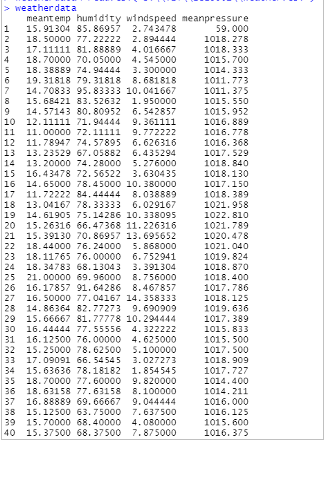
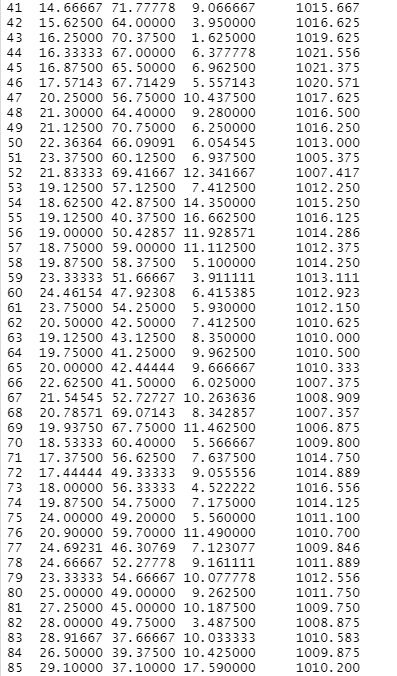
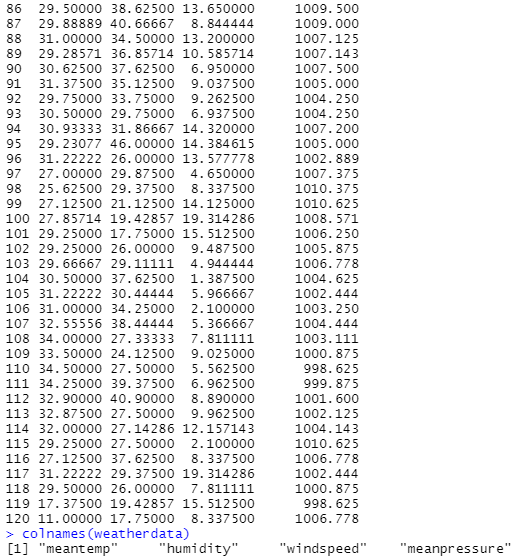
**R PROGRAMMING Output –**

For ‘meantemp’ column –

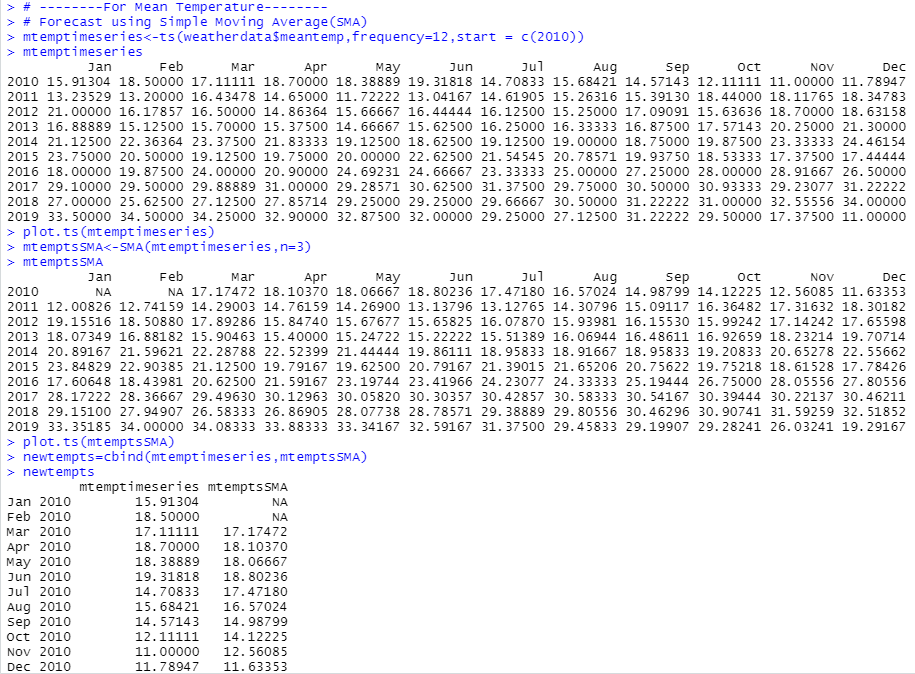
Importing ‘TTR’ and ‘forecast’ packages. Reading weather dataset and viewing the details. The different column names can also be viewed separately.

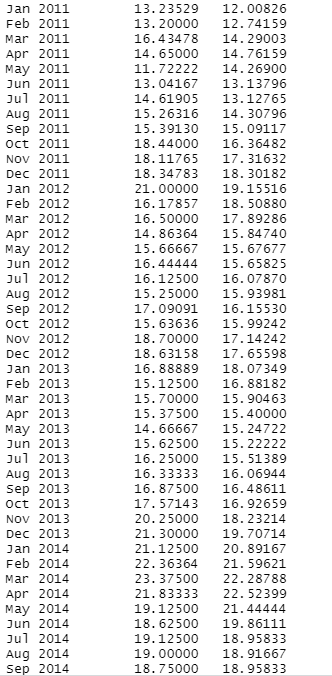


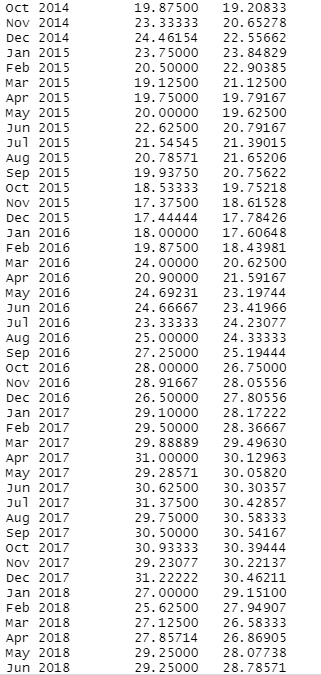




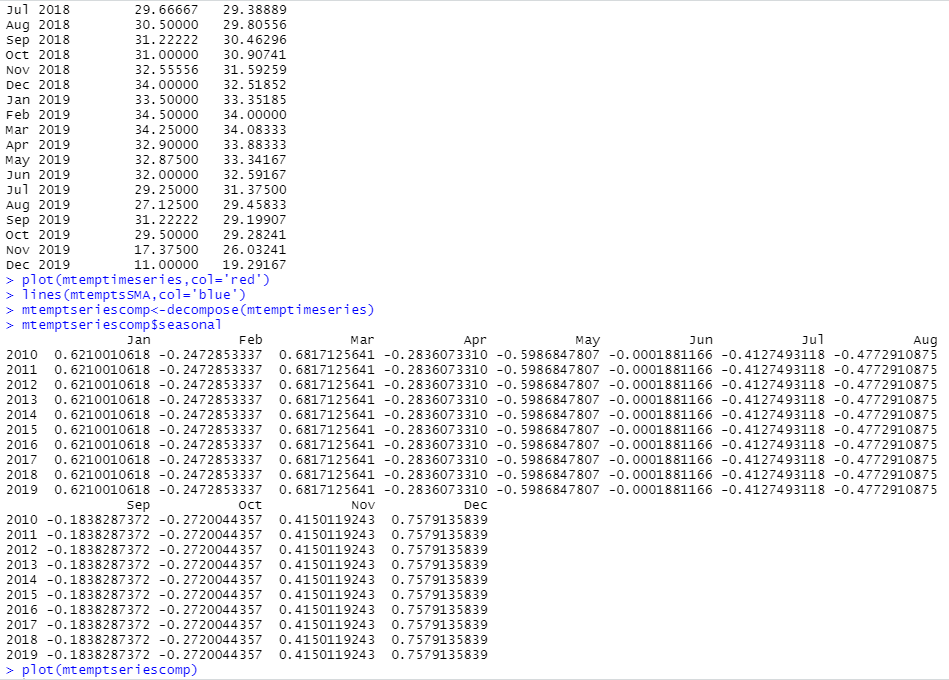
Converting the ‘meantemp’ column as a seasonal time series object with the start year of collecting data as 2010. Then plotting the ‘mtemptimeseries’. Forecasting of values is done using the SMA() function with parameters – the ‘mtemptimeseries’ and order ‘n’ as 3. The forecasted value is plotted to see the smoothing. The actual time series and forecasted time series are added to a new data frame to see the differences between the corresponding values.



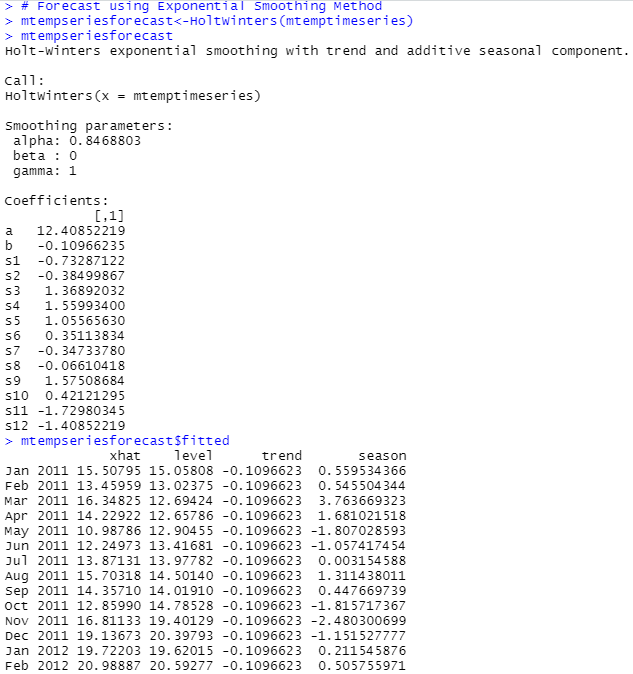


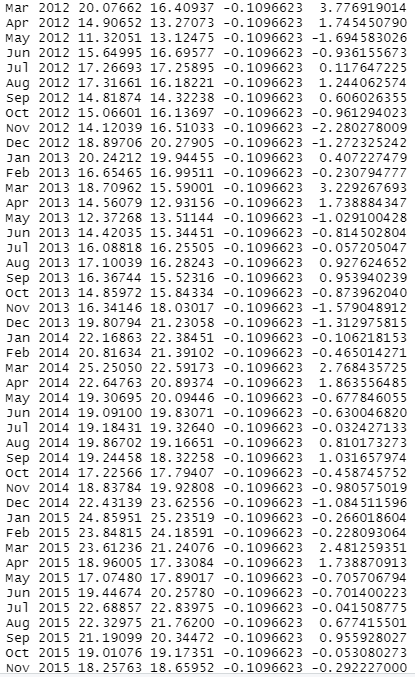


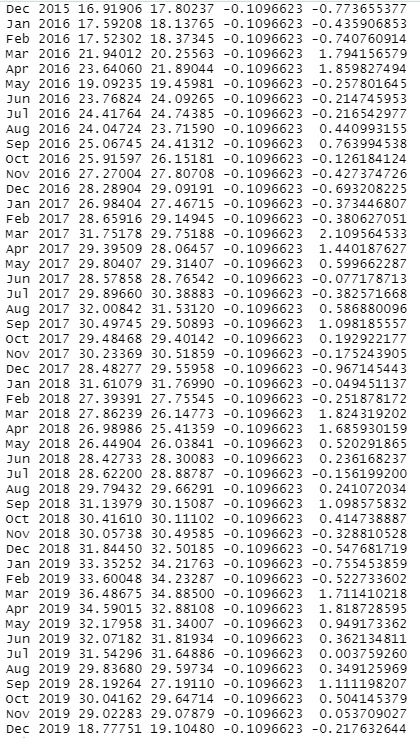
A graph is plotted between the actual time series and forecasted time series to see the differences as a pictorial representation. The actual time series being a seasonal time series can be decomposed into different components and is stored in ‘mtemptseriescomp’. These different components are estimated values. The ‘seasonal’ component is viewed here and the graph of all the components is plotted.



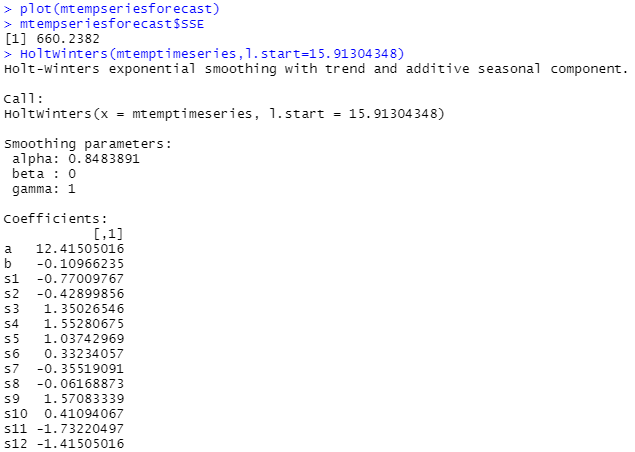
For forecasting using Exponential Smoothing method, ‘HoltWinters()’ function is used. As it is a seasonal time series there is no need of providing values for the ‘beta’ and ‘gamma’ parameters. Only the ‘meantemp’ time series is required. It forecasts the ‘trend’ and ‘seasonal’ component. The forecast can be viewed by using the named element ‘fitted’.



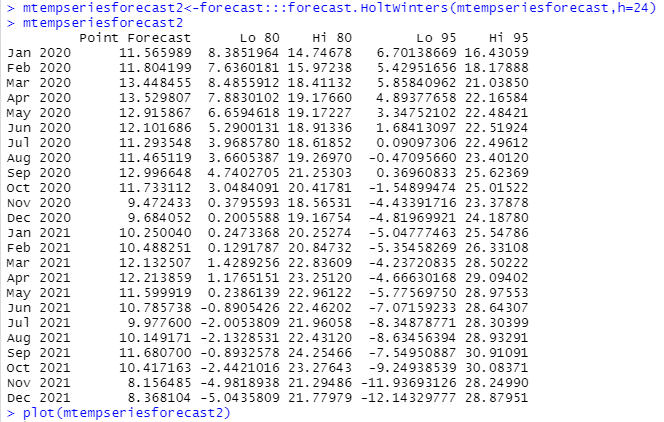




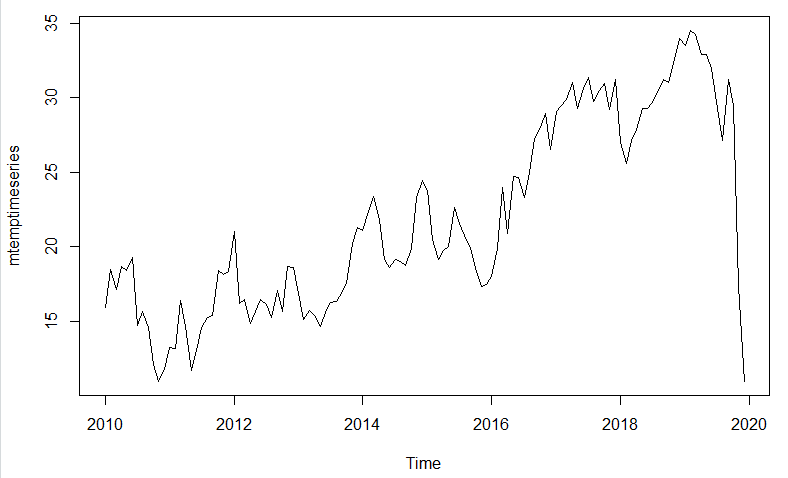
The forecasted values as compared to actual values is plotted. The sum of squared errors for the in-sample forecast errors is stored in a named element of the list variable called “SSE”. In case we want to use the first value in the time series as the initial value for the level in exponential smoothing, we can specify the value using the ‘l.start’ parameter in the HoltWinters() function. Here, the initial value of ‘meantemp’ column is specified as ‘l.start’ value.



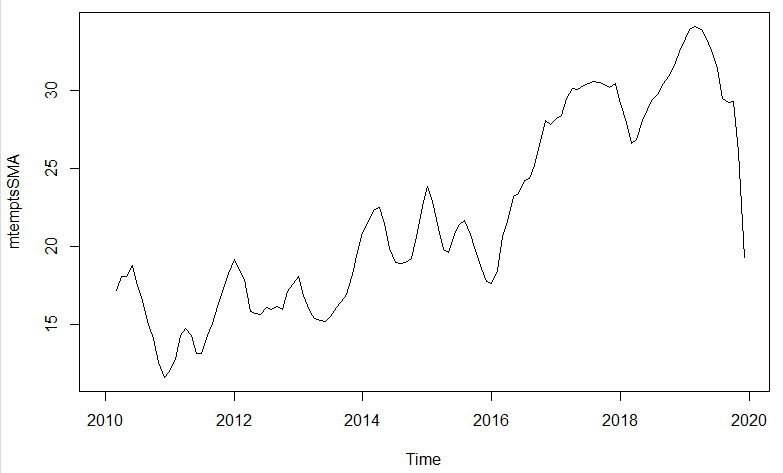
To forecast the ‘meantemp’ values for future months, the ‘forecast.HoltWinters()’ is used. It takes the parameters – previous predicted model and number of months as ‘h’ value, for the forecasting. The forecasted values can also be plotted to see the trend in the values.



Plot for ‘plot.ts(mtemptimeseries)’ –



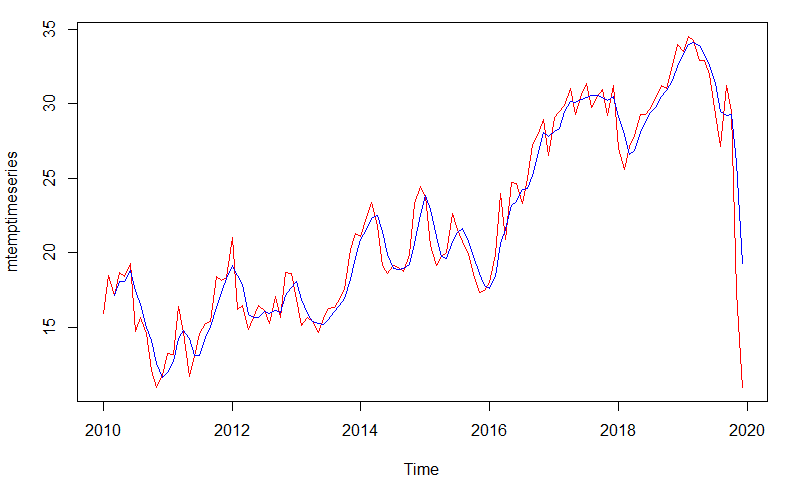
Plot for ‘plot.ts(mtemptsSMA)’ –



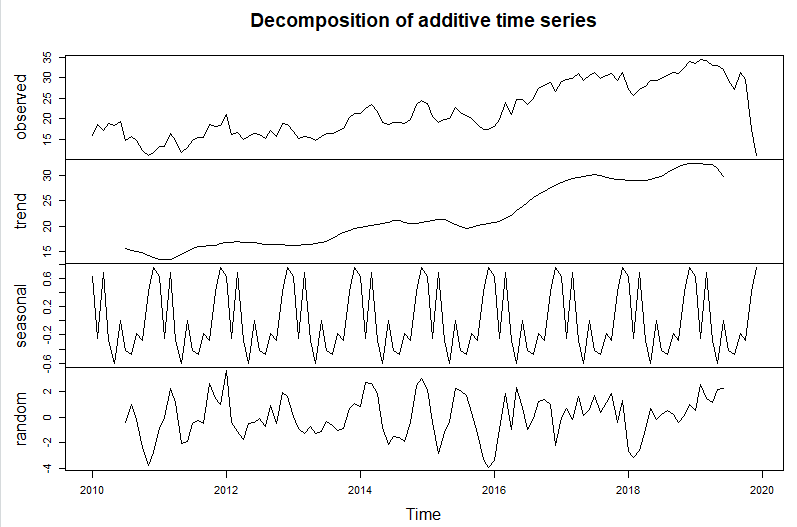
Plot for:

plot(mtemptimeseries,col='red') # Actual time series

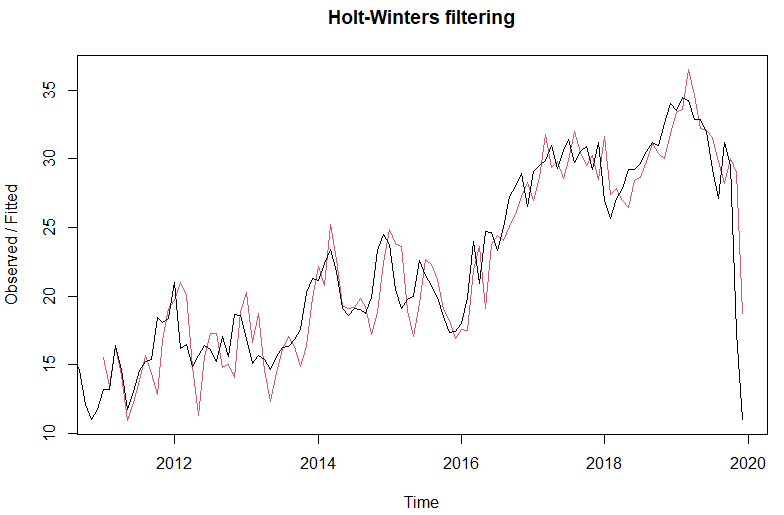
lines(mtemptsSMA,col='blue') # SMA forecasted time series



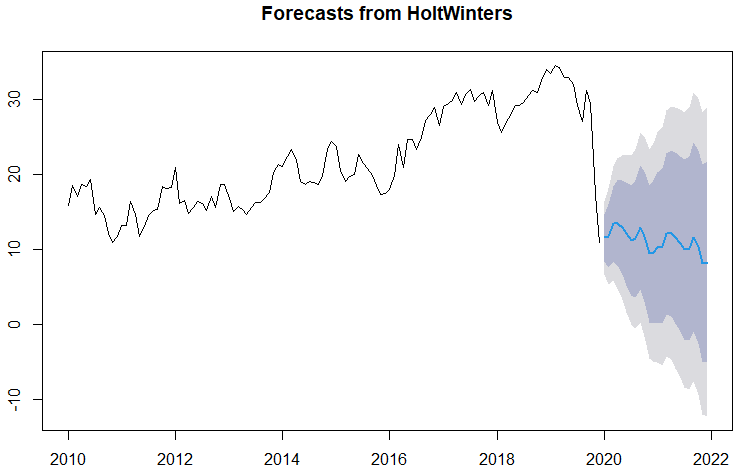
Plot for ‘plot(mtemptseriescomp)’ –



Plot for ‘plot(mtempseriesforecast)’ –

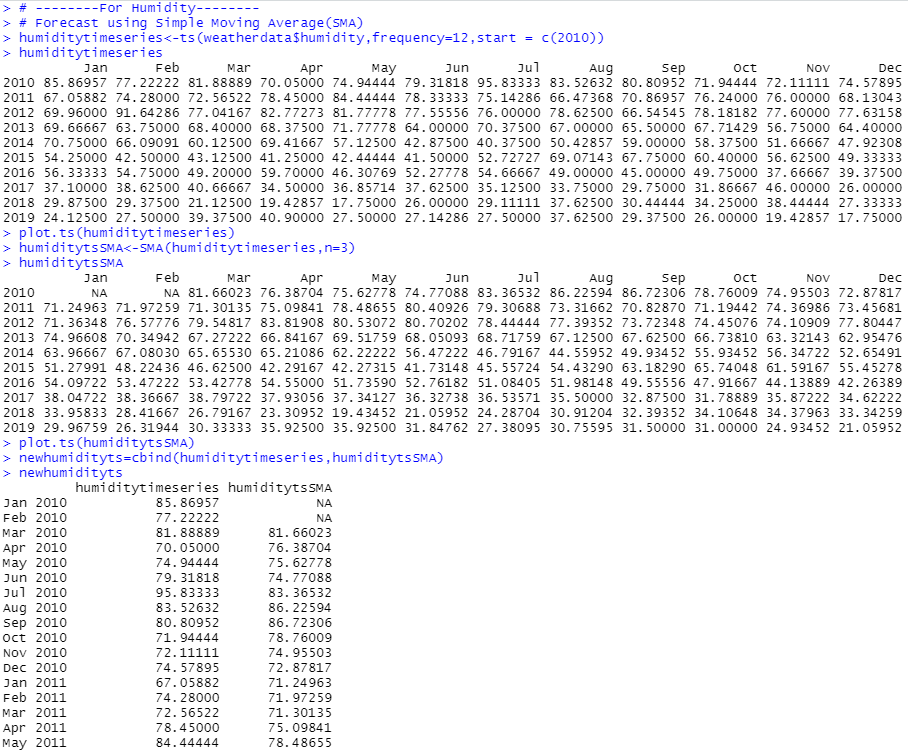


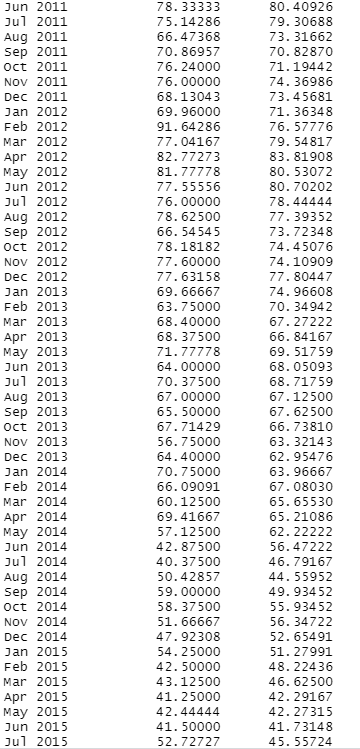
Plot for ‘plot(mtempseriesforecast2)’ –

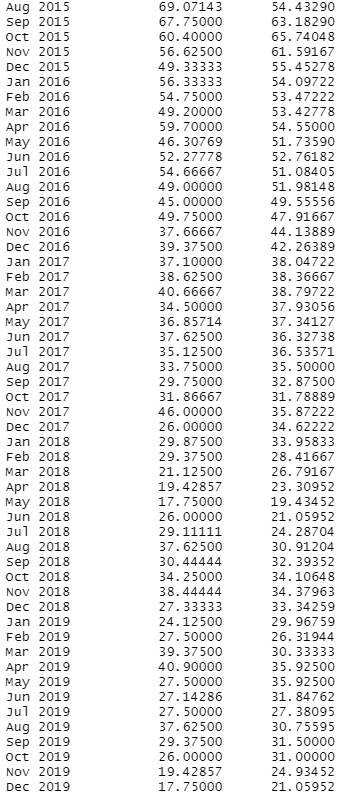


For ‘humidity’ column –

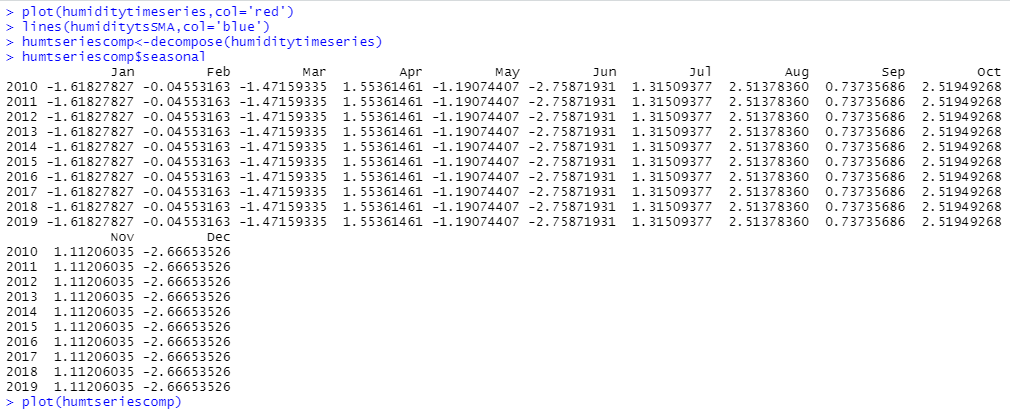
Converting the ‘humidity’ column as a seasonal time series object with the start year of collecting data as 2010. Then plotting the ‘humiditytimeseries’. Forecasting of values is done using the SMA() function with parameters – the ‘humiditytimeseries’ and order ‘n’ as 3. The forecasted value is plotted to see the smoothing. The actual time series and forecasted time series are added to a new data frame to see the differences between the corresponding values.



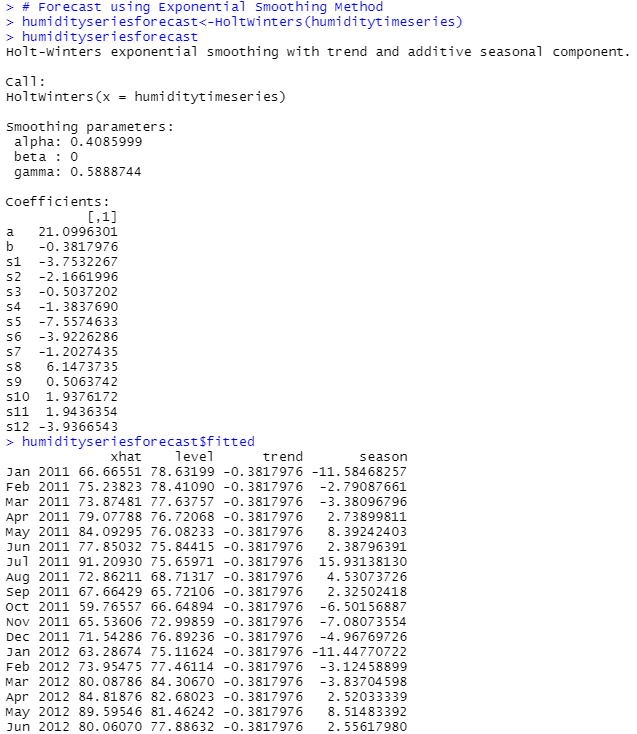


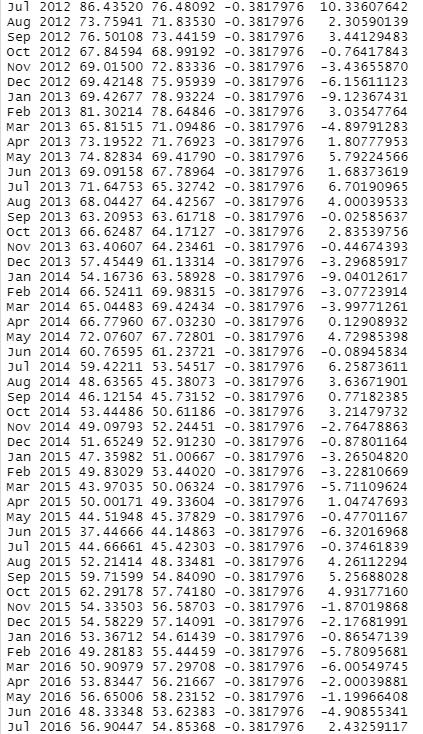


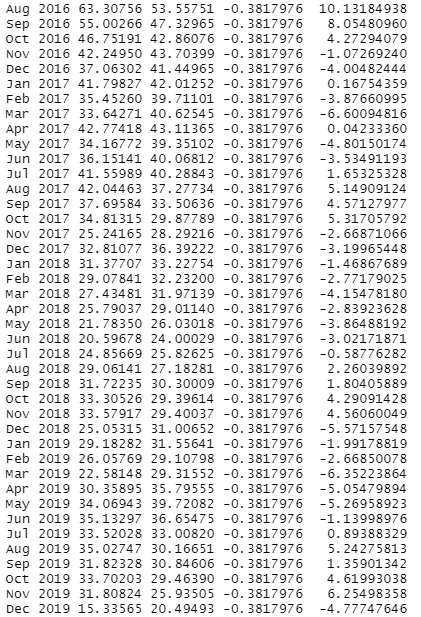
A graph is plotted between the actual time series and forecasted time series to see the differences as a pictorial representation. The actual time series being a seasonal time series can be decomposed into different components and is stored in ‘humtseriescomp’. These different components are estimated values. The ‘seasonal’ component is viewed here and the graph of all the components is plotted.



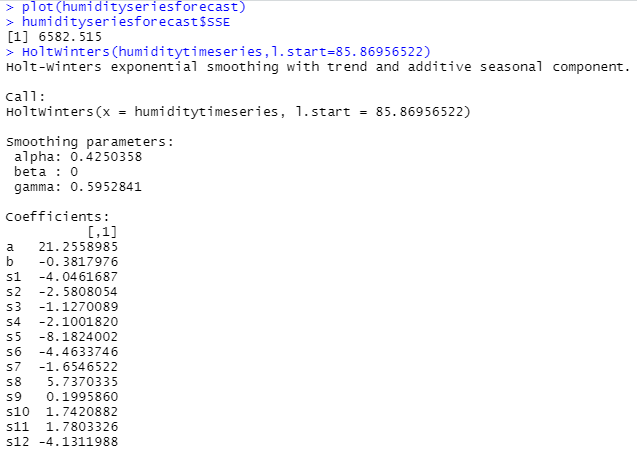
For forecasting using Exponential Smoothing method, ‘HoltWinters()’ function is used. As it is a seasonal time series there is no need of providing values for the ‘beta’ and ‘gamma’ parameters. Only the ‘humidity’ time series is required. It forecasts the ‘trend’ and ‘seasonal’ component. The forecast can be viewed by using the named element ‘fitted’.



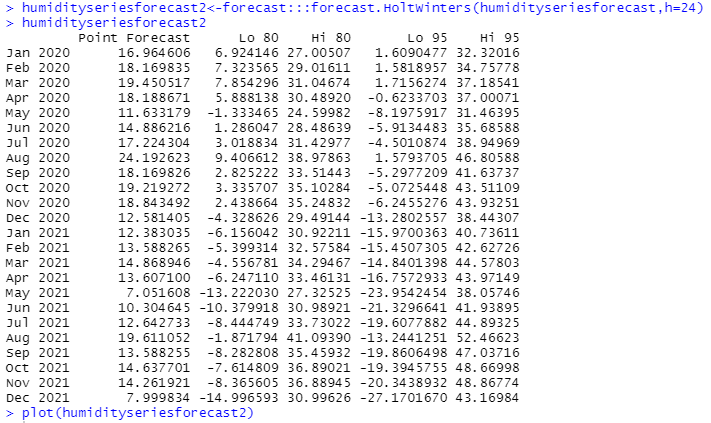




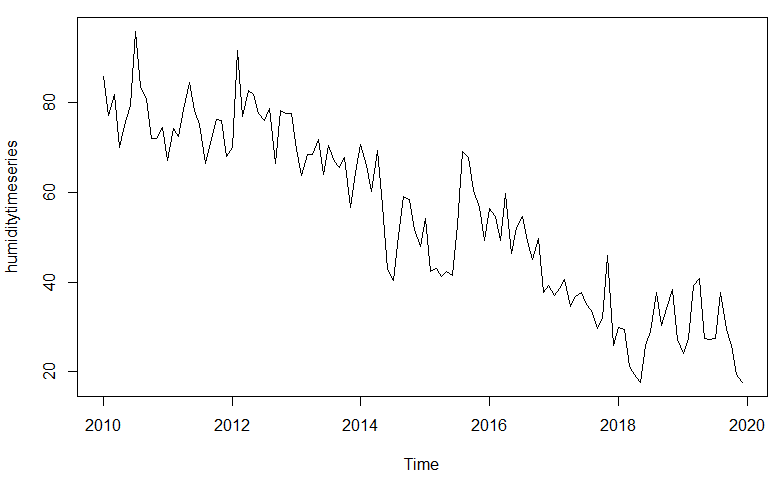
The forecasted values as compared to actual values is plotted. The sum of squared errors for the in-sample forecast errors is stored in a named element of the list variable called “SSE”. In case we want to use the first value in the time series as the initial value for the level in exponential smoothing, we can specify the value using the ‘l.start’ parameter in the HoltWinters() function. Here, the initial value of ‘humidity’ column is specified as ‘l.start’ value.



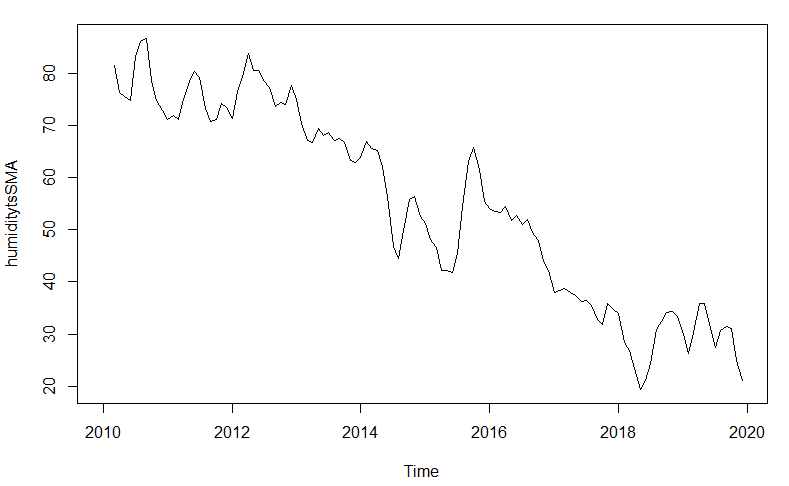
To forecast the ‘humidity’ values for future months, the ‘forecast.HoltWinters()’ is used. It takes the parameters – previous predicted model and number of months as ‘h’ value, for the forecasting. The forecasted values can also be plotted to see the trend in the values.



Plot for ‘plot.ts(humiditytimeseries)’ –



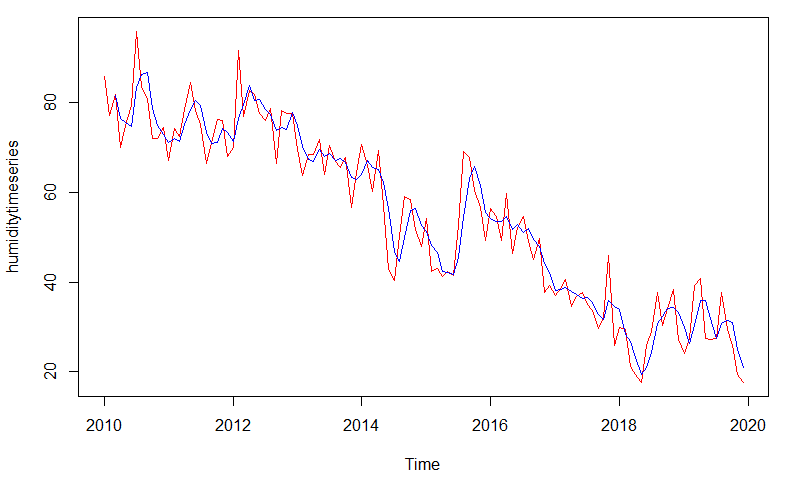
Plot for ‘plot.ts(humiditytsSMA)’ –



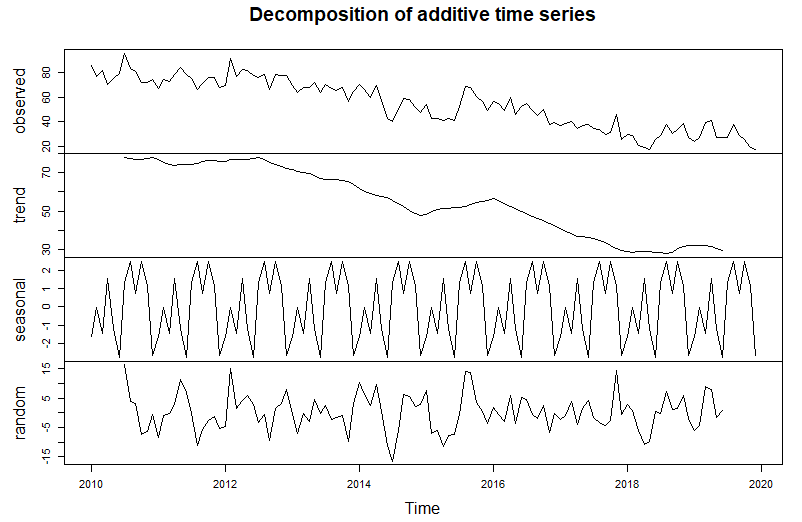
Plot for

plot(humiditytimeseries,col='red') # Actual time series

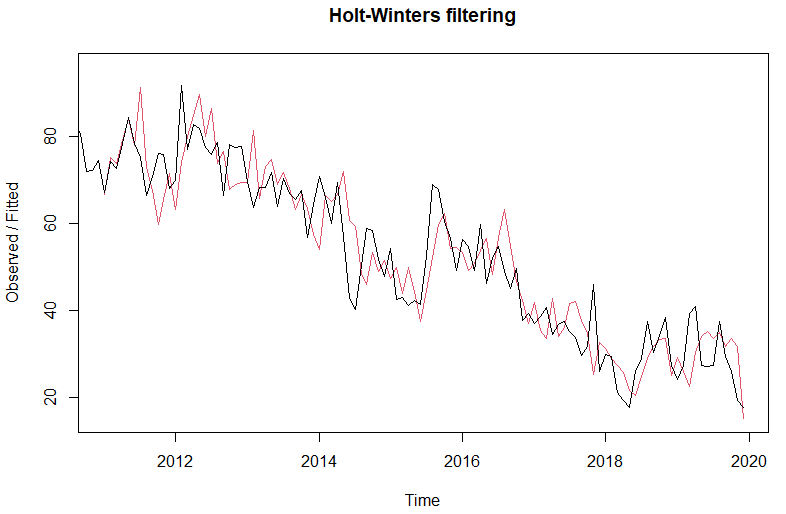
lines(humiditytsSMA,col='blue') # SMA forecasted time series



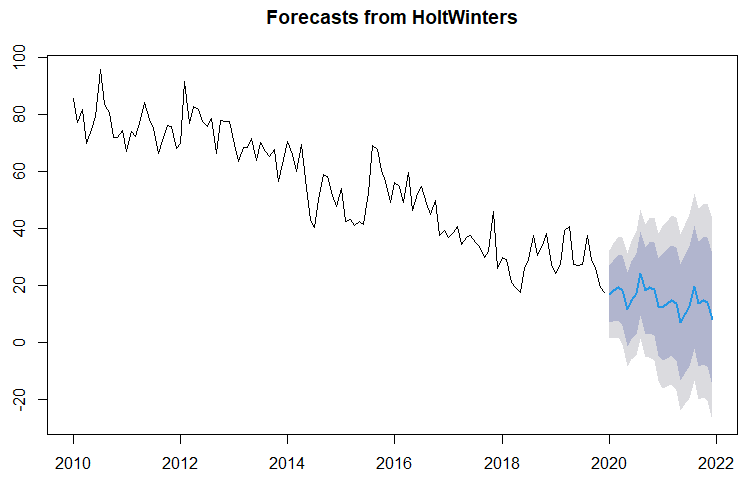
Plot for ‘plot(humtseriescomp)’ –



Plot for ‘plot(humidityseriesforecast)’ –

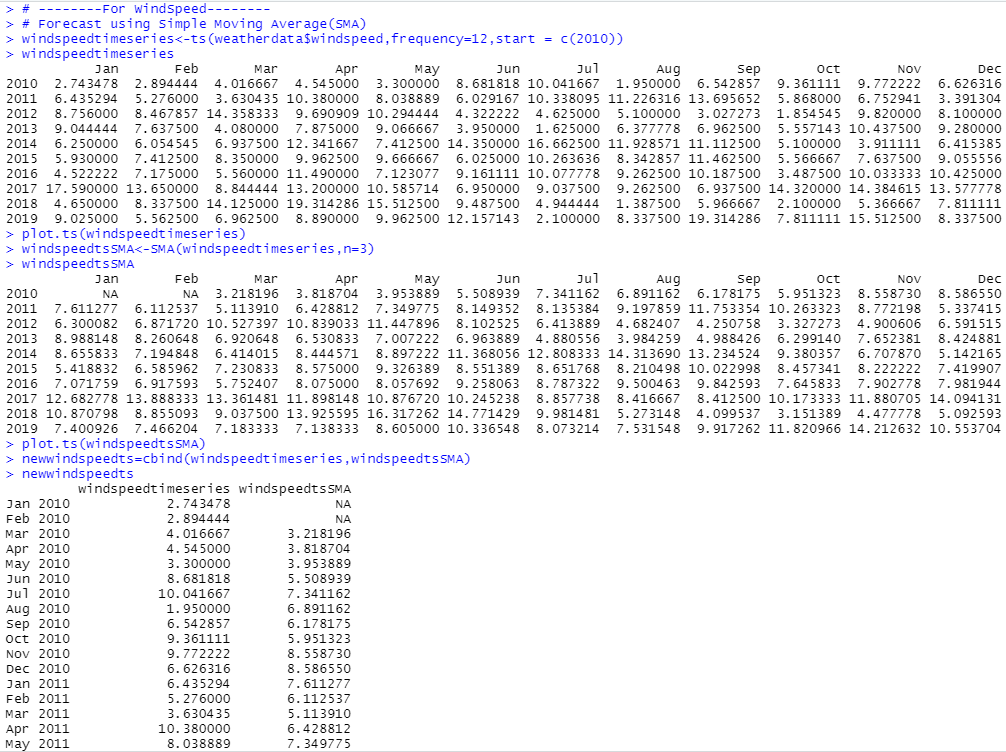


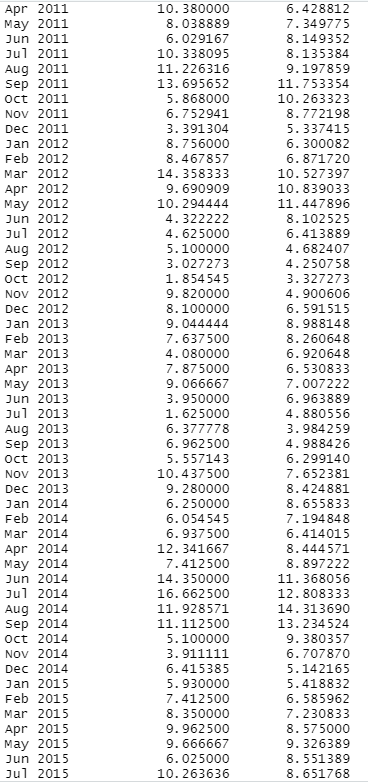
Plot for ‘plot(humidityseriesforecast2)’ –

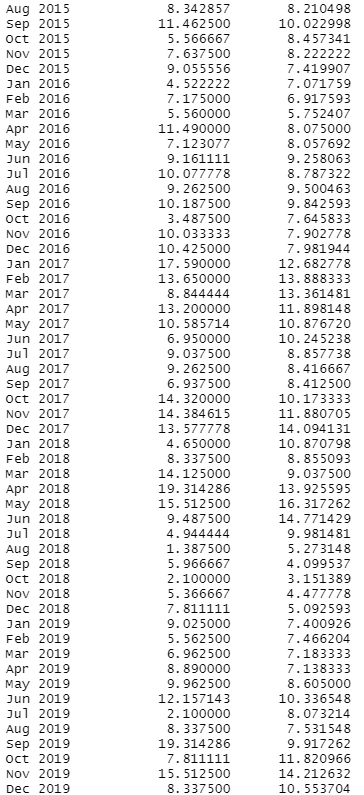


For ‘windspeed’ column –

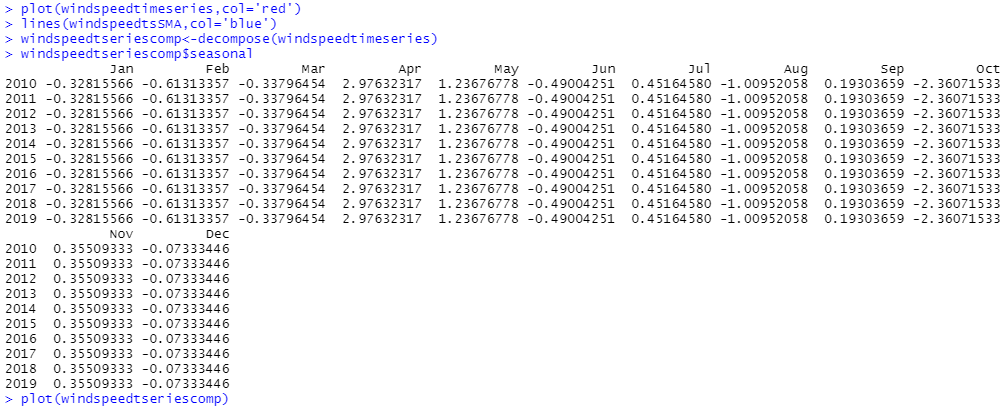
Converting the ‘windspeed’ column as a seasonal time series object with the start year of collecting data as 2010. Then plotting the ‘winspeedtimeseries’. Forecasting of values is done using the SMA() function with parameters – the ‘windspeedtimeseries’ and order ‘n’ as 3. The forecasted value is plotted to see the smoothing. The actual time series and forecasted time series are added to a new data frame to see the differences between the corresponding values.



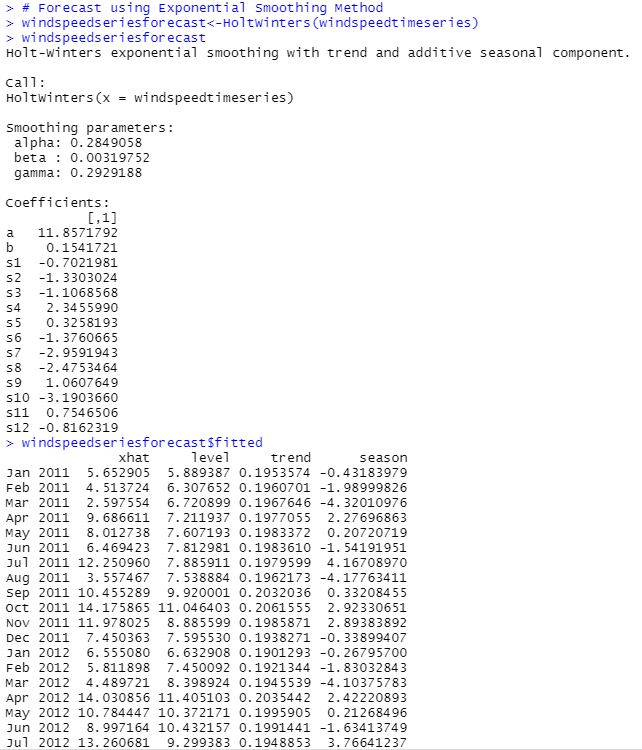




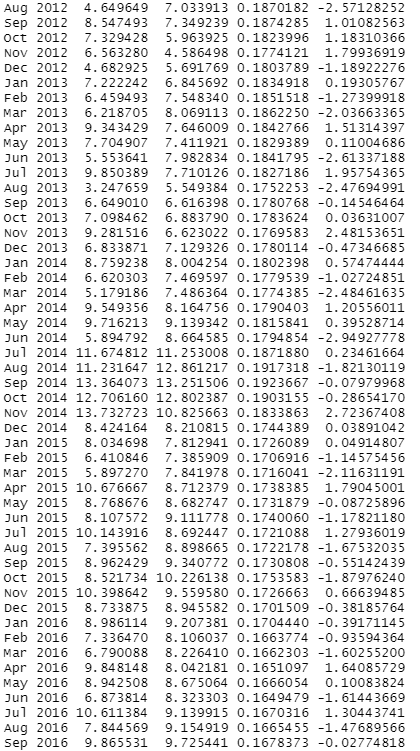
A graph is plotted between the actual time series and forecasted time series to see the differences as a pictorial representation. The actual time series being a seasonal time series can be decomposed into different components and is stored in ‘windspeedtseriescomp’. These different components are estimated values. The ‘seasonal’ component is viewed here and the graph of all the components is plotted.

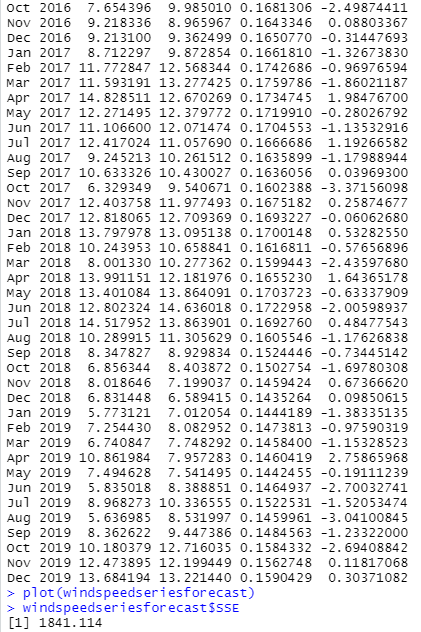


For forecasting using Exponential Smoothing method, ‘HoltWinters()’ function is used. As it is a seasonal time series there is no need of providing values for the ‘beta’ and ‘gamma’ parameters. Only the ‘windspeed’ time series is required. It forecasts the ‘trend’ and ‘seasonal’ component. The forecast can be viewed by using the named element ‘fitted’.

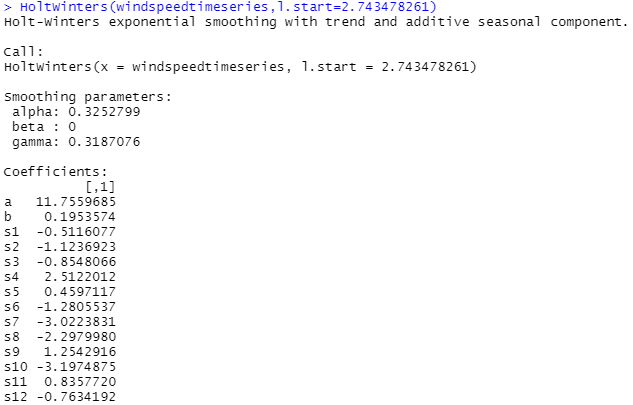


The forecasted values as compared to actual values is plotted. The sum of squared errors for the in-sample forecast errors is stored in a named element of the list variable called “SSE”.

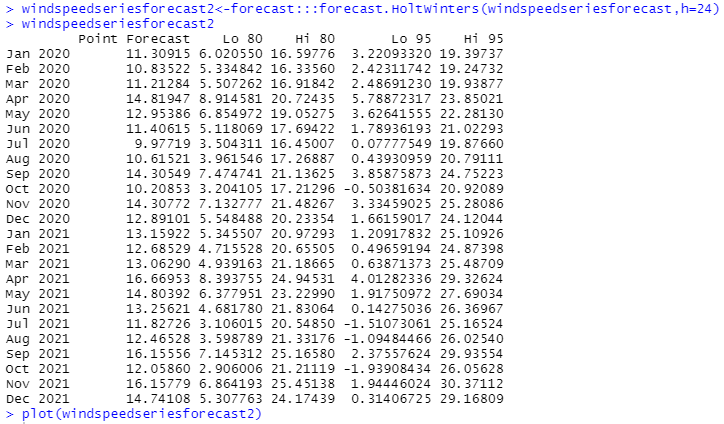




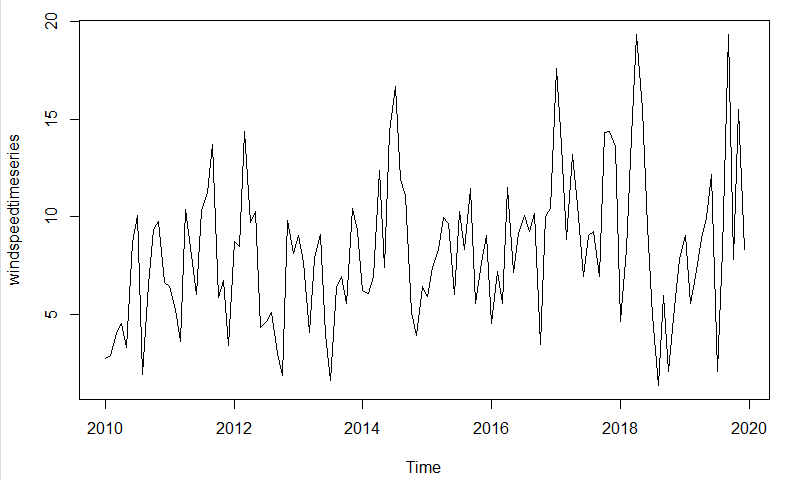
In case we want to use the first value in the time series as the initial value for the level in exponential smoothing, we can specify the value using the ‘l.start’ parameter in the HoltWinters() function. Here, the initial value of ‘windspeed’ column is specified as ‘l.start’ value.



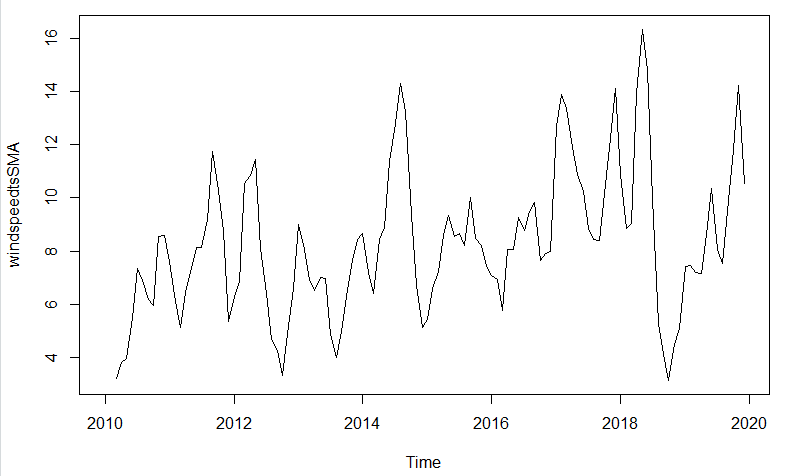
To forecast the ‘windspeed’ values for future months, the ‘forecast.HoltWinters()’ is used. It takes the parameters – previous predicted model and number of months as ‘h’ value, for the forecasting. The forecasted values can also be plotted to see the trend in the values.



Plot for ‘plot.ts(windspeedtimeseries)’ –



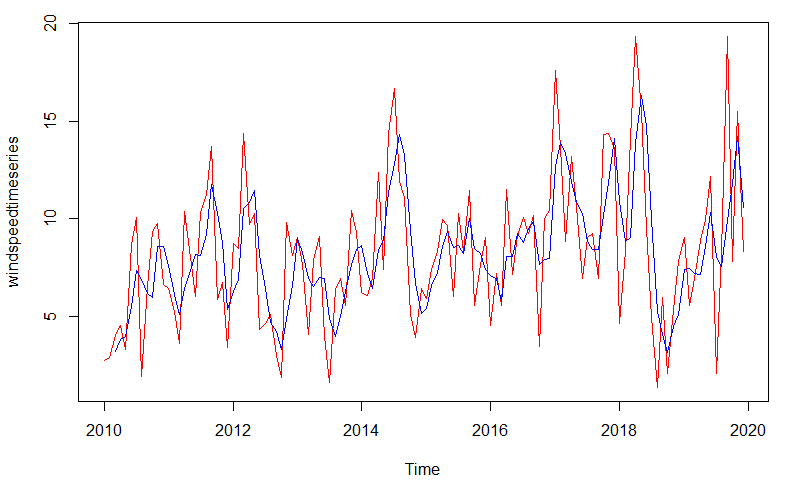
Plot for ‘plot.ts(windspeedtsSMA)’ –



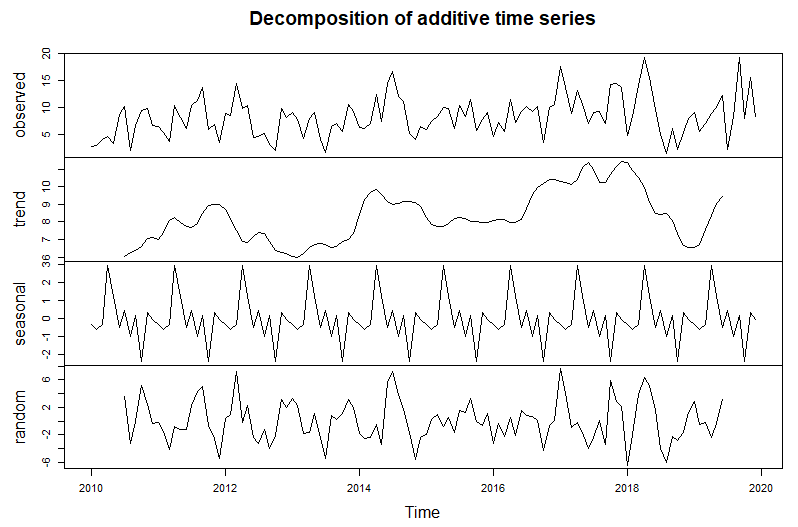
Plot for

plot(windspeedtimeseries,col='red') #Actual time series

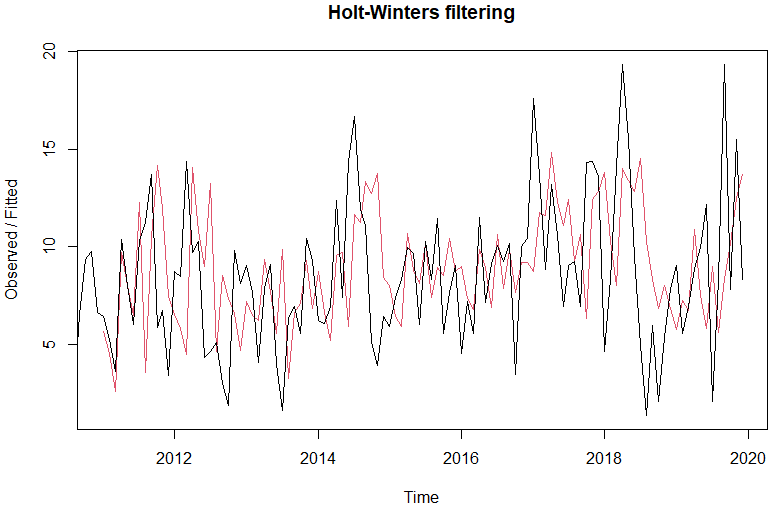
lines(windspeedtsSMA,col='blue') # SMA forecasted time series



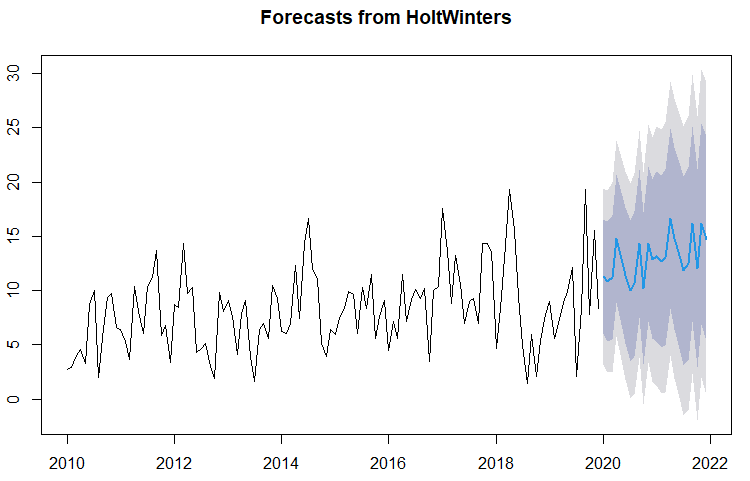
Plot for ‘plot(windspeedtseriescomp)’ –



Plot for ‘plot(windspeedseriesforecast)’ –

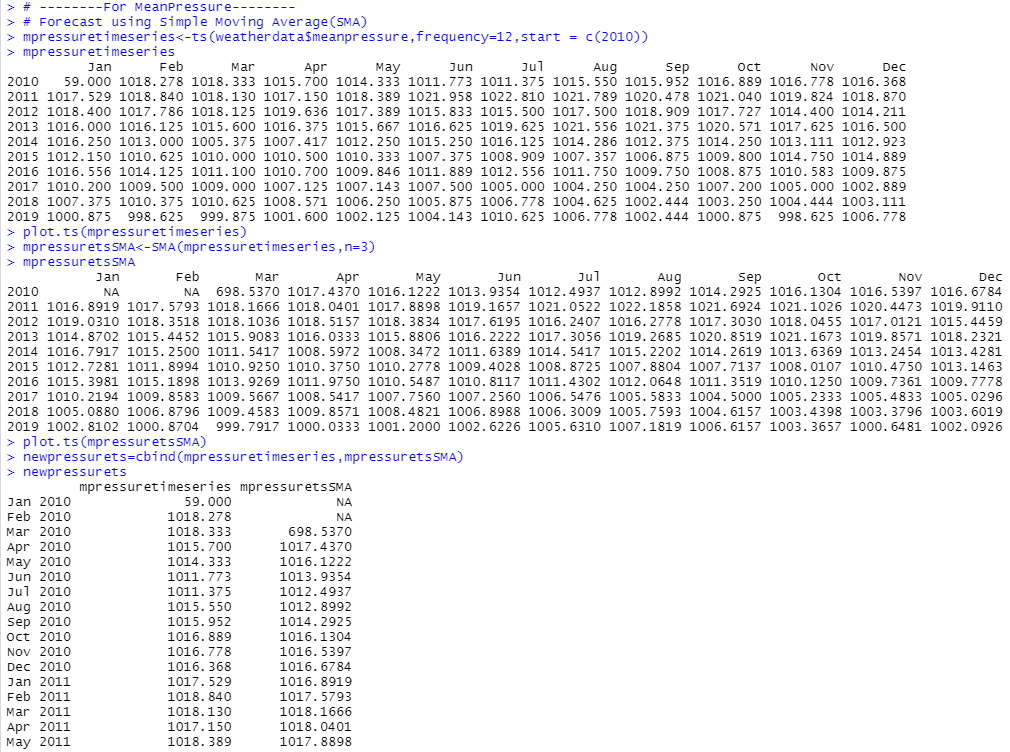


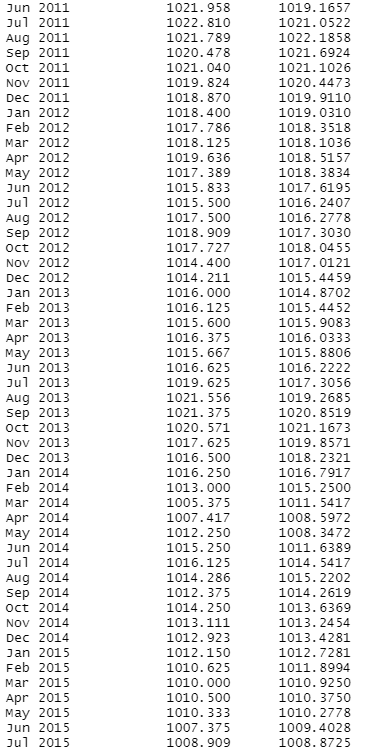
Plot for ‘plot(windspeedseriesforecast2)’ –

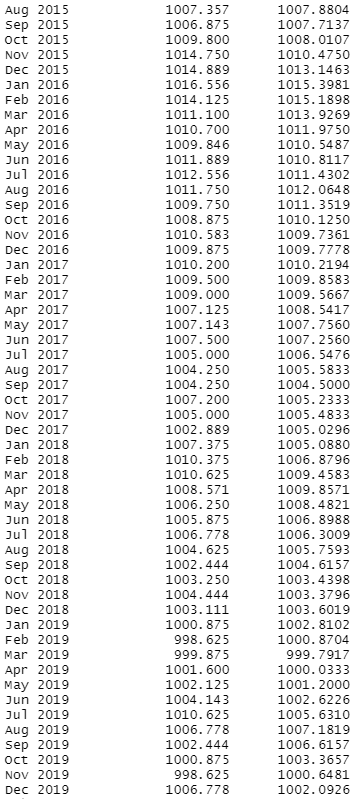


For ‘meanpressure’ column –

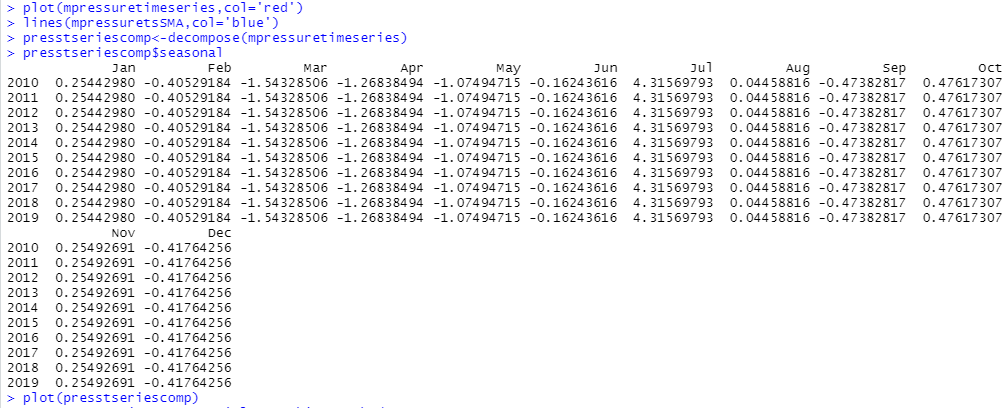
Converting the ‘meanpressure’ column as a seasonal time series object with the start year of collecting data as 2010. Then plotting the ‘mpressuretimeseries’. Forecasting of values is done using the SMA() function with parameters – the ‘mpressuretimeseries’ and order ‘n’ as 3. The forecasted value is plotted to see the smoothing. The actual time series and forecasted time series are added to a new data frame to see the differences between the corresponding values.



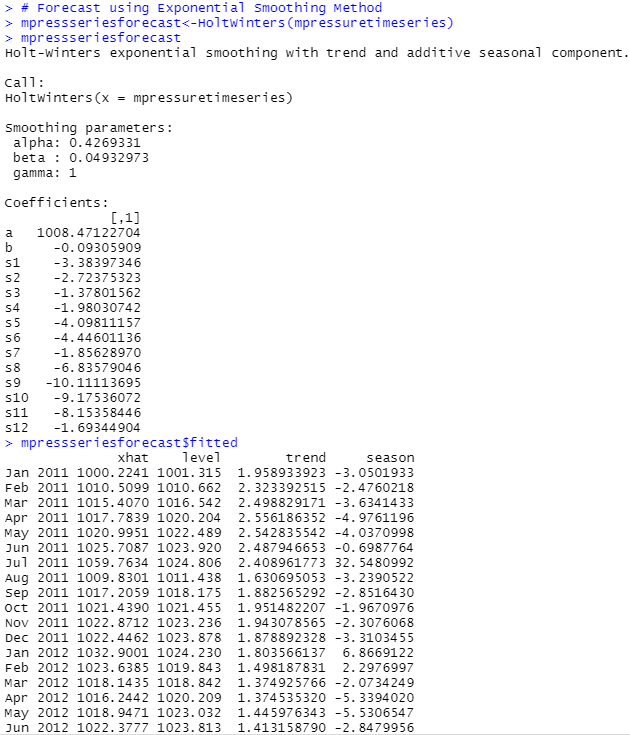




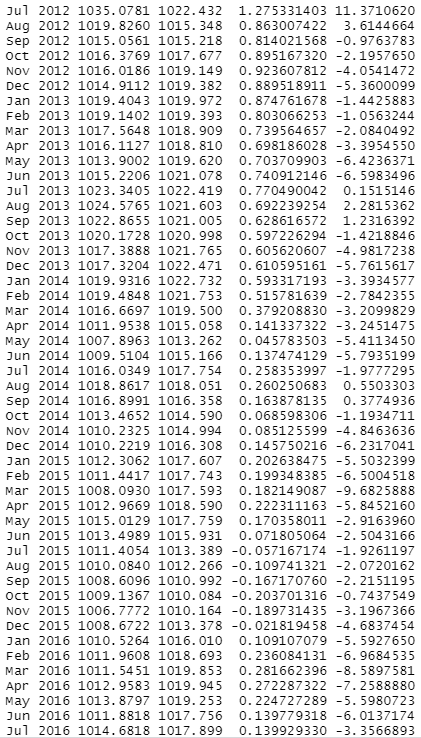
A graph is plotted between the actual time series and forecasted time series to see the differences as a pictorial representation. The actual time series being a seasonal time series can be decomposed into different components and is stored in ‘presstseriescomp’. These different components are estimated values. The ‘seasonal’ component is viewed here and the graph of all the components is plotted.

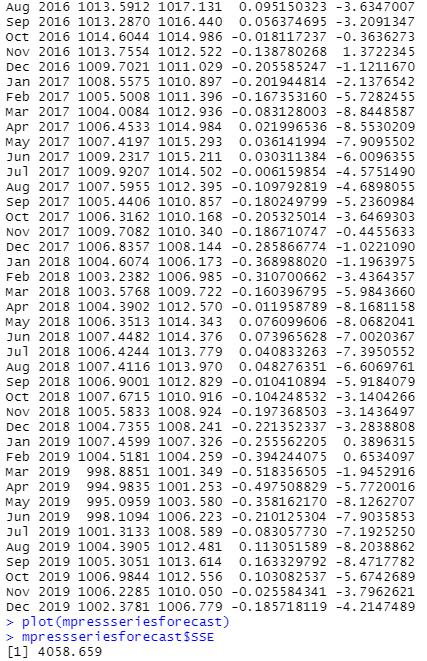


For forecasting using Exponential Smoothing method, ‘HoltWinters()’ function is used. As it is a seasonal time series there is no need of providing values for the ‘beta’ and ‘gamma’ parameters. Only the ‘meanpressure’ time series is required. It forecasts the ‘trend’ and ‘seasonal’ component. The forecast can be viewed by using the named element ‘fitted’.

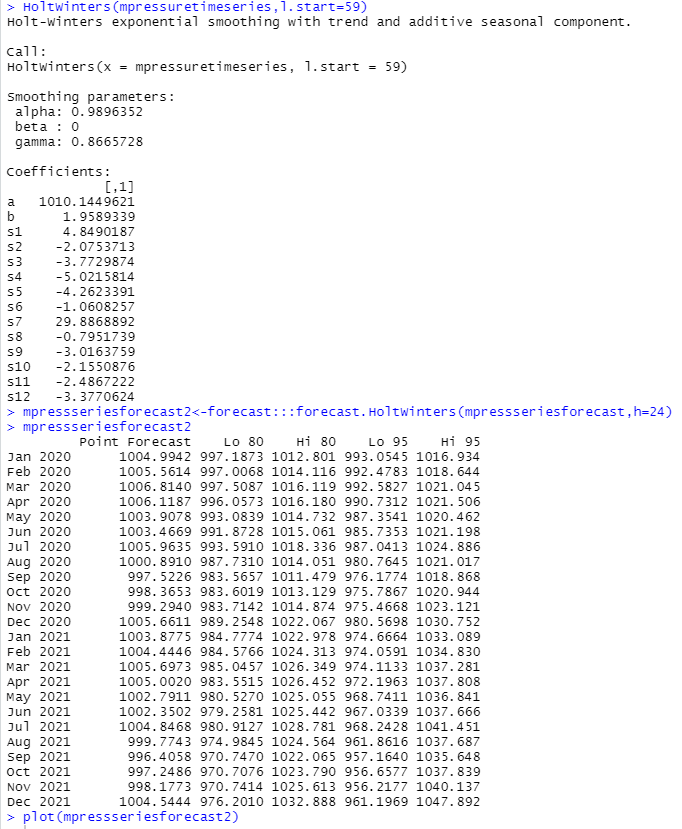


The forecasted values as compared to actual values is plotted. The sum of squared errors for the in-sample forecast errors is stored in a named element of the list variable called “SSE”.





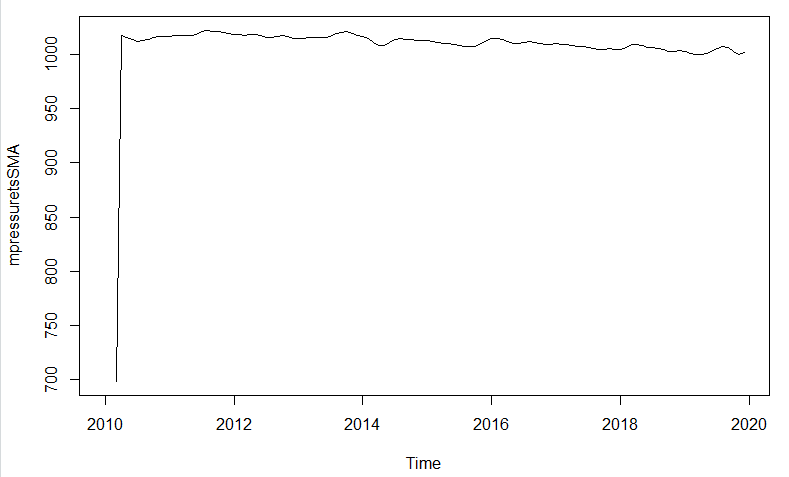
In case we want to use the first value in the time series as the initial value for the level in exponential smoothing, we can specify the value using the ‘l.start’ parameter in the HoltWinters() function. Here, the initial value of ‘meanpressure’ column is specified as ‘l.start’ value. To forecast the ‘meanpressure’ values for future months, the ‘forecast.HoltWinters()’ is used. It takes the parameters – previous predicted model and number of months as ‘h’ value, for the forecasting. The forecasted values can also be plotted to see the trend in the values.



Plot for ‘plot.ts(mpressuretimeseries)’ –



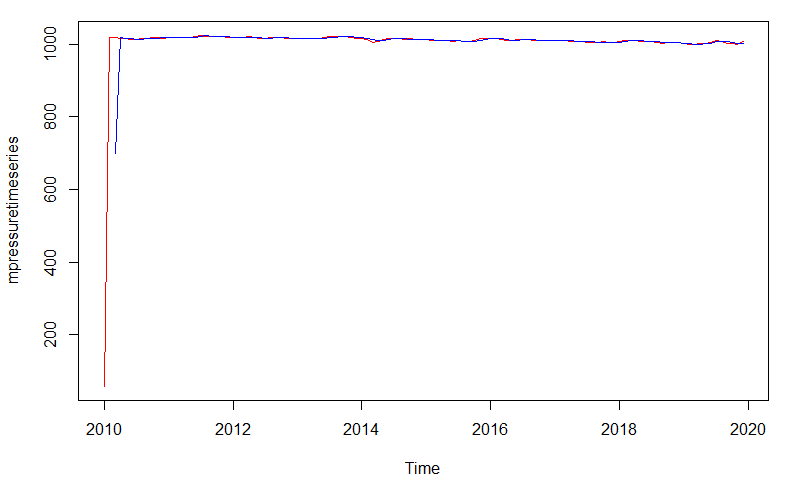
Plot for ‘plot.ts(mpressuretsSMA)’ –



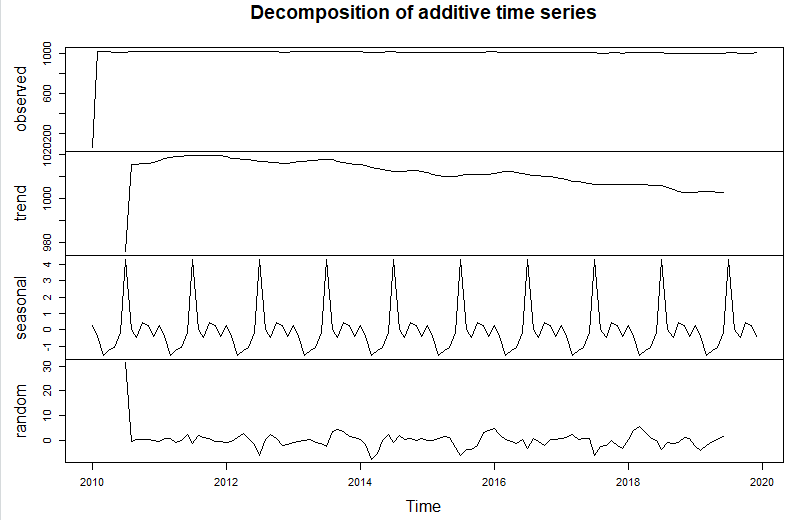
Plot for

plot(mpressuretimeseries,col='red') # Actual time series

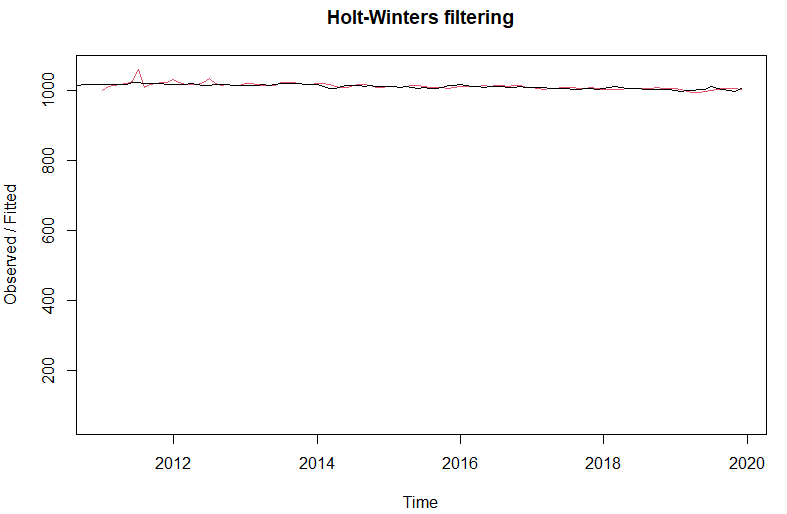
lines(mpressuretsSMA,col='blue') # SMA forecasted time series



Plot for ‘plot(presstseriescomp)’ –



Plot for ‘plot(mpressseriesforecast)’ –



Plot for ‘plot(mpressseriesforecast2)’ –

