---

author: "Audrey Ekuban" date: "8 May 2016" output: word_document ---

```
setwd("C:/Users/Audrey/Data_Science_Workshop_Capstone/Data")

#library(quantmod)

EURGBP = read.csv("EURGBP.csv", colClasses=c(rep("character",
2),rep("numeric",4)))
datetime <- as.POSIXct(paste(EURGBP$DATE, EURGBP$TIME), format = "%Y%m%d
%H%M%OS")
EURGBP = EURGBP[c(3:6)]

summary(EURGBP)
```

```
##       OPEN             HIGH             LOW             CLOSE
##  Min.   :0.5954   Min.   :0.5954   Min.   :0.5954   Min.   :0.5954
##  1st Qu.:0.6811   1st Qu.:0.6811   1st Qu.:0.6810   1st Qu.:0.6811
##  Median :0.7541   Median :0.7542   Median :0.7541   Median :0.7541
##  Mean   :0.7585   Mean   :0.7585   Mean   :0.7584   Mean   :0.7585
##  3rd Qu.:0.8392   3rd Qu.:0.8392   3rd Qu.:0.8391   3rd Qu.:0.8392
##  Max.   :0.9801   Max.   :0.9801   Max.   :0.9796   Max.   :0.9800
```

```
library(xts)
```

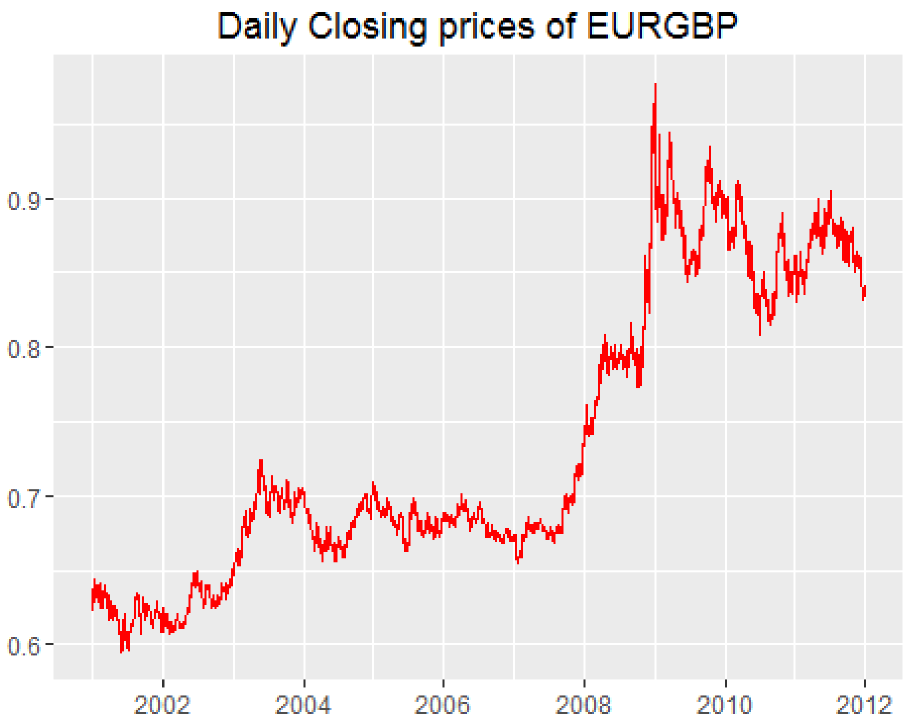
```
## Warning: package 'xts' was built under R version 3.2.5

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.2.4

##
## Attaching package: 'zoo'

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

```
EURGBP.xt <- xts(x = EURGBP, order.by = datetime)
head(EURGBP.xt)
```

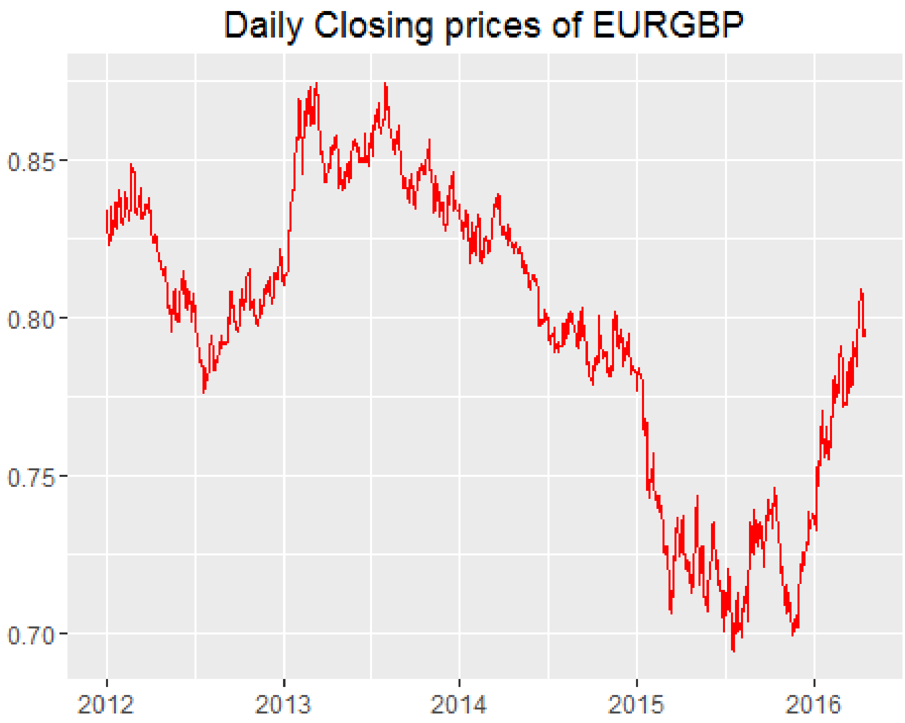
```
##                         OPEN   HIGH    LOW  CLOSE
## 2001-01-03 00:01:00 0.6328 0.6328 0.6328 0.6328
## 2001-01-03 00:02:00 0.6328 0.6328 0.6327 0.6327
## 2001-01-03 00:03:00 0.6327 0.6327 0.6327 0.6327
## 2001-01-03 00:04:00 0.6327 0.6328 0.6327 0.6328
## 2001-01-03 00:05:00 0.6328 0.6328 0.6328 0.6328
## 2001-01-03 00:06:00 0.6327 0.6327 0.6327 0.6327
```

```
#chartSeries(EURGBP.xt, up.col='green',dn.col='red')
#chartSeries(to.daily(EURGBP.xt),up.col='green',dn.col='red')
```

```r
#chartSeries(to.weekly(EURGBP.xt),up.col='green',dn.col='red')
#chartSeries(to.monthly(EURGBP.xt),up.col='green',dn.col='red')
#chartSeries(to.quarterly(EURGBP.xt),up.col='green',dn.col='red')
#chartSeries(to.yearly(EURGBP.xt),up.col='green',dn.col='red')

#seriesLo(EURGBP.xt)
#seriesHi(EURGBP.xt)

# Testing
EURGBP.testing = EURGBP.xt['2012/2016']
#seriesLo(EURGBP.testing)
#seriesHi(EURGBP.testing)

# Training
EURGBP.training = EURGBP.xt['/2011']
#seriesLo(EURGBP.training)
#seriesHi(EURGBP.training)

library(forecast)

## Warning: package 'forecast' was built under R version 3.2.5

## Loading required package: timeDate

## This is forecast 7.1

library(ggfortify)

## Loading required package: proto

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.2.5

x = to.daily(EURGBP.training)[,4]
xtest = to.daily(EURGBP.testing)[,4]

# STEP 1
autoplot(x, ts.colour = 'red', main='Daily Closing prices of EURGBP')
```

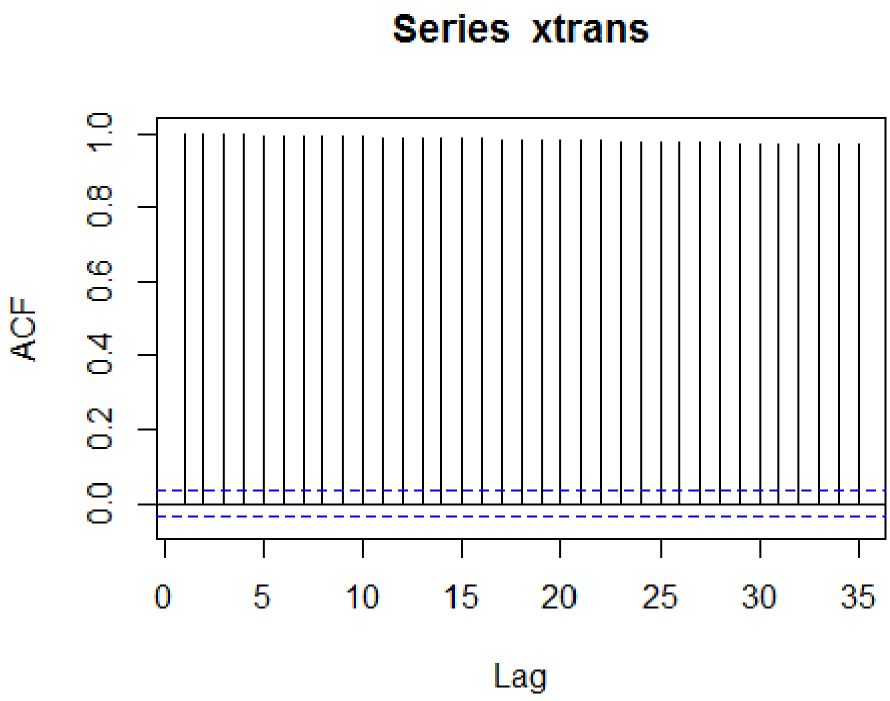


```
autoplot(xtest, ts.colour = 'red', main='Daily Closing prices of EURGBP')
```

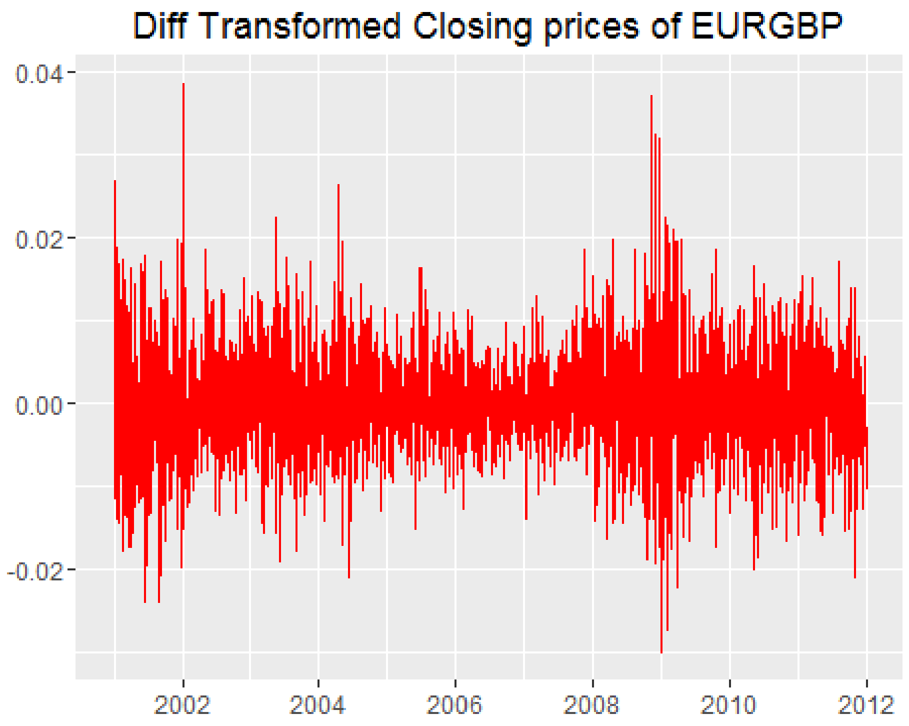


```
lambda = forecast::BoxCox.lambda(x, method="loglik")
xtrans = forecast::BoxCox(x, lambda)
```

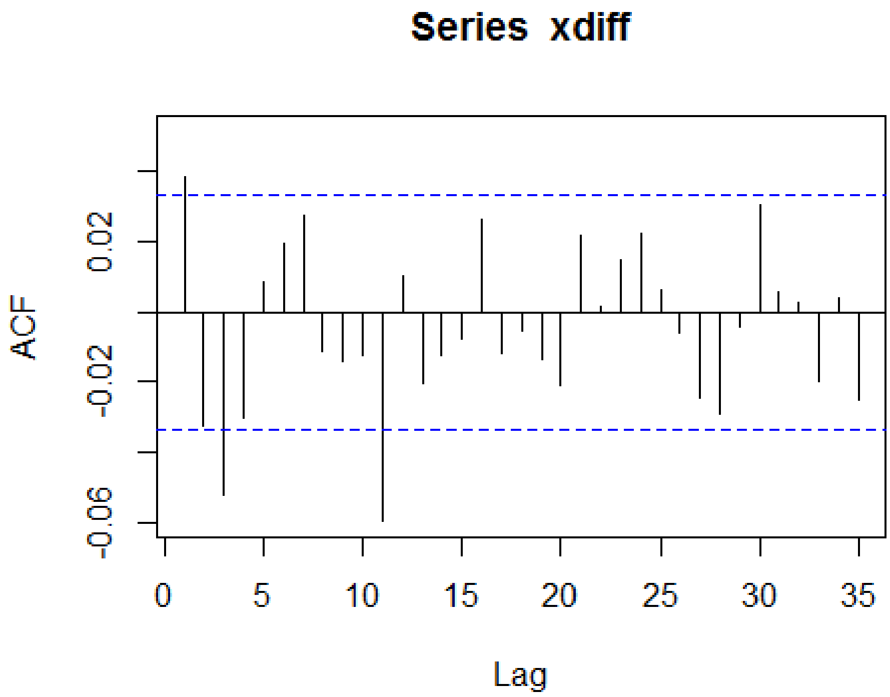
```
xtranstest = forecast::BoxCox(xtest, lambda)

Acf(xtrans)
```



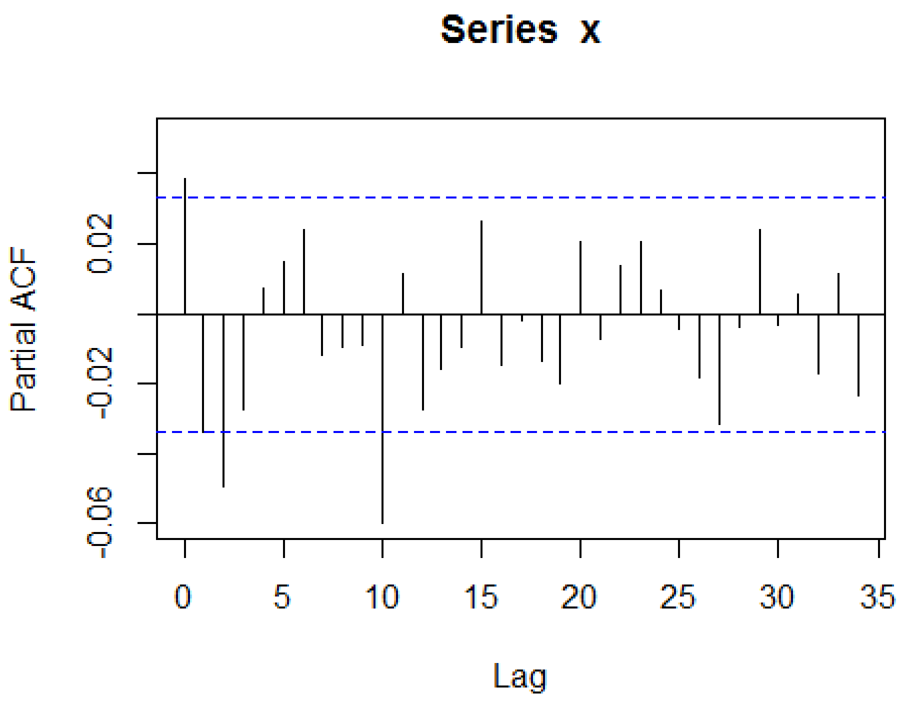
```
tseries::adf.test(xtrans, k = 0)

##
##  Augmented Dickey-Fuller Test
##
## data:  xtrans
## Dickey-Fuller = -2.4005, Lag order = 0, p-value = 0.4088
## alternative hypothesis: stationary

ndiffs(xtrans)

## [1] 1

# nsdiffs(xtrans)

# STEP 2
xdiff = diff(xtrans)
autoplot(xdiff, ts.colour = 'red', main='Diff Transformed Closing prices of 
EURGBP')

## Warning: Removed 1 rows containing missing values (geom_path).
```



```
Acf(xdiff)
```



```
Pacf(xdiff)
```



```
tseries::adf.test(xdiff[-1,], k=0)
```

```
## Warning in tseries::adf.test(xdiff[-1, ], k = 0): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  xdiff[-1, ]
## Dickey-Fuller = -56.437, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

```
summary(auto.arima(xdiff[-1,], stationary=TRUE))
```

```
## Series: xdiff[-1, ]
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.0158  -0.6589  -0.9841  0.5967
## s.e.  0.1950   0.1370   0.2085  0.1476
##
## sigma^2 estimated as 3.81e-05:  log likelihood=12578.14
## AIC=-25146.27   AICc=-25146.25   BIC=-25115.57
##
## Training set error measures:
##                        ME        RMSE         MAE MPE MAPE      MASE
## Training set 0.0001245453 0.006169261 0.004458355 NaN  Inf 0.6913796
##                      ACF1
## Training set 0.004395102
```

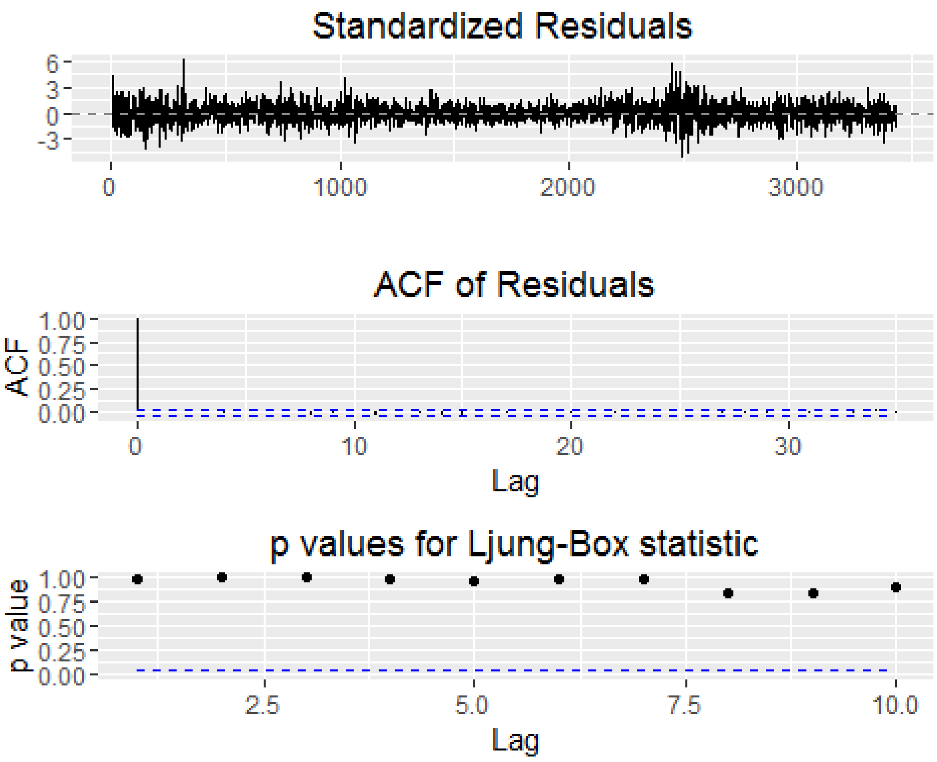
```r
summary(Arima(xtrans, c(2,1,2)))
```

```
## Series: xtrans 
## ARIMA(2,1,2) 
## 
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.0158  -0.6589  -0.9841  0.5967
## s.e.  0.1950   0.1370   0.2085  0.1476
## 
## sigma^2 estimated as 3.81e-05:  log likelihood=12578.14
## AIC=-25146.27   AICc=-25146.25   BIC=-25115.57
## 
## Training set error measures:
##                        ME       RMSE         MAE        MPE     MAPE
## Training set 0.0001243319 0.00616837 0.004457232 -0.1005911 1.792451
##                   MASE        ACF1
## Training set 0.999585 0.004245334
```

```r
summary(Arima(xtrans, c(2,1,5)))
```

```
## Series: xtrans 
## ARIMA(2,1,5) 
## 
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3     ma4      ma5
##       1.3502  -0.9782  -1.3158  0.9013  0.0287  0.0192  -0.0291
## s.e.  0.0151   0.0136   0.0229  0.0314  0.0328  0.0289   0.0174
## 
## sigma^2 estimated as 3.803e-05:  log likelihood=12582.99
## AIC=-25149.97   AICc=-25149.93   BIC=-25100.86
## 
## Training set error measures:
##                        ME        RMSE         MAE         MPE     MAPE
## Training set 0.0001230457 0.006159517 0.004453085 -0.09785779 1.786207
##                    MASE          ACF1
## Training set 0.9986549 -0.0006372632
```

```r
# Model
xtrans.fit <- Arima(xtrans, c(2,1,5))
ggtsdiag(xtrans.fit)
```

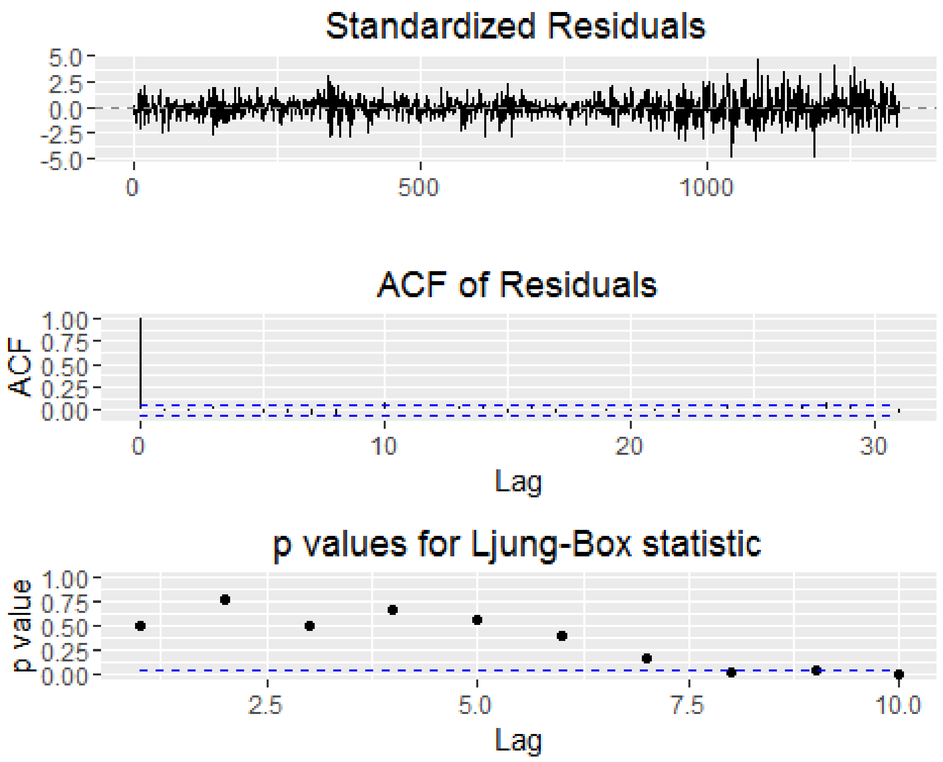


Standardized Residuals

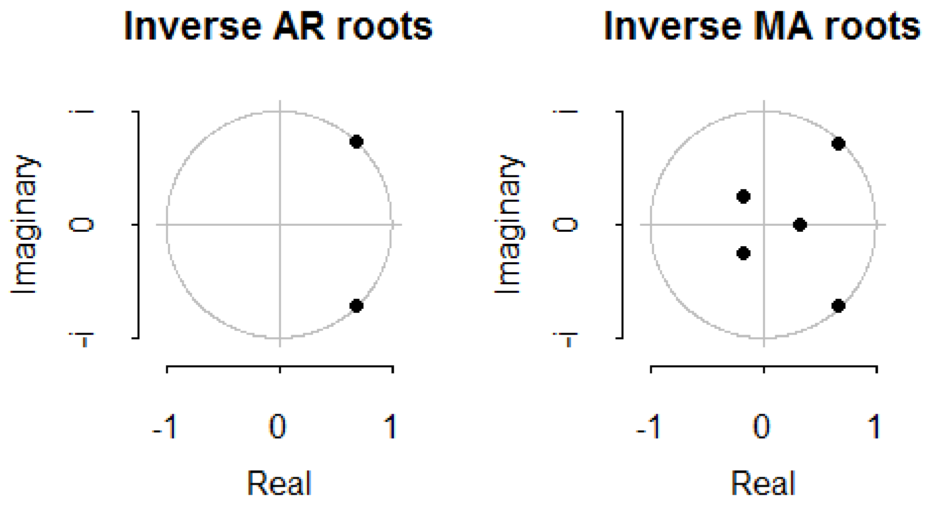
ACF of Residuals

p values for Ljung-Box statistic

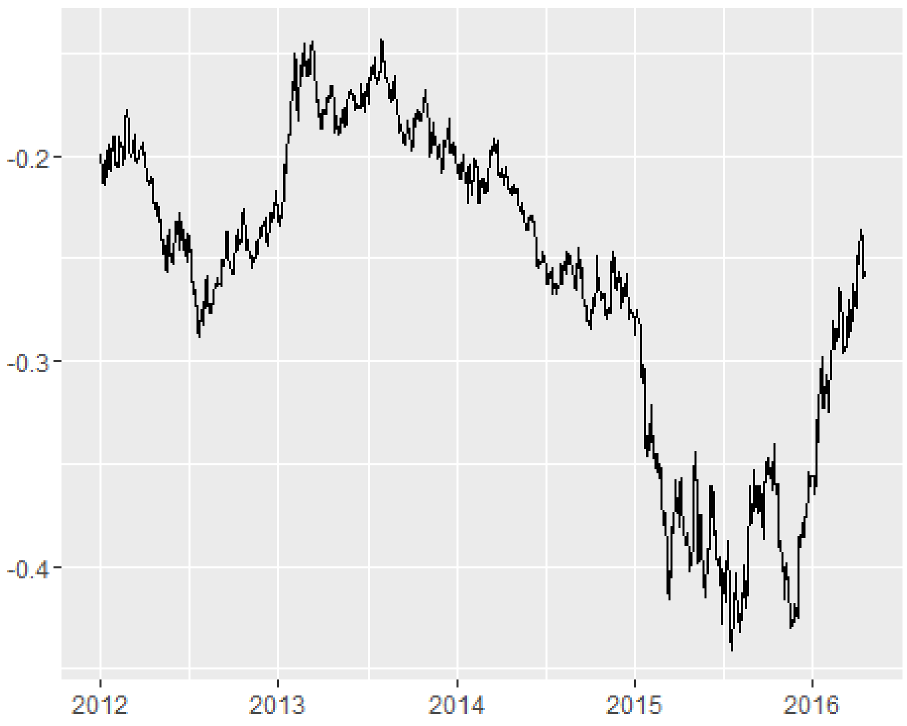
```
# Apply model to test
xtranstest = xtranstest[-1336]
xtranstest.fit = Arima(xtranstest, model=xtrans.fit)
ggtsdiag(xtranstest.fit)
```



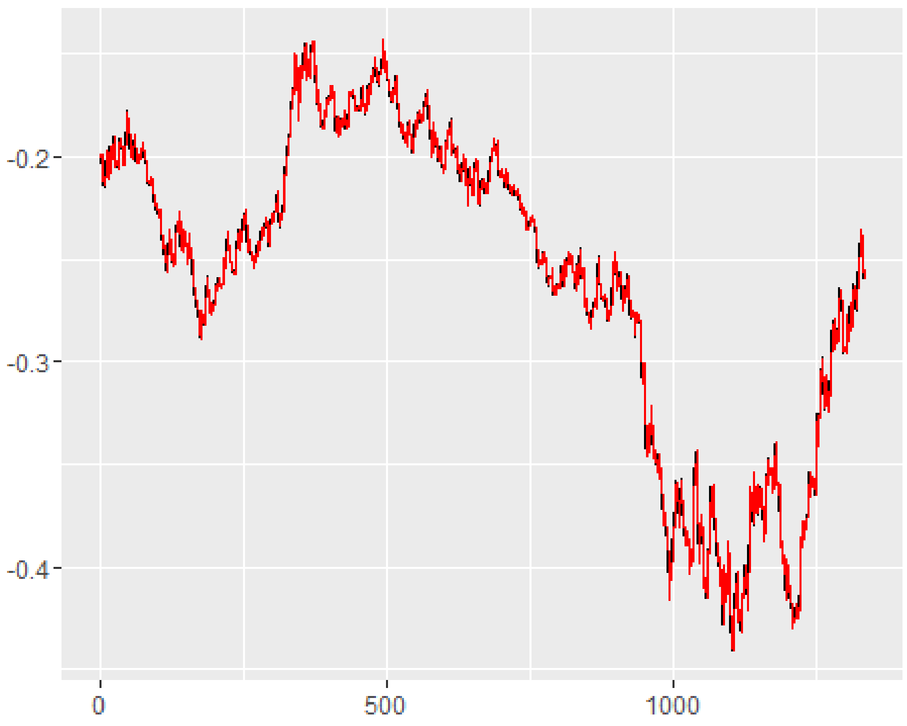
```r
plot.Arima(xtranstest.fit)
```



```r
autoplot(xtranstest)
```



```
autoplot(xtranstest.fit)
```


-0.2
-0.3
-0.4
0
500
1000