Problem4

Arnav, Ushnik, Harshitha

November 23, 2016

Data

Loading data & exploration The first thing that we do is analyse the time series data by reading it into R, and plotting the data over time.

```
exch <- read.csv("C:/Users/Ushnik/Desktop/ExchangeRates.csv", header=TRUE, stringsAsFactors=FALS
E)
View(exch)
names(exch)
```

```
## [1] "Date"            "EUR.to.USD.Rate" "TUR.to.USD.Rate"
```

```
str(exch)
```

```
## 'data.frame':    37 obs. of  3 variables:
##  $ Date           : chr  "11/30/2013" "12/31/2013" "1/31/2014" "2/28/2014" ...
##  $ EUR.to.USD.Rate: num  1.36 1.37 1.36 1.37 1.38 ...
##  $ TUR.to.USD.Rate: num  0.495 0.484 0.45 0.452 0.452 ...
```

```
head(exch)
```

```
##         Date EUR.to.USD.Rate TUR.to.USD.Rate
## 1 11/30/2013        1.357467       0.4951586
## 2 12/31/2013        1.370800       0.4841815
## 3  1/31/2014        1.361795       0.4501760
## 4  2/28/2014        1.366521       0.4524497
## 5  3/31/2014        1.382819       0.4518929
## 6  4/30/2014        1.381018       0.4705613
```

```
summary(exch)
```

```
##      Date           EUR.to.USD.Rate TUR.to.USD.Rate
##  Length:37          Min.   :1.073   Min.   :0.3154
##  Class :character   1st Qu.:1.109   1st Qu.:0.3421
##  Mode  :character   Median :1.123   Median :0.3770
##                     Mean   :1.195   Mean   :0.3948
##                     3rd Qu.:1.332   3rd Qu.:0.4514
##                     Max.   :1.383   Max.   :0.4952
```

```
Date<-as.Date(exch$Date, "%m/%d/%Y")
exch$Date<-Date
str(exch)
```

```
## 'data.frame':    37 obs. of  3 variables:
##  $ Date           : Date, format: "2013-11-30" "2013-12-31" ...
##  $ EUR.to.USD.Rate: num  1.36 1.37 1.36 1.37 1.38 ...
##  $ TUR.to.USD.Rate: num  0.495 0.484 0.45 0.452 0.452 ...
```

```
View(exch)
```

The data set has 37 monthly observations of Exchange Rates of the Eur to the Dollar andd the Lira to the Dollar.

Time Series Objects

We then store both the exchage rates in two time series objects in R. To store the data in a time series object, we use the `ts()` function in R.

```
ts.eur.usd<-ts(exch$EUR.to.USD.Rate,  start=c(2013,11), frequency = 12)
ts.eur.usd
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 2013                                                              
## 2014 1.361795 1.366521 1.382819 1.381018 1.373852 1.359486 1.353336
## 2015 1.161490 1.135026 1.081945 1.082205 1.116735 1.122559 1.099741
## 2016 1.085505 1.109230 1.113352 1.135341 1.129572 1.119571 1.107224
##           Aug      Sep      Oct      Nov      Dec
## 2013                            1.357467 1.370800
## 2014 1.331524 1.288910 1.267732 1.247250 1.232919
## 2015 1.113614 1.122852 1.122795 1.072658 1.088868
## 2016 1.121782 1.121268 1.098850 1.078400
```

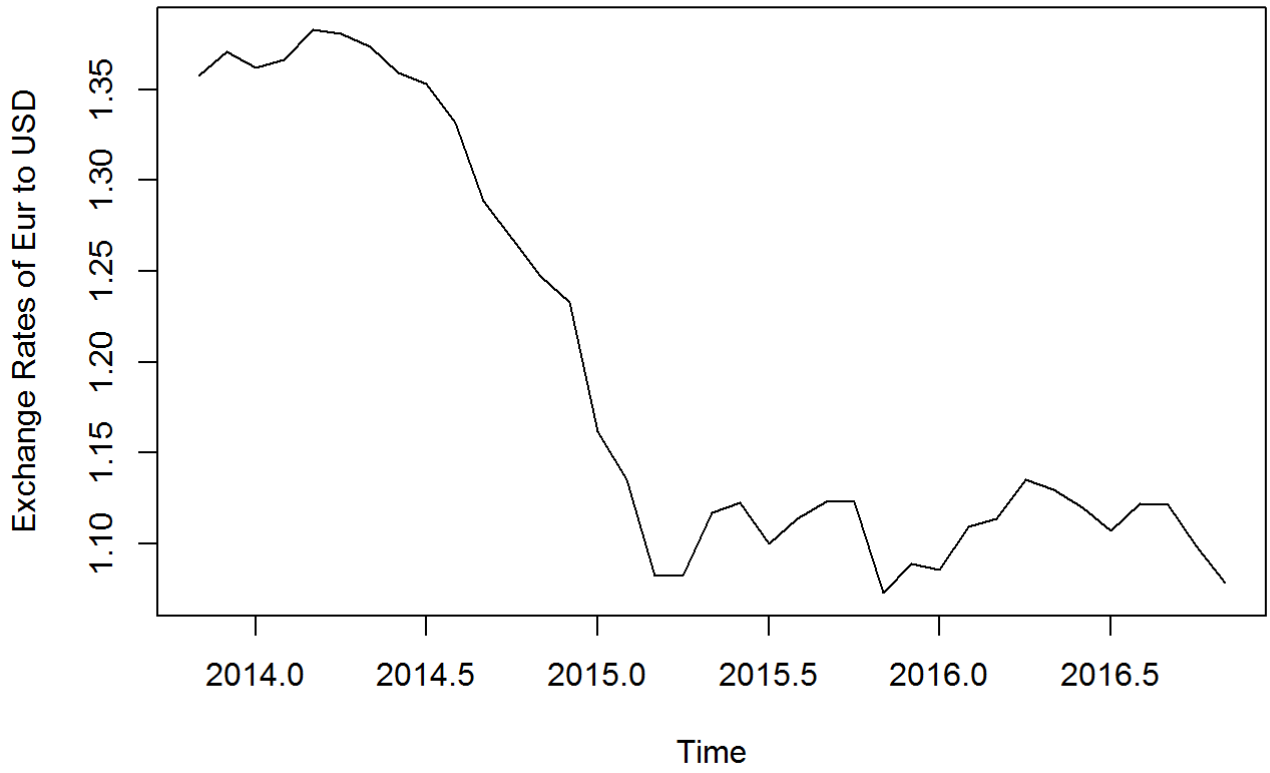
```
ts.tur.usd<-ts(exch$TUR.to.USD.Rate, start=c(2013,11), frequency = 12)
ts.tur.usd
```

```
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 2013                                                                      
## 2014 0.4501760 0.4524497 0.4518929 0.4705613 0.4783469 0.4718146 0.4719974
## 2015 0.4280985 0.4055817 0.3860062 0.3770126 0.3782378 0.3708134 0.3703657
## 2016 0.3322003 0.3397053 0.3473874 0.3528754 0.3402682 0.3436571 0.3364906
##            Aug       Sep       Oct       Nov       Dec
## 2013                                 0.4951586 0.4841815
## 2014 0.4629299 0.4513886 0.4441832 0.4476329 0.4357259
## 2015 0.3502898 0.3315438 0.3420681 0.3480257 0.3424327
## 2016 0.3377845 0.3370873 0.3249257 0.3154117
```

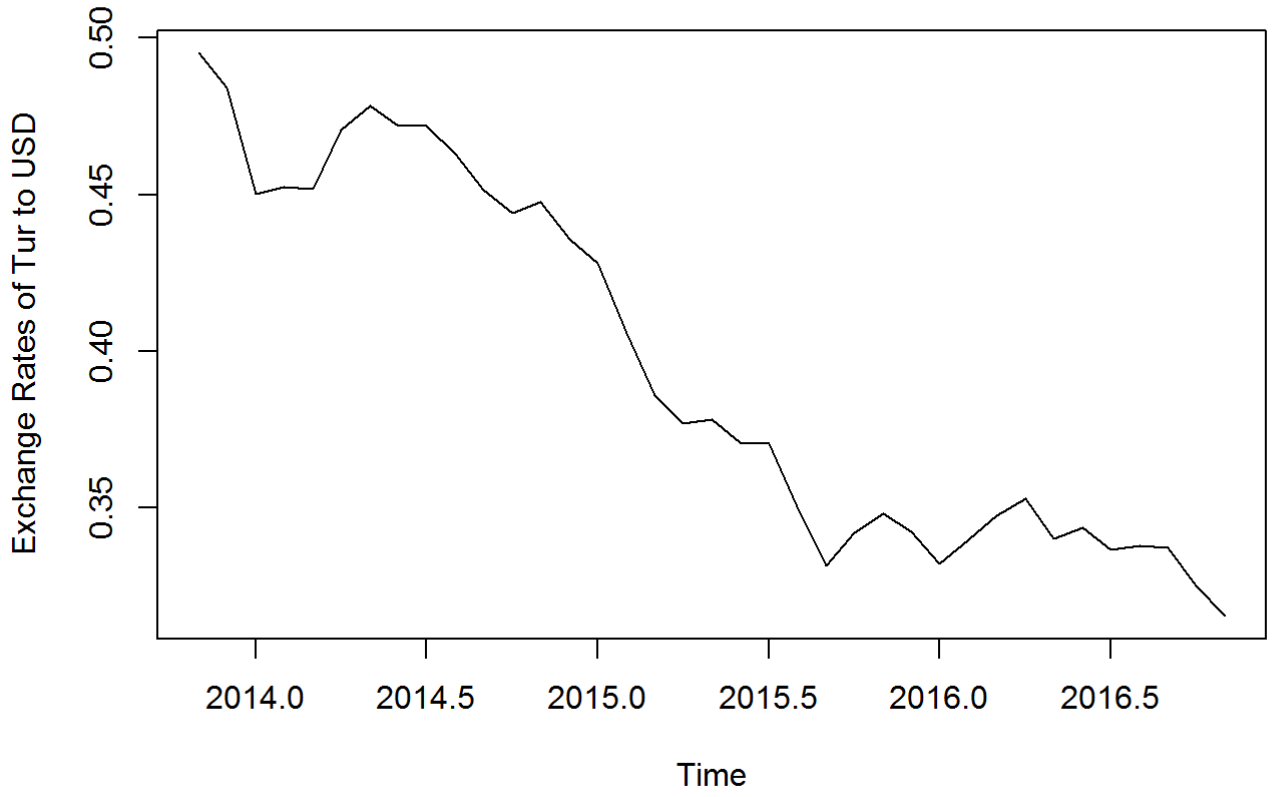
Time Series Plot

We then plot the exchange prices of Euro and Lira from 2013 to 2016.

```
plot.ts(ts.eur.usd, ylab= "Exchange Rates of Eur to USD")
```



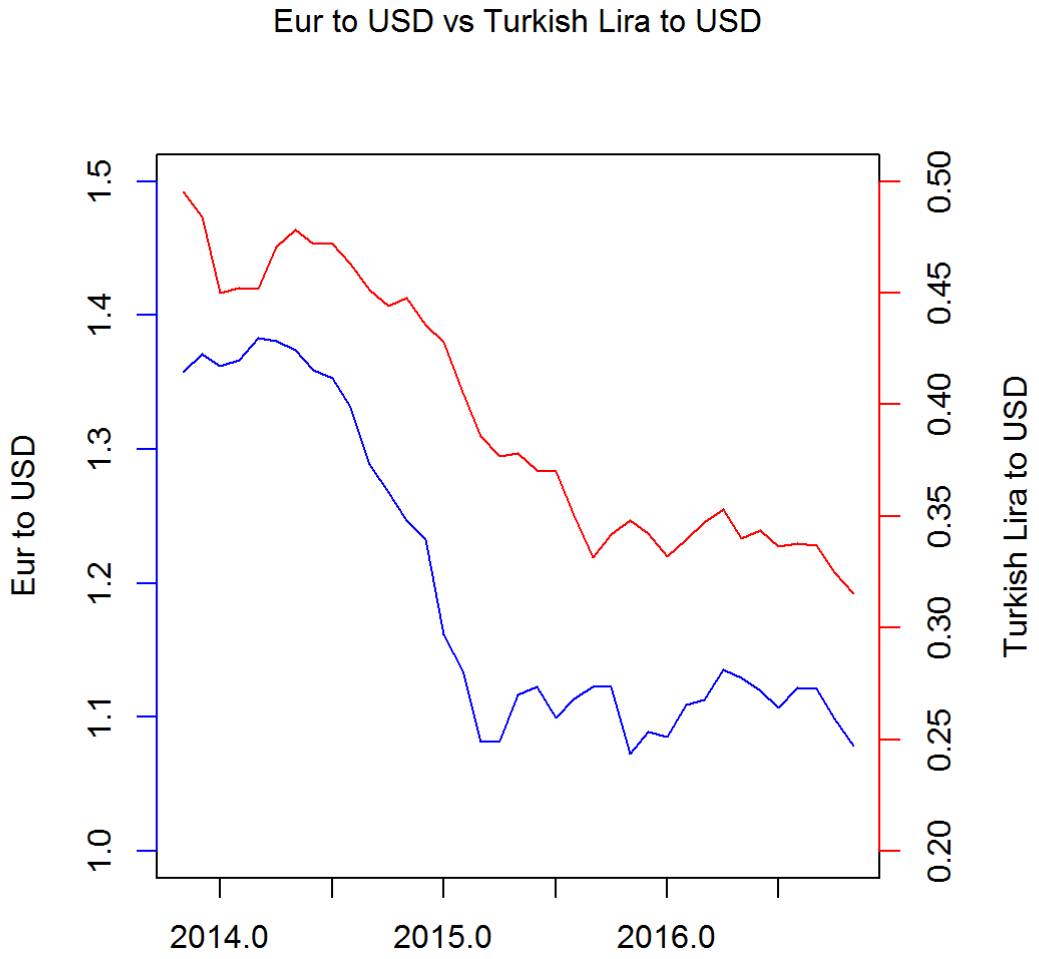

```
plot.ts(ts.tur.usd, ylab= "Exchange Rates of Tur to USD")
```



Consolidated Graphical representation of Euro to USD and Lira to USD

We compare the graphical trends of both the cuurrencies againt the Dollar in a consolidted graph.

```
par(mar=c(2,8,4,8)+0.1)
plot(ts.eur.usd, col="blue", ylim=c(1,1.5), axes=F, ylab="")
box()
axis(2, col="blue")
mtext("Eur to USD", side=2, line=3)
par(new=T)
plot(ts.tur.usd, col="red", ylim=c(0.2,0.5), axes=F, ylab="")
axis(4, col="red")
mtext("Turkish Lira to USD", side=4, line=3)
axis(1, xlim=c(2013, 2016))
mtext("Eur to USD vs Turkish Lira to USD",  line=3)
```



Decomposing, Holt Smoothening and forcasting for Euro

To estimate the trend, seasonal and irregular components of Euro over the given period, we decompose the time series obtained.We then go onto forecast rates for 2017 `(h=12)` which returns values within confidence levels of 80% and 95% respectively for the exchange rate forecasts of the Euro to the Dollar.

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.3.2
```

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

```
## Warning: package 'zoo' was built under R version 3.3.2
```

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

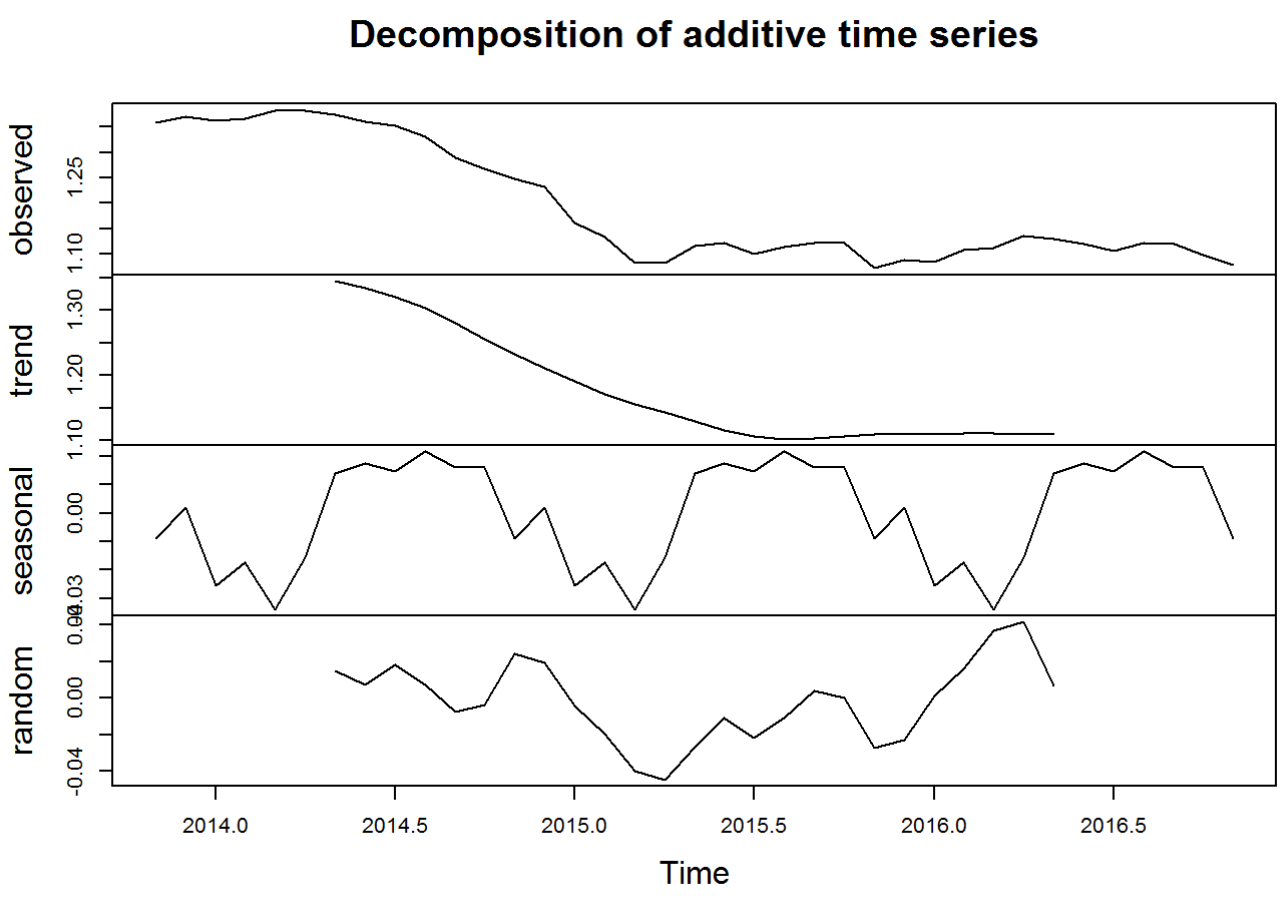
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

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

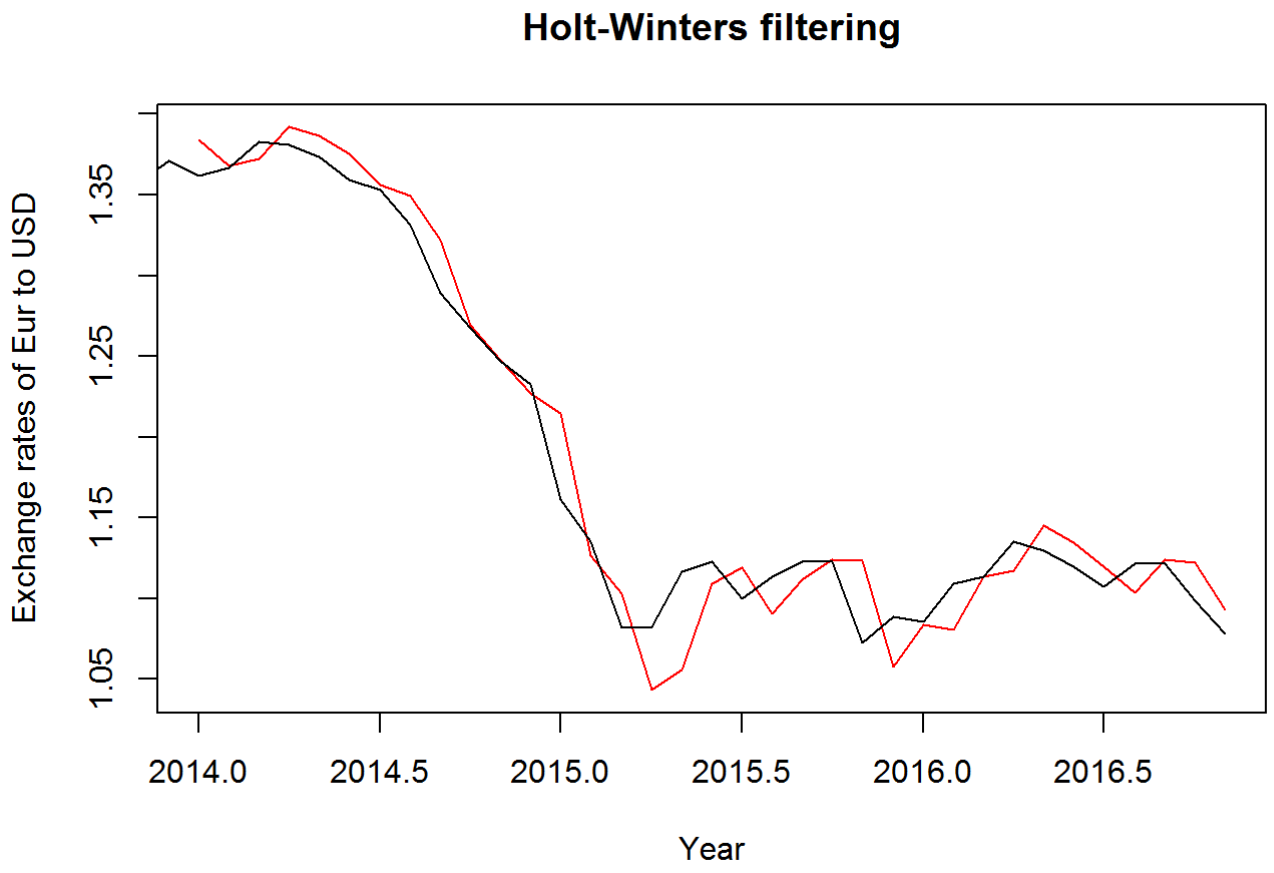
```
## Warning: package 'timeDate' was built under R version 3.3.2
```

```
## This is forecast 7.3
```

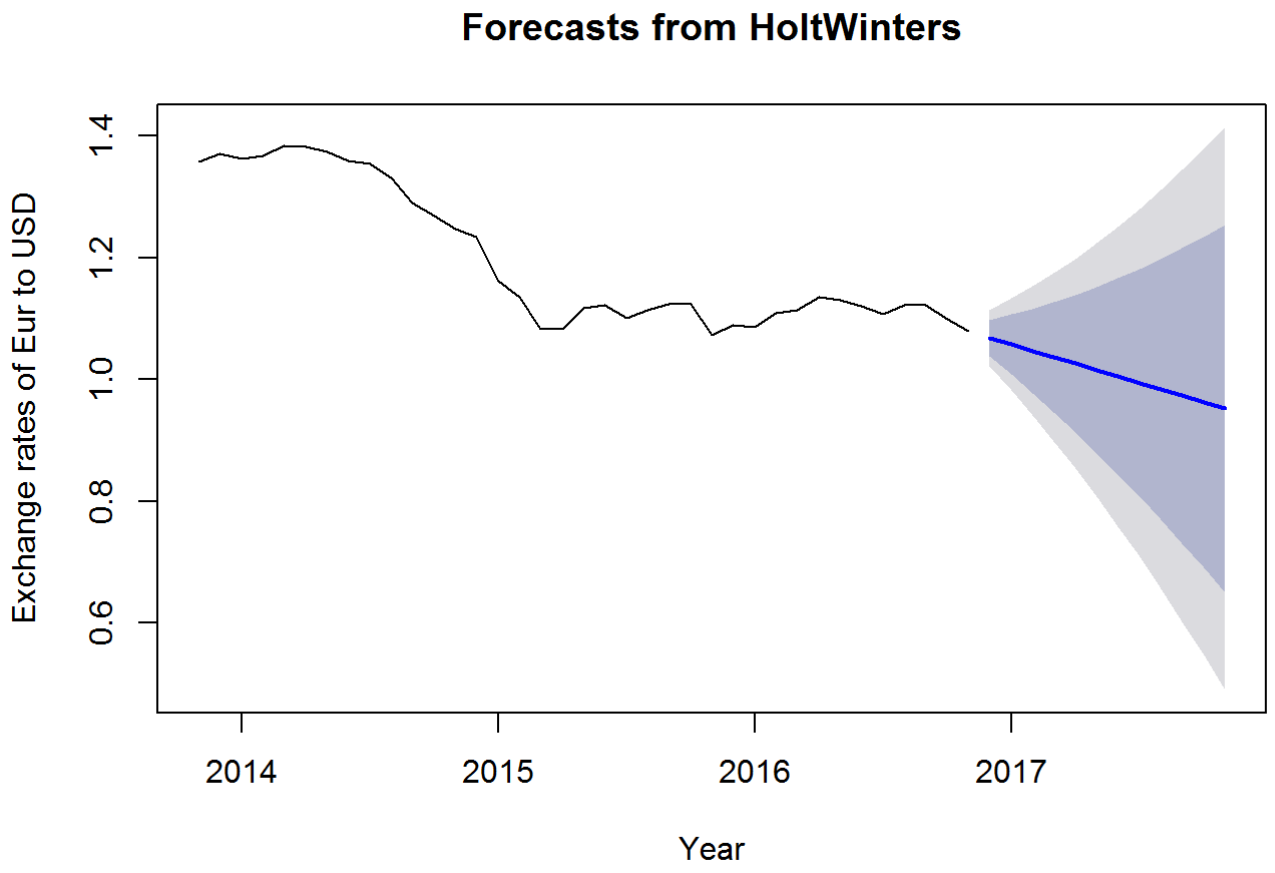
```
ts.eur.usd.d <- decompose(ts.eur.usd)
plot(ts.eur.usd.d)
```




```
ts.eur.holt <- HoltWinters(ts.eur.usd, gamma=FALSE)
plot(ts.eur.holt, ylab="Exchange rates of Eur to USD", xlab="Year")
```



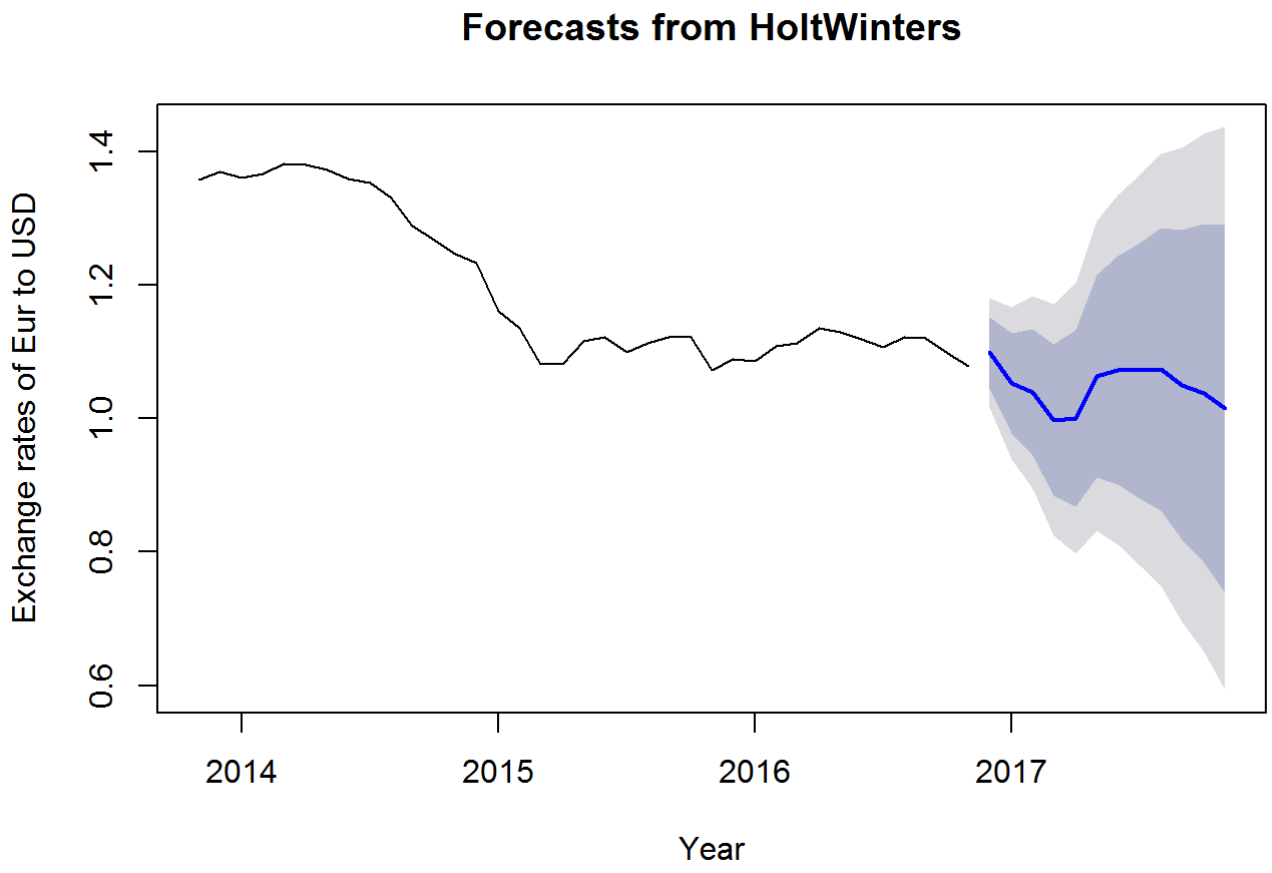

```
ts.eur.forecasts <- forecast.HoltWinters(ts.eur.holt, h=12)
plot.forecast(ts.eur.forecasts, ylab="Exchange rates of Eur to USD", xlab="Year")
```




```
ts.eur.holt <- HoltWinters(ts.eur.usd, gamma=TRUE)
ts.eur.forecasts <- forecast.HoltWinters(ts.eur.holt, h=12)
ts.eur.forecasts
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Dec 2016      1.0991371 1.0453373 1.152937 1.0168574 1.181417
## Jan 2017      1.0530620 0.9783779 1.127746 0.9388425 1.167281
## Feb 2017      1.0393531 0.9451722 1.133534 0.8953158 1.183390
## Mar 2017      0.9978785 0.8845927 1.111164 0.8246228 1.171134
## Apr 2017      1.0008335 0.8684295 1.133238 0.7983390 1.203328
## May 2017      1.0640431 0.9123073 1.215779 0.8319832 1.296103
## Jun 2017      1.0730652 0.9016749 1.244456 0.8109463 1.335184
## Jul 2017      1.0722814 0.8808505 1.263712 0.7795131 1.365050
## Aug 2017      1.0735431 0.8616483 1.285438 0.7494779 1.397608
## Sep 2017      1.0508585 0.8180545 1.283663 0.6948154 1.406902
## Oct 2017      1.0383644 0.7841931 1.292536 0.6496430 1.427086
## Nov 2017      1.0150993 0.7390966 1.291102 0.5929896 1.437209
```

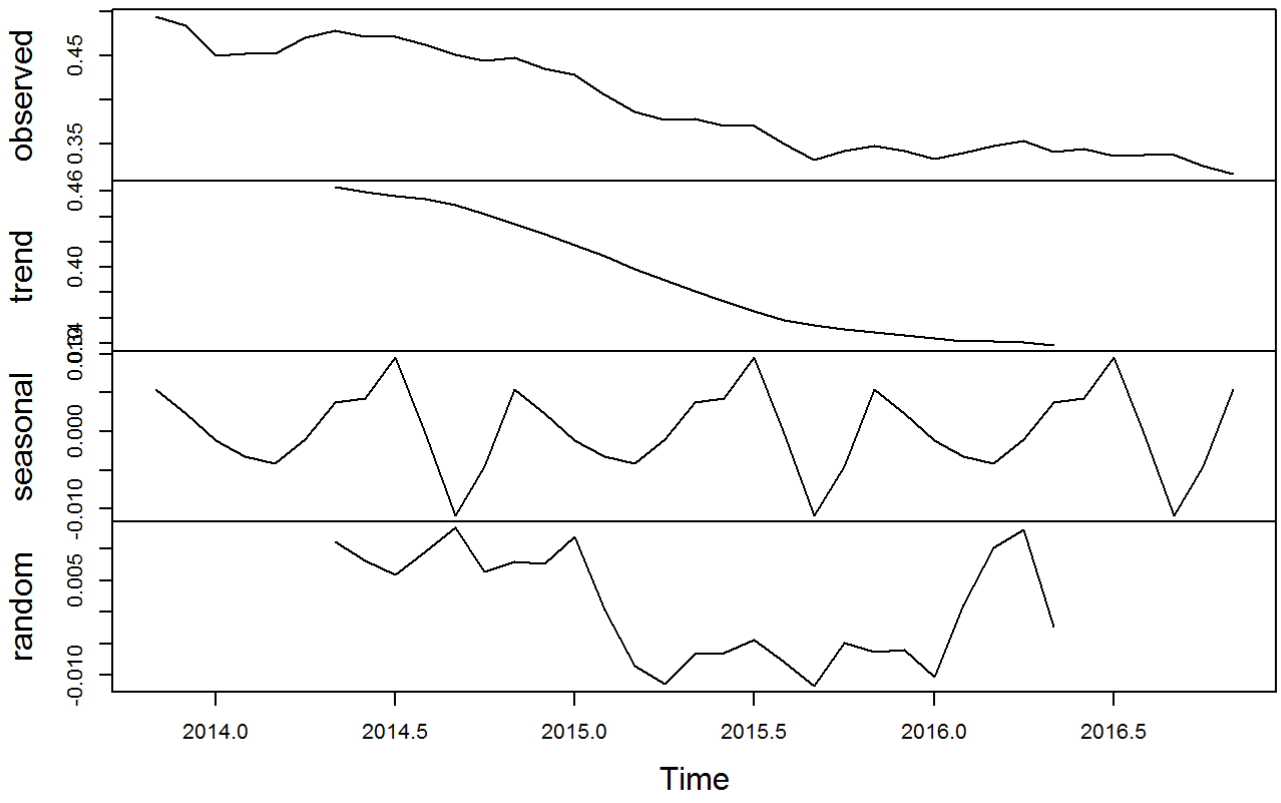
```
plot.forecast(ts.eur.forecasts, ylab="Exchange rates of Eur to USD", xlab="Year")
```



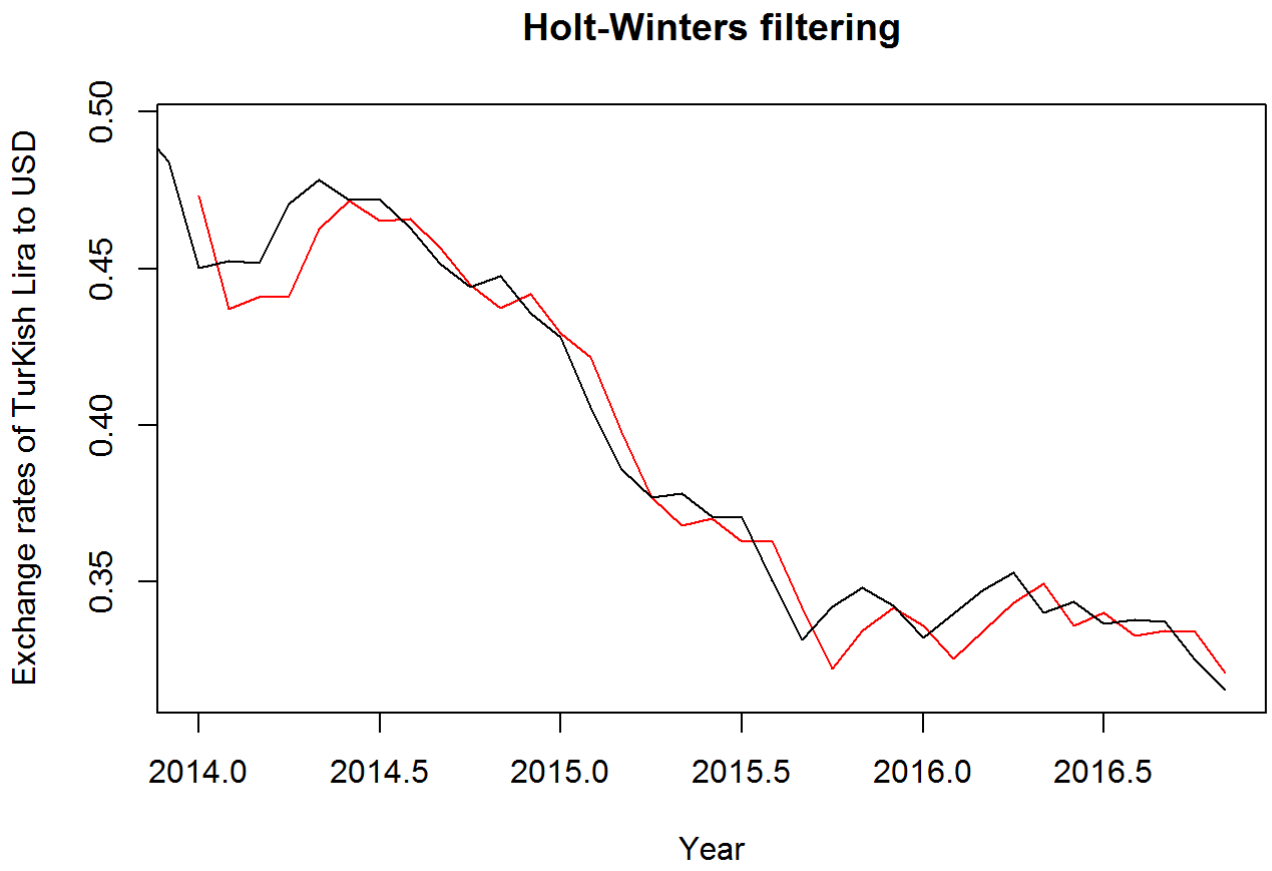
Decomposing, Holt Smoothening and forcasting for Lira

To estimate the trend, seasonal and irregular components of Lira over the given period, we decompose the time series obtained. We then go onto forecast rates for 2017 `(h=12)` which returns values within confidence levels of 80% and 95% respectively for the exchange rate forecasts of the Lira to the Dollar.

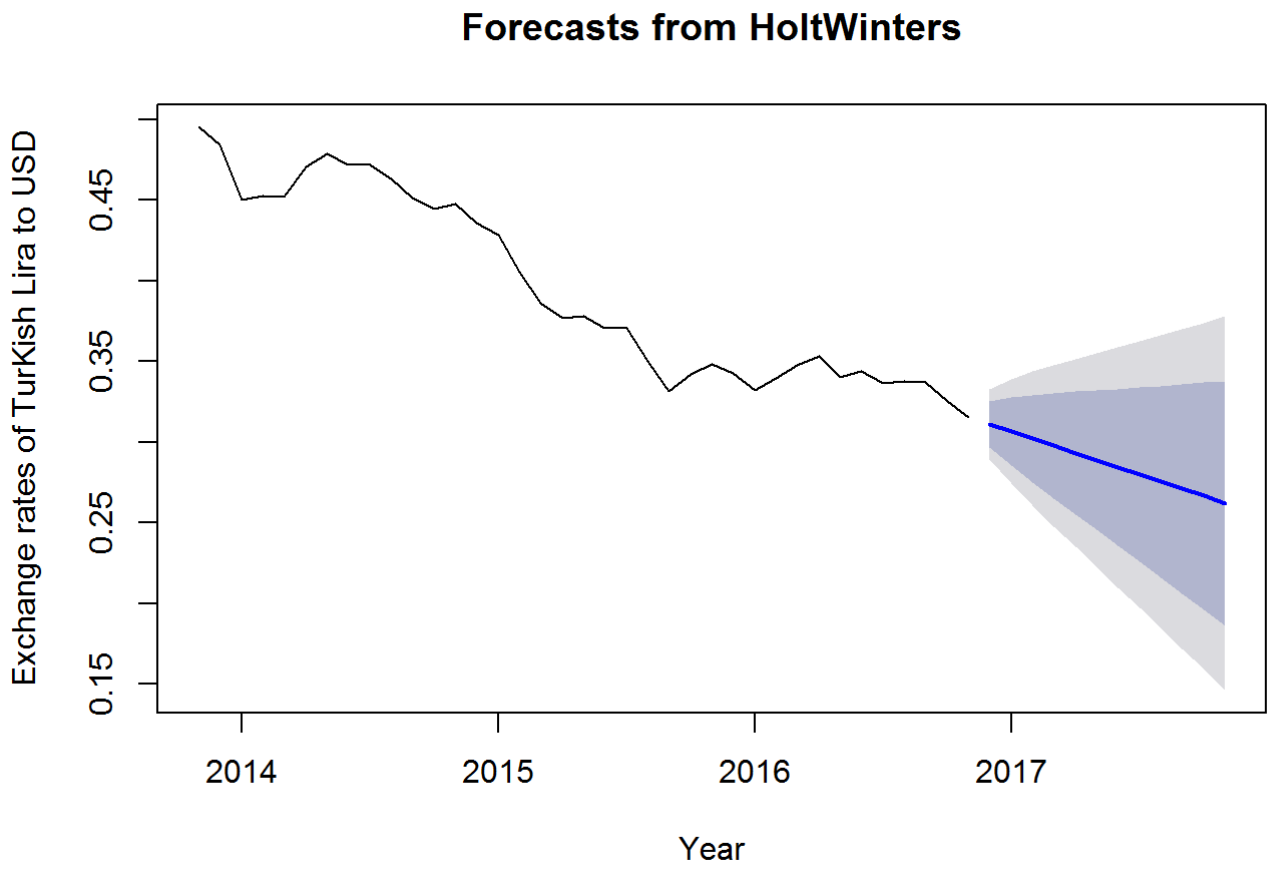
```
ts.tur.usd.d <- decompose(ts.tur.usd)
plot(ts.tur.usd.d)
```




```
ts.tur.holt <- HoltWinters(ts.tur.usd, gamma=FALSE)
plot(ts.tur.holt, ylab="Exchange rates of TurKish Lira to USD", xlab="Year")
```



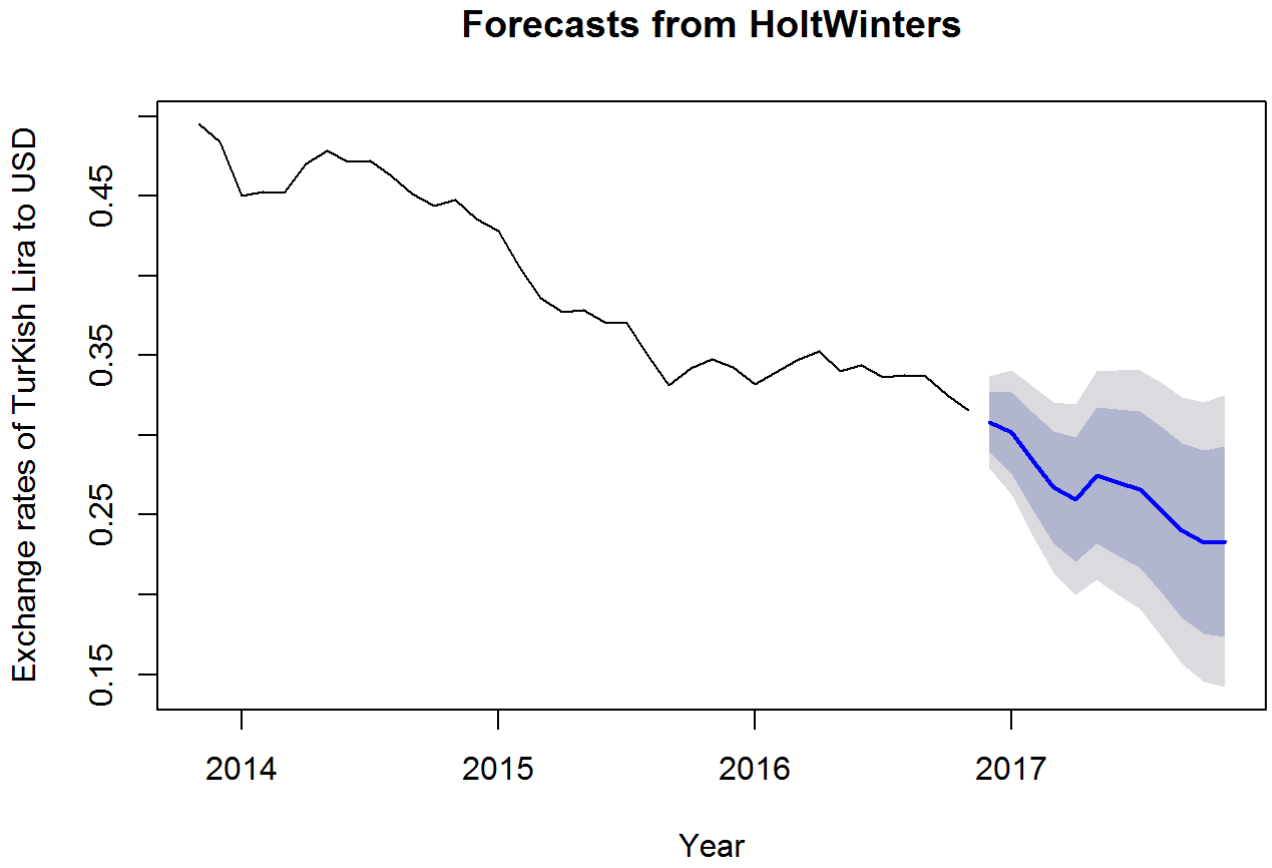

```
ts.tur.forecasts <- forecast.HoltWinters(ts.tur.holt, h=12)
plot.forecast(ts.tur.forecasts, ylab="Exchange rates of TurKish Lira to USD", xlab="Year")
```




```
ts.tur.holt <- HoltWinters(ts.tur.usd, gamma=TRUE)
ts.tur.forecasts <- forecast.HoltWinters(ts.tur.holt, h=12)
ts.tur.forecasts
```

```
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## Dec 2016      0.3081405 0.2894003 0.3268808 0.2794798 0.3368012
## Jan 2017      0.3017106 0.2763391 0.3270821 0.2629082 0.3405130
## Feb 2017      0.2840397 0.2534418 0.3146376 0.2372443 0.3308352
## Mar 2017      0.2671900 0.2321365 0.3022435 0.2135803 0.3207998
## Apr 2017      0.2595542 0.2205508 0.2985577 0.1999037 0.3192048
## May 2017      0.2747545 0.2321659 0.3173430 0.2096209 0.3398880
## Jun 2017      0.2705148 0.2246203 0.3164093 0.2003253 0.3407043
## Jul 2017      0.2660806 0.2171029 0.3150584 0.1911756 0.3409856
## Aug 2017      0.2537437 0.2018656 0.3056217 0.1744030 0.3330844
## Sep 2017      0.2401001 0.1854754 0.2947247 0.1565589 0.3236413
## Oct 2017      0.2329999 0.1757603 0.2902395 0.1454595 0.3205403
## Nov 2017      0.2334400 0.1736999 0.2931802 0.1420753 0.3248048
```

```
plot.forecast(ts.tur.forecasts, ylab="Exchange rates of TurKish Lira to USD", xlab="Year")
```



Forecast for Eur to USD exchange rates using Arima Model

We use the Arima model to further consolidate our Forecast Models achieved as above. We use the `auto.arima` function to predict the (p,d,q) variables and use the returned values for an optimum forecast model.

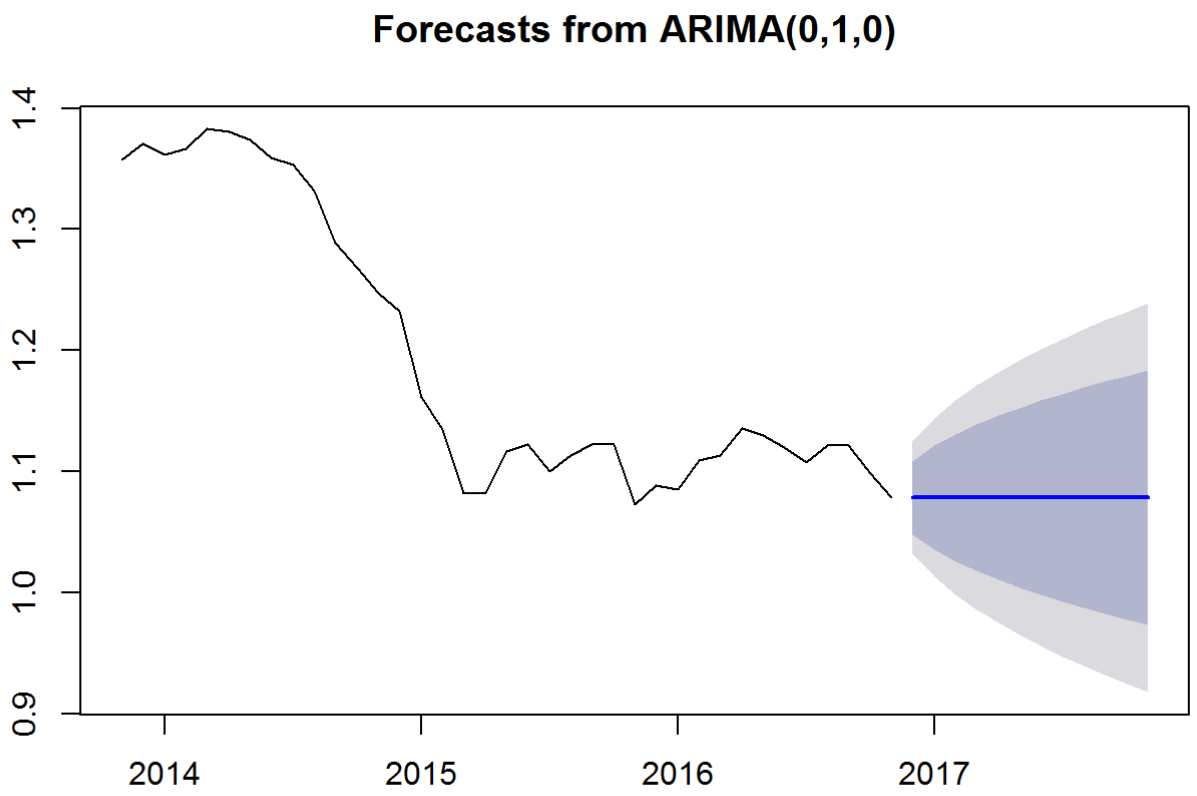
```
auto.arima(ts.eur.usd)
```

```
## Series: ts.eur.usd
## ARIMA(0,1,0) with drift
##
## Coefficients:
##         drift
##       -0.0078
## s.e.   0.0037
##
## sigma^2 estimated as 0.0005133:  log likelihood=85.77
## AIC=-167.54   AICc=-167.18   BIC=-164.38
```

```
eur.arima<-arima(ts.eur.usd, c(0,1,0))
eur.arima.forecasts <- forecast.Arima(eur.arima, h=12)
eur.arima.forecasts
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Dec 2016         1.0784 1.0480989 1.108701 1.0320584 1.124742
## Jan 2017         1.0784 1.0355478 1.121252 1.0128631 1.143937
## Feb 2017         1.0784 1.0259169 1.130883 0.9981341 1.158666
## Mar 2017         1.0784 1.0177978 1.139002 0.9857169 1.171083
## Apr 2017         1.0784 1.0106447 1.146155 0.9747771 1.182023
## May 2017         1.0784 1.0041777 1.152622 0.9648868 1.191913
## Jun 2017         1.0784 0.9982308 1.158569 0.9557918 1.201008
## Jul 2017         1.0784 0.9926955 1.164104 0.9473263 1.209474
## Aug 2017         1.0784 0.9874967 1.169303 0.9393753 1.217425
## Sep 2017         1.0784 0.9825795 1.174221 0.9318551 1.224945
## Oct 2017         1.0784 0.9779026 1.178897 0.9247025 1.232098
## Nov 2017         1.0784 0.9734339 1.183366 0.9178682 1.238932
```

```
plot(eur.arima.forecasts)
```



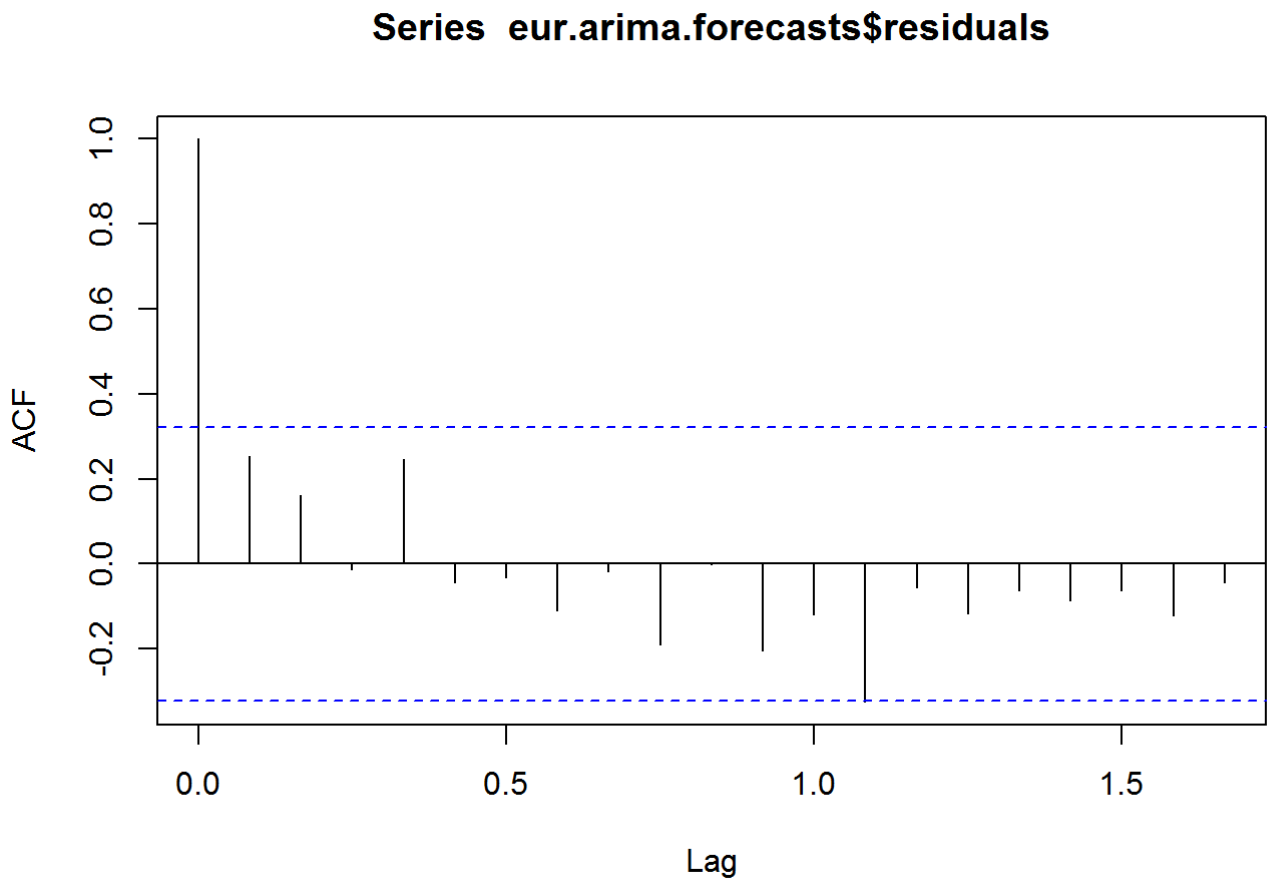
The ‘forecast errors’ are calculated as the observed values minus predicted values, for each time point. We can only calculate the forecast errors for the time period covered by our original time series, which is 2013 to 2016 for the exchange rate data. As mentioned above, one measure of the accuracy of the predictive model is the sum-of-squared-errors (SSE) for the in-sample forecast errors.

The in-sample forecast errors are stored in the named element “residuals” of the list variable returned by `forecast.HoltWinters()`. If the predictive model cannot be improved upon, there should be no correlations between forecast errors for successive predictions. In other words, if there are correlations between forecast errors for successive predictions, it is likely that the simple exponential smoothing forecasts could be improved upon by another forecasting technique.

To figure out whether this is the case, we can obtain a correlogram of the in-sample forecast errors for lags 1-20. We can calculate a correlogram of the forecast errors using the `acf()` function in R. To specify the maximum lag that we want to look at, we use the `lag.max` parameter in `acf()`.

For example, to calculate a correlogram of the forecast errors for the exchange rate data for lags 1-20, we type:

```
acf(eur.arima.forecasts$residuals, lag.max=20)
```



We can see from the correlogram that the autocorrelation at lag 13 is just touching the significance bounds. To test whether there is significant evidence for non-zero correlations at lags 1-20, we can carry out a Ljung-Box test. This can be done in R using the `Box.test()` function. The maximum lag that we want to look at is specified using the “lag” parameter in the `Box.test()` function. For example, to test whether there are non-zero autocorrelations at lags 1-20, for the in-sample forecast errors for exchange rate data, we type:

```
Box.test(eur.arima.forecasts$residuals, lag=20, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  eur.arima.forecasts$residuals
## X-squared = 21.951, df = 20, p-value = 0.3432
```

Here the Ljung-Box test statistic is 21.9, and the p-value is 0.3, so there is little evidence of non-zero autocorrelations in the in-sample forecast errors at lags 1-20.

To be sure that the predictive model cannot be improved upon, it is also a good idea to check whether the forecast errors are normally distributed with mean zero and constant variance. To check whether the forecast errors have constant variance, we can make a time plot of the in-sample forecast errors:

```
plot.ts(eur.arima.forecasts$residuals)
```

The plot shows that the in-sample forecast errors do not seem to have a constant variance over time, although the size of the fluctuations in the start of the time series may be slightly less than that at later dates.

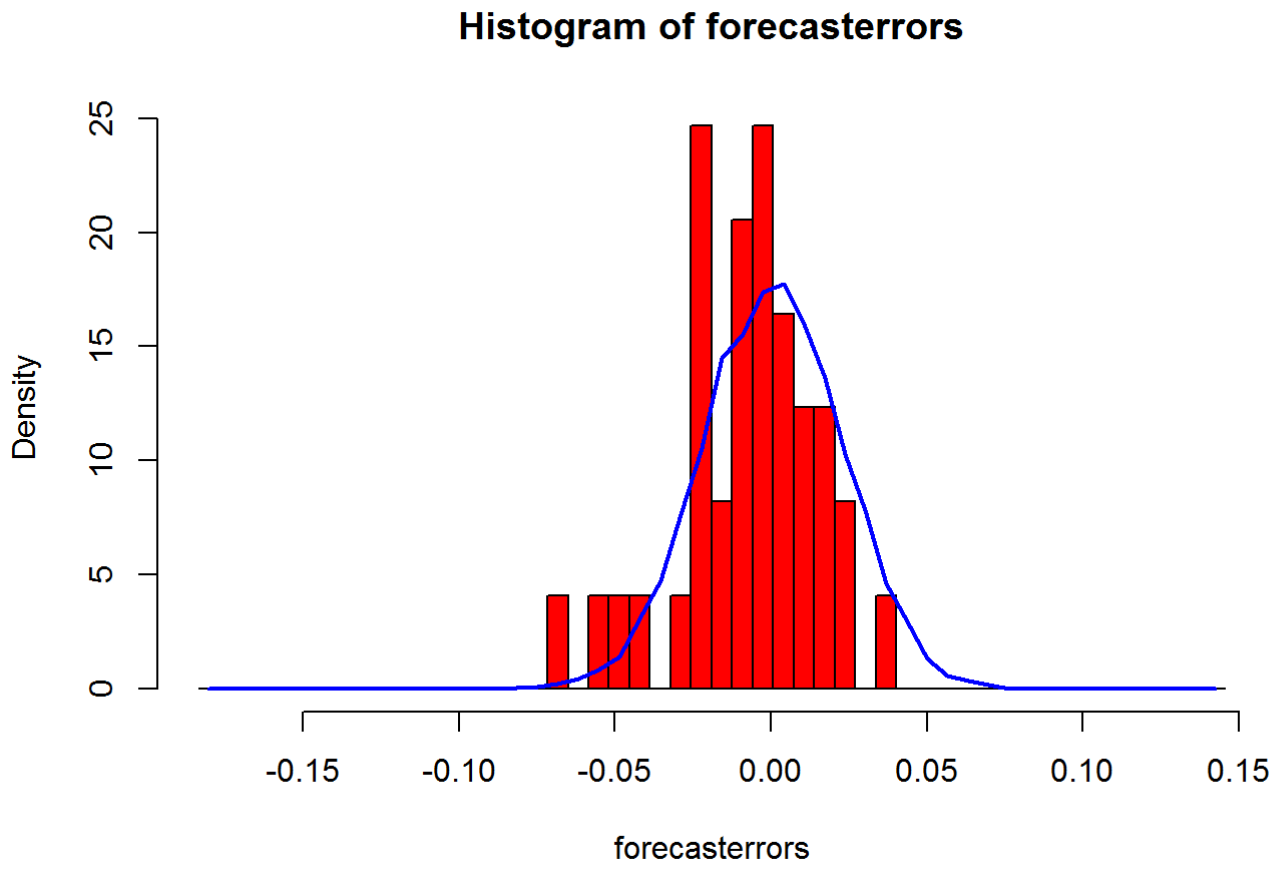
We therefore difference the given forecast once to obtain constant variance over time. This is also supported by the `auto.arima` function predicting a integrated variable “d” as 1.

```
eur.usd.diff <- diff(eur.arima.forecasts$residuals, differences = 1)
plot.ts(eur.usd.diff)
```

To check whether the forecast errors are normally distributed with mean zero, we can plot a histogram of the forecast errors, with an overlaid normal curve that has mean zero and the same standard deviation as the distribution of forecast errors. To do this, we can define an R function “plotForecastErrors()”, below:

```
PlotForecastErrors <- function(forecasterrors)
{
  mybinsize <- IQR(forecasterrors)/4
  mysd <- sd(forecasterrors)
  mymin <- min(forecasterrors)-mysd*5
  mymax <- max(forecasterrors)+mysd*5
  mynorm <- rnorm(10000,mean=0,sd=mysd)
  mymin2 <- min(mynorm)
  mymax2 <- max(mynorm)
  if (mymin2<mymin)
  {
    mymin<-mymin2
  }
  if (mymax2>mymax)
  {
    mymax<-mymax2
  }
  mybins <- seq(mymin,mymax,mybinsize)
  hist(forecasterrors, col="red", freq=FALSE, breaks=mybins)
  myhist <- hist(mynorm, plot=FALSE, breaks=mybins)
  points(myhist$mids, myhist$density, type="l", col="blue", lwd=2)
}

PlotForecastErrors(eur.arima.forecasts$residuals)
```



```
mean(eur.arima.forecasts$residuals)
```

```
## [1] -0.007505654
```

The plot shows that the distribution of forecast errors is roughly centred on zero (supported by the `mean` function), and is more or less normally distributed and so it is plausible that the forecast errors are normally distributed with mean zero.

The Ljung-Box test showed that there is little evidence of non-zero autocorrelations in the in-sample forecast errors, and the distribution of forecast errors seems to be normally distributed with mean zero. This suggests that the simple exponential smoothing method provides an adequate predictive model for the exchange rates, which probably cannot be improved upon. Furthermore, the assumptions that the 80% and 95% predictions intervals were based upon (that there are no autocorrelations in the forecast errors, and the forecast errors are normally distributed with mean zero and constant variance) are probably valid.

Forecast for Turkish Lira to USD exchange rates using Arima Model

We carry out the same procedure to achieve forecast values using the Arima model and check for forecast errors.

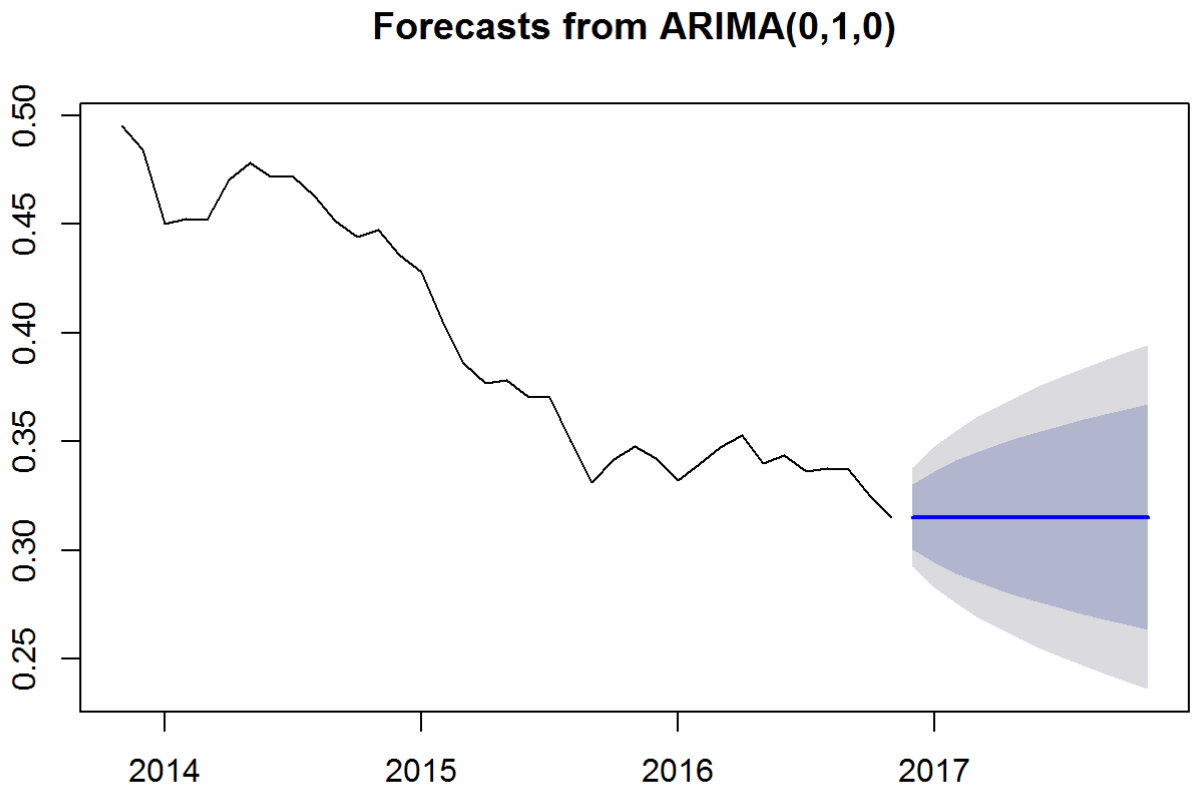
```
auto.arima(ts.tur.usd)
```

```
## Series: ts.tur.usd
## ARIMA(0,1,0) with drift
##
## Coefficients:
##         drift
##       -0.0050
## s.e.   0.0018
##
## sigma^2 estimated as 0.000114:  log likelihood=112.86
## AIC=-221.72   AICc=-221.36   BIC=-218.55
```

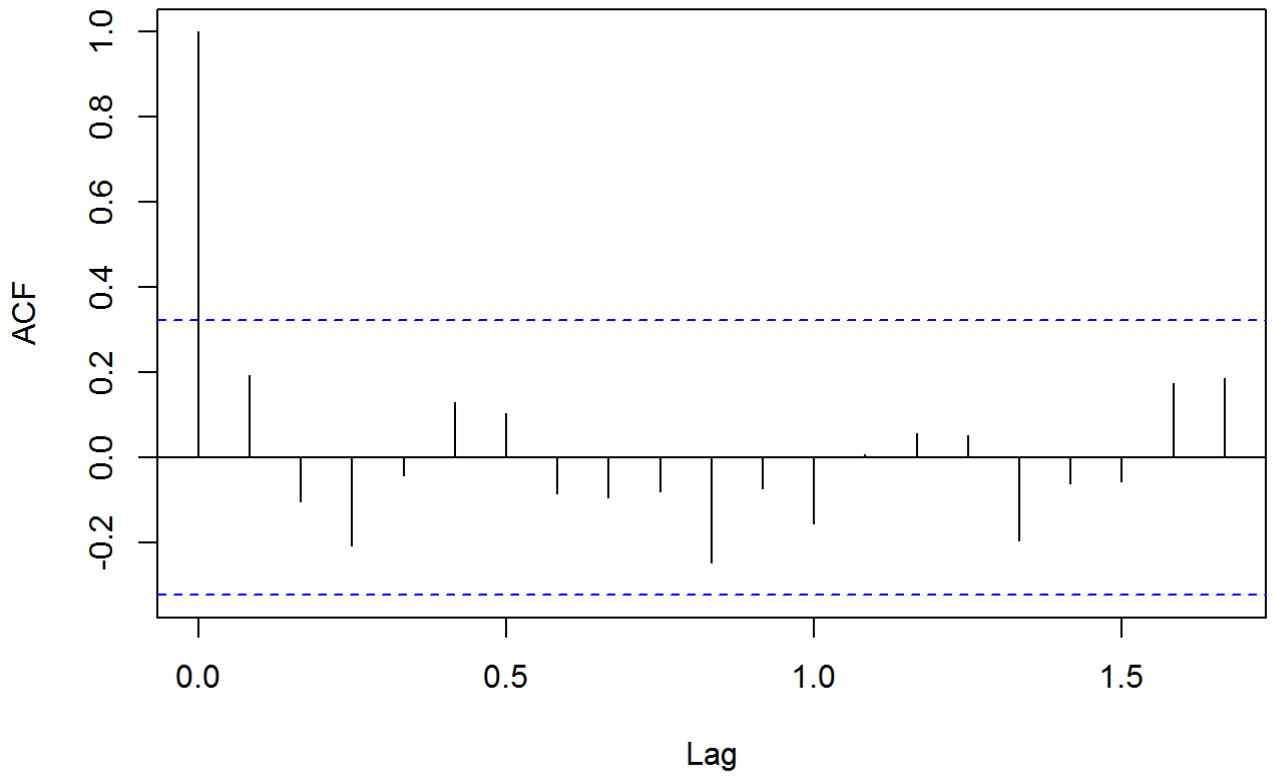
```
tur.arima<-arima(ts.tur.usd, c(0,1,0))
tur.arima.forecasts <- forecast.Arima(tur.arima, h=12)
tur.arima.forecasts
```

```
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## Dec 2016      0.3154117 0.3004819 0.3303414 0.2925786 0.3382448
## Jan 2017      0.3154117 0.2942978 0.3365256 0.2831208 0.3477026
## Feb 2017      0.3154117 0.2895526 0.3412708 0.2758636 0.3549598
## Mar 2017      0.3154117 0.2855521 0.3452712 0.2697455 0.3610779
## Apr 2017      0.3154117 0.2820277 0.3487956 0.2643553 0.3664681
## May 2017      0.3154117 0.2788414 0.3519820 0.2594822 0.3713411
## Jun 2017      0.3154117 0.2759112 0.3549121 0.2550010 0.3758224
## Jul 2017      0.3154117 0.2731839 0.3576394 0.2508299 0.3799935
## Aug 2017      0.3154117 0.2706224 0.3602010 0.2469124 0.3839110
## Sep 2017      0.3154117 0.2681996 0.3626237 0.2432071 0.3876163
## Oct 2017      0.3154117 0.2658953 0.3649281 0.2396828 0.3911405
## Nov 2017      0.3154117 0.2636935 0.3671299 0.2363155 0.3945079
```

```
plot(tur.arima.forecasts)
```



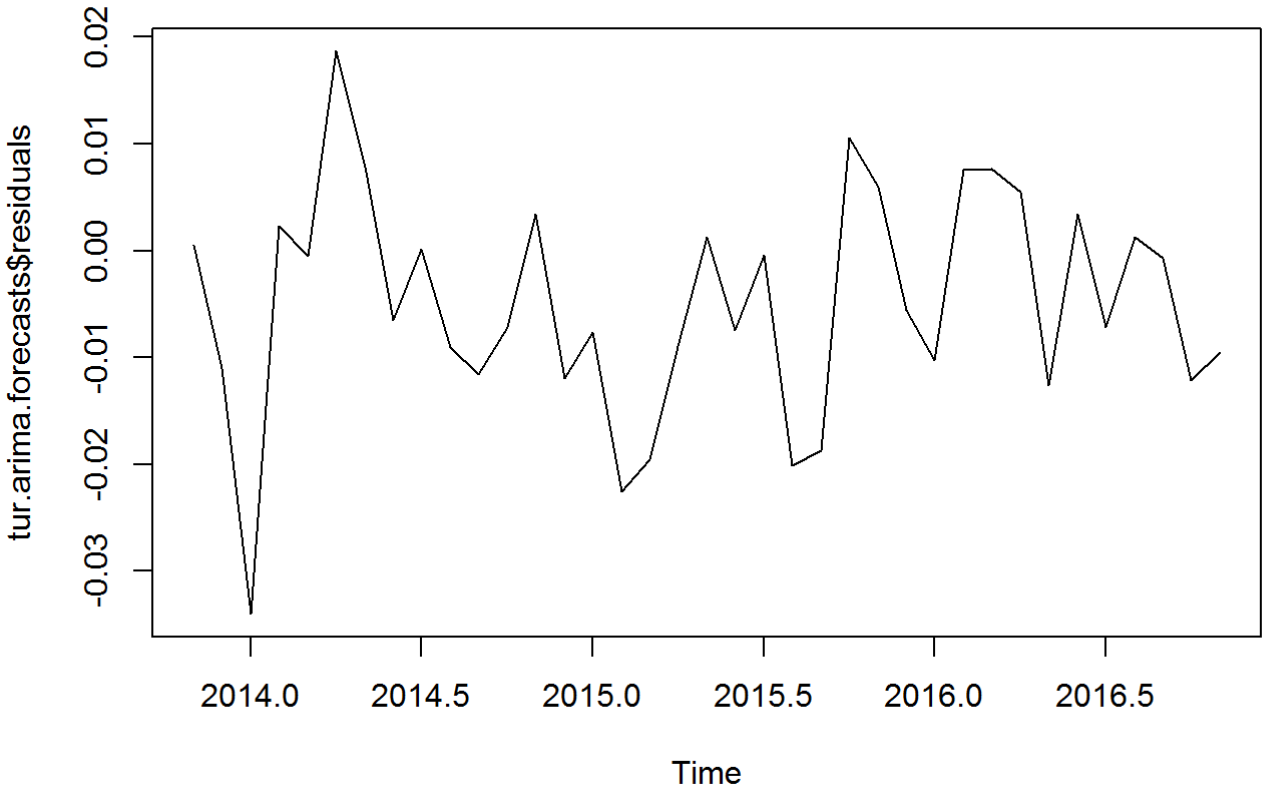

```
acf(tur.arima.forecasts$residuals, lag.max=20)
```



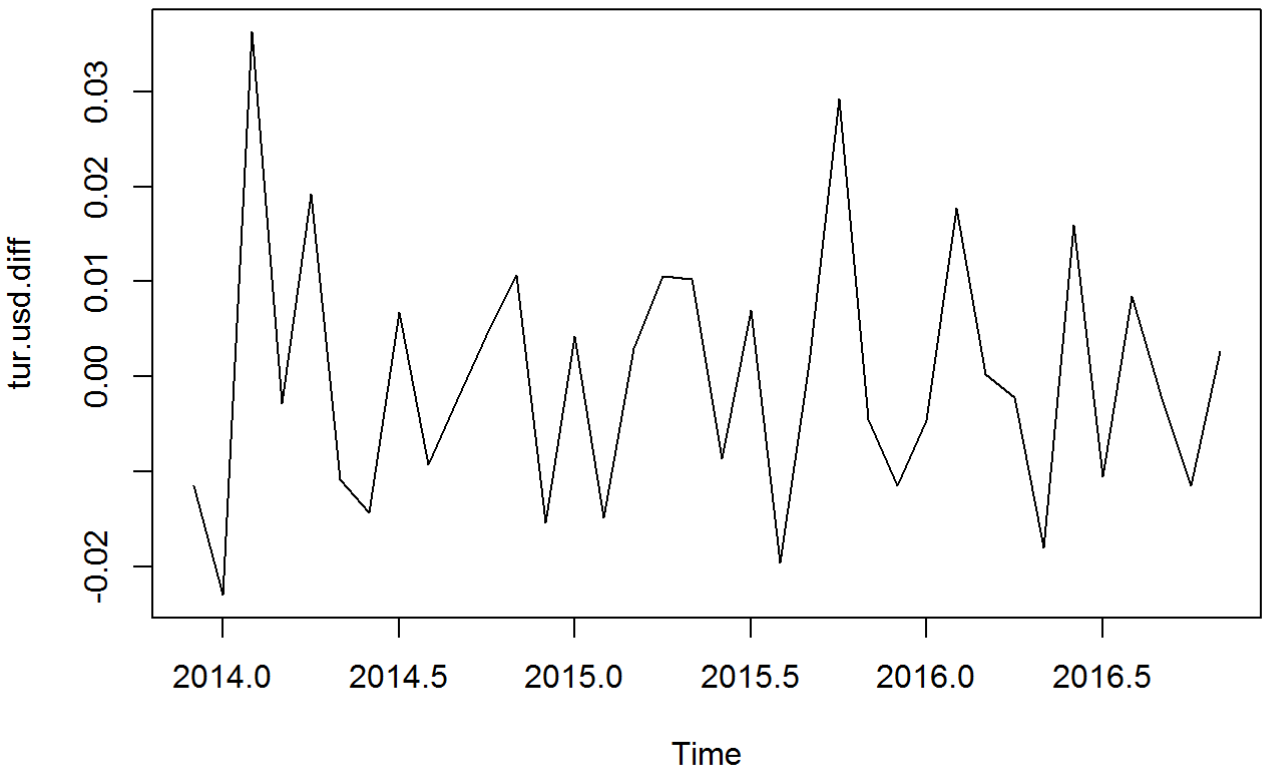

```
Box.test(tur.arima.forecasts$residuals, lag=20, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  tur.arima.forecasts$residuals
## X-squared = 20.023, df = 20, p-value = 0.4565
```

```
plot.ts(tur.arima.forecasts$residuals)
```

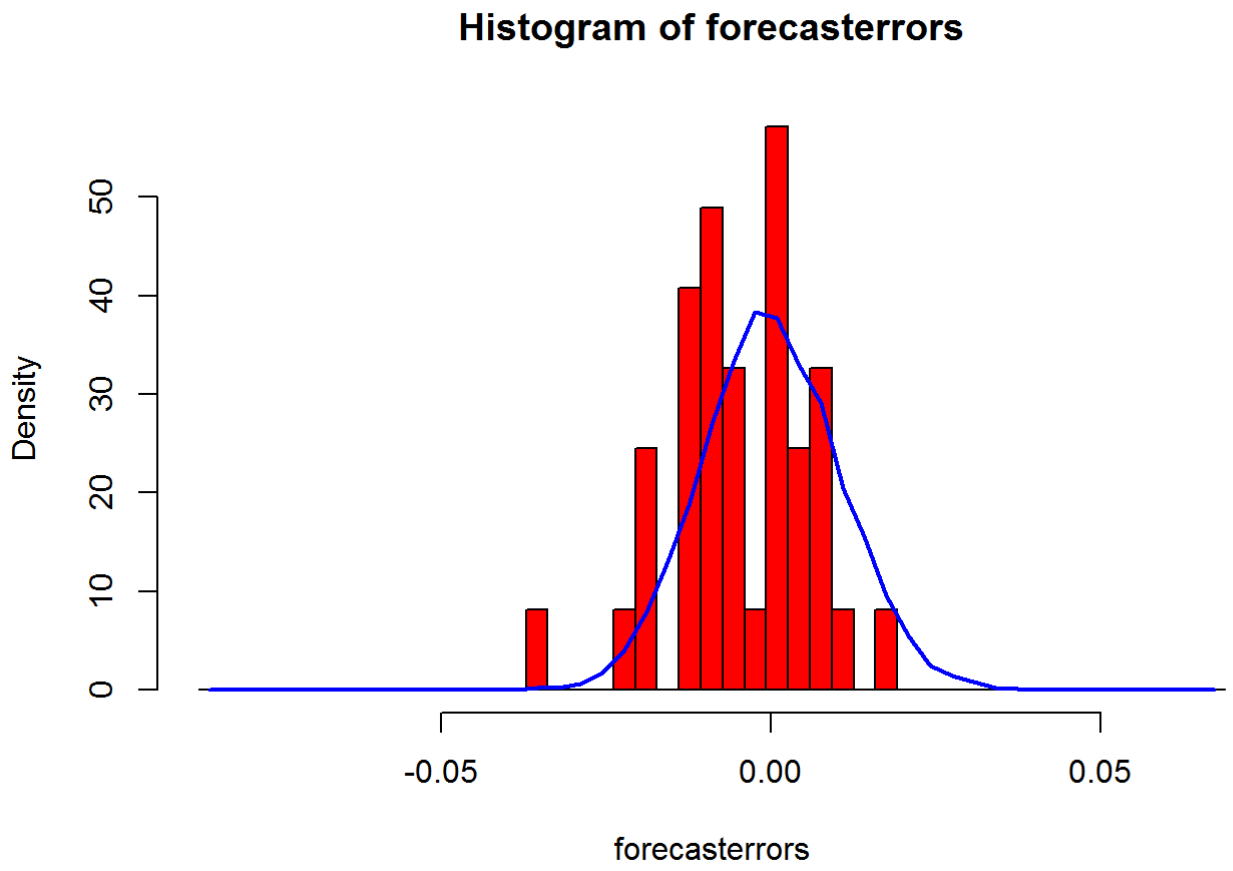



```
tur.usd.diff <- diff(tur.arima.forecasts$residuals, differences = 1)
plot.ts(tur.usd.diff)
```




```
PlotForecastErrors <- function(forecasterrors)
{
  mybinsize <- IQR(forecasterrors)/4
  mysd <- sd(forecasterrors)
  mymin <- min(forecasterrors)-mysd*5
  mymax <- max(forecasterrors)+mysd*5
  mynorm <- rnorm(10000,mean=0,sd=mysd)
  mymin2 <- min(mynorm)
  mymax2 <- max(mynorm)
  if (mymin2<mymin)
  {
    mymin<-mymin2
  }
  if (mymax2>mymax)
  {
    mymax<-mymax2
  }
  mybins <- seq(mymin,mymax,mybinsize)
  hist(forecasterrors, col="red", freq=FALSE, breaks=mybins)
  myhist <- hist(mynorm, plot=FALSE, breaks=mybins)
  points(myhist$mids, myhist$density, type="l", col="blue", lwd=2)
}

PlotForecastErrors(tur.arima.forecasts$residuals)
```




```
mean(tur.arima.forecasts$residuals)
```

```
## [1] -0.004844642
```

Interpretation:

1. Seasonality and trends Euro to USD: In the Decomposition of time series graph of Euros to USD, we observe that there has been a decreasing trend from mid 2014 to mid 2015 and has been roughly constant thereafter. There is also a seasonal effect on the currency exchanges: We see that at the beginning of the year, the exchange rates are low, they increase during March and are highest in June. Again, we see a drop in the rates towards the year end.

LIRA to USD: In the Decomposition of time series graph of Lira to USD, we observe that there has been a decreasing trend from mid 2014 and has been gradually decreasing since then. We can also see a seasonal effect represented by exchange rate highs in June, a sudden drop in September-October and a gradual increase towards the end of the year.

2. The exchange rates are not the same but follow a similar trend. They have been decreasing from 2014. The seasonal effect for both the Lira and Euro show its highest value in June with a decrease towards the end of the year and an increase during the beginning the next year.

3.The company should invest in other markets when the exchange rates are it the lowest.
As per the summary of our forecasts, we should invest in European markets in the month of March, 2017 when, at an 80% confidence level (range of values from 0.8845 to 1.111 Euros to the Dollar), point forecast value is

projected at 0.9978 Euros towards the Dollar. We should invest in Turkey during October as, at a confidence level of 80%, (range of values from 0.1757 to 0.2902 Lira to Dollar) the point forecast value is projected at 0.2329 Lira towards the Dollar.