ARIMA Model EUR And CAD

Jane

01/05/2021

Forcasting Exchange Rate Using ARIMA Model for EUR And CAD

Reading EUR and CAD 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
```

```
EURCADARIMA<-  read.csv ("EURCAD_Candlestick_1_D_BID_01.01.2000-31.12.2020.csv")%>%
  select('GMT.TIME', CLOSE)%>%
  rename(Date = ('GMT.TIME'), RateEURCAD = ("CLOSE"))


head(EURCADARIMA)
```

```
##         Date RateEURCAD
## 1 2000-01-03     1.4817
## 2 2000-01-04     1.4969
## 3 2000-01-05     1.4963
## 4 2000-01-06     1.5064
## 5 2000-01-07     1.4992
## 6 2000-01-10     1.4928
```

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

```
##         Date RateEURCAD
## 1 2000-01-03     1.4817
## 2 2000-01-04     1.4969
## 3 2000-01-05     1.4963
## 4 2000-01-06     1.5064
## 5 2000-01-07     1.4992
## 6 2000-01-10     1.4928
```

##Checking for obvious errors or missingg value

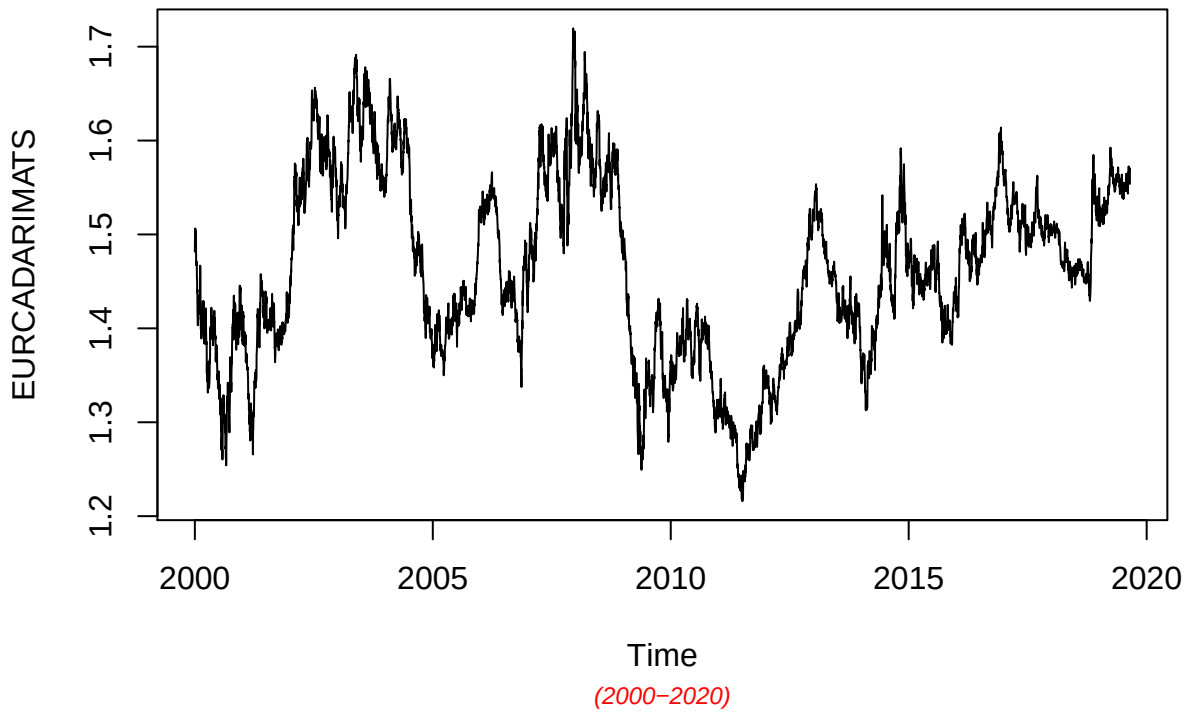
```r
#Checking for obvious errors
which(is.na(EURCADARIMA))
```

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

##Converting the data set into time series object

```r
#Converting the data set into time series object
EURCADARIMATS<- ts(as.vector(EURCADARIMA$Rate),  frequency = 322, start= c(2000,01,03))
plot.ts(EURCADARIMATS)
title("Time Series plot of EURCADTimeseries ", sub = "(2000-2020)",
      cex.main = 1.5,   font.main= 4, col.main= "blue",
      cex.sub = 0.75, font.sub = 3, col.sub = "red")
```

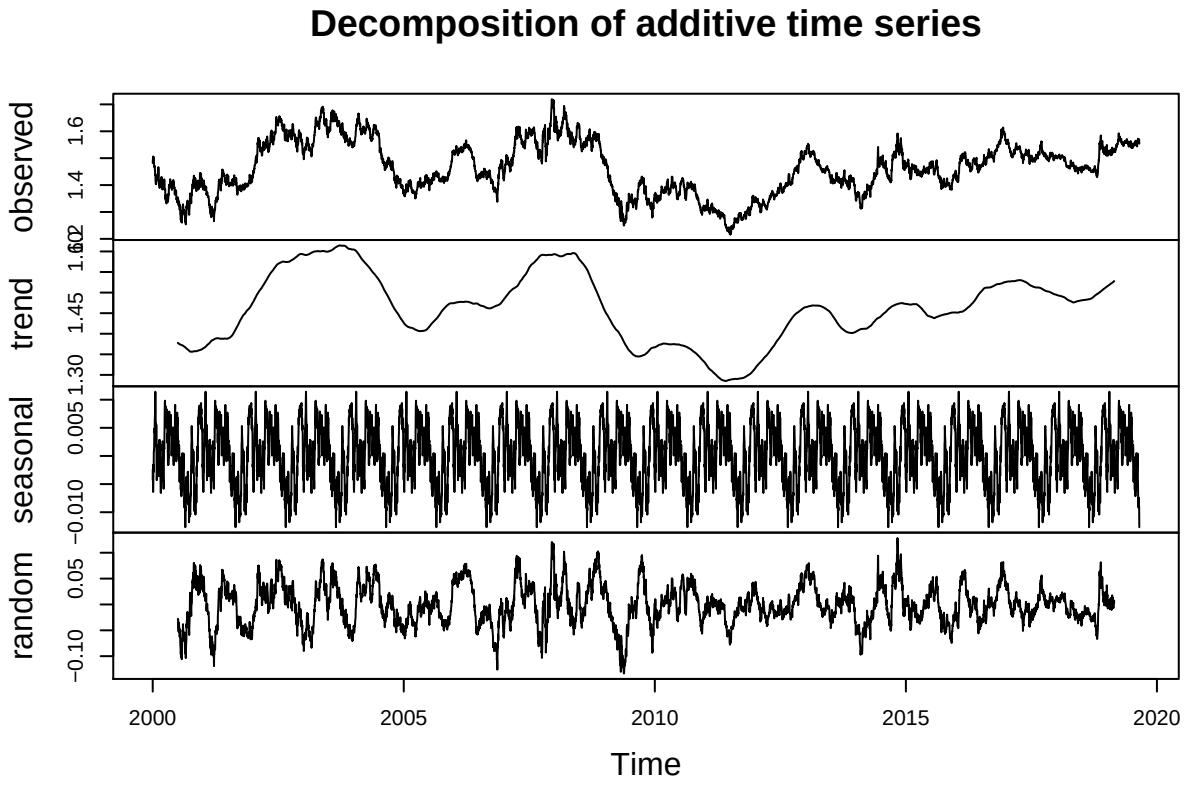
Time Series plot of EURCADTimeseries



(2000−2020)

Finding the component of the Time Series

```
ComponentEURCAD <- decompose(EURCADARIMATS)
plot(ComponentEURCAD)
```



To To achieve stationarity by differencing the data – compute the differences between consecutive observations

```
library("fUnitRoots")
```

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

```
## Loading required package: timeDate
```

```
## Warning: package 'timeDate' was built under R version 4.0.4
```

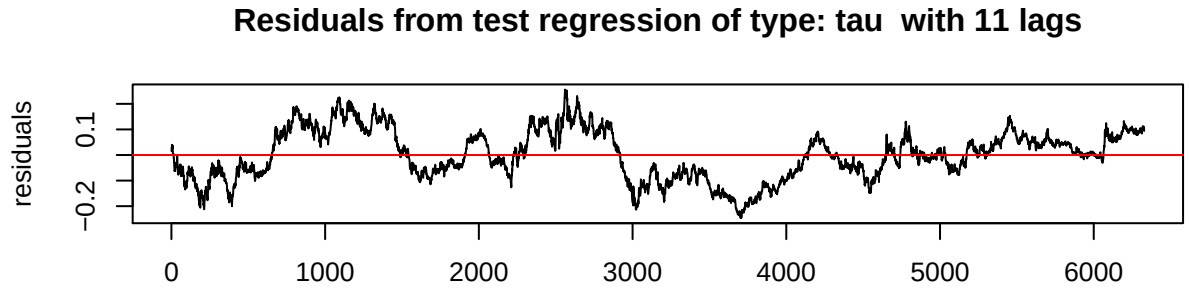
```
## Loading required package: timeSeries
```

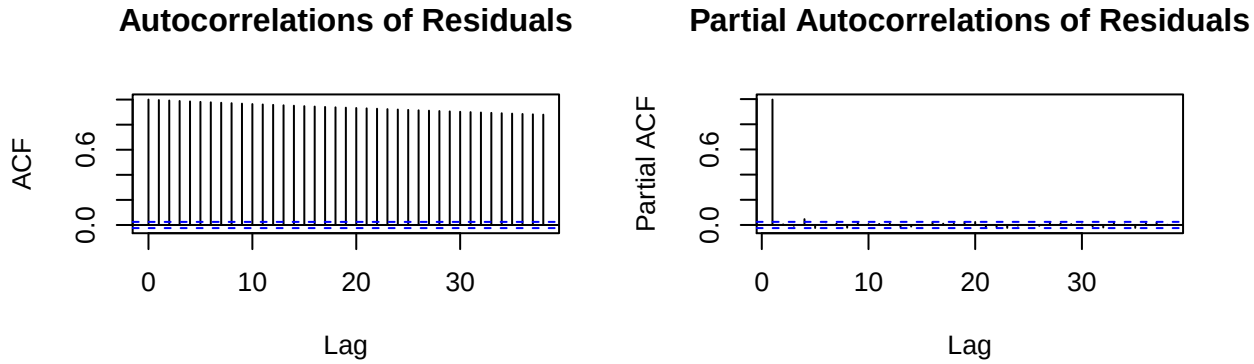
```
## Warning: package 'timeSeries' was built under R version 4.0.5
```

```
## Loading required package: fBasics
```

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

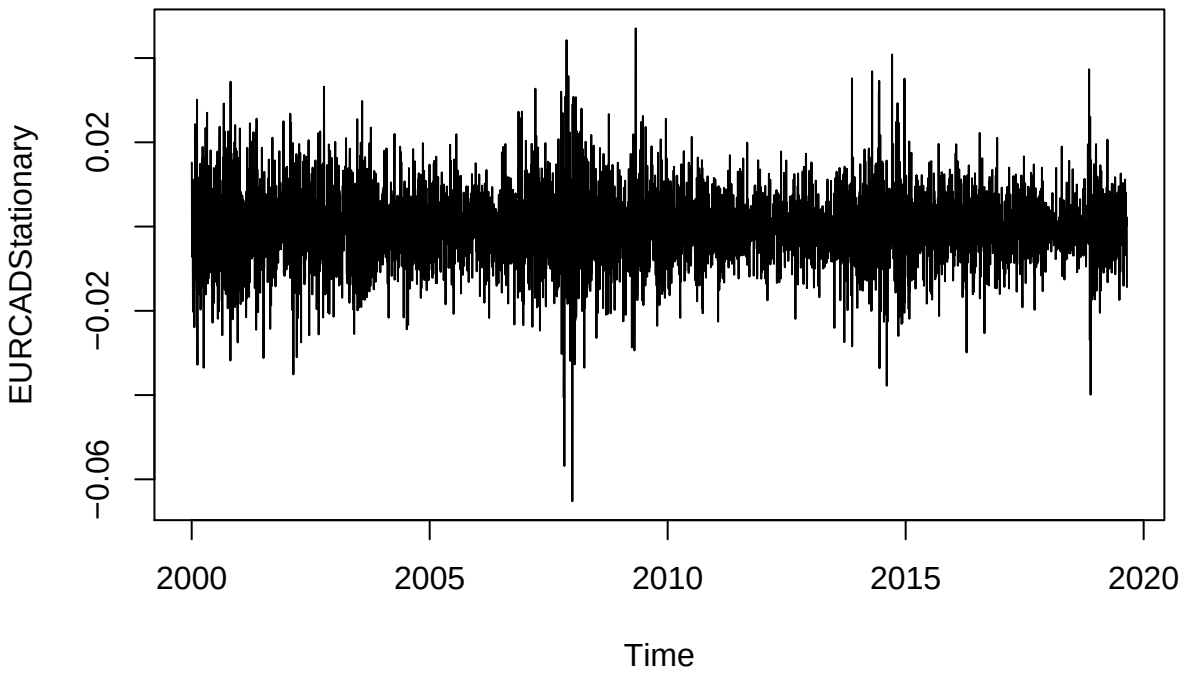
```
urkpssTest(EURCADARIMATS, type = c("tau"), lags = c("short"),use.lag = NULL, doplot = TRUE)
```





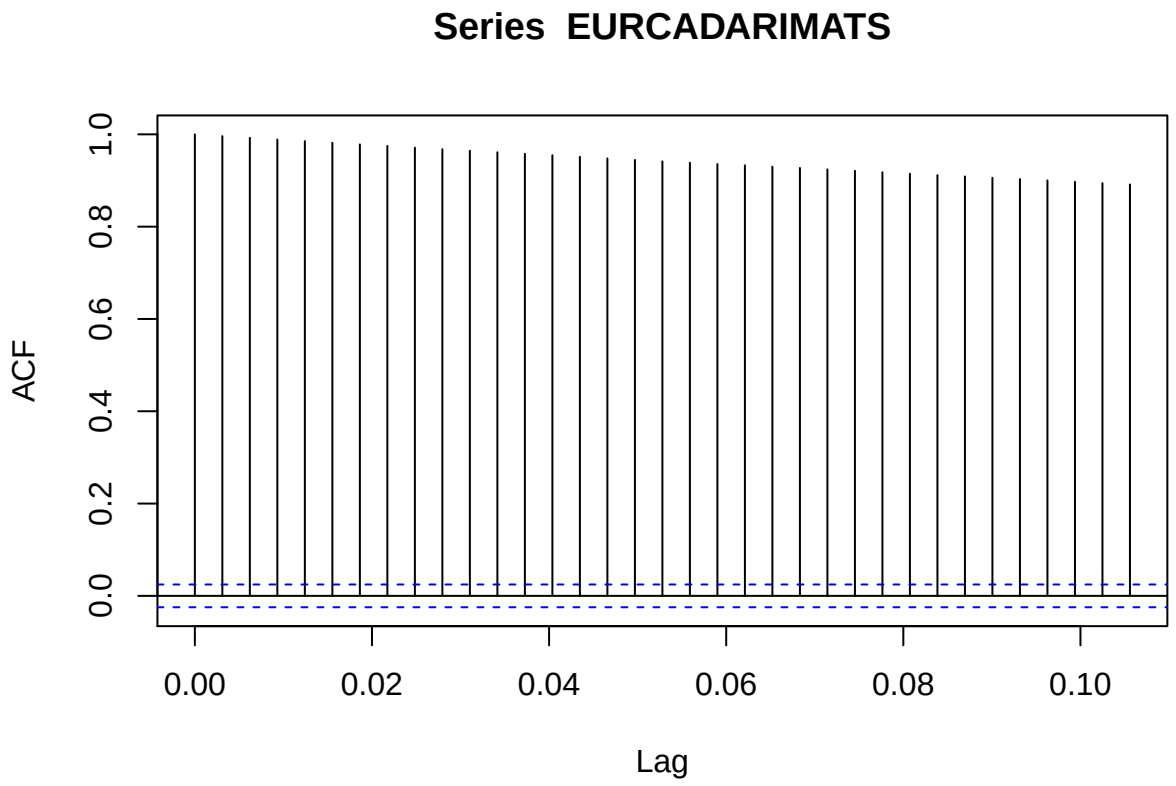

```
##
## Title:
##  KPSS Unit Root Test
##
## Test Results:
##   NA
##
## Description:
##  Mon May 03 23:33:44 2021 by user: janeo
```

```
EURCADStationary= diff(EURCADARIMATS, differences=1)
plot(EURCADStationary)
```

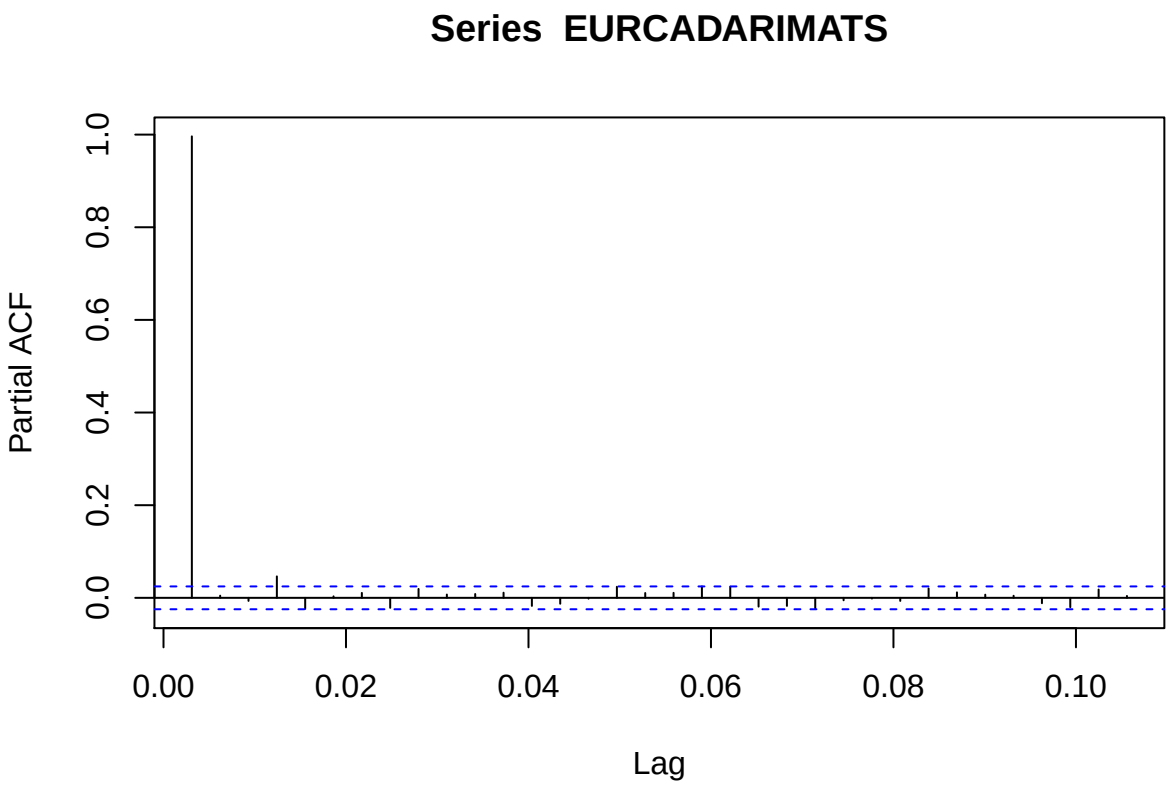


Calculating Autocorrlation function and partil autocorlation function

```
acf(EURCADARIMATS,lag.max=34)
```

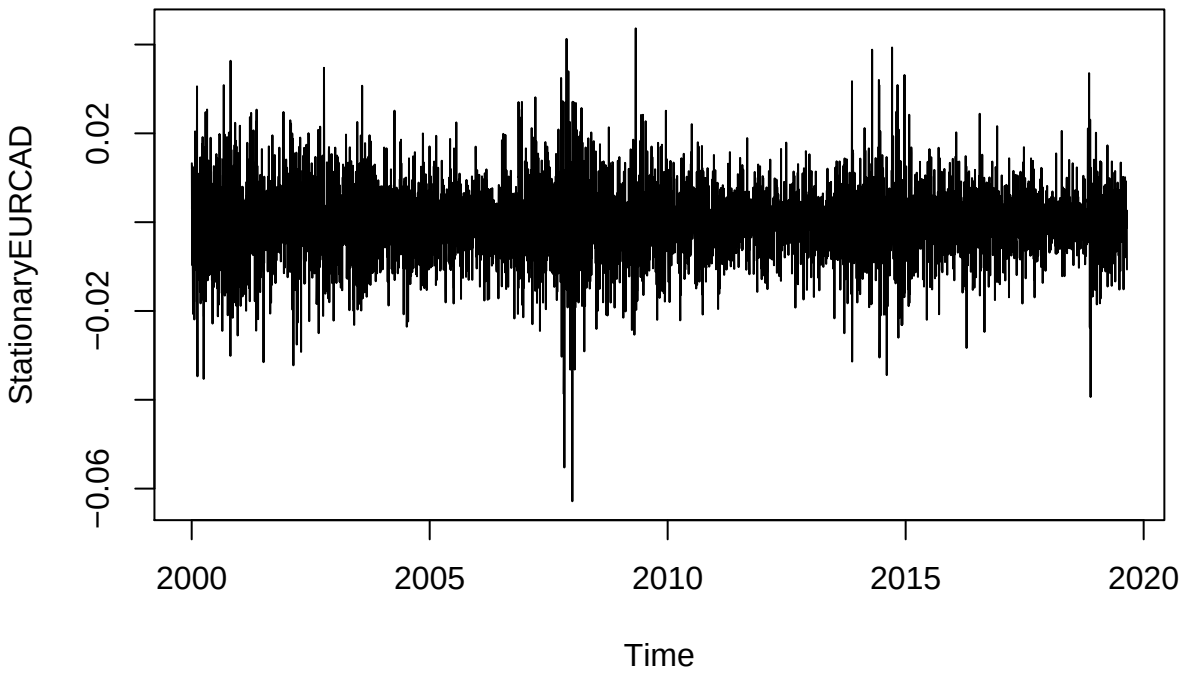



```
pacf(EURCADARIMATS, lag.max = 34)
```



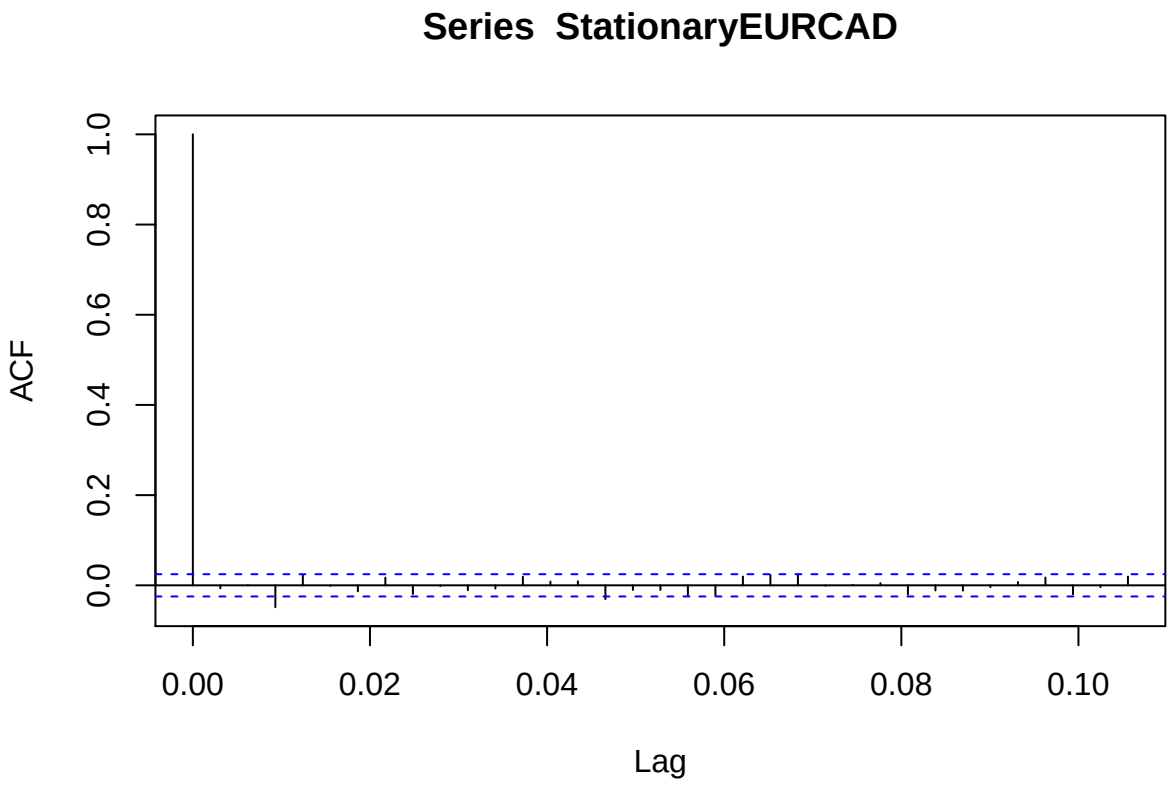
Adjusting and ensuring there are no seasonality

```
TSseasonallyadjustedEURCAD <- EURCADARIMATS- ComponentEURCAD$seasonal
StationaryEURCAD <- diff(TSseasonallyadjustedEURCAD, differences=1)
plot(StationaryEURCAD)
```

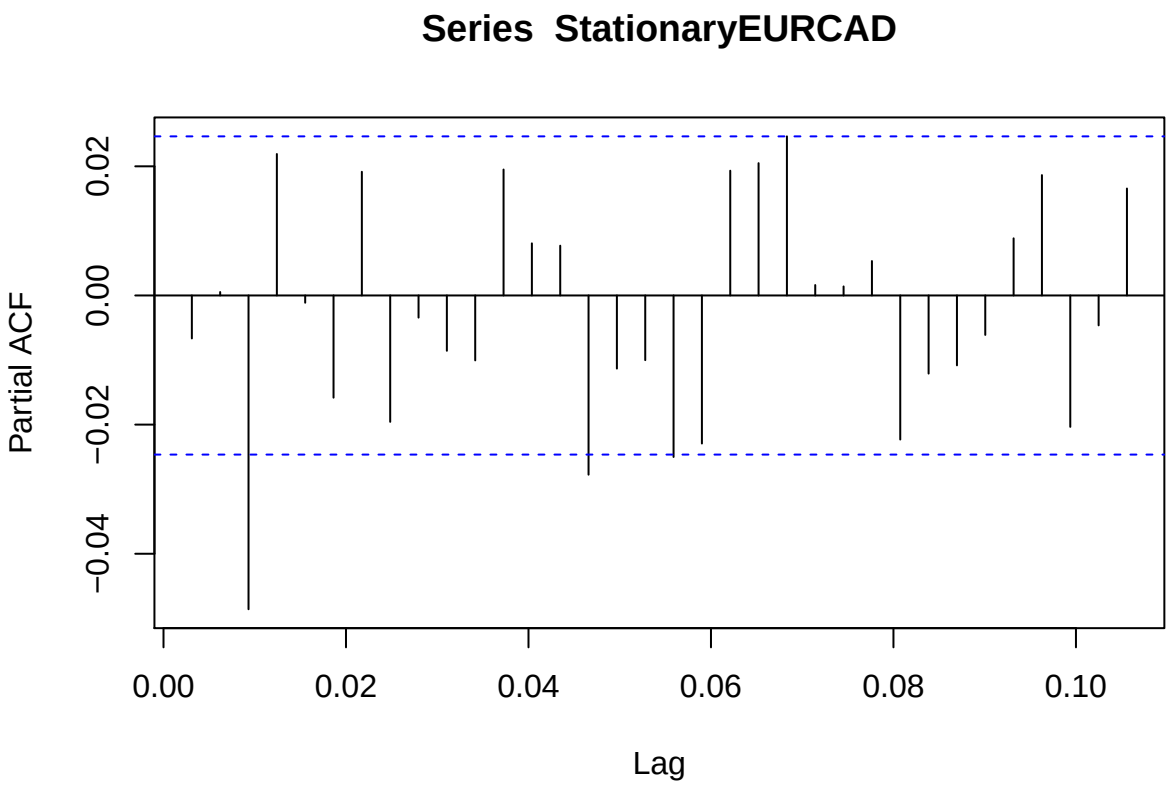


Calculating again for ACF and PACF after finding stationality

```
acf(StationaryEURCAD, lag.max=34)
```




```
pacf(StationaryEURCAD, lag.max=34)
```



Fitting The ARIMA Model

ARIMA fitting (1,1,0)

```
fitArima1EURCAD <- arima(EURCADARIMATS, order =  c(1,0,0), include.mean = TRUE)
fitArima1EURCAD
```

```
##
## Call:
## arima(x = EURCADARIMATS, order = c(1, 0, 0), include.mean = TRUE)
##
## Coefficients:
##          ar1  intercept
##       0.9964     1.4630
## s.e.  0.0011     0.0275
##
## sigma^2 estimated as 6.832e-05:  log likelihood = 21365.44,  aic = -42724.87
```

##Arima Fitting (0,1,0)

```
fitArima2EURCAD <- arima(EURCADARIMATS, order =  c(0,1,0), include.mean = TRUE)
fitArima2EURCAD
```

```
##
## Call:
## arima(x = EURCADARIMATS, order = c(0, 1, 0), include.mean = TRUE)
##
##
## sigma^2 estimated as 6.845e-05:  log likelihood = 21358.4,  aic = -42714.8
```

Arima Fitting (2,1,1)

```
fitArima3EURCAD <- arima(EURCADARIMATS, order = c(2,1,1), include.mean = TRUE)
fitArima3EURCAD
```

```
##
## Call:
## arima(x = EURCADARIMATS, order = c(2, 1, 1), include.mean = TRUE)
##
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

##           ar1     ar2      ma1
##       -0.0055  0.0041  -0.0048
## s.e.      NaN  0.0081      NaN
##
## sigma^2 estimated as 6.844e-05:  log likelihood = 21358.79,  aic = -42709.58
```

##Fitting Arima (3,1,0)

```
fitArima4EURCAD <- arima(EURCADARIMATS, order = c(3,1,0), include.mean = TRUE)
fitArima4EURCAD
```

```
##
## Call:
## arima(x = EURCADARIMATS, order = c(3, 1, 0), include.mean = TRUE)
##
## Coefficients:
##           ar1     ar2      ar3
##       -0.0101  0.0038  -0.0493
## s.e.   0.0126  0.0126   0.0126
##
## sigma^2 estimated as 6.828e-05:  log likelihood = 21366.48,  aic = -42724.96
```

##Best possible model is selected by AIC scores of the models

```
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
```

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

```
## Loading required package: dynlm
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following object is masked from 'package:timeSeries':
##
##     time<-
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
ARIMAModelSelectionEURCAD = AIC(fitArima1EURCAD,fitArima2EURCAD,fitArima3EURCAD,fitArima4EURCAD)
```

```
## Warning in AIC.default(fitArima1EURCAD, fitArima2EURCAD, fitArima3EURCAD, :
## models are not all fitted to the same number of observations
```

```
sortScore(ARIMAModelSelectionEURCAD, score ="aic")
```

```
##                 df       AIC
## fitArima4EURCAD  4 -42724.96
## fitArima1EURCAD  3 -42724.87
## fitArima2EURCAD  1 -42714.80
## fitArima3EURCAD  4 -42709.58
```

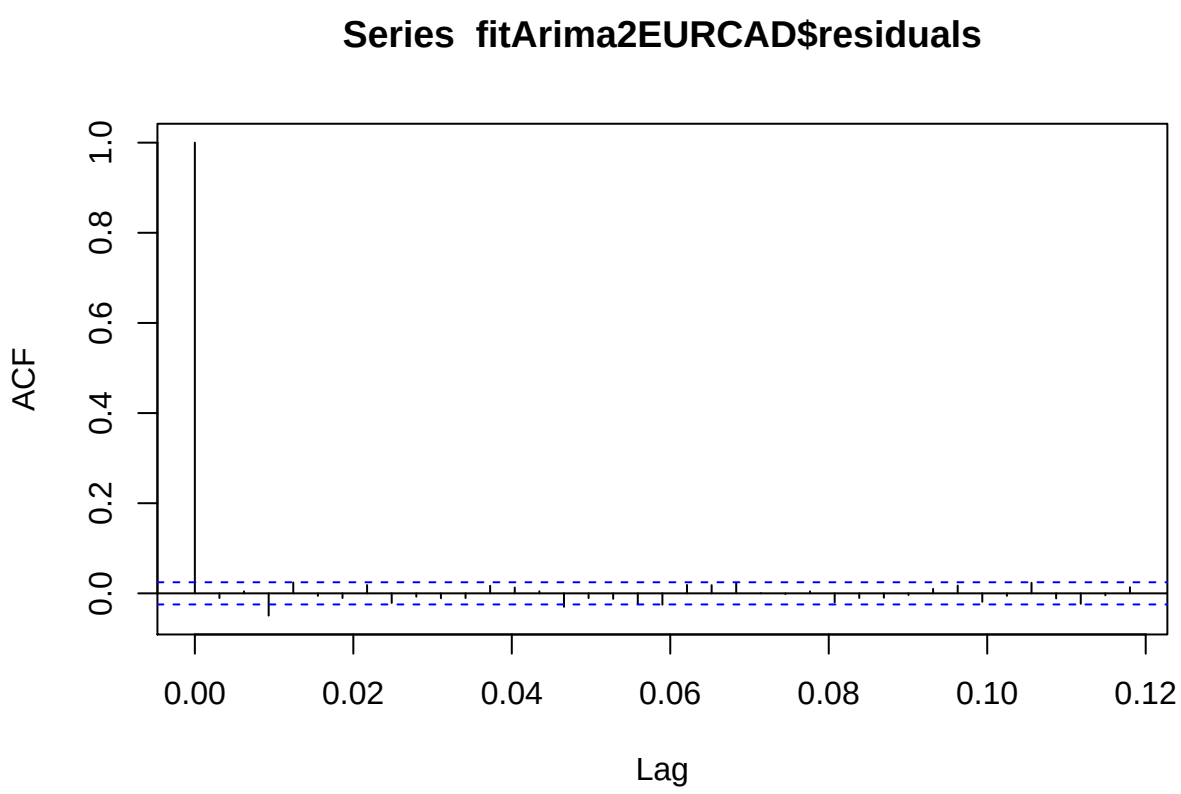
Base on the above the fitArima1CanJap is selected

```
confint(fitArima2EURCAD)
```

```
##      2.5 % 97.5 %
```

Runing code to obtain Box Test Rest

```
acf(fitArima2EURCAD$residuals)
```




```
library(FitAR)
```

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

```
## Loading required package: lattice
```

```
## Loading required package: leaps
```

```
## Loading required package: ltsa
```

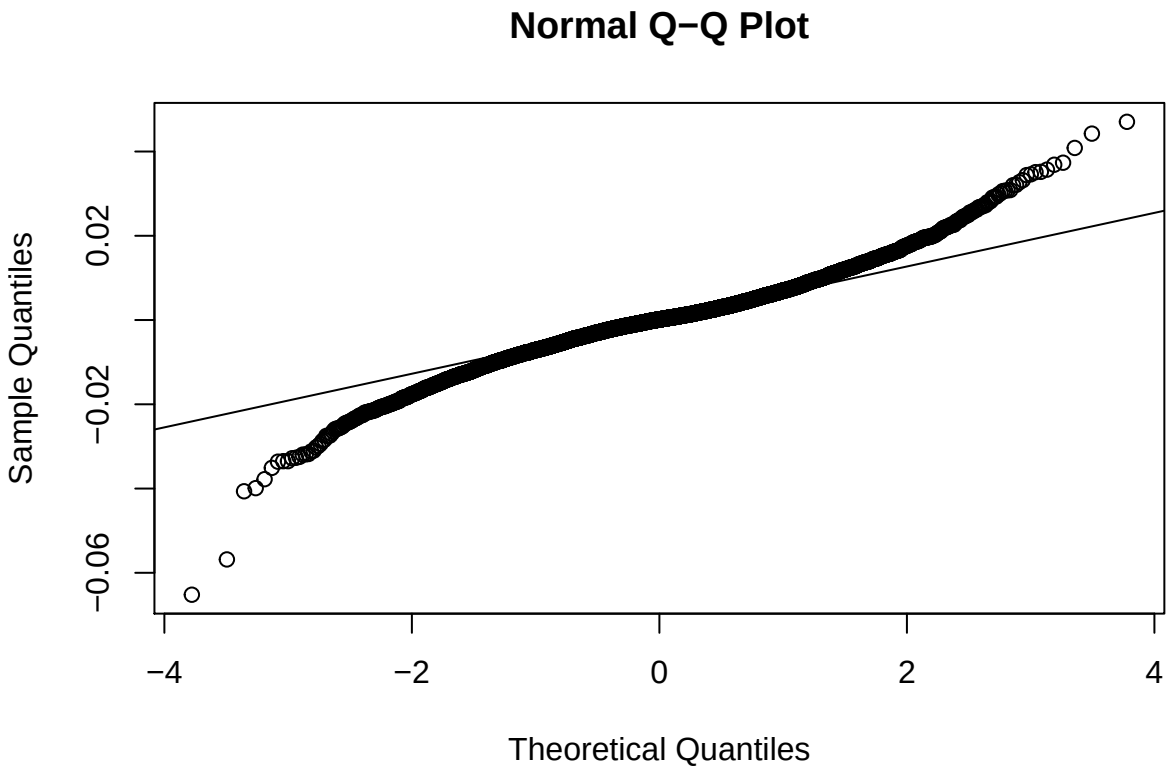
```
## Loading required package: bestglm
```

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

```
library(bestglm)
 Box.test(resid(fitArima2EURCAD),type="Ljung",lag=20,fitdf=1)
```

```
##
##  Box-Ljung test
##
## data:  resid(fitArima2EURCAD)
## X-squared = 47.346, df = 19, p-value = 0.0003186
```

```
qqnorm(fitArima2EURCAD$residuals)
qqline(fitArima2EURCAD$residuals)
```



Using Auto.arima to find the best model fit

```
library(forecast)
```

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

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:FitAR':
##
##     BoxCox
```

```
## The following object is masked from 'package:dLagM':
##
##     forecast
```

```
auto.arima(EURCADARIMATS, trace=TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2)(1,0,1)[322] with drift         : Inf
##  ARIMA(0,1,0)             with drift         : -42703.22
##  ARIMA(1,1,0)(1,0,0)[322] with drift         : -43003.27
##  ARIMA(0,1,1)(0,0,1)[322] with drift         : Inf
##  ARIMA(0,1,0)                                : -42705.21
##  ARIMA(1,1,0)             with drift         : -42704.27
##  ARIMA(1,1,0)(2,0,0)[322] with drift         : Inf
##  ARIMA(1,1,0)(1,0,1)[322] with drift         : Inf
##  ARIMA(1,1,0)(0,0,1)[322] with drift         : Inf
##  ARIMA(1,1,0)(2,0,1)[322] with drift         : Inf
##  ARIMA(0,1,0)(1,0,0)[322] with drift         : Inf
##  ARIMA(2,1,0)(1,0,0)[322] with drift         : Inf
##  ARIMA(1,1,1)(1,0,0)[322] with drift         : Inf
##  ARIMA(0,1,1)(1,0,0)[322] with drift         : Inf
##  ARIMA(2,1,1)(1,0,0)[322] with drift         : -43004.71
##  ARIMA(2,1,1)             with drift         : Inf
##  ARIMA(2,1,1)(2,0,0)[322] with drift         : Inf
##  ARIMA(2,1,1)(1,0,1)[322] with drift         : Inf
##  ARIMA(2,1,1)(0,0,1)[322] with drift         : Inf
##  ARIMA(2,1,1)(2,0,1)[322] with drift         : Inf
##  ARIMA(3,1,1)(1,0,0)[322] with drift         : Inf
##  ARIMA(2,1,2)(1,0,0)[322] with drift         : Inf
##  ARIMA(1,1,2)(1,0,0)[322] with drift         : Inf
##  ARIMA(3,1,0)(1,0,0)[322] with drift         : Inf
##  ARIMA(3,1,2)(1,0,0)[322] with drift         : Inf
##  ARIMA(2,1,1)(1,0,0)[322]                    : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(2,1,1)(1,0,0)[322] with drift         : Inf
##  ARIMA(1,1,0)(1,0,0)[322] with drift         : Inf
##  ARIMA(0,1,0)                                : -42714.8
##
##  Best model: ARIMA(0,1,0)
```

```
## Series: EURCADARIMATS
## ARIMA(0,1,0)
##
## sigma^2 estimated as 6.845e-05:  log likelihood=21358.4
## AIC=-42714.8   AICc=-42714.8   BIC=-42708.05
```

forecasting using Best model: ARIMA(0,1,0)

```
forecastarimaEURCAD<- predict(fitArima2EURCAD,n.ahead = 100)
forecastarimaEURCAD
```

```
## $pred
## Time Series:
## Start = c(2019, 211)
## End = c(2019, 310)
## Frequency = 322
##   [1] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [10] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [19] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [28] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [37] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [46] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [55] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [64] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [73] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [82] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
##  [91] 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384 1.55384
## [100] 1.55384
##
## $se
## Time Series:
## Start = c(2019, 211)
## End = c(2019, 310)
## Frequency = 322
##   [1] 0.008273537 0.011700548 0.014330186 0.016547073 0.018500190 0.020265943
##   [7] 0.021889720 0.023401095 0.024820610 0.026163220 0.027440216 0.028660371
##  [13] 0.029830660 0.030956739 0.032043269 0.033094146 0.034112665 0.035101643
##  [19] 0.036063510 0.037000380 0.037914107 0.038806326 0.039678487 0.040531886
##  [25] 0.041367683 0.042186924 0.042990557 0.043779440 0.044554358 0.045316026
##  [31] 0.046065102 0.046802190 0.047527849 0.048242593 0.048946902 0.049641219
##  [37] 0.050325958 0.051001504 0.051668219 0.052326439 0.052976482 0.053618645
##  [43] 0.054253207 0.054880433 0.055500570 0.056113855 0.056720509 0.057320742
##  [49] 0.057914756 0.058502738 0.059084869 0.059661320 0.060232255 0.060797828
##  [55] 0.061358189 0.061913478 0.062463831 0.063009377 0.063550240 0.064086538
##  [61] 0.064618386 0.065145892 0.065669160 0.066188292 0.066703384 0.067214528
##  [67] 0.067721815 0.068225330 0.068725156 0.069221373 0.069714058 0.070203285
##  [73] 0.070689127 0.071171652 0.071650928 0.072127019 0.072599988 0.073069896
##  [79] 0.073536801 0.074000760 0.074461829 0.074920060 0.075375505 0.075828215
##  [85] 0.076278238 0.076725621 0.077170411 0.077612652 0.078052387 0.078489659
##  [91] 0.078924508 0.079356974 0.079787097 0.080214913 0.080640459 0.081063771
##  [97] 0.081484885 0.081903833 0.082320649 0.082735365
```

```
par(mfrow = c(1,1))
```