# exercis3

*ajay parihar*

*April 26, 2019*

## EXERCISE 3

Q>For this exercise, find a recent and relevant time series to forecast, using the techniques that have been discussed during the lectures. In your analysis, set up a carefully selected forecasting process, taking data considerations and implementation issues into account. Describe your approach, and motivate your choices. The data set should be original (not from R packages), recent, suffciently long and must include a seasonal component. The data should be analyzed using at least two techniques, and results should be compared.

A> In this exercise we are looking to predict amazon stock price based on historic data. The process covers the following steps: 1)data import 2)creating a time series 3)train/test split 4) jung box testing 5) arima and tbats functions

```
library(ggplot2)
library(forecast)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unable to identify current timezone 'C':
## please set environment variable 'TZ'
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

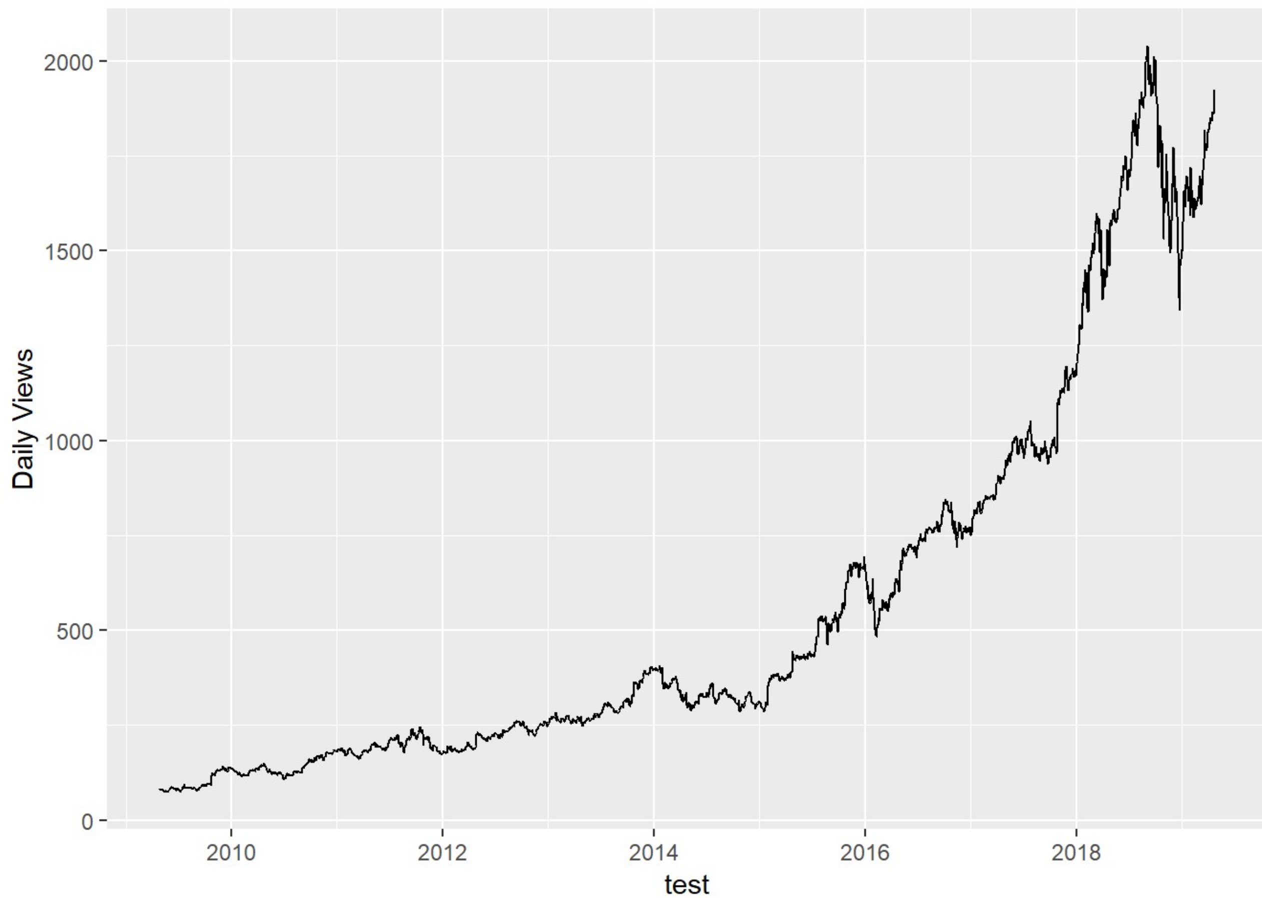
```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
#library(ggfortify)
library(tseries)
library(gridExtra)
#library(docstring)
library(readr)
#library(here)
setwd("C:\\Users\\aparihar\\Downloads")
amazon<-read_csv("HistoricalQuotes.csv")
```

```
## Parsed with column specification:
## cols(
##   date = col_character(),
##   close = col_double(),
##   volume = col_double(),
##   open = col_double(),
##   high = col_double(),
##   low = col_double()
## )
```

```r
amazon[is.na(amazon)] <- 0
amazon<-amazon[,-c(3:6)]
amazon$close<-as.numeric(amazon$close)
amazon$date<-as.Date(amazon$date,format="%m/%d/%Y")
ggplot(amazon, aes(date, close)) + geom_line() +
   xlab("test") + ylab("Daily Views")
```




```
amazon <- ts(amazon$close, start=c(2009, 5), freq=12)
```

Stationery testing

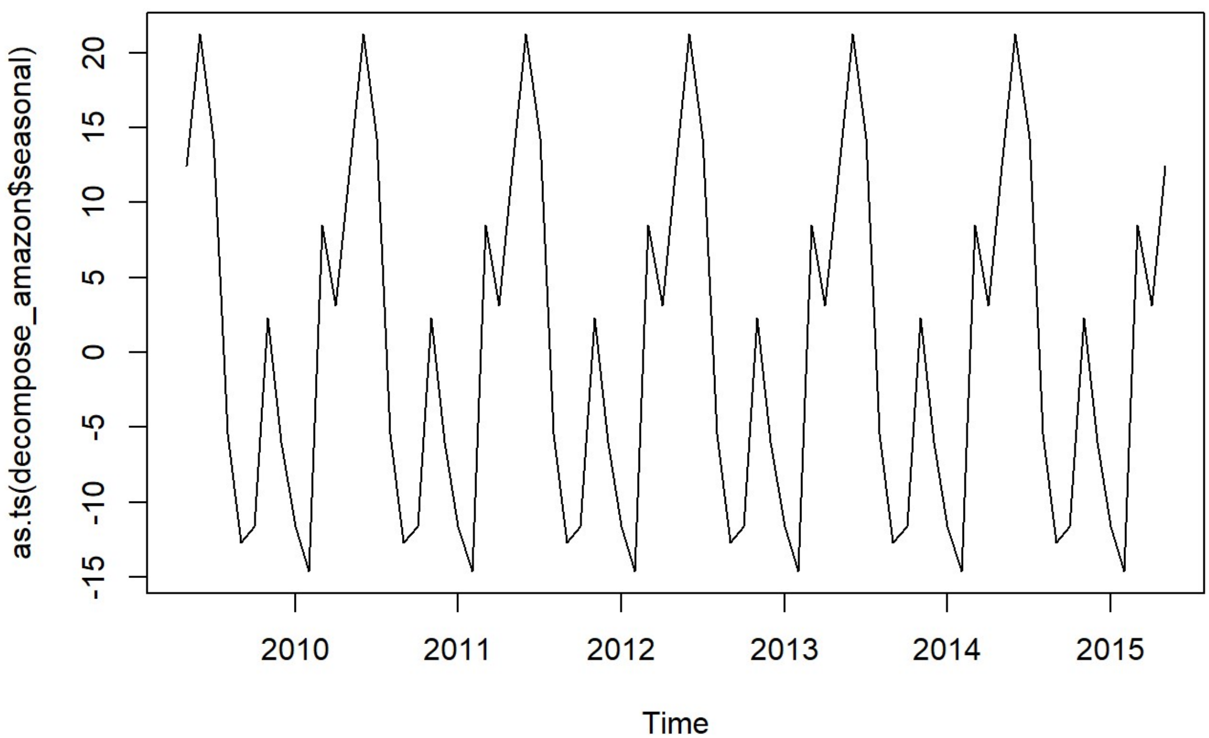
```
amazon_train <- ts(amazon, start=c(2009, 5), end=c(2015, 5), freq=12)
 Box.test(amazon, lag = 20, type = 'Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  amazon
## X-squared = 48636, df = 20, p-value < 2.2e-16
```

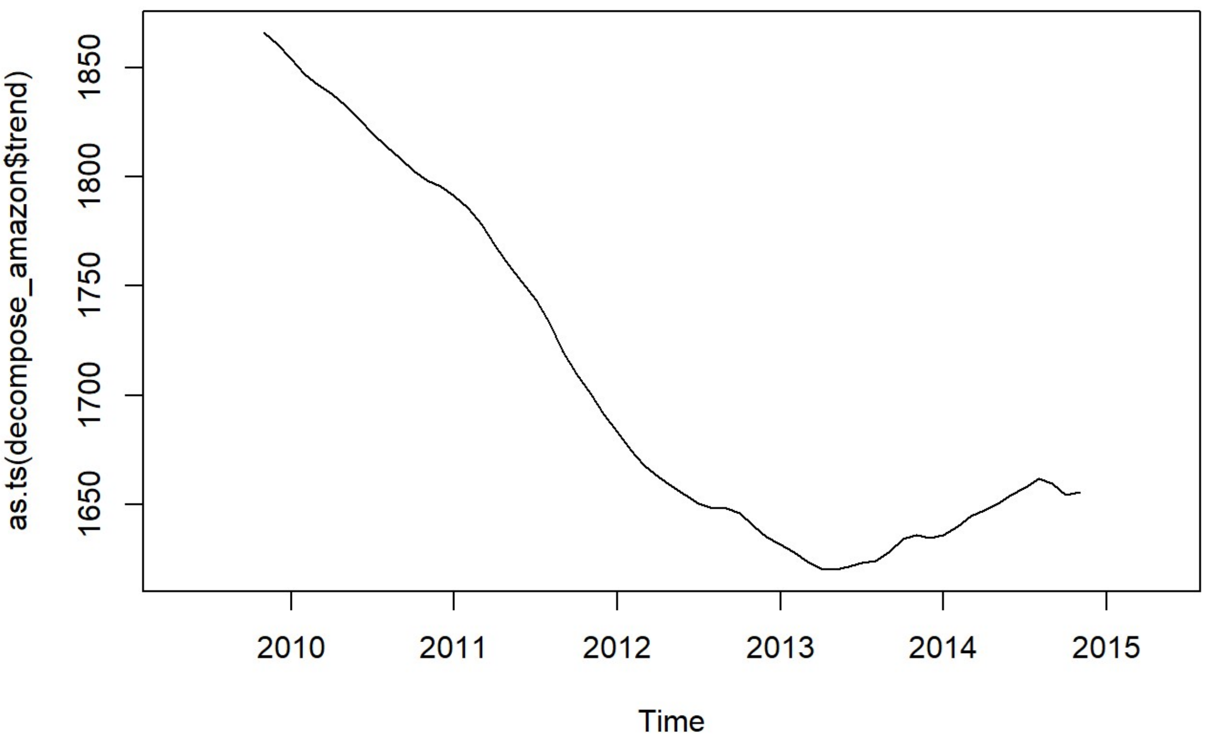
```
 adf.test(amazon)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  amazon
## Dickey-Fuller = -1.9578, Lag order = 13, p-value = 0.5962
## alternative hypothesis: stationary
```

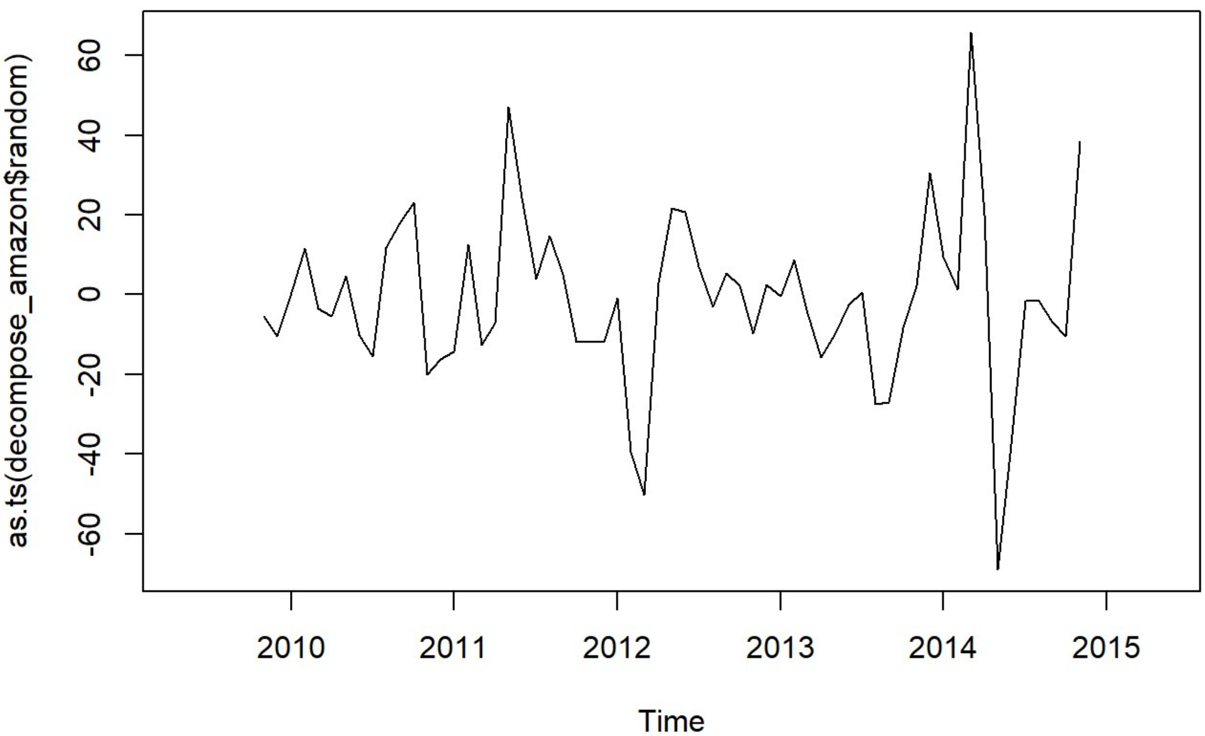
```
###Decomposition
decompose_amazon = decompose(amazon_train, "additive")

plot(as.ts(decompose_amazon$seasonal))
```



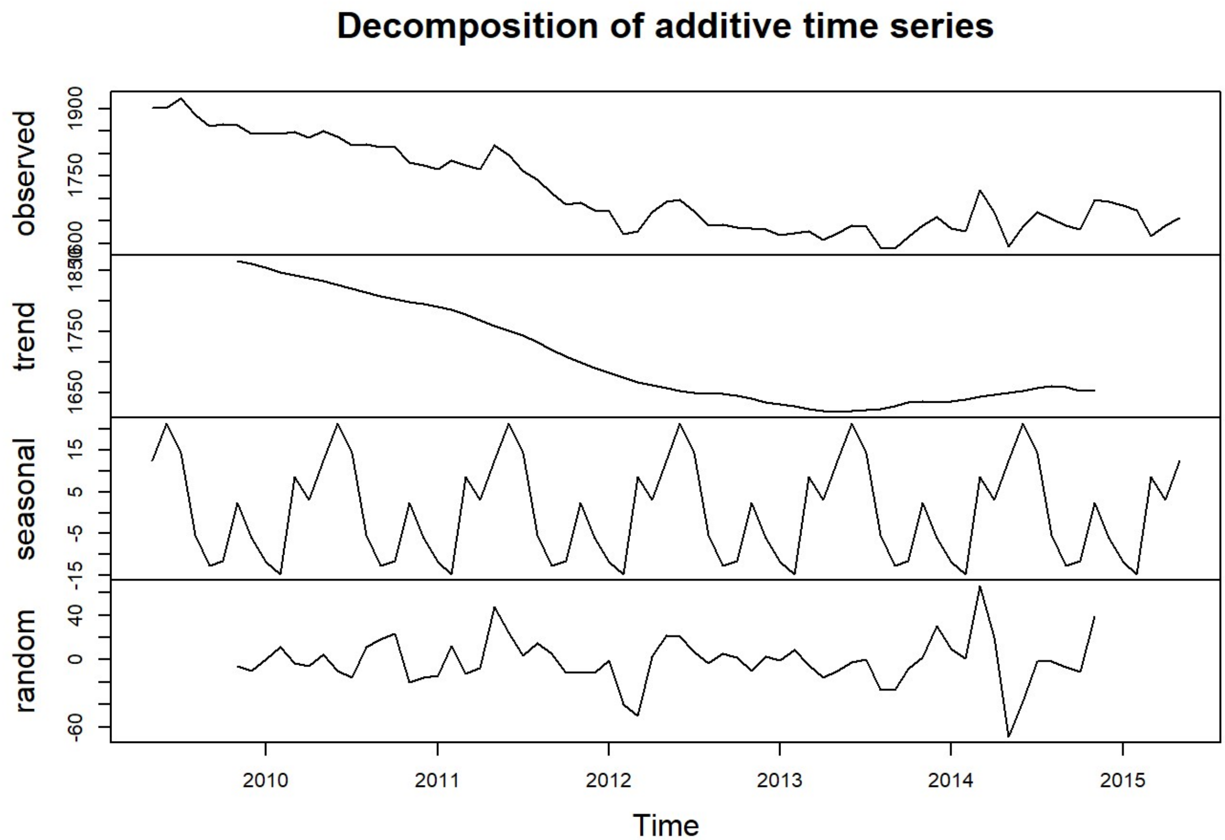

```
plot(as.ts(decompose_amazon$trend))
```



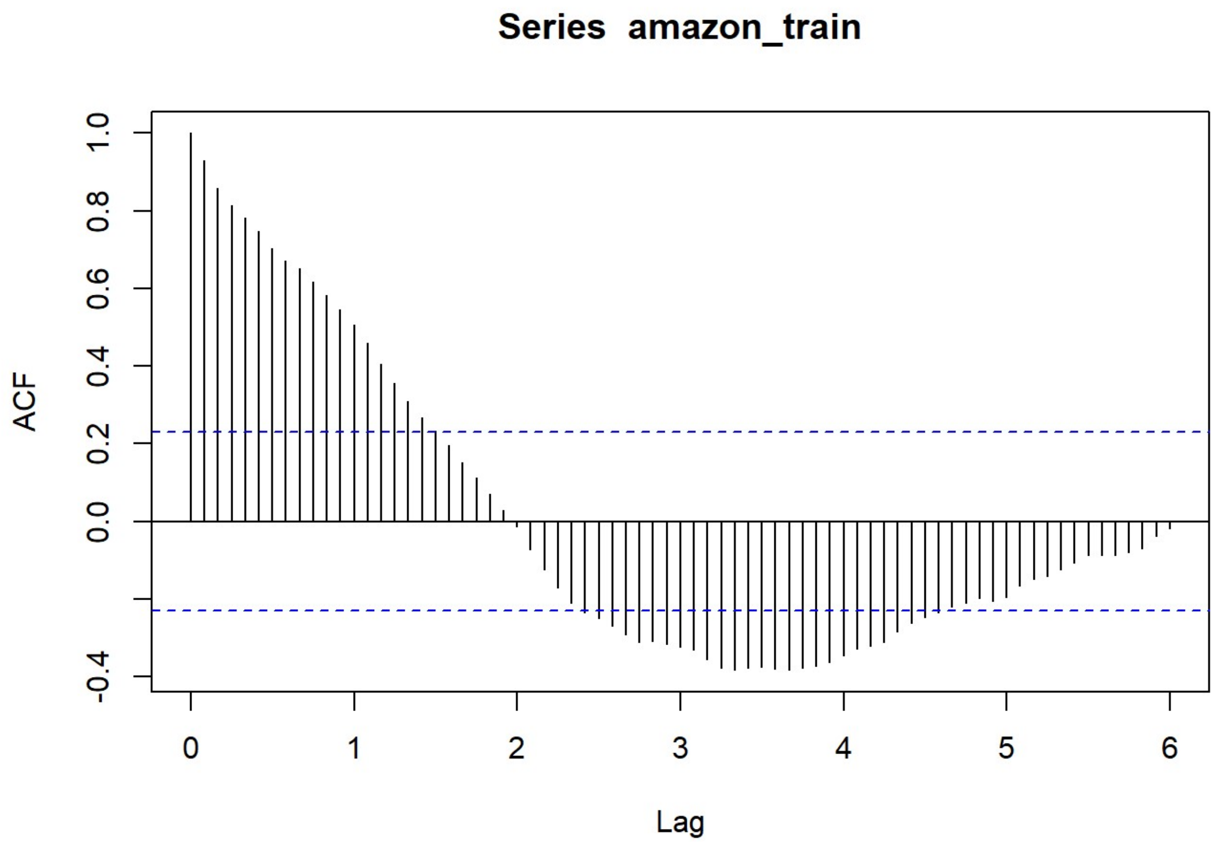

```
plot(as.ts(decompose_amazon$random))
```




```
plot(decompose_amazon)
```



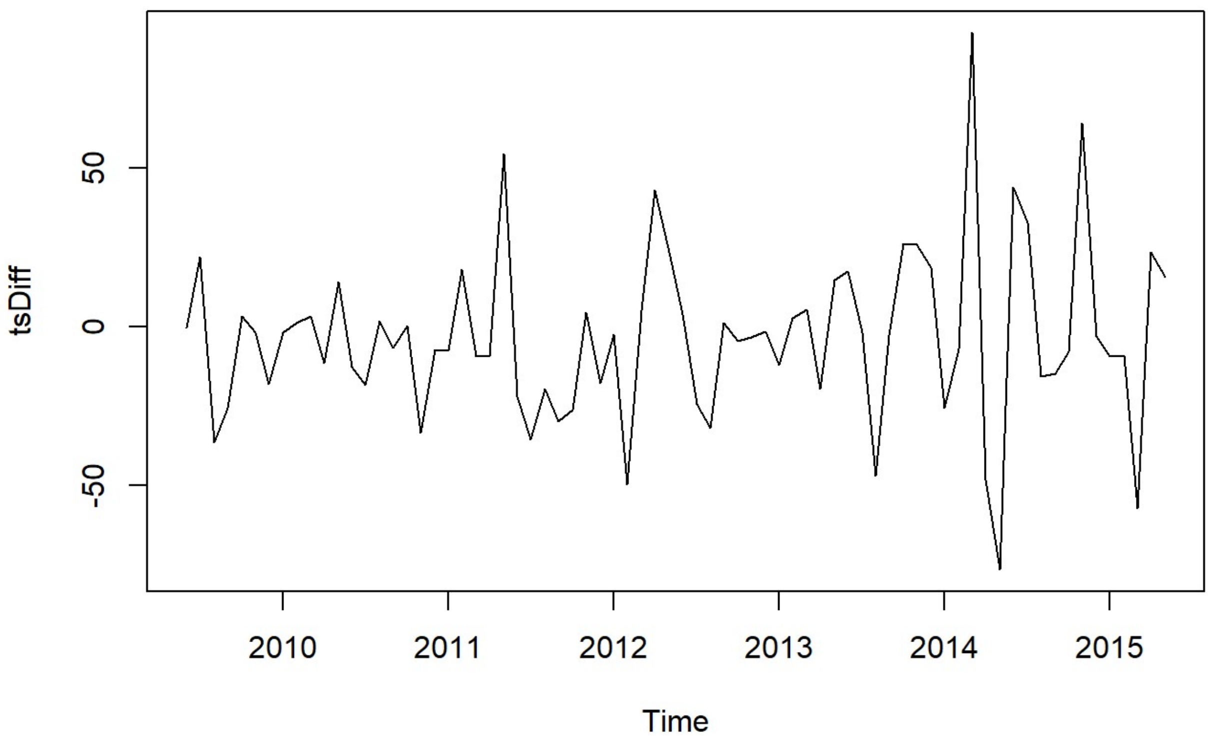

```
acf(amazon_train, 'Amazon')
```



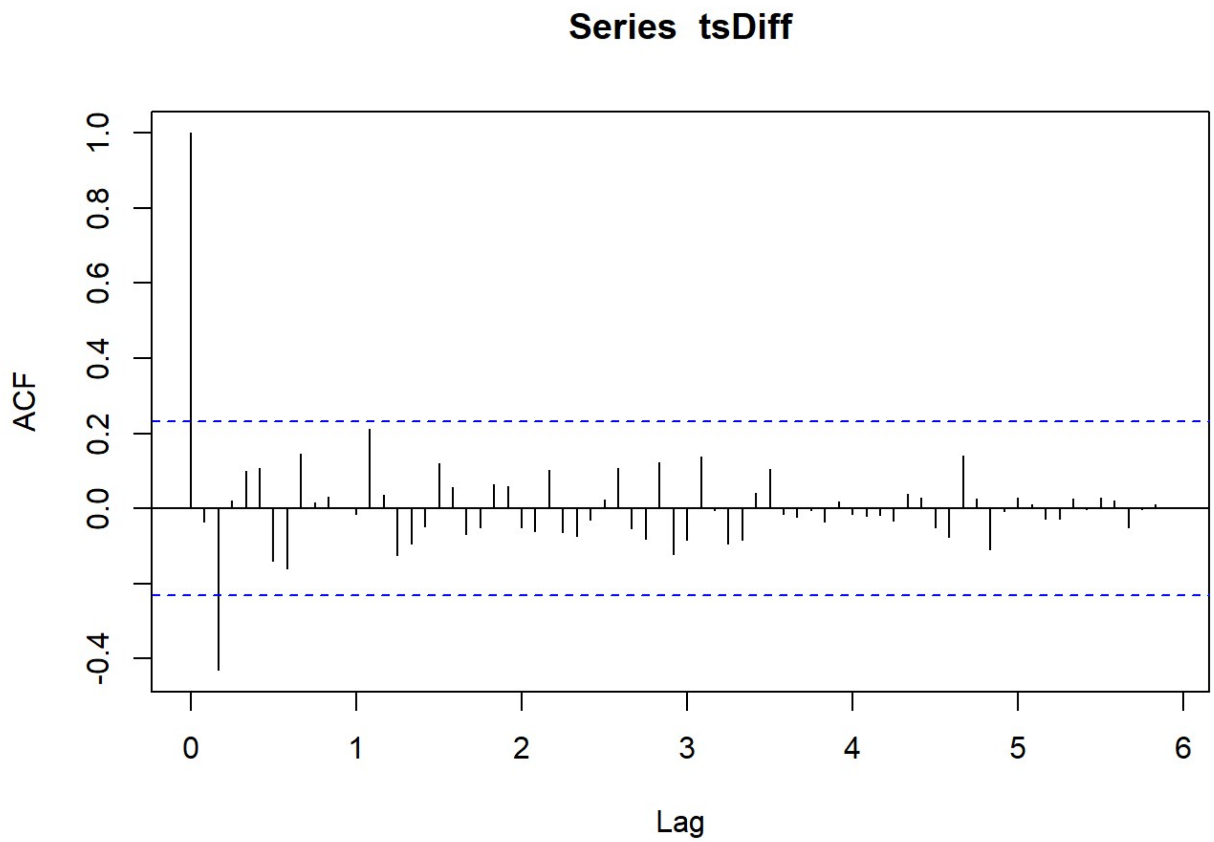

```
amz<-auto.arima(amazon_train)
amz
```

```
## Series: amazon_train
## ARIMA(2,1,0) with drift
##
## Coefficients:
##           ar1      ar2    drift
##       -0.0544  -0.4361  -3.7352
## s.e.   0.1060   0.1056   1.9339
##
## sigma^2 estimated as 613:  log likelihood=-331.9
## AIC=671.8   AICc=672.4   BIC=680.91
```

```
tsDiff <- diff(amazon_train)
plot.ts(tsDiff)
```




```
acf(tsDiff, 'S&P 500')
```




```
amztff<-auto.arima(tsDiff)

refitamz <- Arima(amazon_train, model=amz)
 accuracy(refitamz)
```

```
##                     ME     RMSE      MAE         MPE     MAPE    MASE
## Training set 0.1610269 24.07131 17.36049 0.006074507 1.020861 0.28047
##                      ACF1
## Training set -0.002433989
```

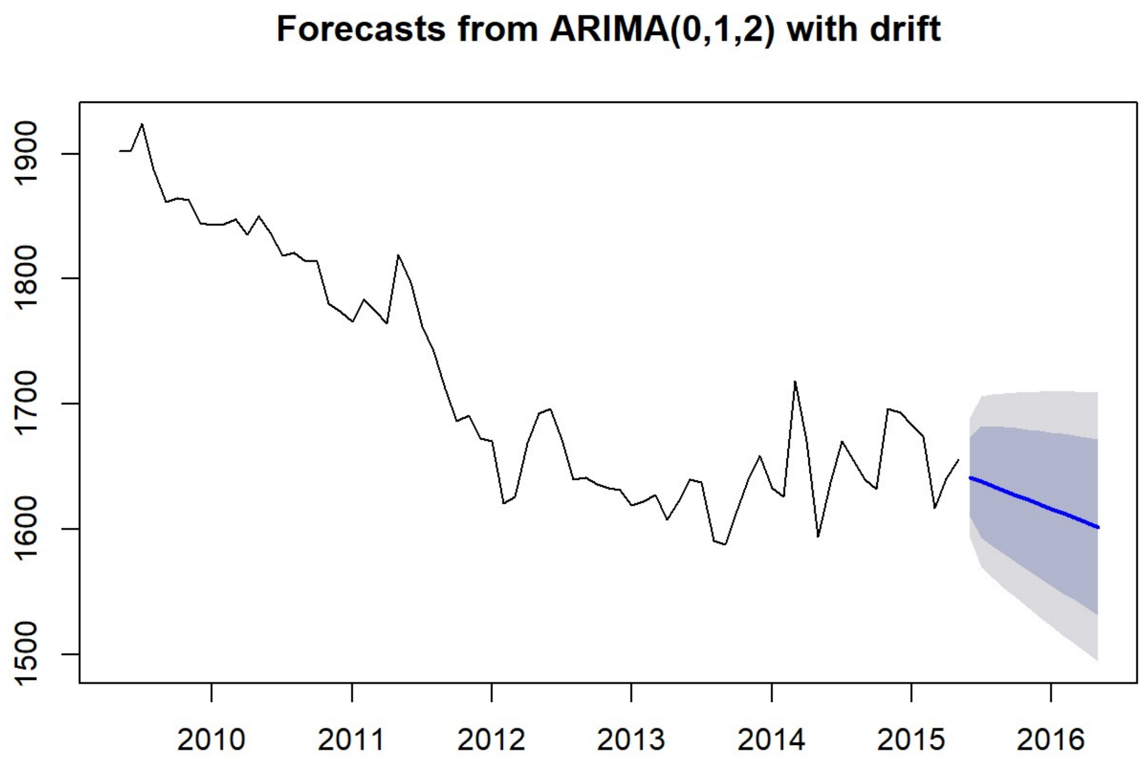
```
refittff <- Arima(tsDiff, model=amztff)
 accuracy(refittff)
```

```
##                     ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.1367866 24.23684 17.57513 85.64637 172.5466 0.6182221
##                     ACF1
## Training set -0.00254791
```

```r
####since RMSE is lower for amazon_train we will use auto arima amz

fit <- Arima(amazon_train, order = c(0,1,2), include.drift = TRUE)
summary(fit)
```

```
## Series: amazon_train
## ARIMA(0,1,2) with drift
##
## Coefficients:
##           ma1      ma2    drift
##       -0.0007  -0.4567  -3.5988
## s.e.   0.1005   0.0936   1.5806
##
## sigma^2 estimated as 607.4:  log likelihood=-331.6
## AIC=671.19   AICc=671.79   BIC=680.3
##
## Training set error measures:
##                      ME     RMSE      MAE           MPE     MAPE      MASE
## Training set 0.01659821 23.96153 16.94971 -0.0004708825 0.996014 0.2738335
##                     ACF1
## Training set -0.03001029
```

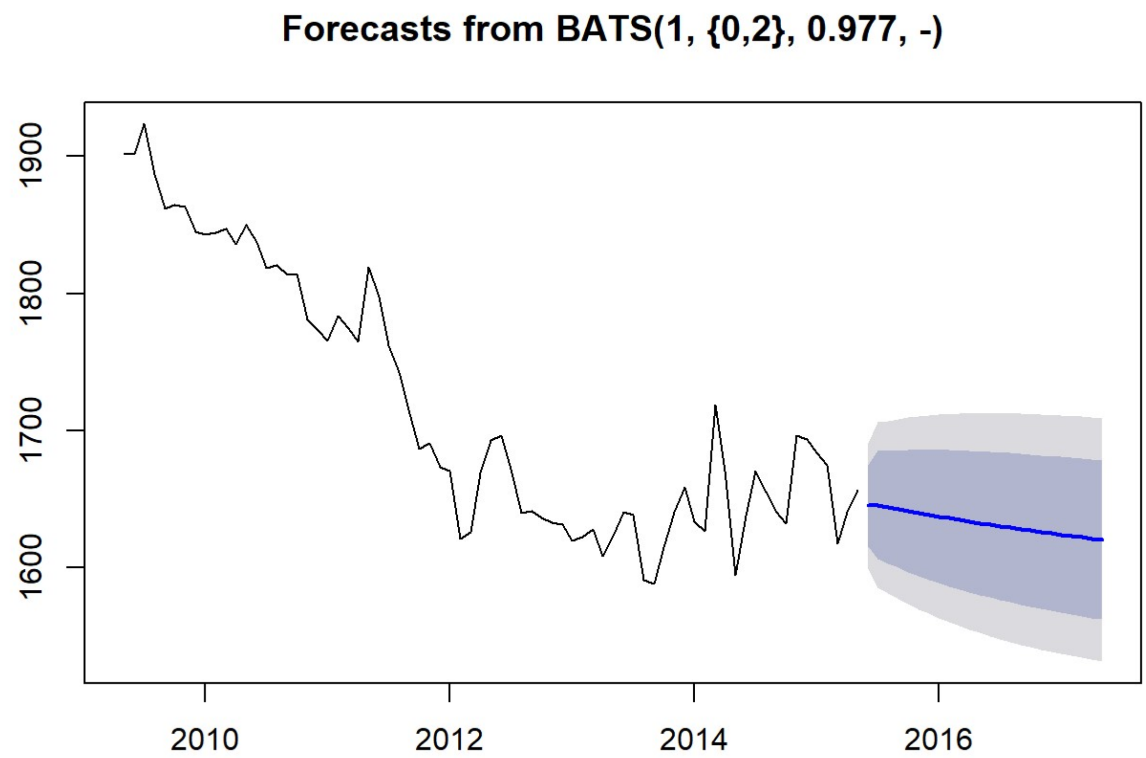
```r
for_amazon_all <- forecast(fit, h = 12)
 plot(for_amazon_all)
```




```
m_tbats = tbats(amazon_train)
accuracy(m_tbats)
```

```
##                     ME     RMSE     MAE         MPE      MAPE      MASE
## Training set -0.396802 23.30304 16.2938 -0.04088253 0.9573467 0.8316879
##                    ACF1
## Training set 0.02458858
```

```
f_tbats = forecast(m_tbats, h=24)
plot(f_tbats)
```


Forecasts from BATS(1, {0,2}, 0.977, -)
1600
1700
1800
1900
2010
2012
2014
2016