Mini – Project

Report on Time Series Forecasting

Report By

Chakradhar Varma

Table of Contents

1. Project Objective

The objective of the project is to perform time series forecasting techniques on the given data, which shows monthly demand of two different types of consumable items in a certain store from January 2002 to September 2017 and to predict sales for the period October 2017 to December 2018 and to conclude the decisions as a store manager by looking at the demand. This is a typical supply chain problem with short term forecasting.

So as part of the project the preliminary analysis, visualization of time series, partition of data, build different models, selection of an appropriate model and prediction for the future is performed.

2. Solutions

This Solutions section will explain each part of the project in the following steps:

1. Preliminary and Exploratory Data Analysis.
2. Decomposition of Time Series.
3. Model Building and Selection.
4. Forecasting.
5. Conclusion.

The Source code for all the above steps is attached in the Appendix A Section.

2.1 Preliminary and Exploratory Data Analysis

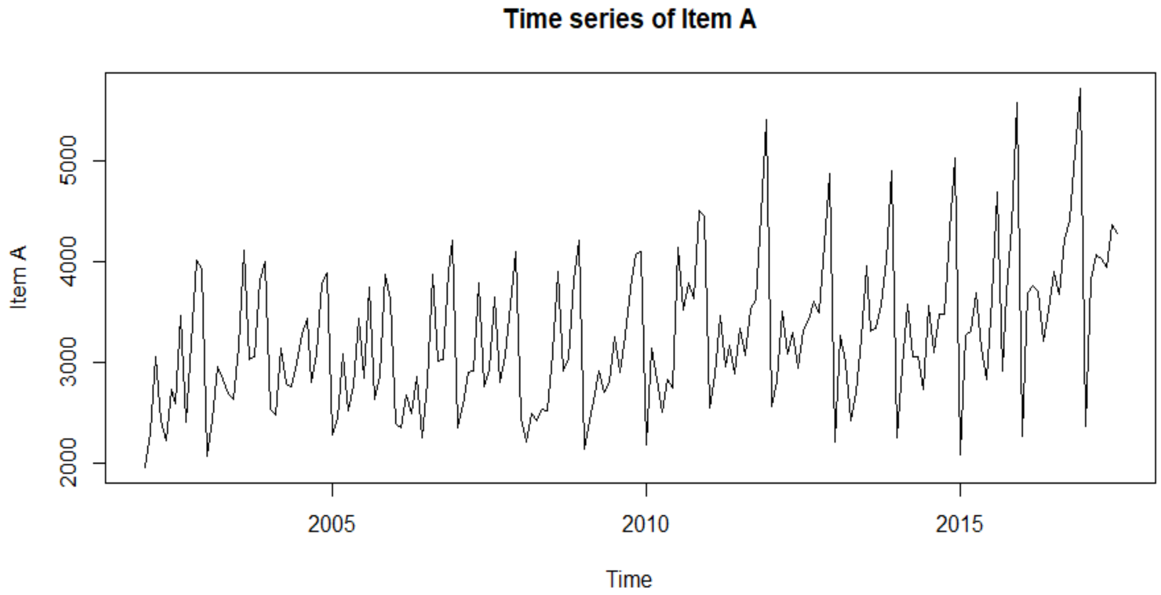
The data provided is of a certain store's monthly demand of two different Items. There are total of 187 data points are present from Jan 2002 to July 2017.

It has all the features of class, numeric type. These features are basically the year, month, demand of Item A and demand of Item B. The data has been tested for any missing values and found out that there is no missing values in the series. Five point summary of both the items across the series is as below.

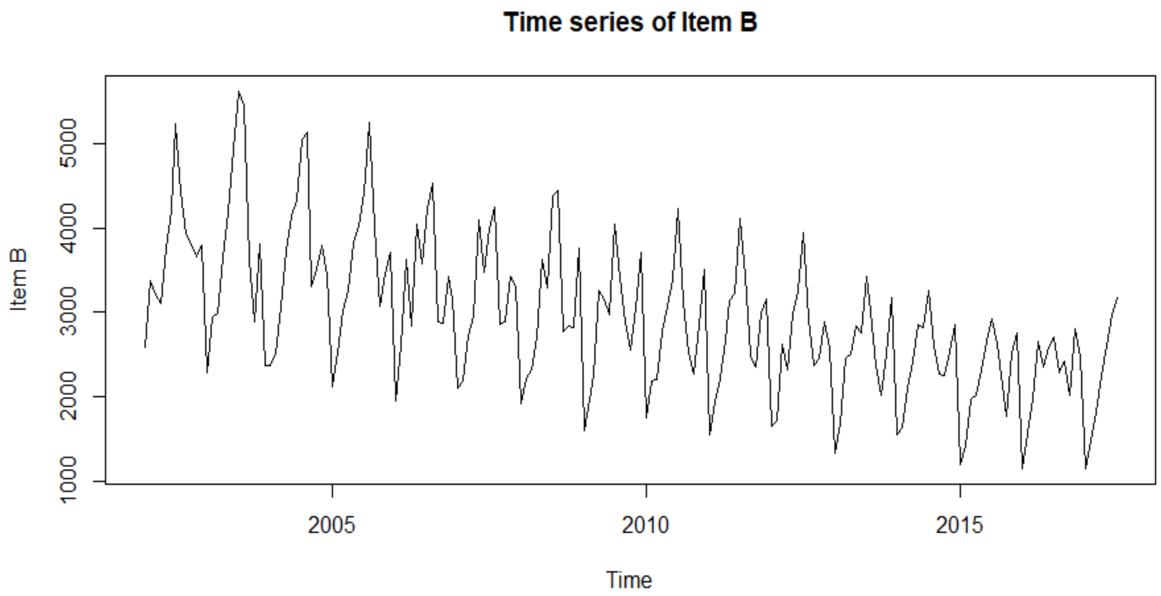
Item A		Item B	
Min.	1954	Min.	1153
1st Qu.	2748	1st Qu.	2362
Median	3134	Median	2876
Mean	3263	Mean	2962
3rd Qu.	3741	3rd Qu.	3468
Max.	5725	Max.	5618

2.1.1 Time Series Exploration

As part of the Preliminary Analysis, time series objects are created for both the items in order to plot and visualize the series. From the below plots it is observed that for Item-A, there is constant demand till 2010 and Increase in demand from 2010. No perceptible trend (signs of it only towards the end) and it has seasonality. There is no cyclicality exists, no abrupt changes and there are no outliers as well.

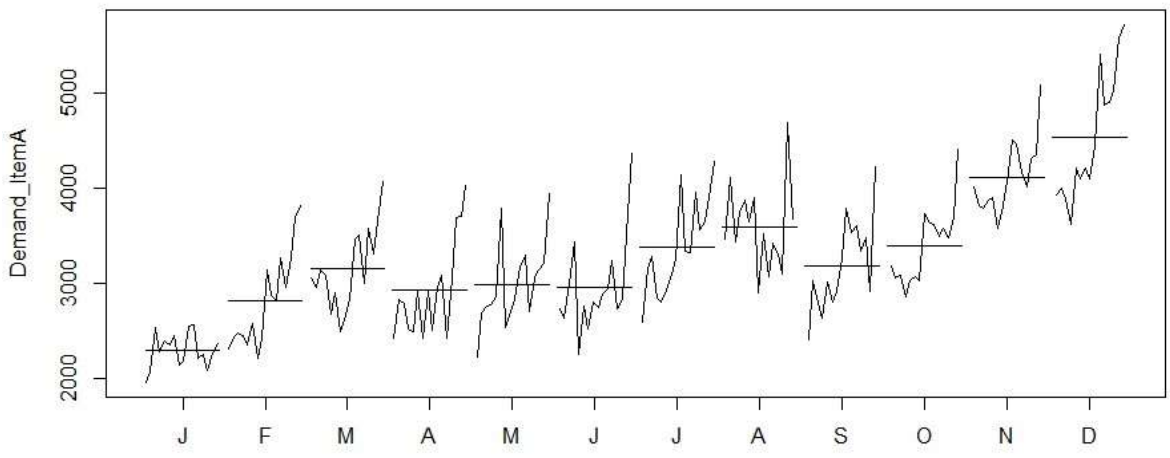


For Item-B, decrease in the demand. Clearly a decreasing trend and seems to have seasonality. There is no cyclicality exists, no abrupt changes and there are no outliers as well.

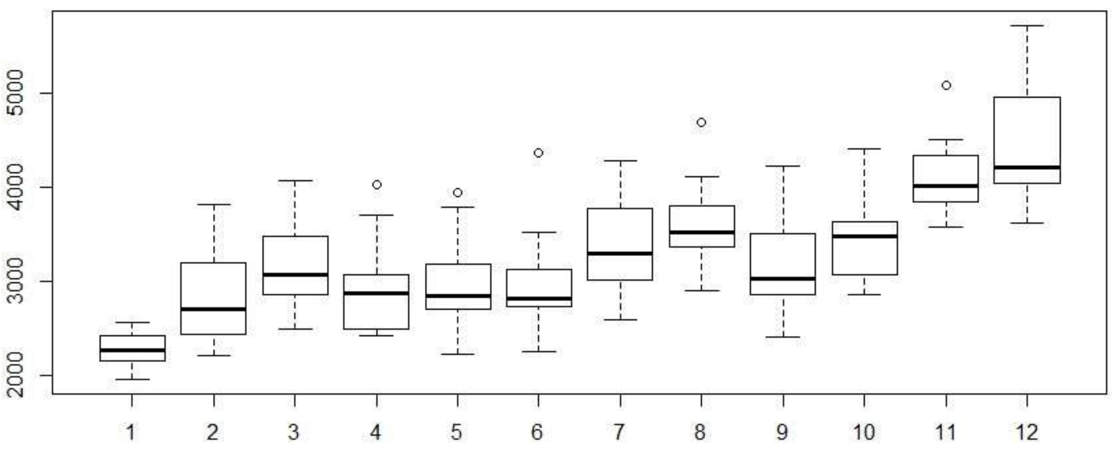


2.1.2 Seasonal Analysis

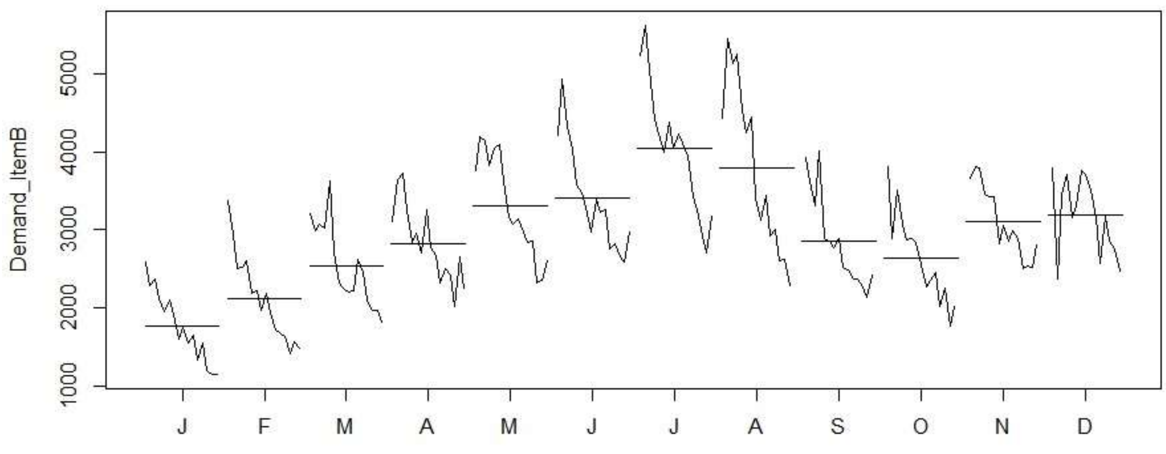
Before a formal extraction of time series components is done, seasonal changes in the data for the two series is explored. Through monthly plot and boxplot for each season, seasonality changes and the variability in the season is explored for both the series. The plots for the same are presented below.

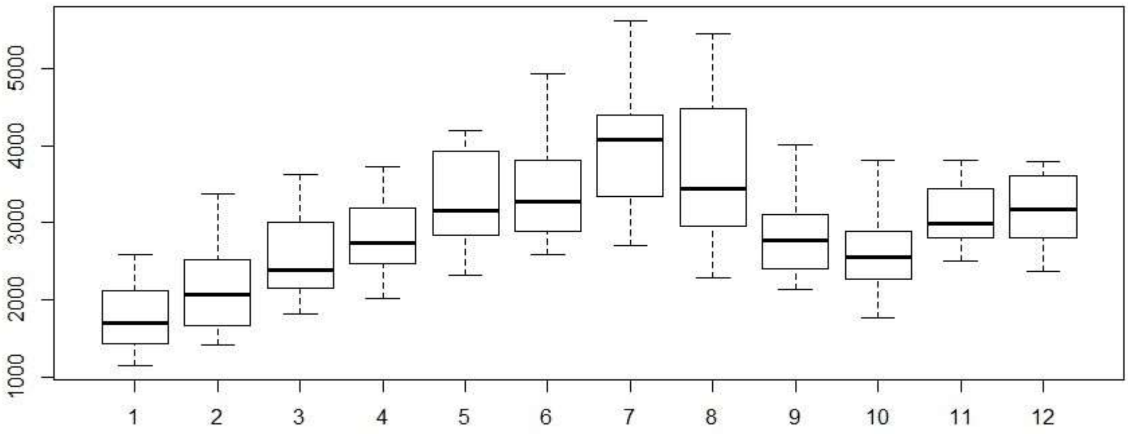


Demand in the month of January is low and it is almost constant from February to October except there is slightly higher demand in August and July. Demand in the months of November and December is comparatively high across all years. Variability in the month of January is low and high in December compared to other months.



For Item-B, overall demand is more in the months of July and August. After 2009 the demand in July month is more. Demand in the month of January is low and it is increasing till July and then drops till October. November and December months looks to maintain constant Demand. There is more variability in the month of August.





2.2 Decomposition of Time Series

A time series decomposition is procedure which transform a time series into multiple different series. The original time series is often computed (decompose) into 3 sub-time series:

1. **Seasonal:** patterns that repeat with fixed period of time.
2. **Trend:** the underlying trend of the metrics.
3. **Random:** (also call “noise”, “Irregular” or “Remainder”) is the residuals of the time series after allocation into the seasonal and trends time series.

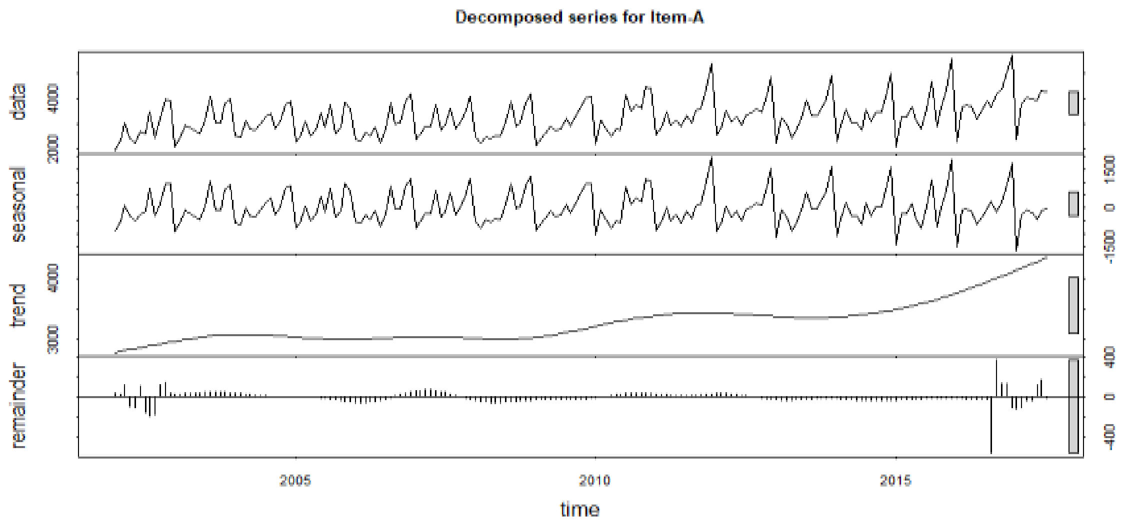
Other than above three component there is **cyclic** component which occurs after long period of time. To get a successful decomposition, it is important to choose between the additive or multiplicative model.

a. The **additive model** is useful when the magnitude of the seasonal pattern in the data does not depend on the magnitude of the data. In other words, the magnitude of the seasonal pattern does not change as the series goes up or down.
b. The **multiplicative model** is useful when the magnitude of the seasonal pattern increases as the data values increase, and decreases as the data values decrease.

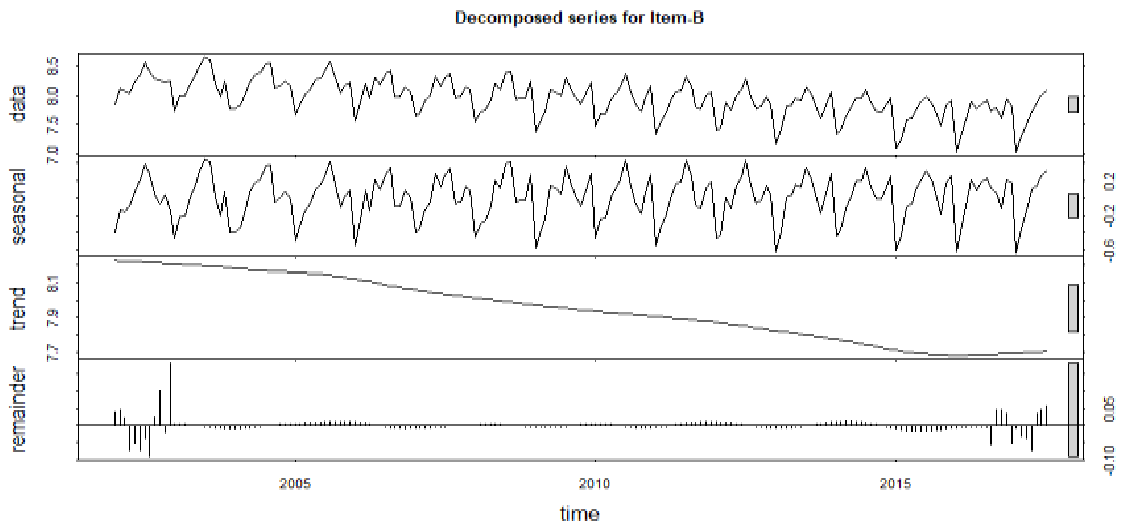
2.2.1 Decomposition – Seasonality Model Selection

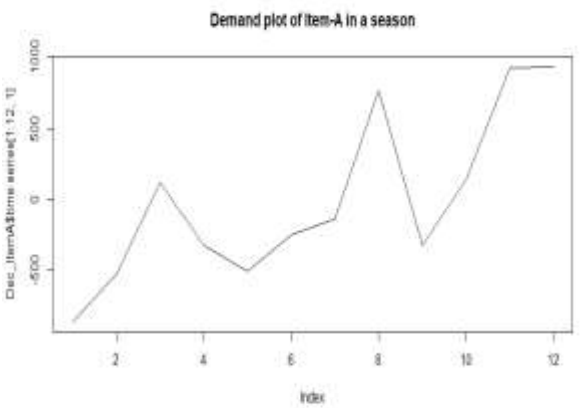
The given data set for two items is decomposed using appropriate method as well as the seasonality window value. For Item-A additive seasonality is chosen as the seasonal variation looks to be constant across the most part of the series and for Item-B, multiplicative model is appropriate as the seasonal variation decreases over the time. Below are the plots showing the decomposed series for both the items.

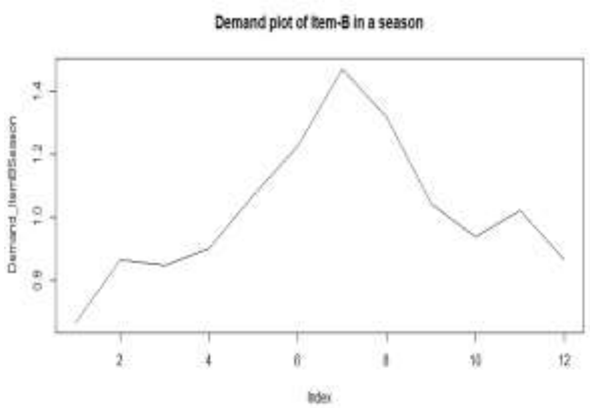
For Item-A using an additive model with window value of 3, the decomposed series is obtained as shown below. From the plot, the grey scale on the right indicates seasonality to be a significant component. It is observed that December month produce higher sales and January month sales are low.



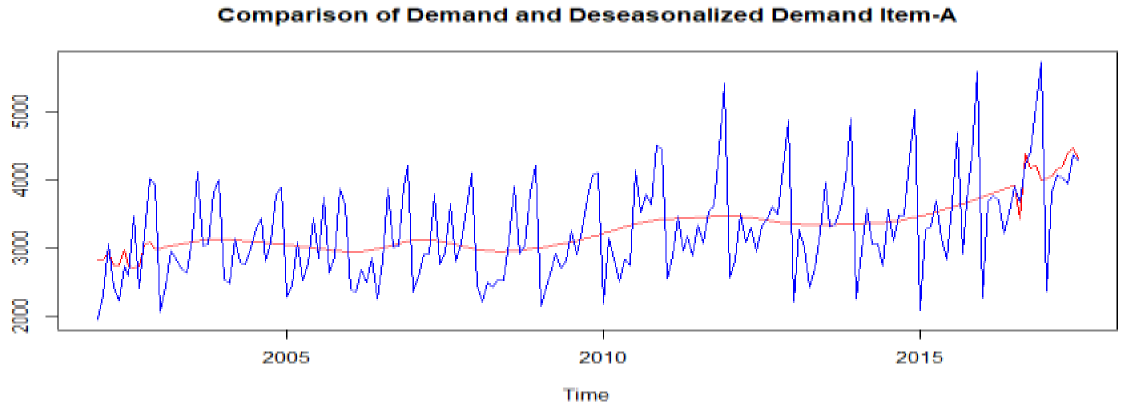
For Item-B using multiplicative model with window value of 3, the decomposed series is obtained as shown below. Since the data is transformed into logarithmic scale the plot shows the same scale and from the plot the grey scale on the right indicates seasonality to be a significant component. It is observed that July month has higher sales and January month sales are low.

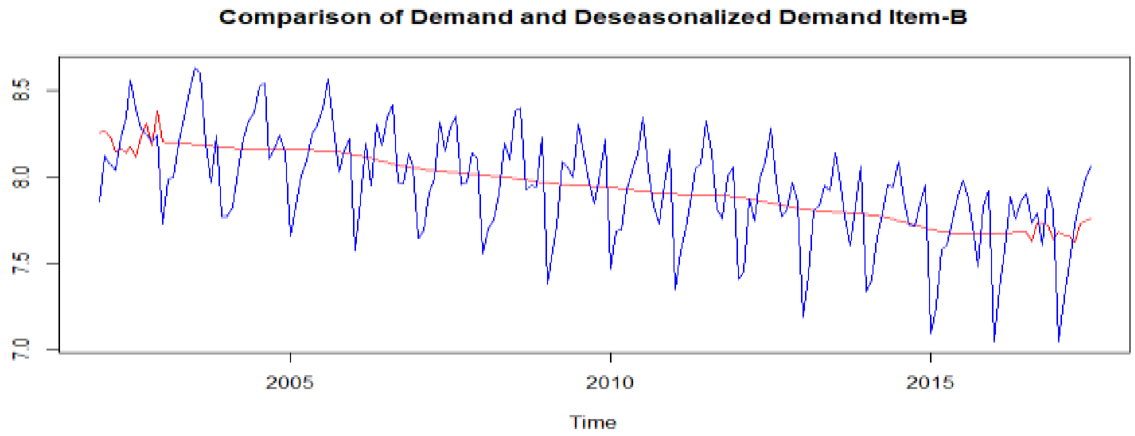






For both the items seasonality is significant and for Item A the demand is higher towards the end and for Item B the demand is higher around the middle of the ear. The demand is low in the month of January for both the items. The demand for Item-A seems to be increasing whereas for Item-B it is decreasing over the period. In the Deseasonalized demand plot the fluctuations are observed only at the start and end of the series due to the residuals. The Deseasonalized plots for both items are as obtained as below.





2.2.2 Stationarity Test of Residuals

A stationary time series is one whose properties do not depend on the time at which the series is observed. So, the series with trends or with seasonality - are not stationary - the trend and seasonality affect the values of time series at different times. A series is stationary when no matter when you observe it, it should look much the same at any period of time.

A time series with cyclic behavior (but not trend or seasonality) is stationary. The cycles are not always of the same length, so we cannot be sure of the peaks and troughs of the cycle before we observe the series. In general, a stationary time series will have no predictable patterns in the long-term. Time plots will show the series to be roughly horizontal (except for some cyclic behavior) with constant variance.

It is asked to extract the residuals from the decomposed series of both items and check for stationarity. The residuals are nothing but the remainder obtained from the decomposed series data and to check stationarity, ADF test is used by assuming Null hypothesis as ‘series is non-stationary’ and alternative hypothesis as ‘series is stationary’. Below is the results of the ADF test for both Item’s residuals.

Ho = Series is non-stationary.
Ha = Series is stationary.

```
Augmented Dickey-Fuller Test

data:  Dec_ItemA$time.series[, 3]
Dickey-Fuller = -4.4097, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

For Item-A, p-value comes out to be less than 0.05, so we reject the Null hypothesis. Hence the residuals form a stationary series.

```
Augmented Dickey-Fuller Test

data:  exp(logDec_ItemB$time.series[, 3])
Dickey-Fuller = -8.2618, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

Similarly for Item-B also p-value comes out to be less than 0.05, so we reject the null hypothesis and accept alternative hypothesis. Hence the residuals form a stationary series.

2.3 Model Building

The following models are built by partitioning the time series to create a training and hold-out samples and to forecast the demand for the hold-out period and to choose the model based on the performance.

- Forecast by Decomposition.
- Exponential smoothing (Holt-Winters Model)

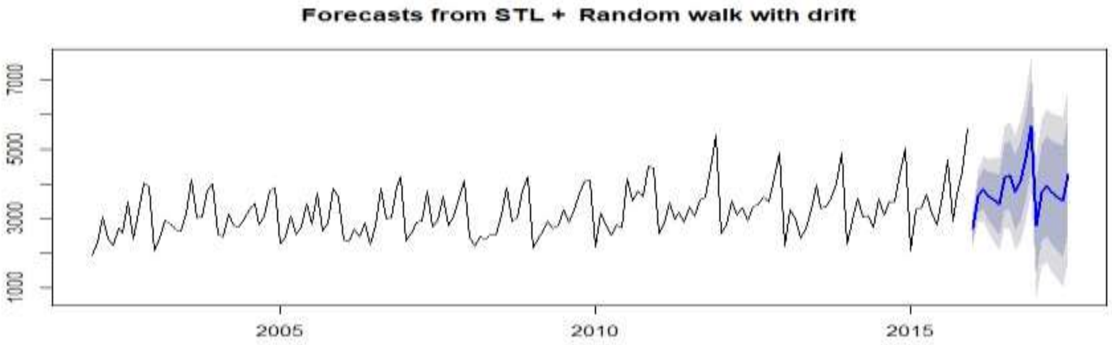
2.3.1 Data Partition

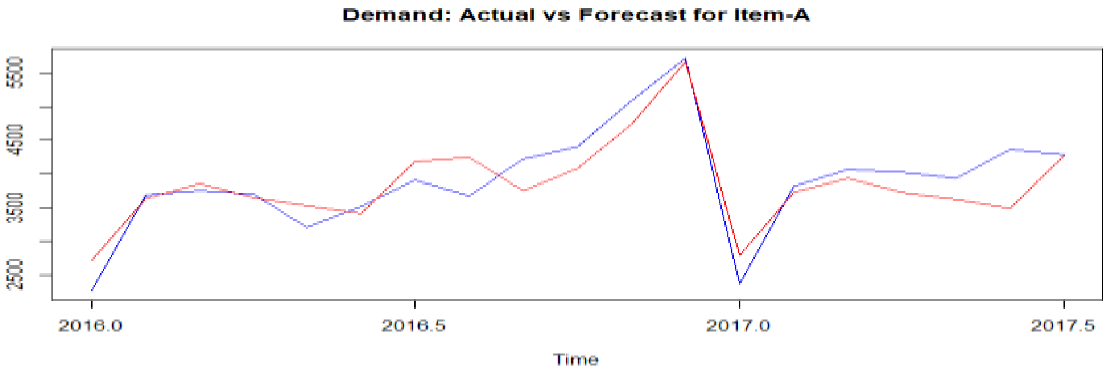
To partition the time series, at least one complete season should be present in the hold-out sample or it can be divided on the 80:20 rule also. So the last 19 months of the series is considered as hold-out sample and the rest as the training series. Two objects are created one for the training sample and other for the testing purpose using window method.

2.3.2 Forecast by Decomposition

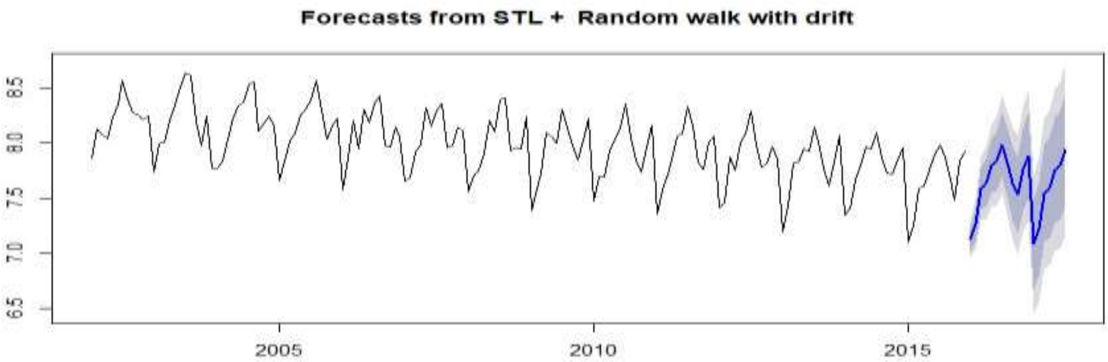
The training series is decomposed using appropriate seasonality model and the forecasting model is built using Random walk with drift method, which is a naïve method. The below plot shows the forecasted series for 19 months i.e., from 2016 Jan month to 2017 Sep month with 80% and 95% confidence interval.

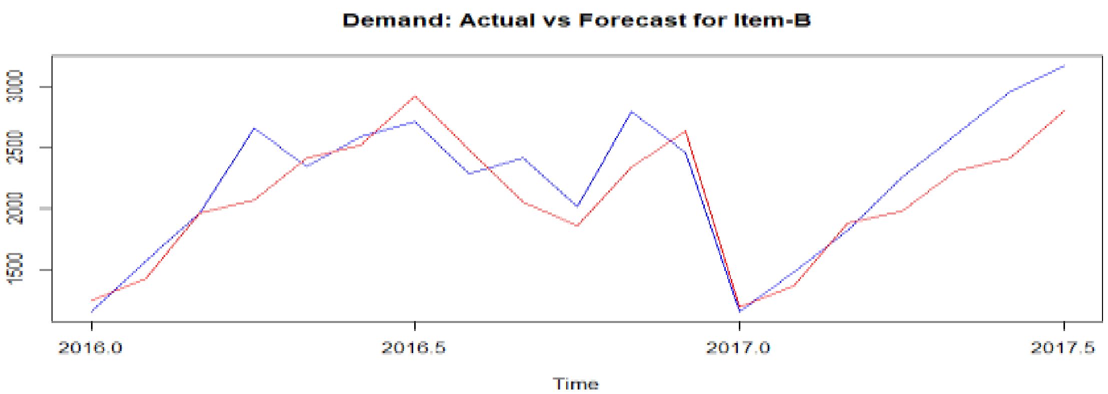
Further the forecasted series is plotted against the original hold-out period and the plot obtained is as below. Most of the forecasted series is in line with the original series except at the end, there is a big margin of difference.





So to gauge the performance of the model Mean Absolute Percentage Error (MAPE) is calculated and it came out to be **0.0774(7.7%)** which is considered to be a good value indicating a possible working model. In the similar way the demand for Item-B is also forecasted for the hold-out sample and the plots are as shown below.



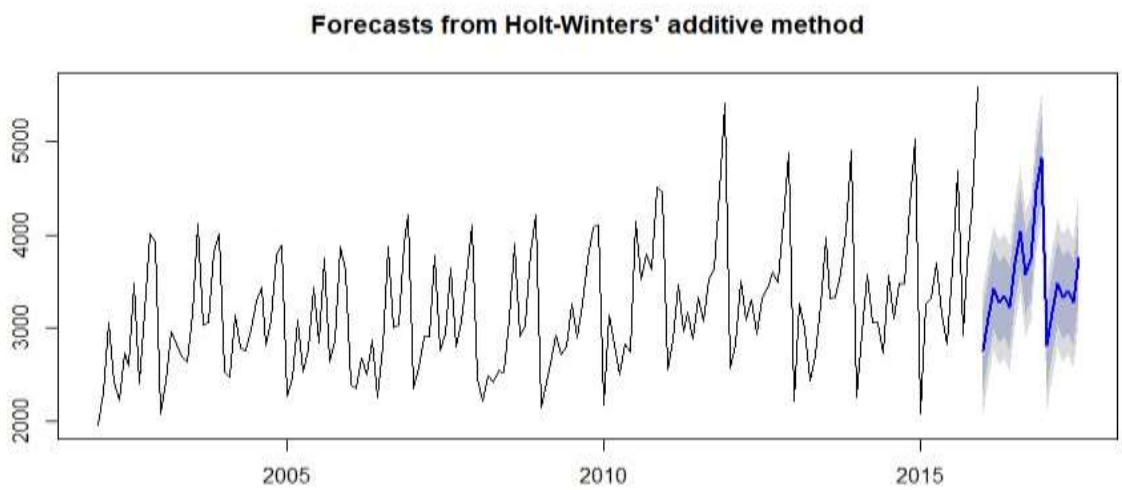


The MAPE value for forecasting decomposition model on Item-B has come out to be **0.0922 (9.22%)**, which is less than 10% and indicates a good working model. The forecasted values for the hold-out sample looks to be close to the original values for most of the series except for at the end. Now before forecasting the series for the future using decomposition model, exponential smoothing model also should be tried out as asked and compare both models.

2.3.3 Exponential Smoothing Model – Holt Winter's Model

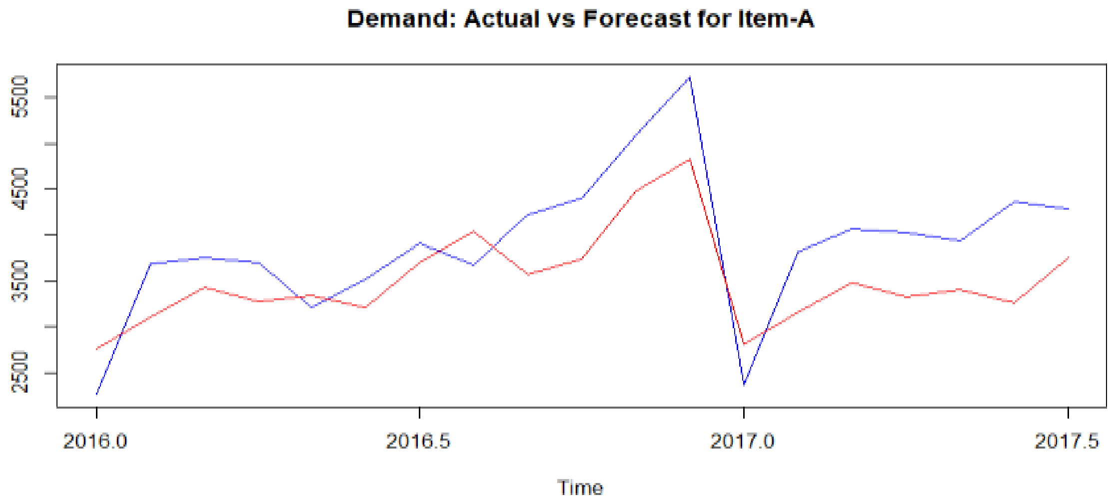
The Holt-Winters method is an exponential smoothing approach used for handling seasonal data. It comprises the forecast equation and three smoothing parameters — one for the level, one for trend, and one for the seasonal component, with smoothing parameters alpha, beta and gamma.

The Model is built on the same training series used earlier and with appropriate seasonal model for both the items using hw method. The below are the plots and model output for Item-A.

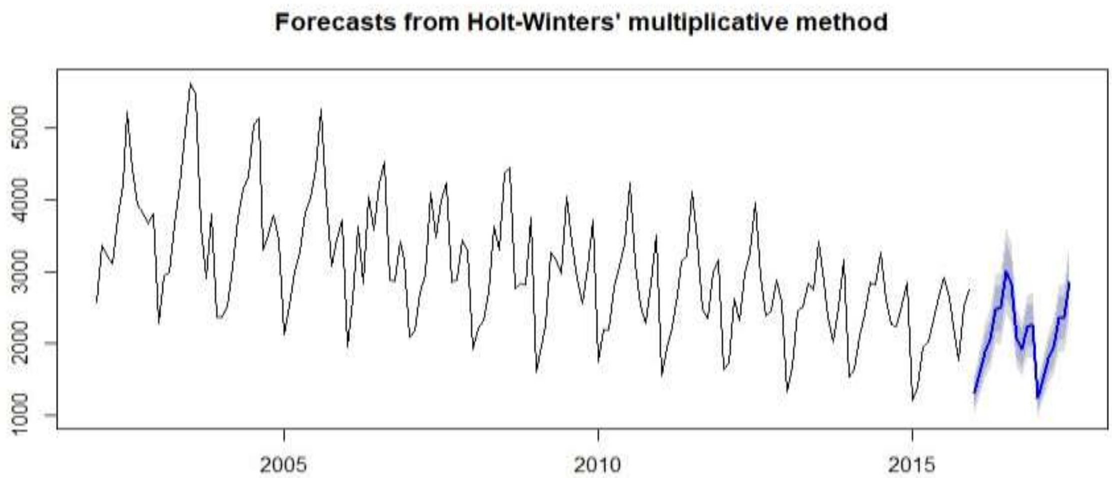



```
Holt-Winters' additive method

Call:
 hw(y = DataATrain, h = 19)

  Smoothing parameters:
    alpha = 0.0728
    beta  = 1e-04
    gamma = 1e-04

  Initial states:
    l = 3007.3727
    b = 4.6452
    s = 1182.799 836.5771 102.7667 -57.2941 404.6823 71.8966
           -407.3715 -269.9908 -342.3852 -179.8529 -496.6008 -845.226

  sigma:  336.7443

     AIC     AICc      BIC
2833.305 2837.385 2886.412
```

The model estimates alpha, beta and gamma are all come out to be very small and it indicates that there is almost a straight line trend as well as in-significant seasonality. The model is tested on the hold-out sample and the forecasted series is plotted against the original hold-out series, the plot obtained is as below.

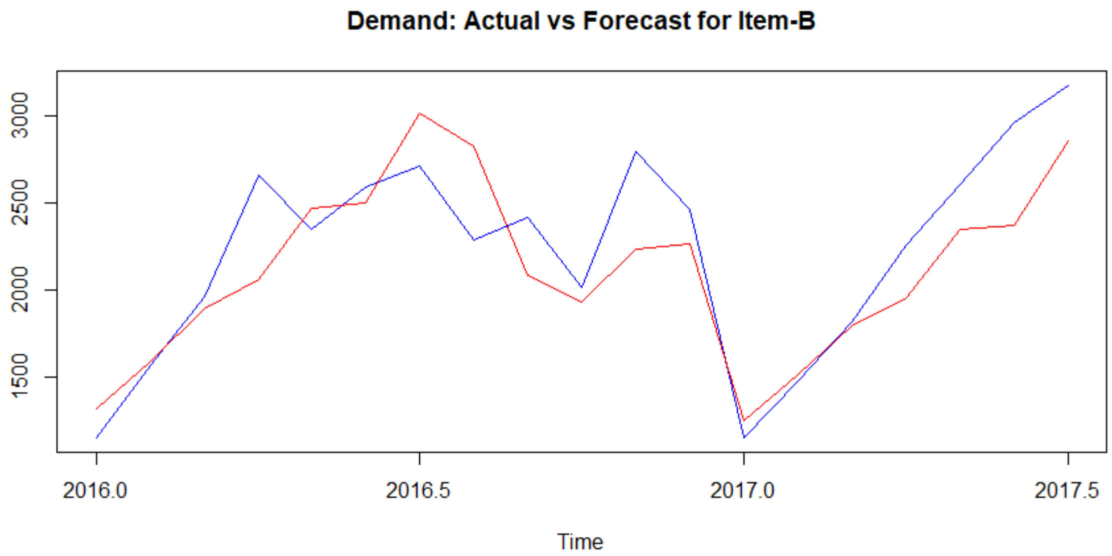


It can clearly be seen that the forecasted series is not so appropriate when compared to the forecasted model by decomposition. The MAPE value for the model is come out to be **0.1383 (13.8%)**, which is more than the desired value of 10% to be rated it as a good model. Similarly the model is built for Item-B as well using multiplicative seasonal method and the obtained results are as below.



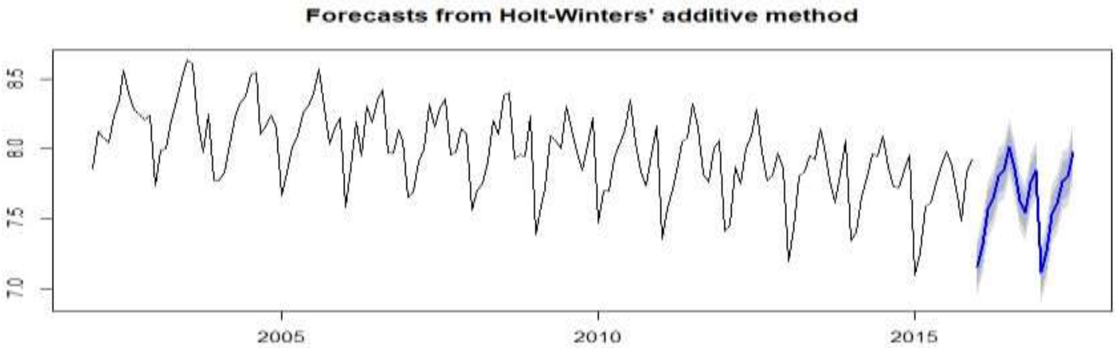
The model estimates alpha, beta and gamma are all come out to be very small, the beta value is higher than the gamma value which indicates slight significance of trend over seasonality. The model is tested on the hold-out sample and the forecasted series is plotted against the original hold-out series, the plot obtained is as below.

```
Holt-Winters' multiplicative method

Call:
 hw(y = DataBTrain, h = 19, seasonal = "multiplicative")

  Smoothing parameters:
    alpha = 0.0212
    beta  = 0.0016
    gamma = 1e-04

  Initial states:
    l = 4061.8957
    b = -12.621
    s = 1.0607 1.0409 0.8957 0.9634 1.3019 1.3818
           1.1411 1.1234 0.9337 0.8529 0.7151 0.5895

  sigma:  0.0992

     AIC     AICc      BIC
2782.834 2786.914 2835.942
```



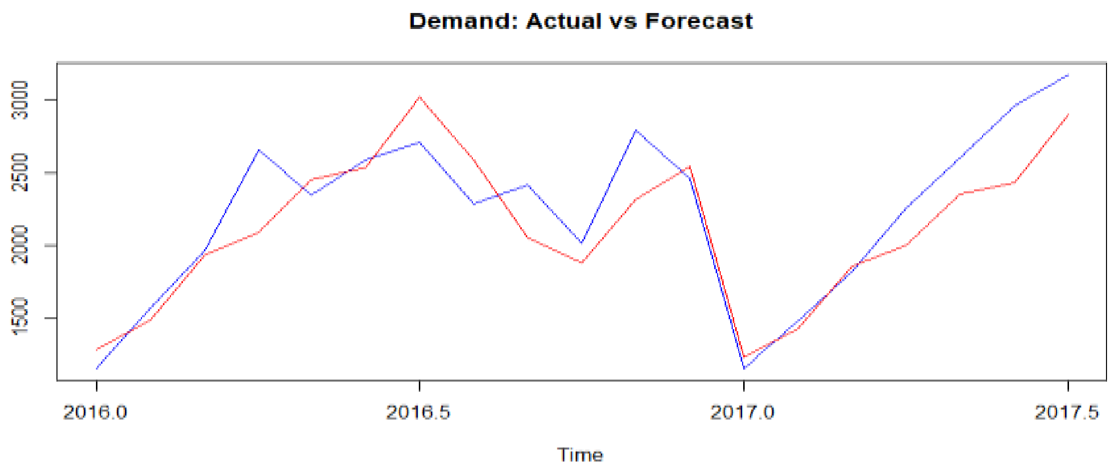
The model seems to be performed well for Item-B, as it is almost similar to the one that is obtained by forecasting by decomposition model and the MAPE value comes out to be **0.1035767(10.35%)**, which is acceptable and it indicates a working model.

Since multiplicative seasonal method is selected for the model, it is tried to see if transforming the series to logarithmic scale and use an additive seasonality to build the model would lead a more accurate model with low MAPE value. The results are as below.



```
Holt-Winters' additive method

Call:
 hw(y = log(DataBTrain), h = 19)

  Smoothing parameters:
    alpha = 1e-04
    beta  = 1e-04
    gamma = 0.2396

  Initial states:
    l = 8.2809
    b = -0.0035
    s = 0.0789 0.0222 -0.0417 -0.0314 0.2754 0.4015
           0.1728 0.1317 -0.0066 -0.1359 -0.3187 -0.5481

  sigma:  0.1021

     AIC     AICc      BIC
111.4846 115.5646 164.5920
```

The model estimates alpha, beta come out to be very small, the gamma value is higher than other values which indicates significant seasonality. The model is tested on the hold-out sample and the forecasted series is plotted against the original hold-out series, the plot obtained is as below.



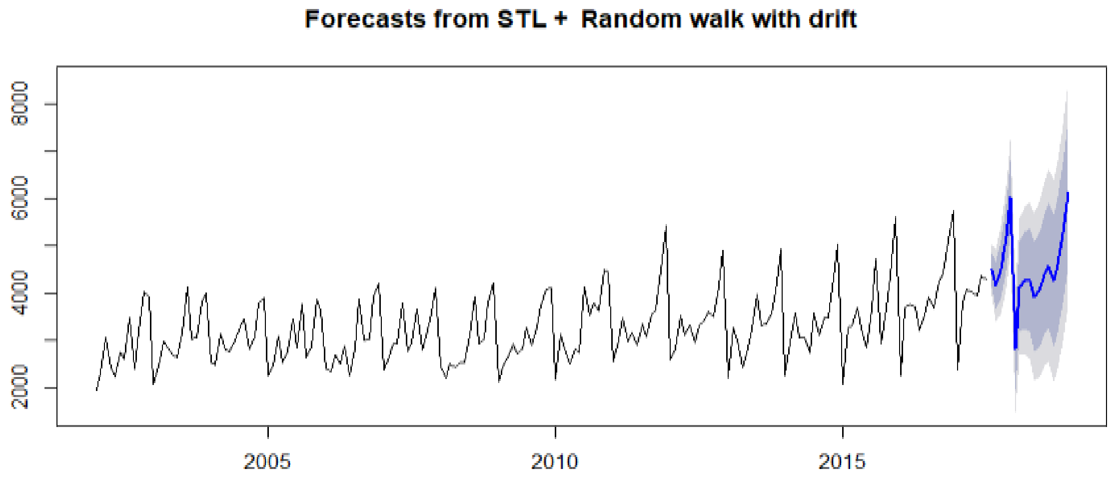
The model seems to be performed well for Item-B, with the MAPE value of **0.09030 (9.03%)**, which is less than 10% acceptance and it indicates a working model. So the value is less than what is obtained for Forecast decomposition model.

2.3.4 Conclusion

The forecasting models for both items has been built and the model accuracy in from of MAPE value is observed and by comparing the MAPE values for both models, **Forecasting by decomposition model** with value of **7.7%** is better working model for **Item-A** demand series and for **Item-B**, the **Holt-Winter's model** with logarithmic transformation and with additive seasonal method having MAPE of **9.03%** is the better performed model. So these models will be used to forecast for the future period from Aug 2017 to Dec 2018.

2.4 Forecasting for the Future Period

The best performing model is chosen for each item to forecast the future demand from Aug 2017 to Dec 2018. The plots and the forecasted data with 80% - 95% confidence interval are presented below.

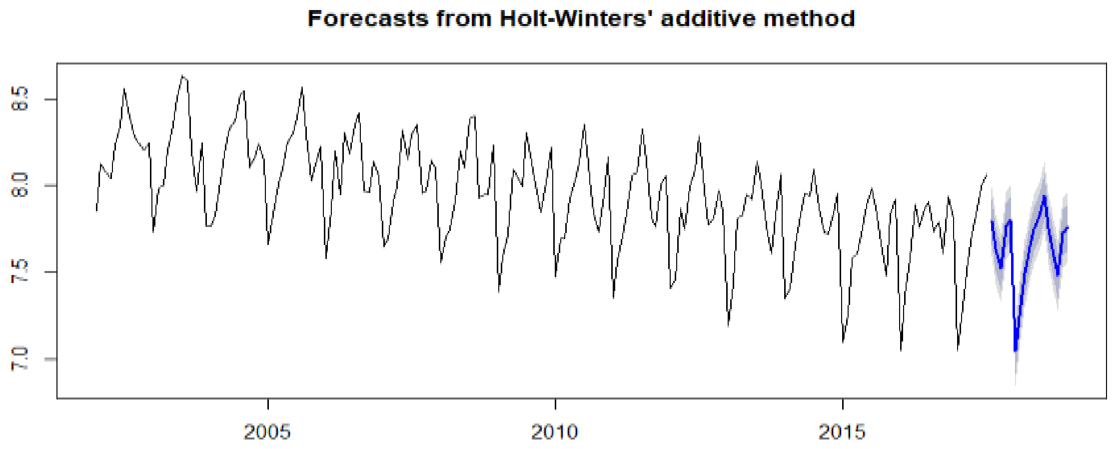


	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Aug-17	4482.661	4135.757	4829.564	3952.118	5013.204
Sep-17	4155.016	3661.804	4648.228	3400.713	4909.319
Oct-17	4485.163	3877.916	5092.409	3556.459	5413.866
Nov-17	5165.715	4460.866	5870.564	4087.741	6243.689
Dec-17	**6024.258**	5232.14	6816.377	4812.818	7235.699
Jan-18	**2806.48**	1934.319	3678.64	1472.625	4140.334
Feb-18	4114.391	3167.58	5061.202	2666.369	5562.413
Mar-18	4259.724	3242.467	5276.981	2703.964	5815.485
Apr-18	4263.406	3179.087	5347.725	2605.084	5921.728
May-18	3914.072	2765.483	5062.661	2157.457	5670.687
Jun-18	4053.246	2842.734	5263.758	2201.927	5904.564
Jul-18	4390.061	3119.629	5660.493	2447.103	6333.019
Aug-18	4582.722	3254.102	5911.341	2550.774	6614.67
Sep-18	4255.077	2869.785	5640.369	2136.455	6373.699
Oct-18	4585.224	3144.593	6025.854	2381.97	6788.477
Nov-18	5265.776	3770.995	6760.557	2979.706	7551.846
Dec-18	**6124.319**	4576.45	7672.188	3757.058	8491.581

Table 2.4.1 – Forecasted values for Item-A

As said earlier for Item-A demand for Dec month is high, so is the forecasted values for 2017 and 2018. For rest of the months the demand varies in range of 1000 units expect for the Jan month, which is less than half of the demand for December.

For Item-B, since the log transformation is used and performed a Holt-winter's model, the output plot is in the logarithmic scale. The forecasted values from the model are exponentiated to get the values in normal scale and presented in the table below.



	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Aug-17	2421.31	2126.058	2757.574	1984.625	2954.09
Sep-17	2068.049	1815.871	2355.249	1695.073	2523.094
Oct-17	1848.267	1622.888	2104.945	1514.928	2254.952
Nov-17	2359.432	2071.722	2687.098	1933.904	2878.592
Dec-17	2449.169	2150.516	2789.297	2007.457	2988.074
Jan-18	**1144.52**	1004.952	1303.459	938.0992	1396.349
Feb-18	1405.022	1233.693	1600.144	1151.623	1714.177
Mar-18	1790.641	1572.289	2039.316	1467.695	2184.646
Apr-18	2089.994	1835.139	2380.242	1713.059	2549.869
May-18	2316.845	2034.327	2638.597	1898.996	2826.634
Jun-18	2487.255	2183.957	2832.673	2038.673	3034.542
Jul-18	**2809.093**	2466.55	3199.207	2302.466	3427.196
Aug-18	2327.144	2034.785	2661.509	1895.195	2857.542
Sep-18	1987.619	1737.914	2273.201	1618.69	2440.633
Oct-18	1776.384	1553.217	2031.616	1446.663	2181.254
Nov-18	2267.669	1982.781	2593.49	1846.758	2784.513
Dec-18	2353.916	2058.192	2692.129	1916.996	2890.418

The demand for Item-B is observed to be high in the month of July, so is the forecasted value for 2018. For rest of the months the demand varies in range of 1000 units expect for the Jan month, which is also less than half of the highest demand in the season.

3. Conclusion – Decision Making

The main objective of the project is to predict sales for the period Aug 2017 to Dec 2018. This has been achieved by the analysing the time series, decomposing the series, building various models to see the performance and select appropriate final model to forecast for the given time period.

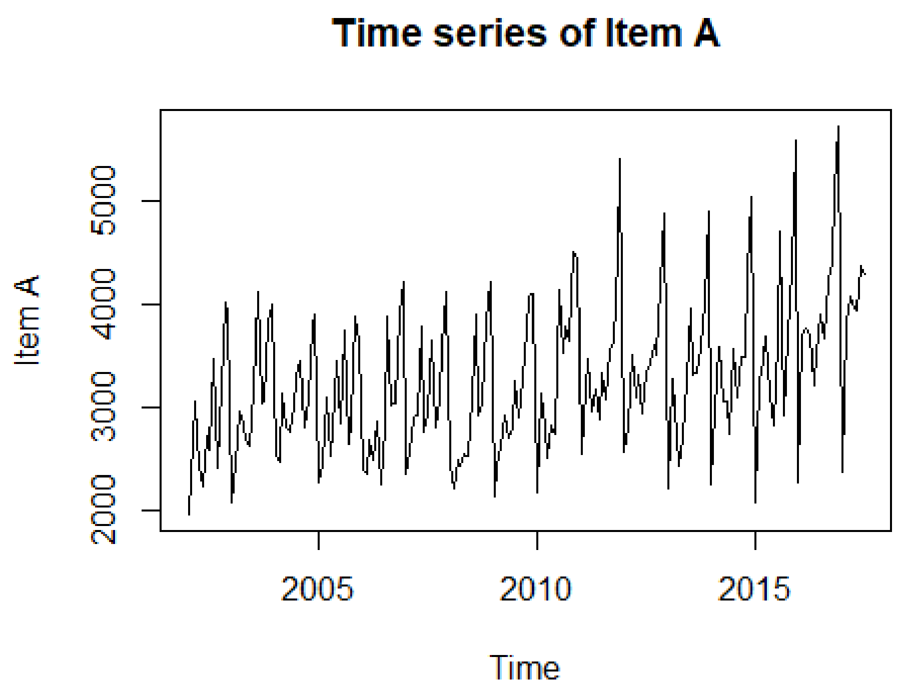
As a Store Manager after looking at the time series and the predicted sales for the year ahead, I would keep the stocks of Item-A available for the whole year starting from Feb to Oct as the demand for Item-A has been increased over the years and maintain constant demand across most of the months in an year. Additional stocks might be procured for the months of Nov and Dec as the demand is high in those months.

Similarly for Item-B more stocks would be required for the middle and towards final months of the year, so I would procure more stocks in the months of June, July, August and November, December as well to up the sales and meet the demand needs. Less stocks of Item-B would be suffice when compared to Item-A as the demand for Item-B is decreasing over the time period. So, an optimum stocks (by maximization) for Item-A and Item-B would be maintained to attain maximum profits.

4. Appendix A – Source Code

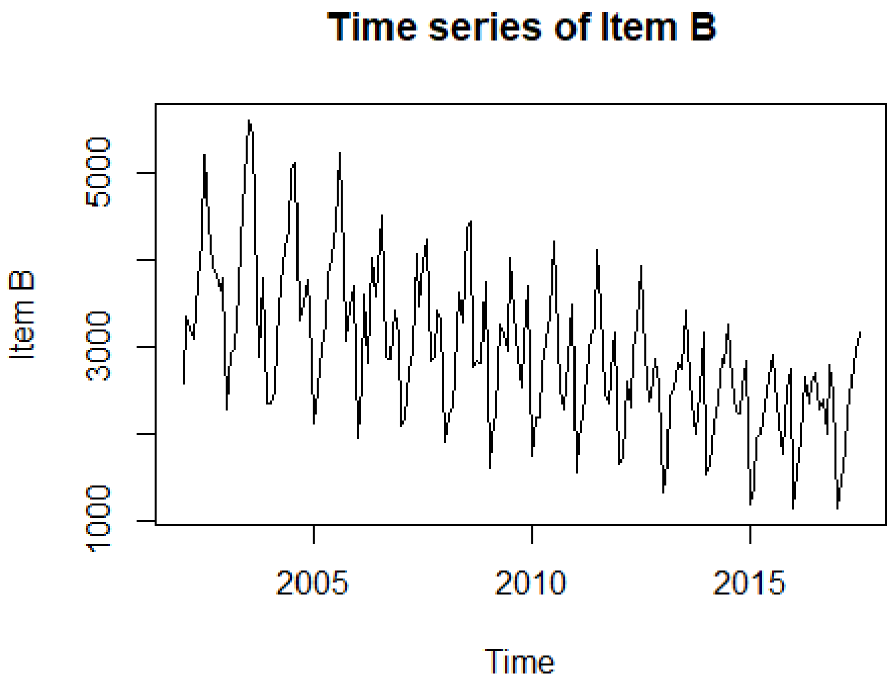
```
#set up of working directory
setwd("D:/BACP Program/R Directory")

#libraries

library(readxl)

library(forecast)

library(tseries)

#supply chain problem - short-term forecasting

#preliminary Analysis

##Importing the Demand data
Demand_data <- read_excel("Demand.xlsx",skip=1)
#View(Demand_data)

#checking for missing values
colSums(is.na(Demand_data))

##   Year  Month Item A Item B
##      0      0      0      0

#No missing values in the series

str(Demand_data)

## Classes 'tbl_df', 'tbl' and 'data.frame':    187 obs. of  4 variables:
##  $ Year  : num  2002 2002 2002 2002 2002 ...
##  $ Month : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ Item A: num  1954 2302 3054 2414 2226 ...
##  $ Item B: num  2585 3368 3210 3111 3756 ...

#187 observations of 4 variables
#Monthly series

summary(Demand_data[3])

##      Item A
##  Min.   :1954
```

```
##   1st Qu.:2748
##   Median :3134
##   Mean   :3263
##   3rd Qu.:3741
##   Max.   :5725

#Min. demand - 1954 units, Max. demand - 5725

summary(Demand_data[4])

##       Item B
##   Min.   :1153
##   1st Qu.:2362
##   Median :2876
##   Mean   :2962
##   3rd Qu.:3468
##   Max.   :5618

#Min. demand - 1153 units, Max. demand - 5618


#creating Time series Objects
Demand_ItemA <- ts(Demand_data[3], start=c(2002,1), end=c(2017,7), frequen
cy=12)
plot(Demand_ItemA, main="Time series of Item A")
```

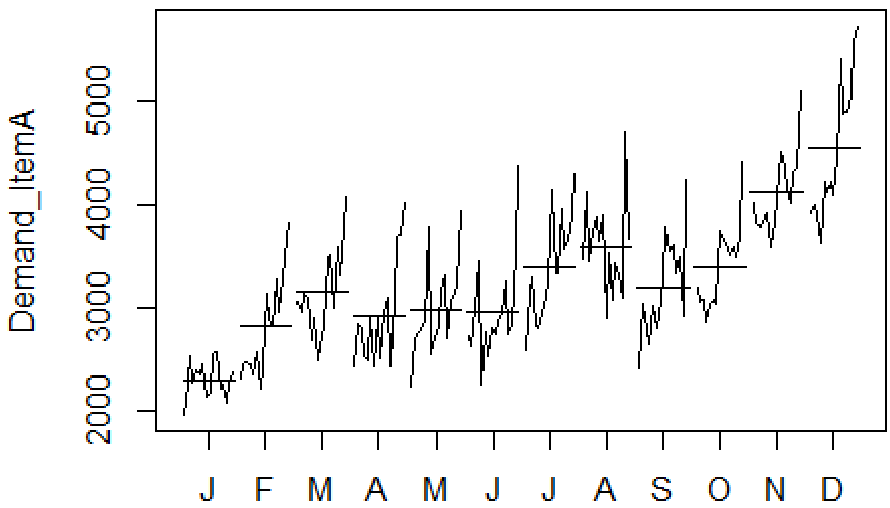



```
#Constant Demand till 2010 and Increase in Demand from 2010. No perceptibl
e trend (signs of it only towards the end), has seasonality.
#No cyclicality exists. No outliers and No abrupt Change.

Demand_ItemB <- ts(Demand_data[4], start=c(2002,1), end=c(2017,7), frequen
```

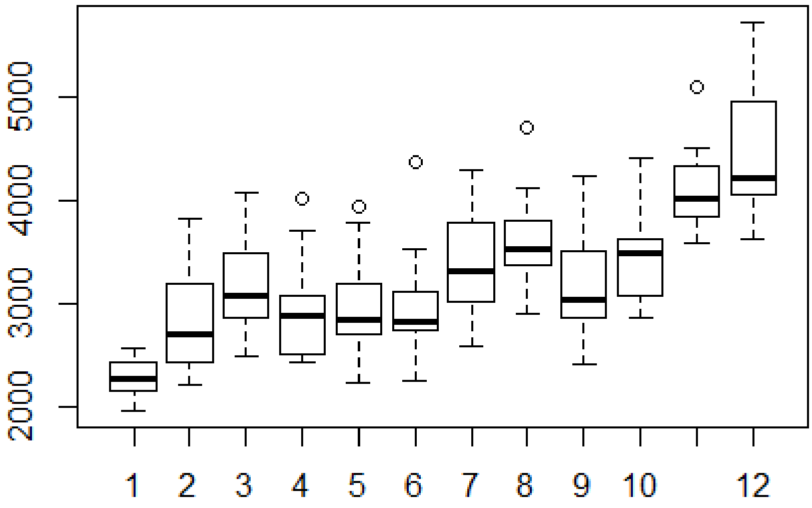
```
cy=12)
plot(Demand_ItemB, main="Time series of Item B")
```



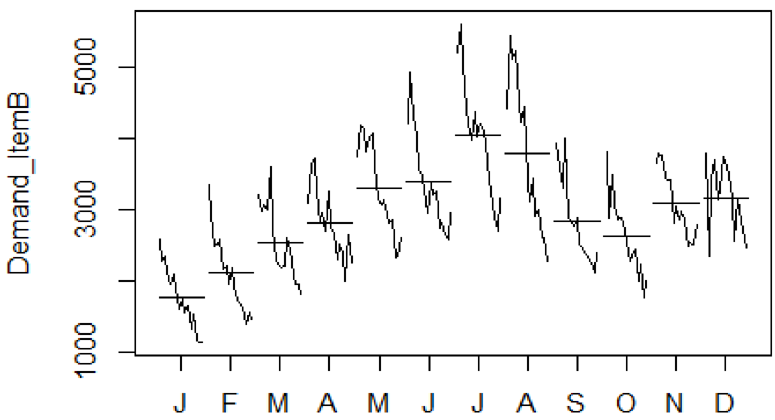

```
#Decrease in Demand. Clearly a decreasing trend and seams to have seasonal
ity. No cyclicality exists.
#No outliers and No abrupt change.

monthplot(Demand_ItemA)
```



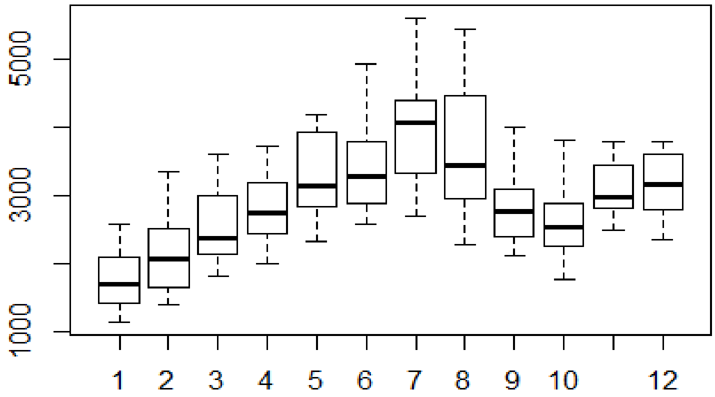
```
boxplot(Demand_ItemA~cycle(Demand_ItemA))
```



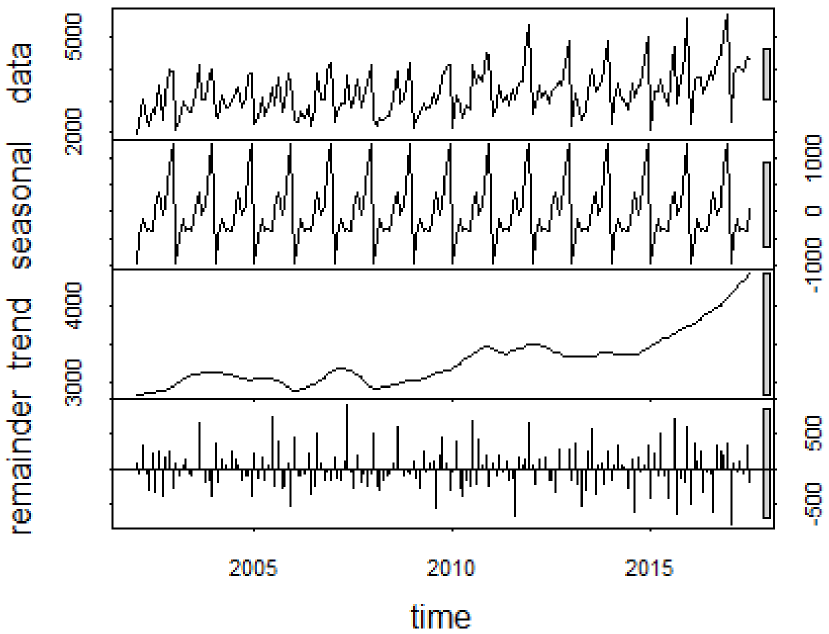

```
#Demand in the month of January is low and it is almost constant from feb to octomber except there is slightly
#higher demand in august and july. Demand in the months of Nov and Dec is comparatively high across all years.
#variability in the month of jan is low and high in Dec compared to other months.

monthplot(Demand_ItemB)
```



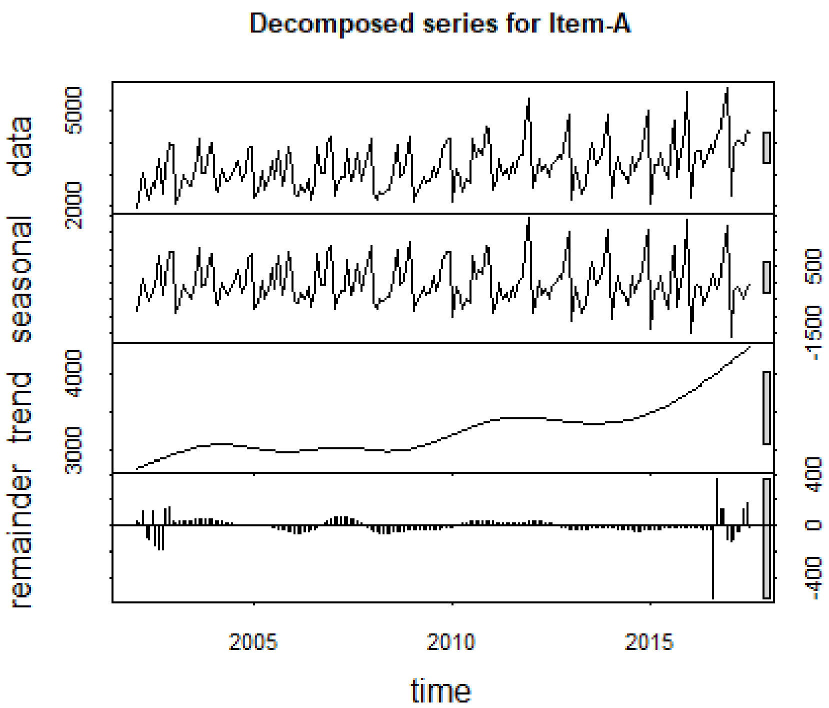

```
boxplot(Demand_ItemB~cycle(Demand_ItemB))
```



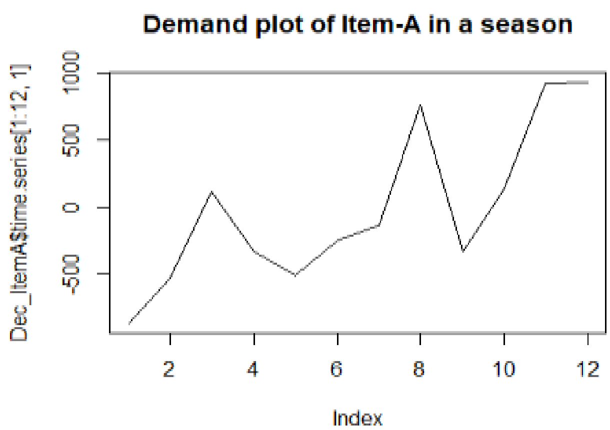

```r
#overall Demand is more in the months of july and august. After 2009 the d
emand in july month is more.
#Demand in the month of January is low and it is increasing till july and
then drops till october.
#Nov and Dec looks to maintain constant Demand.
#More variability in the months of august.


#Choosing and fitting Models

#Item A - Additive Model
Dec_ItemA <- stl(Demand_ItemA[,1], s.window="p") #constant seasonality
plot(Dec_ItemA)
```



```r
#Dec_ItemA

#seasonality changes
Dec_ItemA <- stl(Demand_ItemA[,1], s.window=3)
plot(Dec_ItemA, main="Decomposed series for Item-A")
```

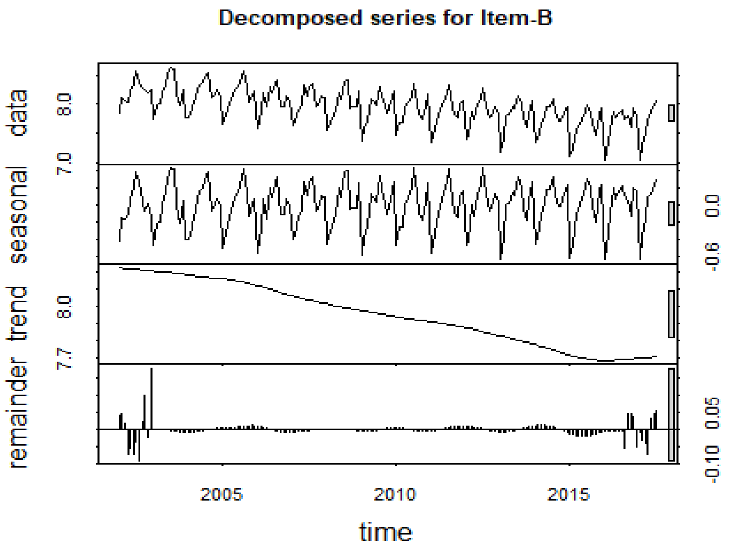



```r
#Dec_ItemA
#seasonality comes out to be significant
#Dec Month higher sales and Jan Month Sales are low.

plot(Dec_ItemA$time.series[1:12,1], type="l", main="Demand plot of Item-A
in a season")
```

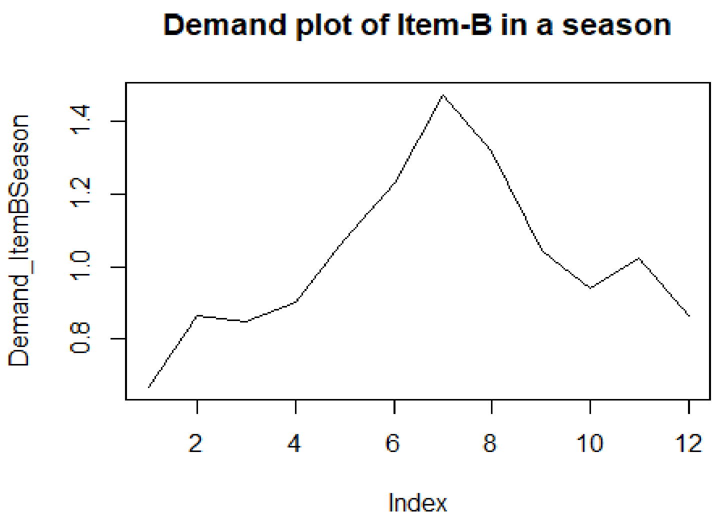



```r
#Item B - Multiplicative Model

logDem_Item_B <- log(Demand_ItemB)
```

```
logDec_ItemB <- stl(logDem_Item_B[,1], s.window=3)
plot(logDec_ItemB, main="Decomposed series for Item-B")
```

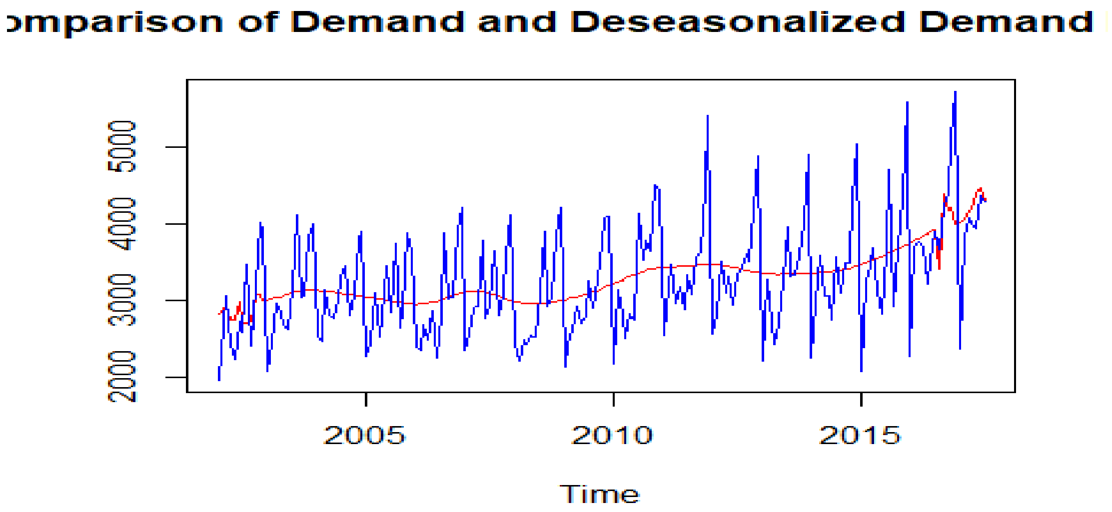



```
#seasonality comes out to be significant
#July Month higher sales and Jan Month sales are low.

logDec_ItemB$time.series[1:12,1]

##  [1] -0.40431263 -0.14416903 -0.16462410 -0.10181432  0.06889026
##  [6]  0.20442112  0.38637146  0.27603861  0.04211517 -0.06391801
## [11]  0.02175431 -0.14464991

Demand_ItemBSeason <- exp(logDec_ItemB$time.series[1:12,1])
plot(Demand_ItemBSeason, type="l", main="Demand plot of Item-B in a season
")
```

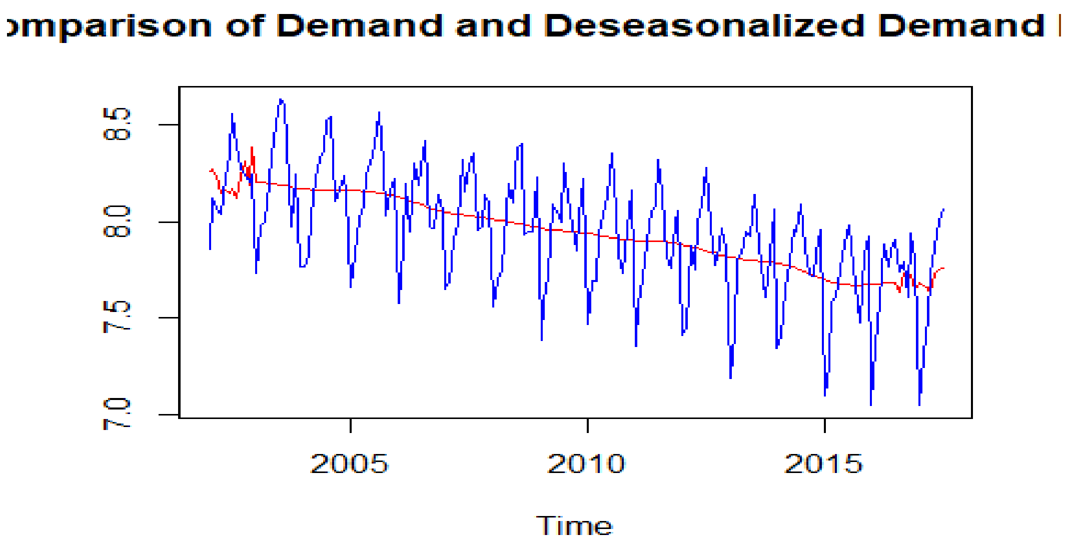



```
#For both the items seasonility is significant and for Item A the demand i
s higher towards the end,
#and for Item B the demand is higher around the middle of the ear.
```

```
#The demand is low in the Month of Jan for both the items.
#The demand for ItemA seems to be increasing whereas for ItemB it is decre
asing over the period.

#Deseasonalized plots
DeseasonDemand_ItemA <- (Dec_ItemA$time.series[,2]+Dec_ItemA$time.series[,
3])
ts.plot(DeseasonDemand_ItemA, Demand_ItemA, col=c("red", "blue"), main="Co
mparison of Demand and Deseasonalized Demand Item-A")
```

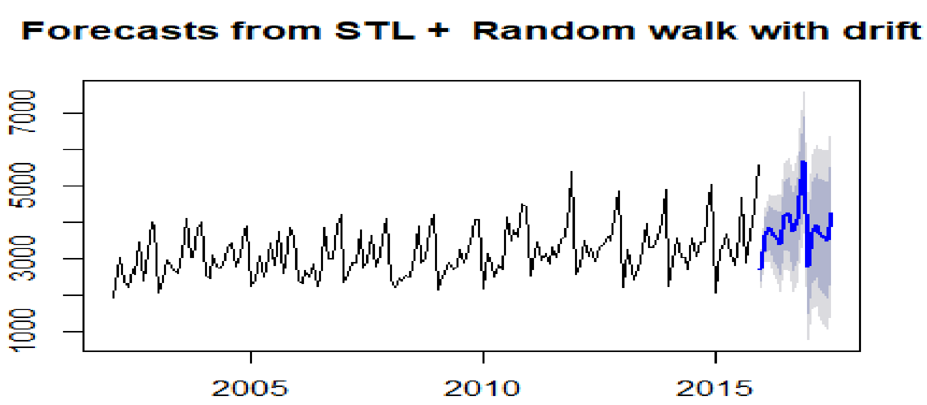



```
DeseasonDemand_ItemB <- (logDec_ItemB$time.series[,2]+logDec_ItemB$time.se
ries[,3])
ts.plot(DeseasonDemand_ItemB, logDem_Item_B, col=c("red", "blue"), main="C
omparison of Demand and Deseasonalized Demand Item-B")
```

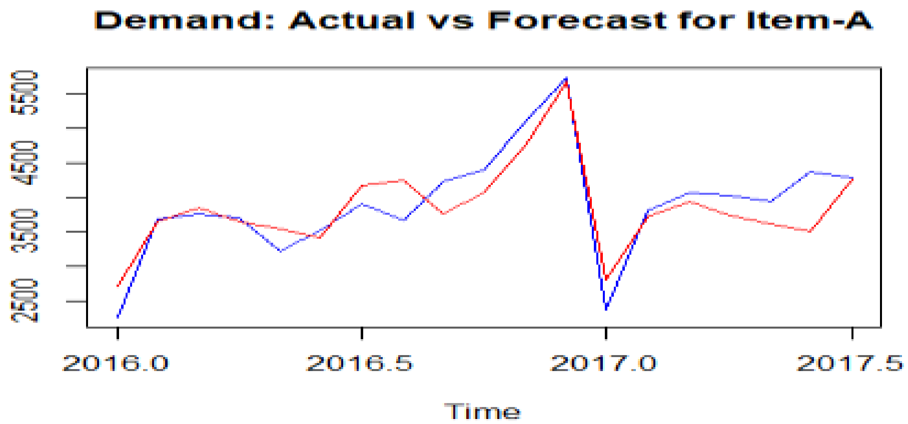



```
#stationarity test
adf.test(Dec_ItemA$time.series[,3])
```

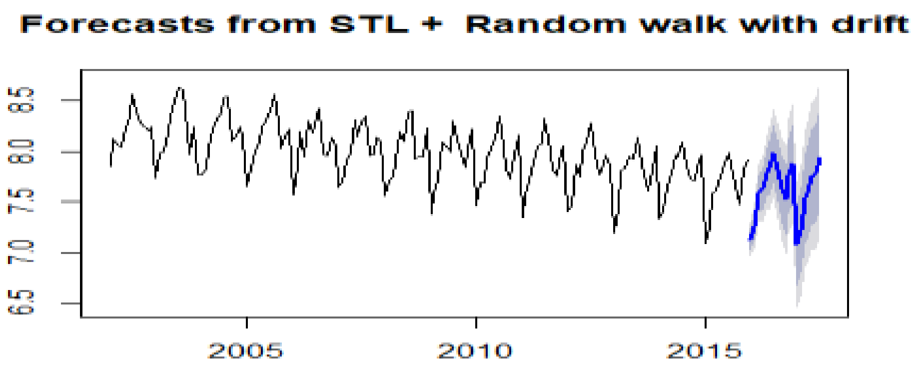
```
## Warning in adf.test(Dec_ItemA$time.series[, 3]): p-value smaller than
## printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  Dec_ItemA$time.series[, 3]
## Dickey-Fuller = -4.4097, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary

#reject null hypothesis. so the residuals form a stationary series

adf.test(exp(logDec_ItemB$time.series[,3]))

## Warning in adf.test(exp(logDec_ItemB$time.series[, 3])): p-value smalle
r
## than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  exp(logDec_ItemB$time.series[, 3])
## Dickey-Fuller = -8.2618, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary

#reject null hypothesis. so the residuals form a stationary series

#Forecasting
#Forecast using decomposition
DataATrain <- window(Demand_ItemA, start=c(2002,1), end=c(2015,12), freque
ncy=12)
DataATest <- window(Demand_ItemA, start=c(2016,1), frequency=12)

DataBTrain <- window(Demand_ItemB, start=c(2002,1), end=c(2015,12), freque
ncy=12)
DataBTest <- window(Demand_ItemB, start=c(2016,1), frequency=12)
#Item-A
Dec_ItemATrain <-stl(DataATrain[,1], s.window=5)
fcst.Dem_ItemA.stl <- forecast(Dec_ItemATrain, method="rwdrift", h=19)
plot(fcst.Dem_ItemA.stl)
```



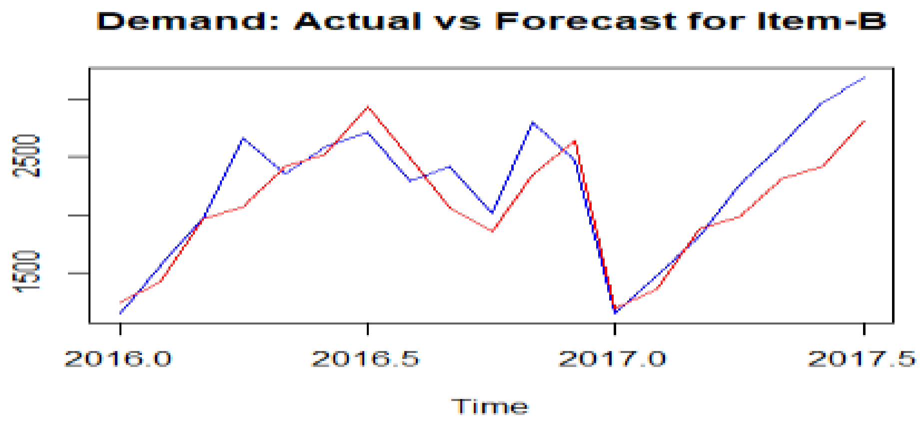
```
#fcst.Dem_ItemA.stl$residuals
#Dec_ItemATrain
#80% and 95% confidence interval

Vec<- cbind(DataATest,fcst.Dem_ItemA.stl$mean)
ts.plot(Vec, col=c("blue", "red"), main="Demand: Actual vs Forecast for It
em-A")
```



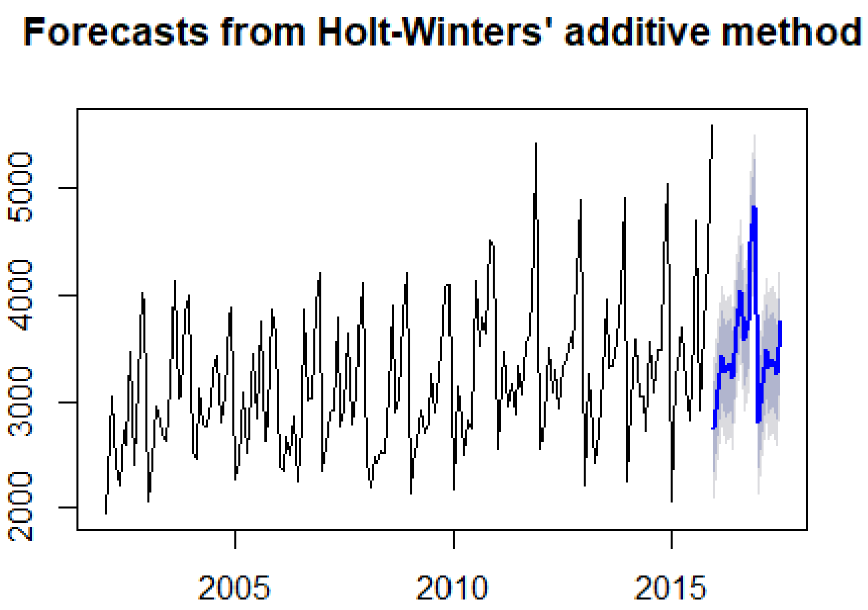

```
MAPE_Dec_ItemA <- mean(abs(Vec[,1]-Vec[,2])/Vec[,1])
MAPE_Dec_ItemA

## [1] 0.07736876

#0.07736876

#Item-B
Dec_ItemBTrain <-stl(log(DataBTrain)[,1], s.window=5)
fcst.Dem_ItemB.stl <- forecast(Dec_ItemBTrain, method="rwdrift", h=19)
plot(fcst.Dem_ItemB.stl)
```




```
Vec<- cbind(DataBTest,exp(fcst.Dem_ItemB.stl$mean))
ts.plot(Vec, col=c("blue", "red"), main="Demand: Actual vs Forecast for It
em-B")
```



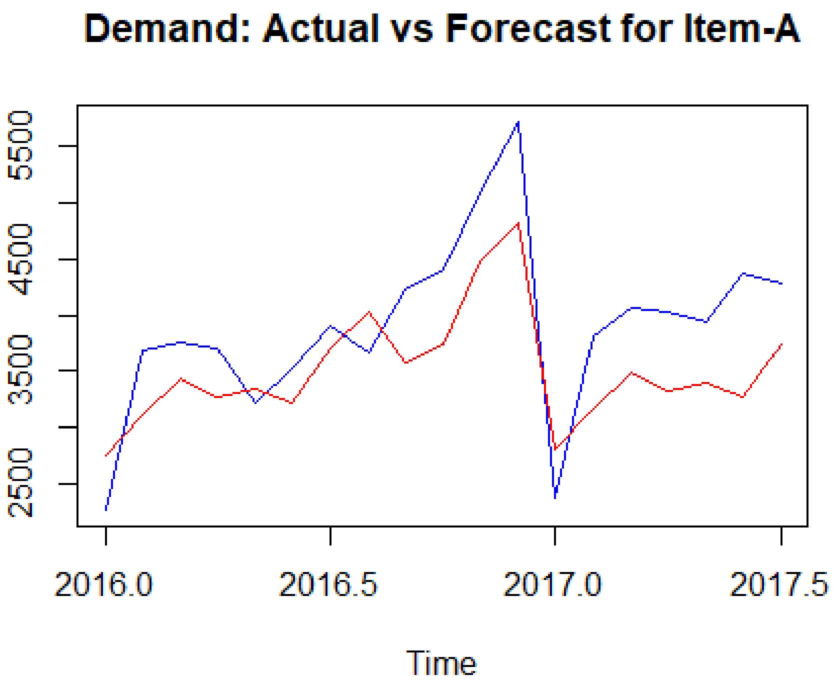

```
MAPE_Dec_ItemB <- mean(abs(Vec[,1]-Vec[,2])/Vec[,1])
MAPE_Dec_ItemB

## [1] 0.09226125

#0.09226125

#MAPE is less than 10 percent which indicates a working model.

#Exponential smoothening Model - Holt-winter's Method

fcst.Dem_ItemA.hw <- hw(DataATrain, h=19)
plot(fcst.Dem_ItemA.hw)
```




```
fcst.Dem_ItemA.hw$model
```

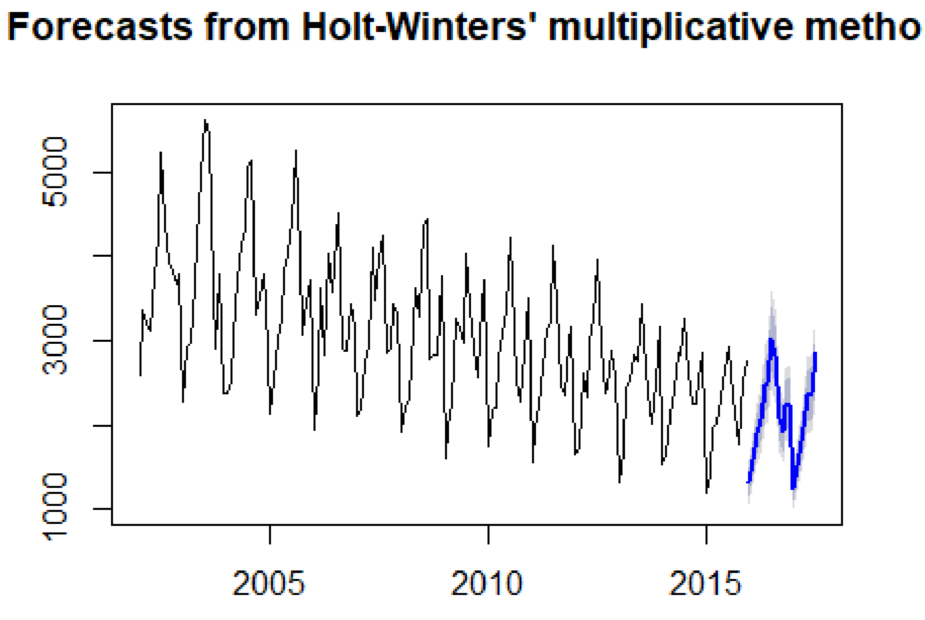
```
## Holt-Winters' additive method
##
## Call:
##  hw(y = DataATrain, h = 19)
##
##   Smoothing parameters:
##     alpha = 0.0728
##     beta  = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 3007.3727
##     b = 4.6452
##     s = 1182.799 836.5771 102.7667 -57.2941 404.6823 71.8966
##            -407.3715 -269.9908 -342.3852 -179.8529 -496.6008 -845.226
##
##   sigma:  336.7443
##
##      AIC     AICc      BIC
## 2833.305 2837.385 2886.412
```

```
Vec<- cbind(DataATest,fcst.Dem_ItemA.hw$mean)
ts.plot(Vec, col=c("blue", "red"), main="Demand: Actual vs Forecast for It
em-A")
```

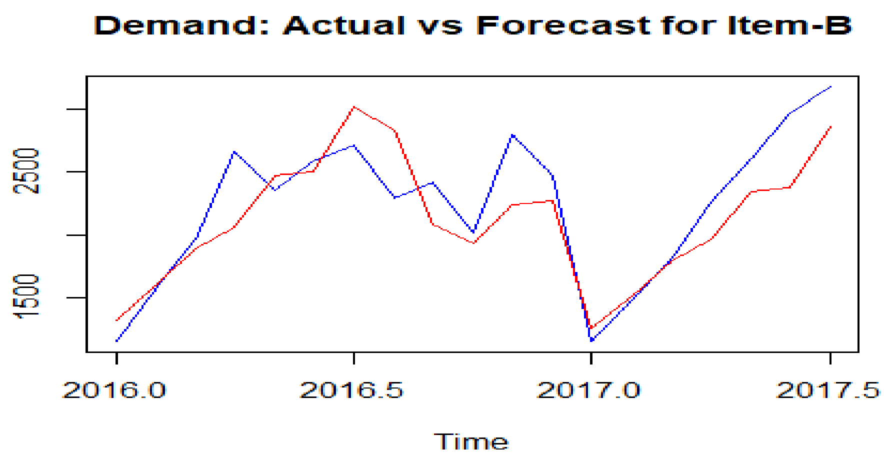



```
MAPE <- mean(abs(Vec[,1]-Vec[,2])/Vec[,1])
MAPE

## [1] 0.1383859

#0.1383859
#round(accuracy(fcst.Dem_ItemA.hw),2)
```

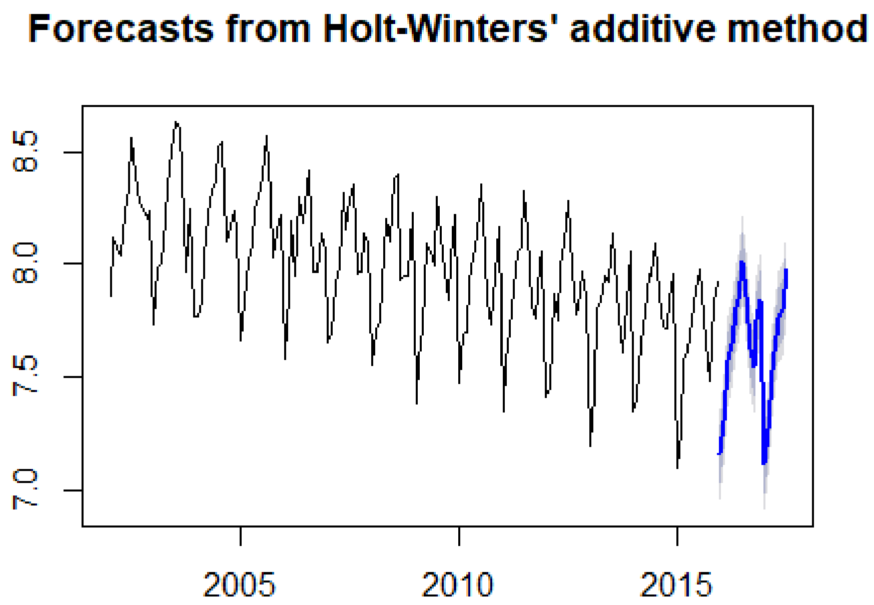
```
fcst.Dem_ItemB.hw <- hw(DataBTrain, seasonal = "multiplicative",  h=19)
plot(fcst.Dem_ItemB.hw)
```




```
fcst.Dem_ItemB.hw$model

## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = DataBTrain, h = 19, seasonal = "multiplicative")
##
##   Smoothing parameters:
##     alpha = 0.0212
##     beta  = 0.0016
##     gamma = 1e-04
##
##   Initial states:
##     l = 4061.8957
##     b = -12.621
##     s = 1.0607 1.0409 0.8957 0.9634 1.3019 1.3818
##            1.1411 1.1234 0.9337 0.8529 0.7151 0.5895
##
##   sigma:  0.0992
##
##      AIC     AICc      BIC
## 2782.834 2786.914 2835.942

Vec<- cbind(DataBTest,fcst.Dem_ItemB.hw$mean)
ts.plot(Vec, col=c("blue", "red"), main="Demand: Actual vs Forecast for It
em-B")
```

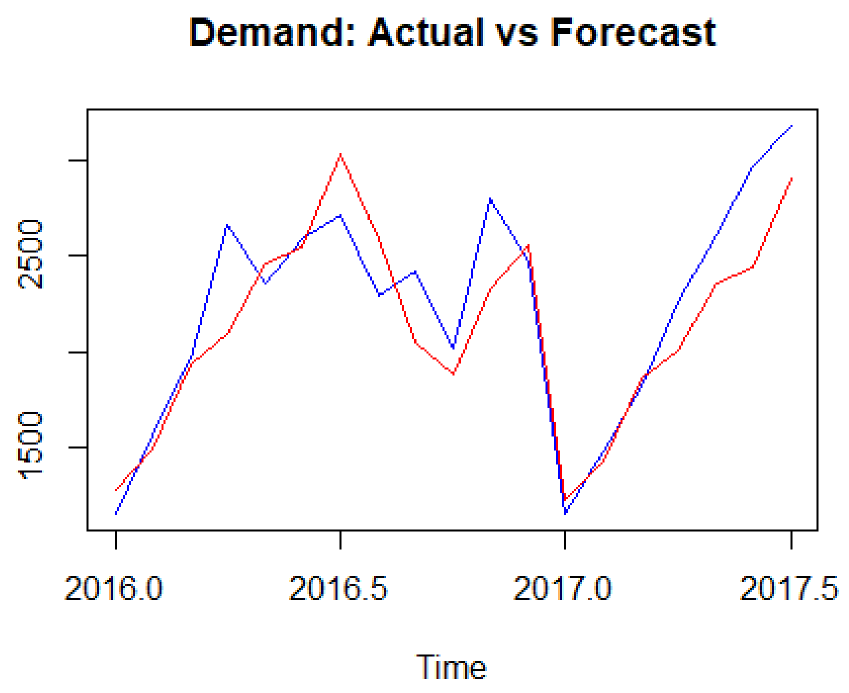



```
MAPE <- mean(abs(Vec[,1]-Vec[,2])/Vec[,1])
MAPE

## [1] 0.1035767

#0.1035767

#By transforming and doing an additive model
fcst.Dem_ItemB.hw <- hw(log(DataBTrain), h=19)
plot(fcst.Dem_ItemB.hw)
```



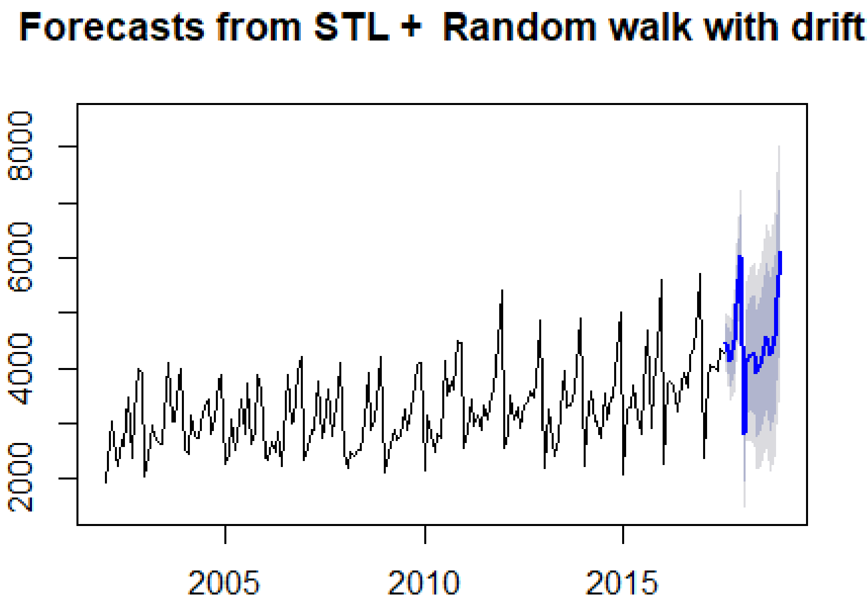

```
fcst.Dem_ItemB.hw$model

## Holt-Winters' additive method
##
## Call:
```

```
## hw(y = log(DataBTrain), h = 19)
##
##   Smoothing parameters:
##     alpha = 1e-04
##     beta  = 1e-04
##     gamma = 0.2396
##
##   Initial states:
##     l = 8.2809
##     b = -0.0035
##     s = 0.0789 0.0222 -0.0417 -0.0314 0.2754 0.4015
##            0.1728 0.1317 -0.0066 -0.1359 -0.3187 -0.5481
##
##   sigma:  0.1021
##
##      AIC     AICc      BIC
## 111.4846 115.5646 164.5920

Vec<- cbind(DataBTest,exp(fcst.Dem_ItemB.hw$mean))
ts.plot(Vec, col=c("blue", "red"), main="Demand: Actual vs Forecast")
```

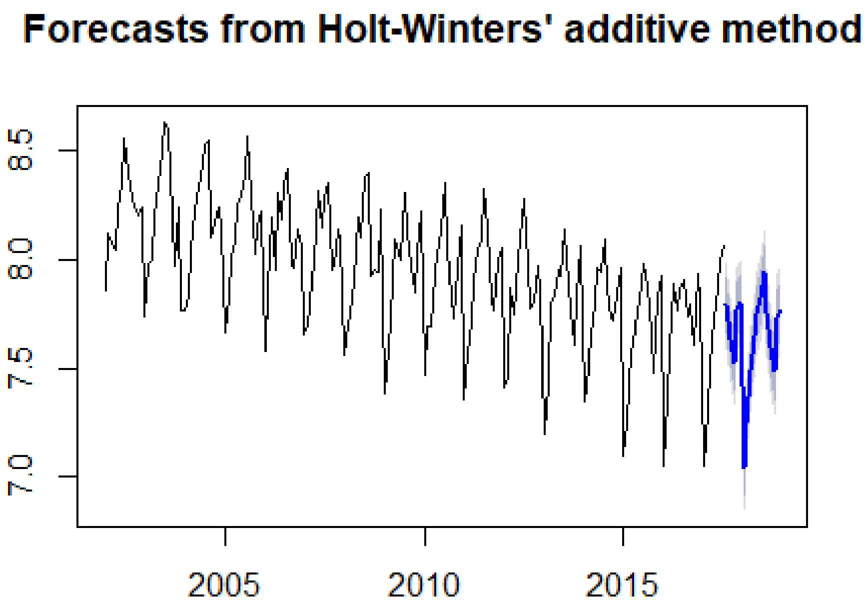



```
MAPE <- mean(abs(Vec[,1]-Vec[,2])/Vec[,1])
MAPE

## [1] 0.09030823

#0.09030823

#Final Forecasting
#Item-A
Dec_ItemAFinal <-stl(Demand_ItemA[,1], s.window=5)
fcst.Dem_ItemA.stl.final <- forecast(Dec_ItemAFinal, method="rwdrift", h=1
```

```
7)
plot(fcst.Dem_ItemA.stl.final)
```




```
fcst.Dem_ItemA.stl.final

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Aug 2017       4482.661 4135.757 4829.564 3952.118 5013.204
## Sep 2017       4155.016 3661.804 4648.228 3400.713 4909.319
## Oct 2017       4485.163 3877.916 5092.409 3556.459 5413.866
## Nov 2017       5165.715 4460.866 5870.564 4087.741 6243.689
## Dec 2017       6024.258 5232.140 6816.377 4812.818 7235.699
## Jan 2018       2806.480 1934.319 3678.640 1472.625 4140.334
## Feb 2018       4114.391 3167.580 5061.202 2666.369 5562.413
## Mar 2018       4259.724 3242.467 5276.981 2703.964 5815.485
## Apr 2018       4263.406 3179.087 5347.725 2605.084 5921.728
## May 2018       3914.072 2765.483 5062.661 2157.457 5670.687
## Jun 2018       4053.246 2842.734 5263.758 2201.927 5904.564
## Jul 2018       4390.061 3119.629 5660.493 2447.103 6333.019
## Aug 2018       4582.722 3254.102 5911.341 2550.774 6614.670
## Sep 2018       4255.077 2869.785 5640.369 2136.455 6373.699
## Oct 2018       4585.224 3144.593 6025.854 2381.970 6788.477
## Nov 2018       5265.776 3770.995 6760.557 2979.706 7551.846
## Dec 2018       6124.319 4576.450 7672.188 3757.058 8491.581

#80% and 95% confidence interval

#Item-B
fcst.Dem_ItemB.hw.final <- hw(log(Demand_ItemB), h=17)
plot(fcst.Dem_ItemB.hw.final)
```




```
exp(fcst.Dem_ItemB.hw.final$mean)

##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 2017
## 2018 1144.515 1405.022 1790.641 2089.994 2316.845 2487.255 2809.093
##           Aug      Sep      Oct      Nov      Dec
## 2017 2421.314 2068.049 1848.267 2359.432 2449.169
## 2018 2327.144 1987.619 1776.384 2267.669 2353.916

exp(fcst.Dem_ItemB.hw.final$lower)

##               80%       95%
## Aug 2017 2126.058 1984.6254
## Sep 2017 1815.871 1695.0727
## Oct 2017 1622.888 1514.9283
## Nov 2017 2071.722 1933.9039
## Dec 2017 2150.516 2007.4565
## Jan 2018 1004.952  938.0992
## Feb 2018 1233.693 1151.6233
## Mar 2018 1572.289 1467.6948
## Apr 2018 1835.139 1713.0592
## May 2018 2034.327 1898.9964
## Jun 2018 2183.957 2038.6726
## Jul 2018 2466.550 2302.4661
## Aug 2018 2034.785 1895.1954
## Sep 2018 1737.914 1618.6900
## Oct 2018 1553.217 1446.6629
## Nov 2018 1982.781 1846.7582
## Dec 2018 2058.192 1916.9959

exp(fcst.Dem_ItemB.hw.final$upper)
```

```
##                 80%      95%
## Aug 2017 2757.574 2954.090
## Sep 2017 2355.249 2523.094
## Oct 2017 2104.945 2254.952
## Nov 2017 2687.098 2878.592
## Dec 2017 2789.297 2988.074
## Jan 2018 1303.459 1396.349
## Feb 2018 1600.144 1714.177
## Mar 2018 2039.316 2184.646
## Apr 2018 2380.242 2549.869
## May 2018 2638.597 2826.634
## Jun 2018 2832.673 3034.542
## Jul 2018 3199.207 3427.196
## Aug 2018 2661.509 2857.542
## Sep 2018 2273.201 2440.633
## Oct 2018 2031.616 2181.254
## Nov 2018 2593.490 2784.513
## Dec 2018 2692.129 2890.418
```