHFit.21 4 -44856.77  
## CanJapGARCHFit.31 5 -44847.13  
## CanJapGARCHFit.42 7 -44841.23  
## CanJapGARCHFit.33 7 -44759.72  
## CanJapGARCHFit.32 6 -44742.18  
## CanJapGARCHFit.22 5 -44722.43

# Model Fitting:

library(rugarch)

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

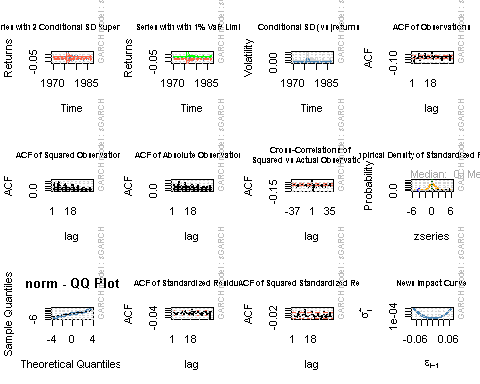
## Loading required package: parallel

##   
## Attaching package: 'rugarch'

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

CanJapmodel2.1<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2, 2)),   
 mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),   
 distribution.model = "norm")  
   
MODEL2.1<-ugarchfit(spec=CanJapmodel2.1,data=DiffCanJapLogTran, out.sample = 100)  
plot(MODEL2.1,which="all")

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



##Model Diagnostics

MODEL2.1

##   
## \*---------------------------------\*  
## \* 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 0.000131 0.000075 1.7556 0.079150  
## ar1 -0.504004 0.191868 -2.6268 0.008619  
## ma1 0.464021 0.196842 2.3573 0.018407  
## omega 0.000001 0.000000 3.1854 0.001445  
## alpha1 0.079586 0.011885 6.6962 0.000000  
## alpha2 0.014674 0.011863 1.2369 0.216113  
## beta1 0.377091 0.105705 3.5674 0.000361  
## beta2 0.516465 0.103434 4.9932 0.000001  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## mu 0.000131 0.000076 1.72955 0.083710  
## ar1 -0.504004 0.161718 -3.11656 0.001830  
## ma1 0.464021 0.165919 2.79667 0.005163  
## omega 0.000001 0.000001 0.69825 0.485024  
## alpha1 0.079586 0.018972 4.19492 0.000027  
## alpha2 0.014674 0.048781 0.30081 0.763558  
## beta1 0.377091 0.265727 1.41909 0.155871  
## beta2 0.516465 0.216243 2.38835 0.016924  
##   
## LogLikelihood : 22051.85   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike -7.1339  
## Bayes -7.1252  
## Shibata -7.1339  
## Hannan-Quinn -7.1309  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.1791 0.6721  
## Lag[2\*(p+q)+(p+q)-1][5] 1.1888 0.9999  
## Lag[4\*(p+q)+(p+q)-1][9] 1.7362 0.9930  
## d.o.f=2  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.004282 0.94782  
## Lag[2\*(p+q)+(p+q)-1][11] 8.955797 0.14773  
## Lag[4\*(p+q)+(p+q)-1][19] 20.491812 0.01024  
## d.o.f=4  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[5] 1.207 0.500 2.000 0.27185  
## ARCH Lag[7] 8.241 1.473 1.746 0.02289  
## ARCH Lag[9] 8.807 2.402 1.619 0.04782  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 89.7066  
## Individual Statistics:   
## mu 0.1669  
## ar1 0.1380  
## ma1 0.1322  
## omega 26.2362  
## alpha1 0.2563  
## alpha2 0.1871  
## beta1 0.2467  
## beta2 0.2525  
##   
## 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 0.902 3.671e-01   
## Negative Sign Bias 2.544 1.100e-02 \*\*  
## Positive Sign Bias 1.444 1.487e-01   
## Joint Effect 22.401 5.383e-05 \*\*\*  
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 258.9 4.914e-44  
## 2 30 286.2 4.343e-44  
## 3 40 308.4 1.344e-43  
## 4 50 316.6 8.138e-41  
##   
##   
## Elapsed time : 0.6878271

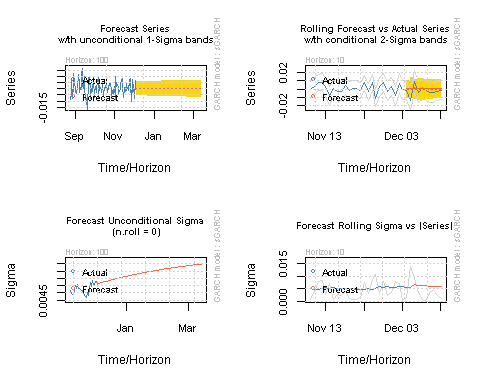
## Forecasting

forc = ugarchforecast(MODEL2.1, data = DiffCanJapLogTran, n.ahead = 100, n.roll =10)  
print(forc)

##   
## \*------------------------------------\*  
## \* 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 3.659e-04 0.005572  
## T+2 1.321e-05 0.005672  
## T+3 1.910e-04 0.005652  
## T+4 1.014e-04 0.005696  
## T+5 1.465e-04 0.005705  
## T+6 1.238e-04 0.005733  
## T+7 1.353e-04 0.005750  
## T+8 1.295e-04 0.005772  
## T+9 1.324e-04 0.005792  
## T+10 1.309e-04 0.005812  
## T+11 1.317e-04 0.005832  
## T+12 1.313e-04 0.005851  
## T+13 1.315e-04 0.005871  
## T+14 1.314e-04 0.005890  
## T+15 1.314e-04 0.005909  
## T+16 1.314e-04 0.005927  
## T+17 1.314e-04 0.005946  
## T+18 1.314e-04 0.005964  
## T+19 1.314e-04 0.005982  
## T+20 1.314e-04 0.006000  
## T+21 1.314e-04 0.006018  
## T+22 1.314e-04 0.006036  
## T+23 1.314e-04 0.006053  
## T+24 1.314e-04 0.006070  
## T+25 1.314e-04 0.006087  
## T+26 1.314e-04 0.006104  
## T+27 1.314e-04 0.006120  
## T+28 1.314e-04 0.006137  
## T+29 1.314e-04 0.006153  
## T+30 1.314e-04 0.006169  
## T+31 1.314e-04 0.006185  
## T+32 1.314e-04 0.006201  
## T+33 1.314e-04 0.006216  
## T+34 1.314e-04 0.006232  
## T+35 1.314e-04 0.006247  
## T+36 1.314e-04 0.006262  
## T+37 1.314e-04 0.006277  
## T+38 1.314e-04 0.006292  
## T+39 1.314e-04 0.006307  
## T+40 1.314e-04 0.006321  
## T+41 1.314e-04 0.006335  
## T+42 1.314e-04 0.006350  
## T+43 1.314e-04 0.006364  
## T+44 1.314e-04 0.006378  
## T+45 1.314e-04 0.006391  
## T+46 1.314e-04 0.006405  
## T+47 1.314e-04 0.006418  
## T+48 1.314e-04 0.006432  
## T+49 1.314e-04 0.006445  
## T+50 1.314e-04 0.006458  
## T+51 1.314e-04 0.006471  
## T+52 1.314e-04 0.006484  
## T+53 1.314e-04 0.006497  
## T+54 1.314e-04 0.006509  
## T+55 1.314e-04 0.006522  
## T+56 1.314e-04 0.006534  
## T+57 1.314e-04 0.006546  
## T+58 1.314e-04 0.006558  
## T+59 1.314e-04 0.006570  
## T+60 1.314e-04 0.006582  
## T+61 1.314e-04 0.006594  
## T+62 1.314e-04 0.006606  
## T+63 1.314e-04 0.006617  
## T+64 1.314e-04 0.006629  
## T+65 1.314e-04 0.006640  
## T+66 1.314e-04 0.006651  
## T+67 1.314e-04 0.006662  
## T+68 1.314e-04 0.006673  
## T+69 1.314e-04 0.006684  
## T+70 1.314e-04 0.006695  
## T+71 1.314e-04 0.006705  
## T+72 1.314e-04 0.006716  
## T+73 1.314e-04 0.006726  
## T+74 1.314e-04 0.006737  
## T+75 1.314e-04 0.006747  
## T+76 1.314e-04 0.006757  
## T+77 1.314e-04 0.006767  
## T+78 1.314e-04 0.006777  
## T+79 1.314e-04 0.006787  
## T+80 1.314e-04 0.006797  
## T+81 1.314e-04 0.006807  
## T+82 1.314e-04 0.006816  
## T+83 1.314e-04 0.006826  
## T+84 1.314e-04 0.006835  
## T+85 1.314e-04 0.006845  
## T+86 1.314e-04 0.006854  
## T+87 1.314e-04 0.006863  
## T+88 1.314e-04 0.006872  
## T+89 1.314e-04 0.006881  
## T+90 1.314e-04 0.006890  
## T+91 1.314e-04 0.006899  
## T+92 1.314e-04 0.006908  
## T+93 1.314e-04 0.006916  
## T+94 1.314e-04 0.006925  
## T+95 1.314e-04 0.006934  
## T+96 1.314e-04 0.006942  
## T+97 1.314e-04 0.006950  
## T+98 1.314e-04 0.006959  
## T+99 1.314e-04 0.006967  
## T+100 1.314e-04 0.006975

## plotting

plot(forc, which= "all")

 ## Forecasting the rate

p.t\_1 = 81.074  
 R\_t <-c(0.005572, 0.005672, 0.005652, 0.005696, 0.005705, 0.005733, 0.005750, 0.005772, 0.005792, 0.005812, 0.005832, 0.005851, 0.005871, 0.005890, 0.005909, 0.005927, 0.005946, 0.005964, 0.005982, 0.006000, 0.006018, 0.006036, 0.006053, 0.006070, 0.006087, 0.006104, 0.006120, 0.006137, 0.006153, 0.006169, 0.006185, 0.006201, 0.006216, 0.006232, 0.006247, 0.006262, 0.006277, 0.006292, 0.006307, 0.006321, 0.006335, 0.006350, 0.006364, 0.006378, 0.006391, 0.006405, 0.006418, 0.006432, 0.006445, 0.006458, 0.006471, 0.006484, 0.006497, 0.006509, 0.006522, 0.006534, 0.006546, 0.006558, 0.006570, 0.006582, 0.006594, 0.006606, 0.006617, 0.006629, 0.006640, 0.006651, 0.006662, 0.006673, 0.006684, 0.006695, 0.006705, 0.006716, 0.006726, 0.006737, 0.006747, 0.006757, 0.006767, 0.006777, 0.006787, 0.006797, 0.006807, 0.006816, 0.006826, 0.006835, 0.006845, 0.006854, 0.006863, 0.006872, 0.006881, 0.006890, 0.006899, 0.006908, 0.006916, 0.006925, 0.006934, 0.006942, 0.006950, 0.006959, 0.006967, 0.006975)  
 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] 81.07852  
## [1] 81.08312  
## [1] 81.0877  
## [1] 81.09232  
## [1] 81.09694  
## [1] 81.10159  
## [1] 81.10626  
## [1] 81.11094  
## [1] 81.11564  
## [1] 81.12035  
## [1] 81.12508  
## [1] 81.12983  
## [1] 81.13459  
## [1] 81.13937  
## [1] 81.14417  
## [1] 81.14898  
## [1] 81.1538  
## [1] 81.15864  
## [1] 81.1635  
## [1] 81.16837  
## [1] 81.17325  
## [1] 81.17815  
## [1] 81.18307  
## [1] 81.18799  
## [1] 81.19294  
## [1] 81.19789  
## [1] 81.20286  
## [1] 81.20784  
## [1] 81.21284  
## [1] 81.21785  
## [1] 81.22287  
## [1] 81.22791  
## [1] 81.23296  
## [1] 81.23802  
## [1] 81.2431  
## [1] 81.24819  
## [1] 81.25329  
## [1] 81.2584  
## [1] 81.26352  
## [1] 81.26866  
## [1] 81.27381  
## [1] 81.27897  
## [1] 81.28414  
## [1] 81.28933  
## [1] 81.29452  
## [1] 81.29973  
## [1] 81.30495  
## [1] 81.31018  
## [1] 81.31542  
## [1] 81.32067  
## [1] 81.32593  
## [1] 81.33121  
## [1] 81.33649  
## [1] 81.34178  
## [1] 81.34709  
## [1] 81.3524  
## [1] 81.35773  
## [1] 81.36307  
## [1] 81.36841  
## [1] 81.37377  
## [1] 81.37913  
## [1] 81.38451  
## [1] 81.38989  
## [1] 81.39529  
## [1] 81.4007  
## [1] 81.40611  
## [1] 81.41153  
## [1] 81.41697  
## [1] 81.42241  
## [1] 81.42786  
## [1] 81.43332  
## [1] 81.43879  
## [1] 81.44427  
## [1] 81.44975  
## [1] 81.45525  
## [1] 81.46075  
## [1] 81.46627  
## [1] 81.47179  
## [1] 81.47732  
## [1] 81.48285  
## [1] 81.4884  
## [1] 81.49396  
## [1] 81.49952  
## [1] 81.50509  
## [1] 81.51067  
## [1] 81.51626  
## [1] 81.52185  
## [1] 81.52745  
## [1] 81.53306  
## [1] 81.53868  
## [1] 81.54431  
## [1] 81.54994  
## [1] 81.55558  
## [1] 81.56123  
## [1] 81.56688  
## [1] 81.57255  
## [1] 81.57822  
## [1] 81.58389  
## [1] 81.58958  
## [1] 81.59527