ARIMA Model GBP And CAD

Jane

28/04/2021

Forcasting Exchange Rate Using ARIMA Model for Bristish Pound And canadian Dollar

Reading GBP 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
```

```
GBPCADARIMA <-  read.csv ("GBPCAD_Candlestick_1_D_BID_01.01.2000-31.12.2020.csv")%>%
  select('GMT.TIME', CLOSE)%>%
  rename(Date = ('GMT.TIME'), RateGBPCAD = ("CLOSE"))


head(GBPCADARIMA)
```

```
##         Date RateGBPCAD
## 1 2000-01-03     2.3675
## 2 2000-01-04     2.3778
## 3 2000-01-05     2.3822
## 4 2000-01-06     2.4037
## 5 2000-01-07     2.3867
## 6 2000-01-10     2.3835
```

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

```
##         Date RateGBPCAD
## 1 2000-01-03     2.3675
## 2 2000-01-04     2.3778
## 3 2000-01-05     2.3822
## 4 2000-01-06     2.4037
## 5 2000-01-07     2.3867
## 6 2000-01-10     2.3835
```

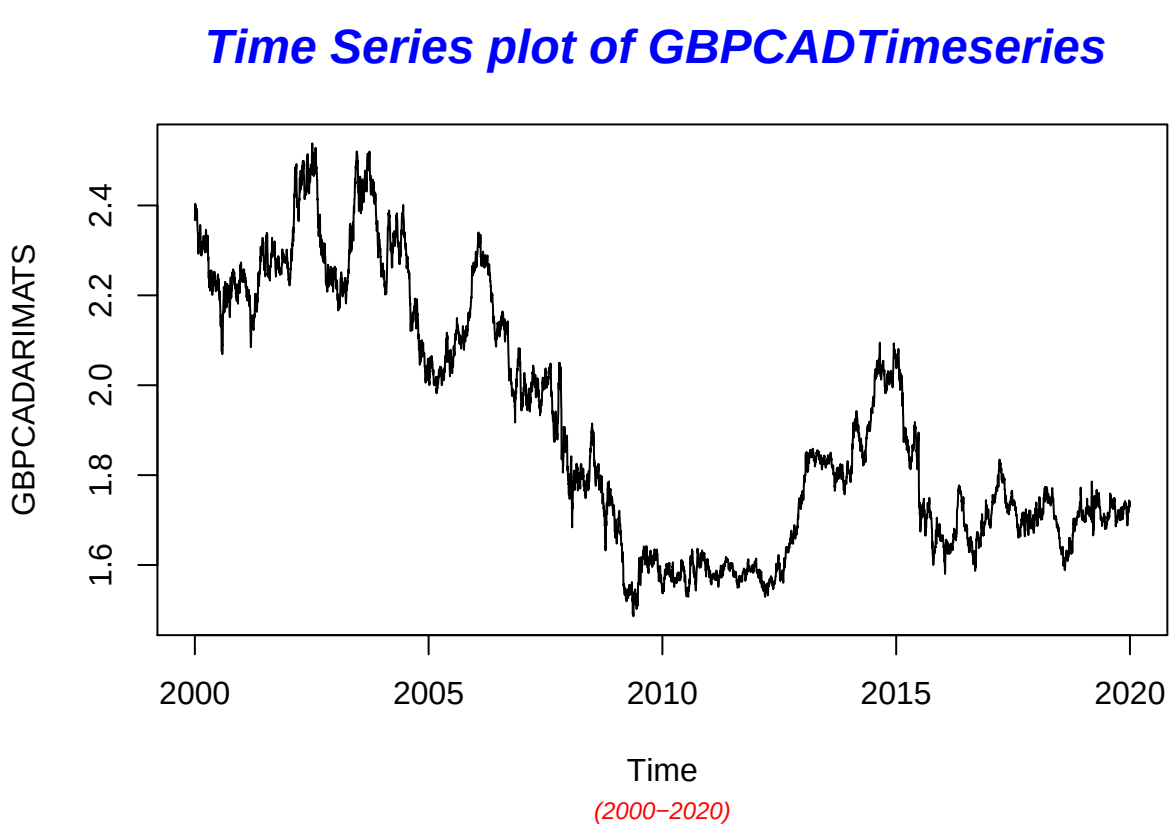
##Checking for obvious errors or missingg value

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

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

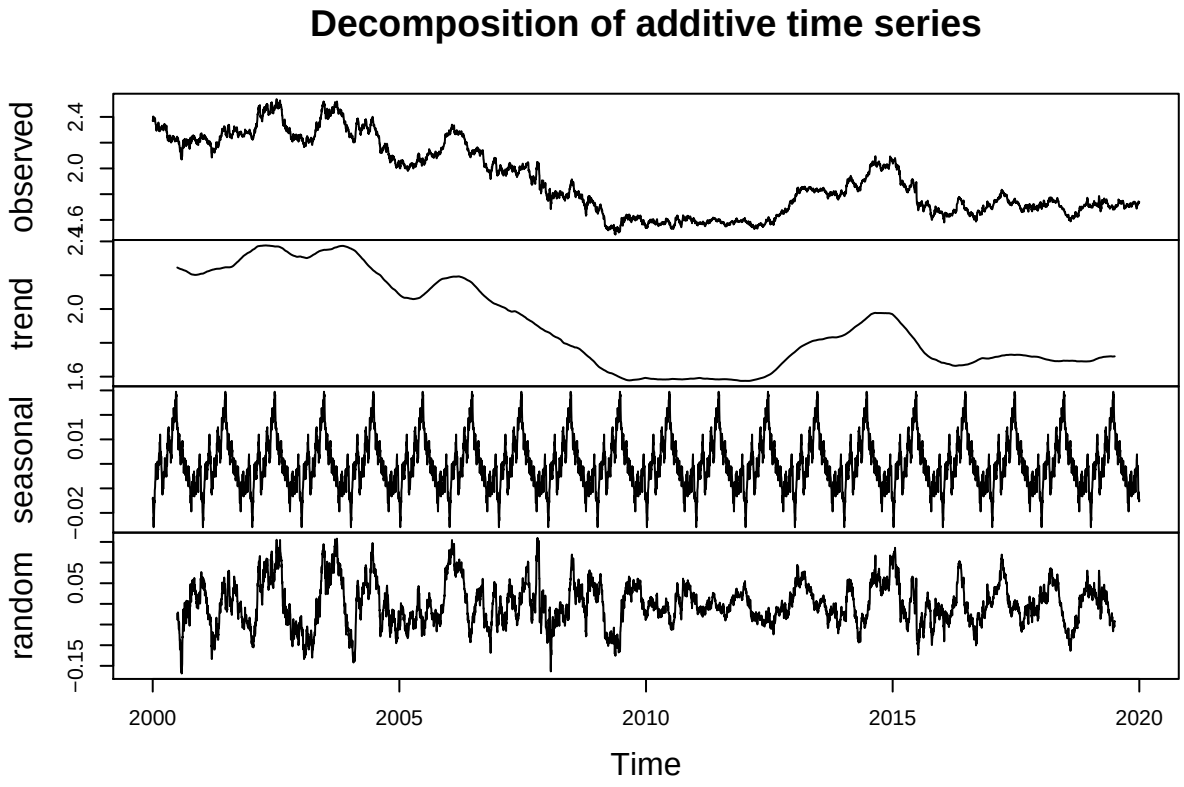
##Converting the data set into time series object

```r
#Converting the data set into time series object
GBPCADARIMATS<- ts(as.vector(GBPCADARIMA$Rate),  frequency = 313, start= c(2000,01,03))
plot.ts(GBPCADARIMATS)
title("Time Series plot of GBPCADTimeseries ", sub = "(2000-2020)",
      cex.main = 1.5,   font.main= 4, col.main= "blue",
      cex.sub = 0.75, font.sub = 3, col.sub = "red")
```



Finding the component of the Time Series

```
ComponentGBPCAD <- decompose(GBPCADARIMATS)
plot(ComponentGBPCAD)
```



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
```

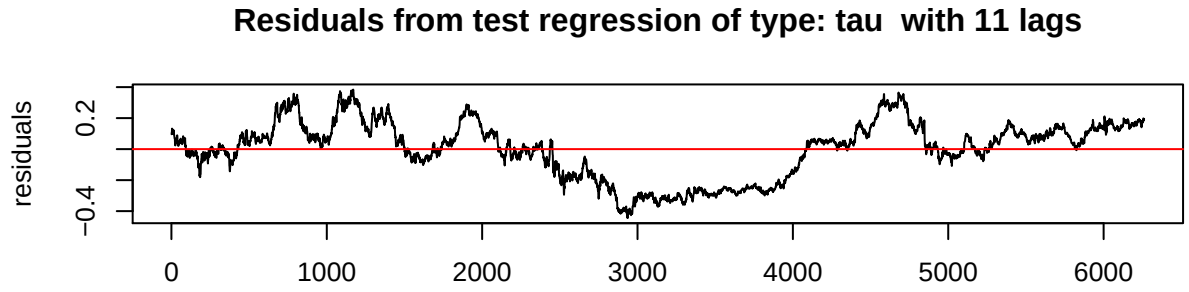
```
## Loading required package: timeSeries
```

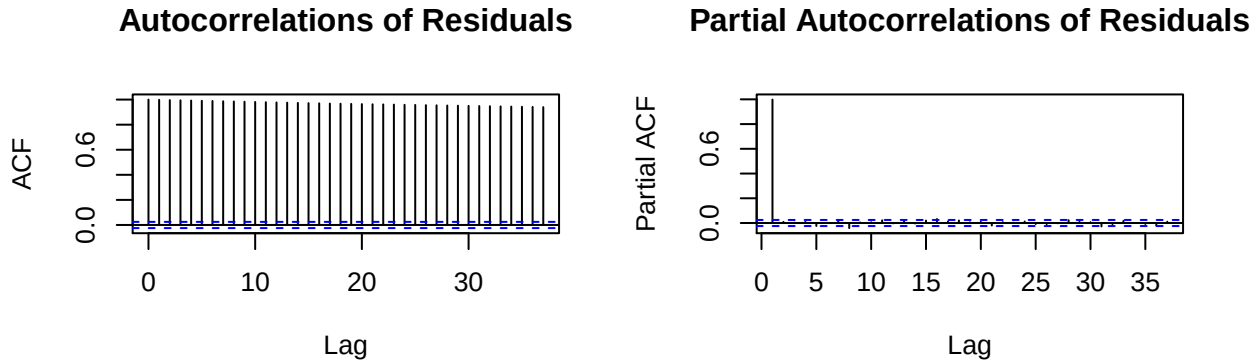
```
## Warning: package 'timeSeries' was built under R version 4.0.5
```

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

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

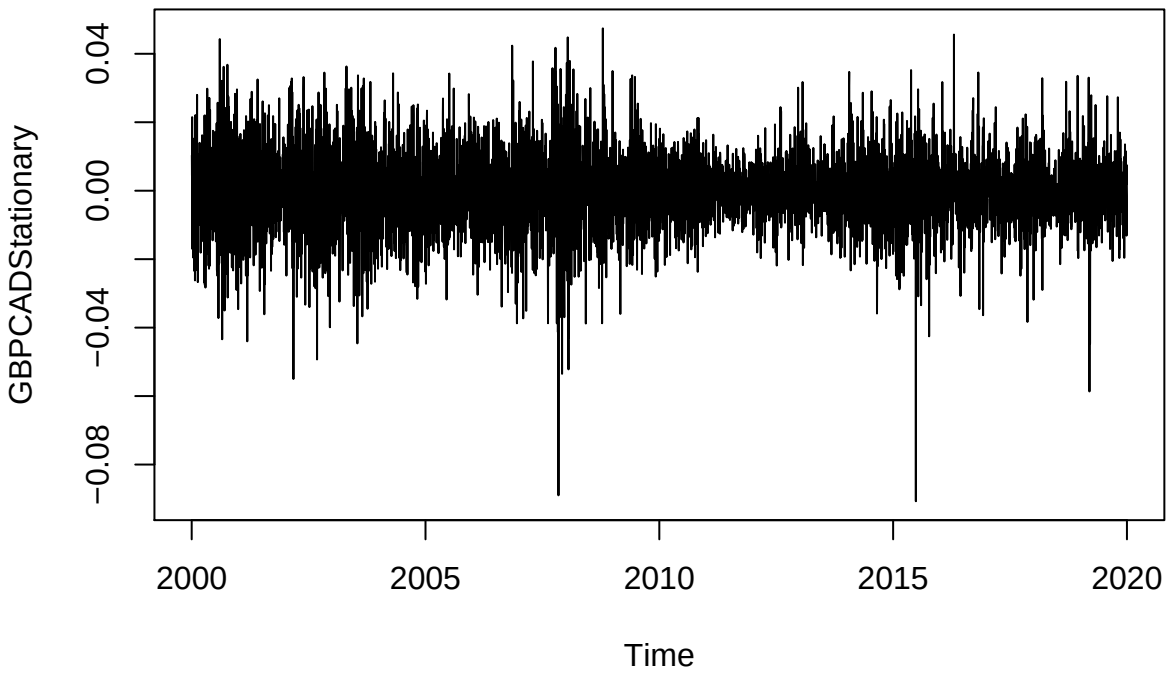
```
urkpssTest(GBPCADARIMATS, type = c("tau"), lags = c("short"),use.lag = NULL, doplot = TRUE)
```





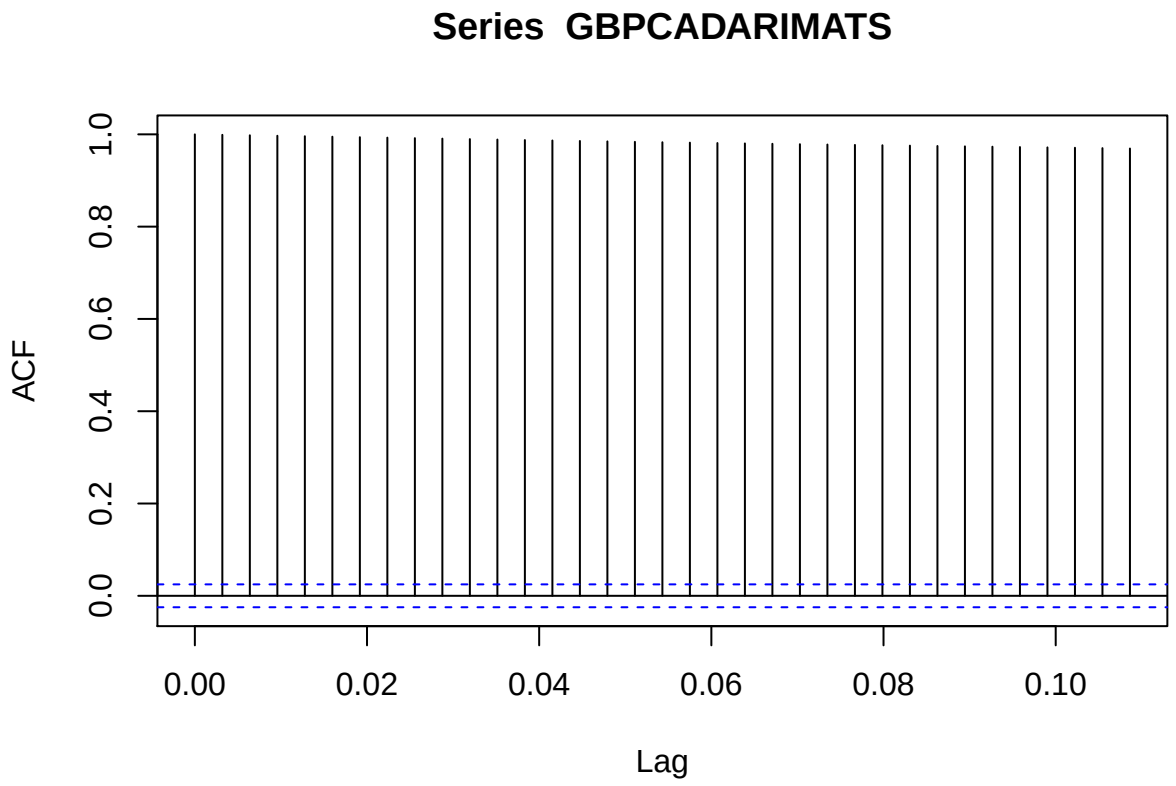

```
##
## Title:
##  KPSS Unit Root Test
##
## Test Results:
##   NA
##
## Description:
##  Tue May 04 00:27:45 2021 by user: janeo
```

```
GBPCADStationary= diff(GBPCADARIMATS, differences=1)
plot(GBPCADStationary)
```

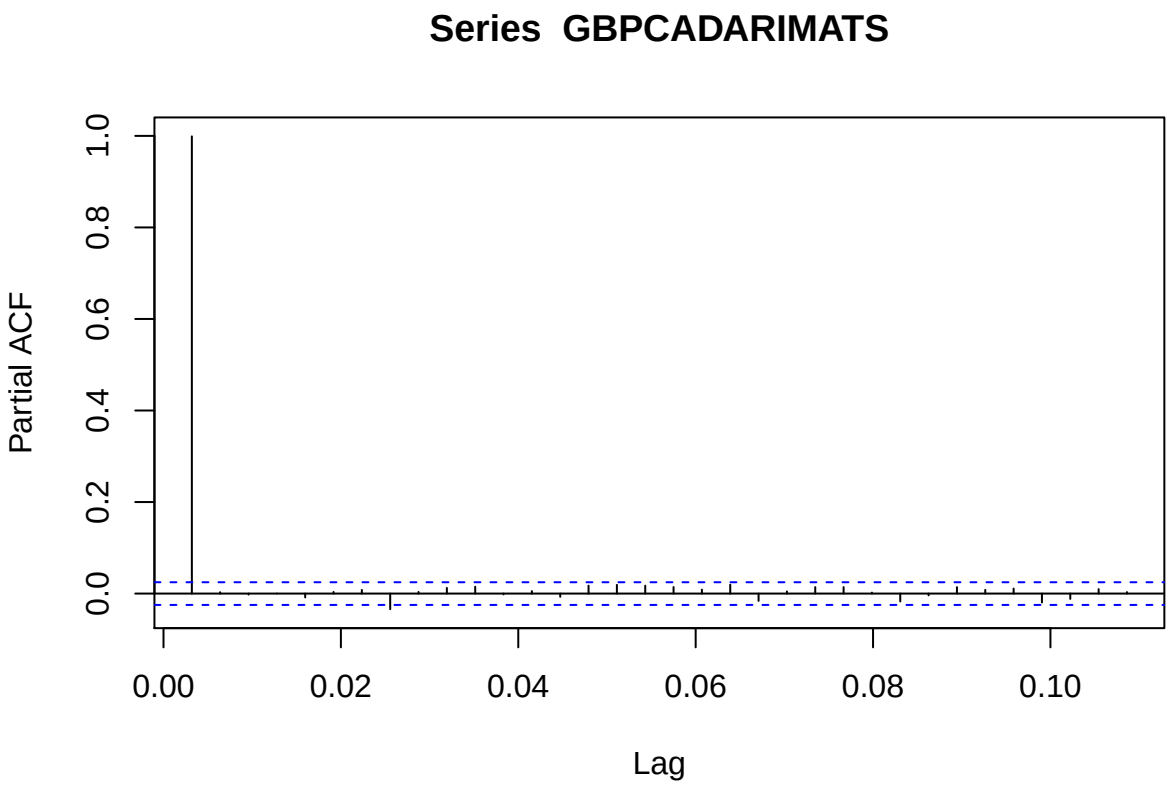


Calculating Autocorrlation function and partil autocorlation function

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

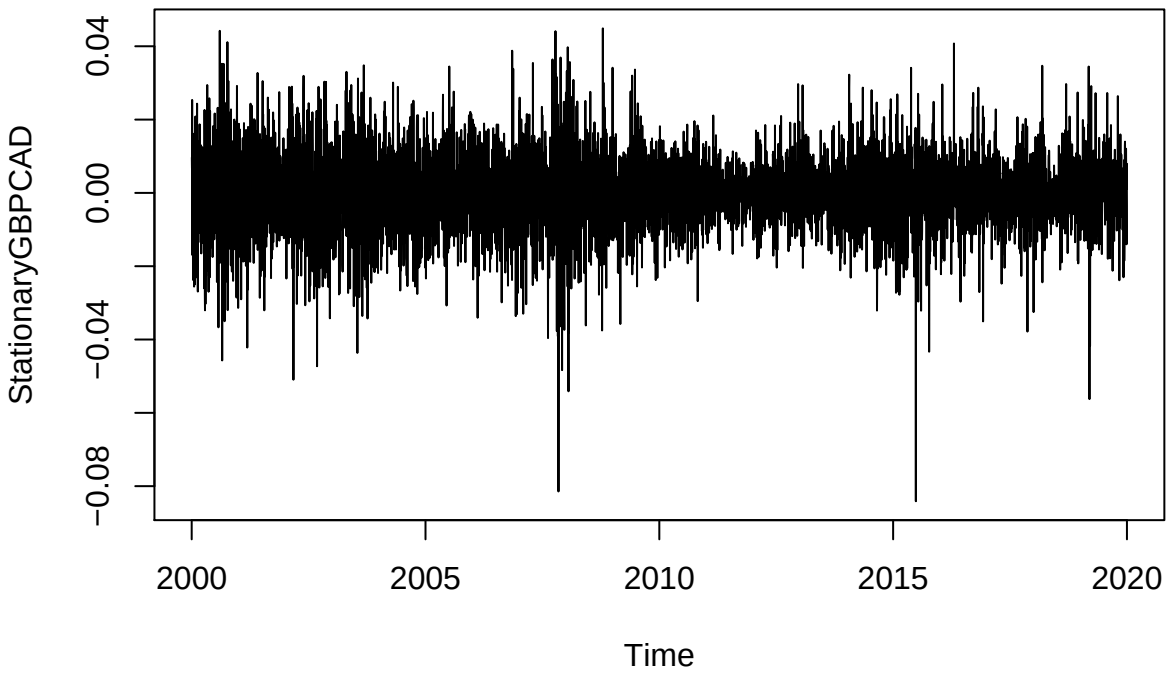



```
pacf(GBPCADARIMATS, lag.max = 34)
```



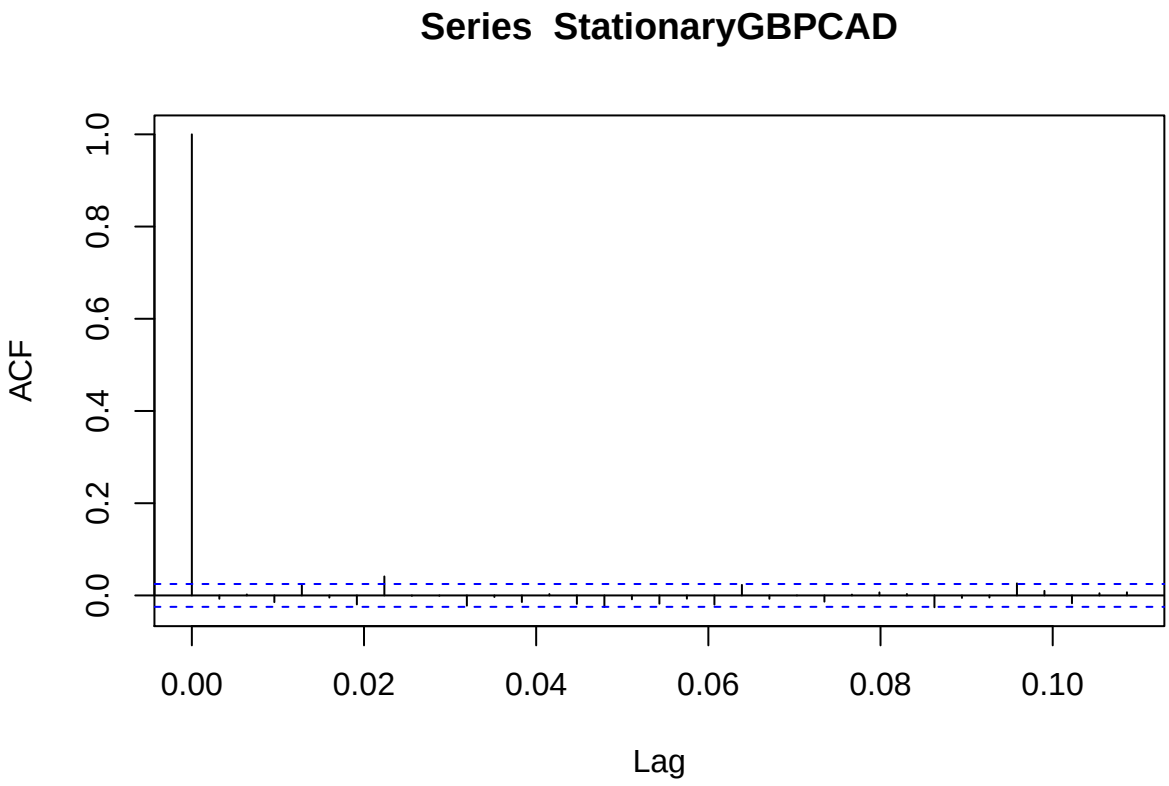
Adjusting and ensuring there are no seasonality

```
TSseasonallyadjustedGBPCAD <- GBPCADARIMATS- ComponentGBPCAD$seasonal
StationaryGBPCAD<- diff(TSseasonallyadjustedGBPCAD, differences=1)
plot(StationaryGBPCAD)
```

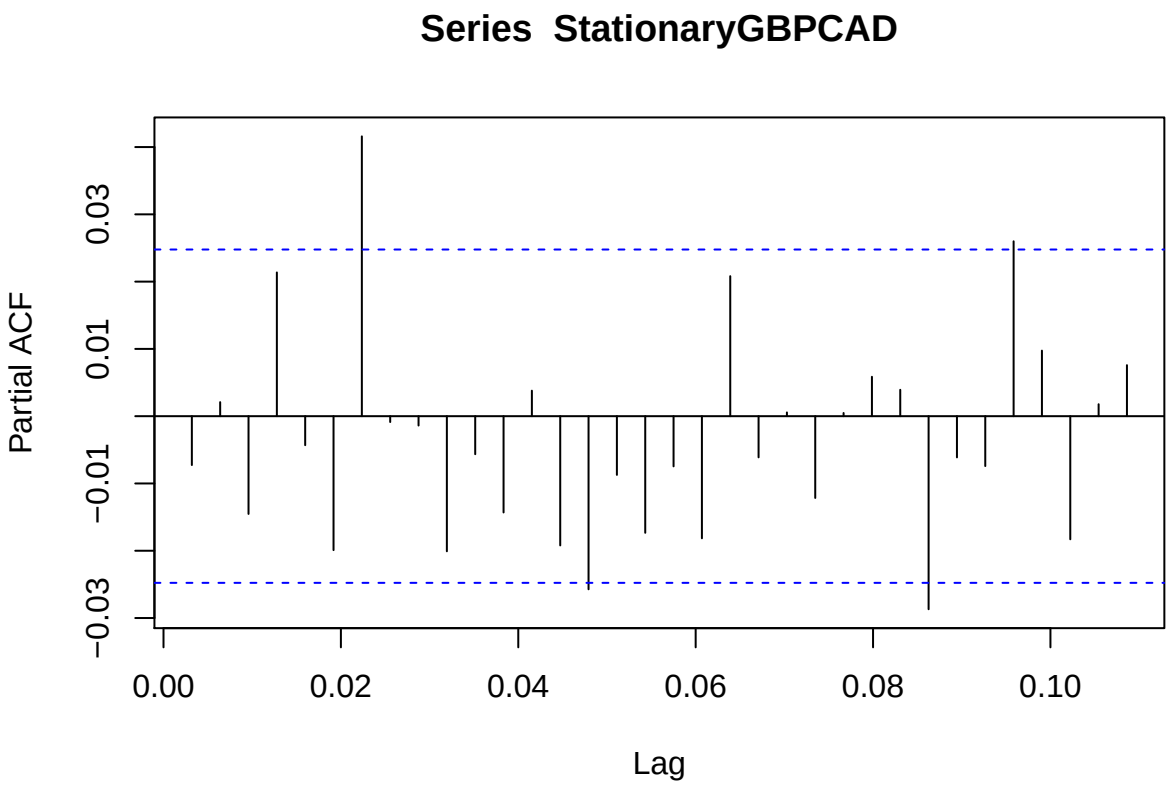


Calculating again for ACF and PACF after finding stationality

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




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



Fitting The ARIMA Model

ARIMA fitting (1,1,0)

```
fitArima1GBPCAD <- arima(GBPCADARIMATS, order =  c(1,0,0), include.mean = TRUE)
fitArima1GBPCAD
```

```
##
## Call:
## arima(x = GBPCADARIMATS, order = c(1, 0, 0), include.mean = TRUE)
##
## Coefficients:
##          ar1  intercept
##       0.9993     1.9085
## s.e.  0.0004     0.1659
##
## sigma^2 estimated as 0.0001177:  log likelihood = 19436.28,  aic = -38866.56
```

##Arima Fitting (0,1,0)

```
fitArima2GBPCAD <- arima(GBPCADARIMATS, order =  c(0,1,0), include.mean = TRUE)
fitArima2GBPCAD
```

```
##
## Call:
## arima(x = GBPCADARIMATS, order = c(0, 1, 0), include.mean = TRUE)
##
##
## sigma^2 estimated as 0.0001177:  log likelihood = 19435.48,  aic = -38868.96
```

Arima Fitting (2,1,1)

```
fitArima3GBPCAD <- arima(GBPCADARIMATS, order = c(2,1,1), include.mean = TRUE)
fitArima3GBPCAD
```

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

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

##           ar1      ar2     ma1
##       -0.0055  -0.0023  -0.006
## s.e.      NaN   0.0121     NaN
##
## sigma^2 estimated as 0.0001177:  log likelihood = 19435.92,  aic = -38863.84
```

##Fitting Arima (0,1,3)

```
fitArima4GBPCAD <- arima(GBPCADARIMATS, order = c(3,1,0), include.mean = TRUE)
fitArima4GBPCAD
```

```
##
## Call:
## arima(x = GBPCADARIMATS, order = c(3, 1, 0), include.mean = TRUE)
##
## Coefficients:
##           ar1      ar2      ar3
##       -0.0117  -0.0025  -0.0147
## s.e.   0.0126   0.0126   0.0126
##
## sigma^2 estimated as 0.0001177:  log likelihood = 19436.59,  aic = -38865.19
```

##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
```

```
ARIMAModelSelectionGBPCAD = AIC(fitArima1GBPCAD,fitArima2GBPCAD,fitArima3GBPCAD,fitArima4GBPCAD)
```

```
## Warning in AIC.default(fitArima1GBPCAD, fitArima2GBPCAD, fitArima3GBPCAD, :
## models are not all fitted to the same number of observations
```

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

```
##                 df       AIC
## fitArima2GBPCAD  1 -38868.96
## fitArima1GBPCAD  3 -38866.56
## fitArima4GBPCAD  4 -38865.19
## fitArima3GBPCAD  4 -38863.84
```

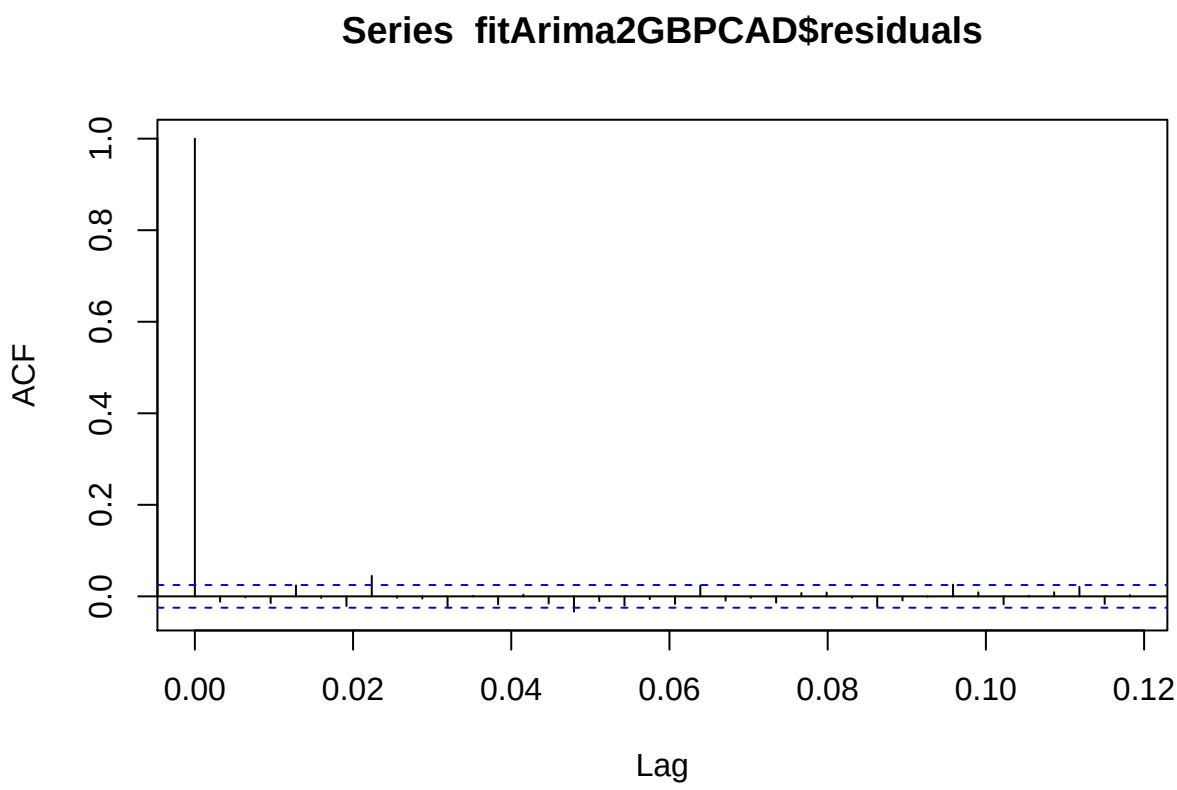
Base on the above the fitArima1CanJap is selected

```
confint(fitArima2GBPCAD)
```

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

Runing code to obtain Box Test Rest

```
acf(fitArima2GBPCAD$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
```

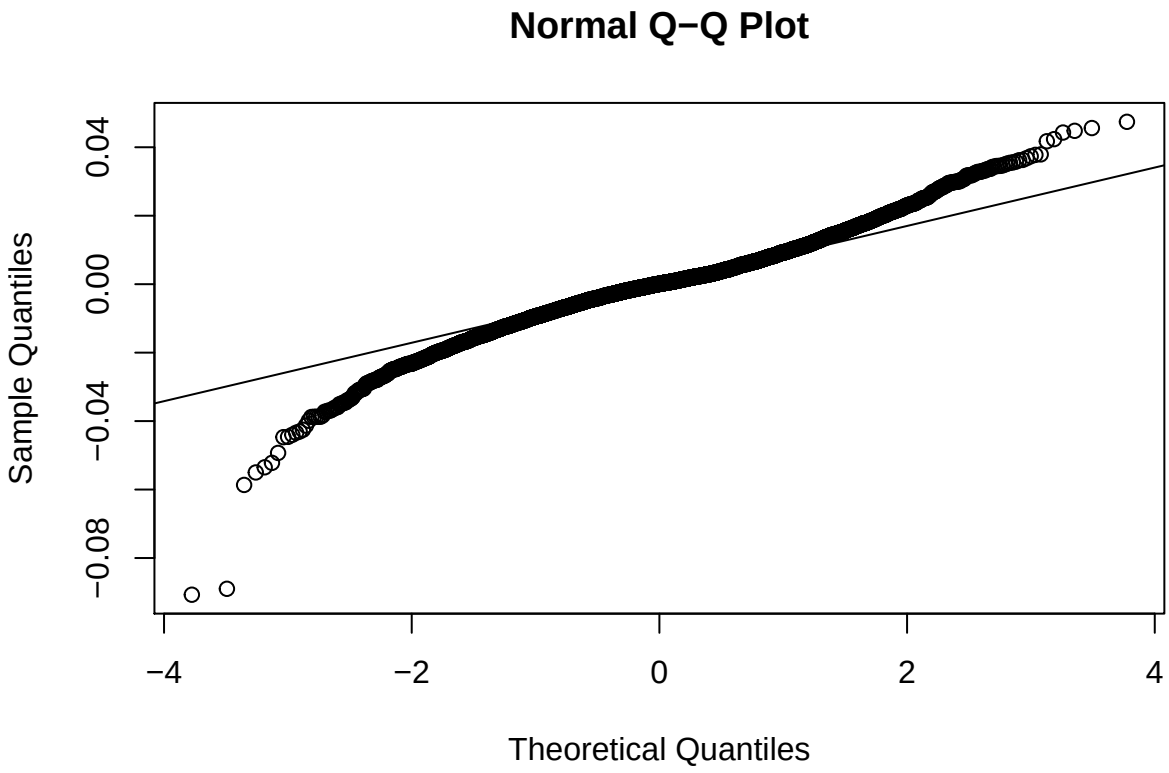
```
## Loading required package: bestglm
```

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

```
library(bestglm)
 Box.test(resid(fitArima2GBPCAD),type="Ljung",lag=20,fitdf=1)
```

```
##
##  Box-Ljung test
##
## data:  resid(fitArima2GBPCAD)
## X-squared = 43.625, df = 19, p-value = 0.001064
```

```
qqnorm(fitArima2GBPCAD$residuals)
qqline(fitArima2GBPCAD$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(GBPCADARIMATS, trace=TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2)(1,0,1)[313] with drift         : Inf
##  ARIMA(0,1,0)             with drift         : -38858.45
##  ARIMA(1,1,0)(1,0,0)[313] with drift         : Inf
##  ARIMA(0,1,1)(0,0,1)[313] with drift         : Inf
##  ARIMA(0,1,0)                                : -38859.91
##  ARIMA(0,1,0)(1,0,0)[313] with drift         : Inf
##  ARIMA(0,1,0)(0,0,1)[313] with drift         : Inf
##  ARIMA(0,1,0)(1,0,1)[313] with drift         : Inf
##  ARIMA(1,1,0)             with drift         : -38857.22
##  ARIMA(0,1,1)             with drift         : -38857.31
##  ARIMA(1,1,1)             with drift         : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(0,1,0)                                : -38868.96
##
##  Best model: ARIMA(0,1,0)
```

```
## Series: GBPCADARIMATS
## ARIMA(0,1,0)
##
## sigma^2 estimated as 0.0001177:  log likelihood=19435.48
## AIC=-38868.96   AICc=-38868.96   BIC=-38862.22
```

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

```
forecastarimaGBPCAD<- predict(fitArima2GBPCAD,n.ahead = 100)
forecastarimaGBPCAD
```

```
## $pred
## Time Series:
## Start = c(2020, 2)
## End = c(2020, 101)
## Frequency = 313
##   [1] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [10] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [19] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [28] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [37] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [46] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [55] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [64] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
```

```
##  [73] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [82] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
##  [91] 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964 1.73964
## [100] 1.73964
##
## $se
## Time Series:
## Start = c(2020, 2)
## End = c(2020, 101)
## Frequency = 313
##   [1] 0.01084937 0.01534333 0.01879167 0.02169875 0.02425994 0.02657543
##   [7] 0.02870475 0.03068667 0.03254812 0.03430873 0.03598330 0.03758334
##  [13] 0.03911798 0.04059464 0.04201945 0.04339750 0.04473312 0.04603000
##  [19] 0.04729133 0.04851988 0.04971808 0.05088808 0.05203177 0.05315086
##  [25] 0.05424687 0.05532117 0.05637500 0.05740949 0.05842567 0.05942447
##  [31] 0.06040676 0.06137333 0.06232491 0.06326218 0.06418577 0.06509625
##  [37] 0.06599417 0.06688004 0.06775432 0.06861747 0.06946989 0.07031198
##  [43] 0.07114411 0.07196661 0.07277982 0.07358404 0.07437956 0.07516667
##  [49] 0.07594562 0.07671666 0.07748003 0.07823595 0.07898464 0.07972630
##  [55] 0.08046112 0.08118928 0.08191098 0.08262637 0.08333563 0.08403889
##  [61] 0.08473632 0.08542806 0.08611424 0.08679500 0.08747045 0.08814074
##  [67] 0.08880596 0.08946623 0.09012167 0.09077238 0.09141845 0.09206000
##  [73] 0.09269710 0.09332985 0.09395834 0.09458265 0.09520288 0.09581908
##  [79] 0.09643135 0.09703976 0.09764437 0.09824527 0.09884251 0.09943616
##  [85] 0.10002629 0.10061296 0.10119623 0.10177615 0.10235279 0.10292620
##  [91] 0.10349644 0.10406354 0.10462758 0.10518859 0.10574662 0.10630173
##  [97] 0.10685395 0.10740333 0.10794991 0.10849375
```

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