**FORECASTING OF MONTHLY MILK PRODUCTION DATA [1962-1965]**

**INTRODUCTION**

***Problem Description:***

Here in this problem we are interested in

Choosing a seasonal data set and answering the following questions,

1. Is the model additive or multiplicative in nature?
2. Perform seasonal differencing and moving average smoothing to extract the stationary component from your data set.
3. Plot acf and pacf for the stationary component and comment about its behaviour.
4. Predict the next 2 values using a suitable forecasting technique for the chosen data set.

***Objective:***

The main objective of this problem is to choose seasonal component and identify if it is an additive or multiplicative model. Further we want to extract stationary component from your data set and we also want to check it’s behaviour using acf and pacf. We also want to predict the next 2 values using a simple exponential smoothing forecasting technique

***Seasonal Differencing:***

Seasonal differencing is a crude form of additive seasonal adjustment: the "index" which is subtracted from each value of the time series is simply the value that was observed in the same season one year earlier.

***Moving average smoothing technique:***

*We* use a simple smoothing technique called “moving average” to help determine the underlying trend in housing permits and other volatile data. A moving average smoothes a series by consolidating the monthly data points into longer units of time—namely an average of several months' data. It is used to estimate as well as eliminate the trend component from the dataset.

***Autocorrelation function:***

The autocorrelation function (ACF) defines how data points in a time series are related, on average, to the preceding data points (Box, Jenkins, & Reinsel, 1994). In other words, it measures the self-similarity of the signal over different delay times.

***Partial autocorrelation function****:*

In time series analysis, the partial autocorrelation function (PACF) gives the partial correlation of a stationary time series with its own lagged values, regressed the values of the time series at all shorter lags. It contrasts with the autocorrelation function, which does not control for other lags.

***Simple Exponential smoothing:***

Exponential smoothing may readily be generalized to deal with time series containing trend and seasonal variation. The version for handling a non-trend & non-seasonal data is usually called simple exponential smoothing.

***In sample forecast:***

In-sample forecast is the process of formally evaluating the predictive capabilities of the models developed using observed data to see how effective the algorithms are in reproducing data. It is kind of similar to a training set in a machine learning algorithm.

***Out of Sample forecast:***

An out of sample forecast instead uses all available data in the sample to estimate a models. For example, here estimation would be performed over 11, Jan- 12, jan, and the forecast(s) would commence in March, 13th jan and 14th jan 1975.

*#Setting and getting the current working directory.*  
**setwd**("E:/M.Sc/SEM III/TIME\_SERIES\_ANALYSIS(MST371)/Practical Labs")  
**getwd**()

## [1] "E:/M.Sc/SEM III/TIME\_SERIES\_ANALYSIS(MST371)/Practical Labs"

***Data Description:***

Here, in this problem we are taking into consideration monthly milk production dataset recorded from 1962-1975. It has total 168 observations.

There are two variables in the dataset i.e.

1. Date denoted by ***t***
2. Milk production value denoted by **zt**

*#Loading the package required to load the dataset.*  
**library**(readxl)  
  
*#Loading the 'milk production' dataset.*  
milk<- **read\_excel**("E:/M.Sc/SEM III/TIME\_SERIES\_ANALYSIS(MST371)/milkProduction.xlsx")  
  
*#Obtaining the first few records of the dataset.*  
**head**(milk)

## # A tibble: 6 x 2  
## Month Milk\_Production  
## <dttm> <dbl>  
## 1 1962-01-01 00:00:00 589  
## 2 1962-02-01 00:00:00 561  
## 3 1962-03-01 00:00:00 640  
## 4 1962-04-01 00:00:00 656  
## 5 1962-05-01 00:00:00 727  
## 6 1962-06-01 00:00:00 697

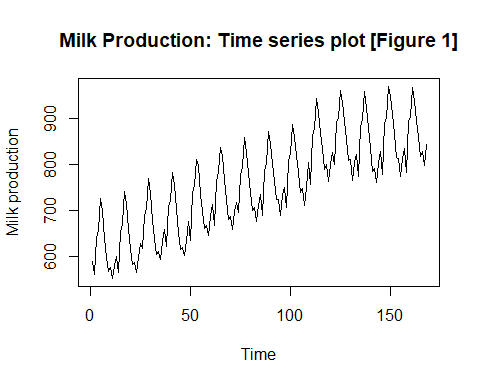
**ANALYSIS**

*#Extracing the data for the milk.production variable which we are interested in.*  
production <- milk**$**Milk\_Production  
  
*#Now converting the it into a time series data.*  
production=**ts**(production)  
  
*#Here. we are checking if the dataset has been converted into a timeseries data.*  
**class**(production)

## [1] "ts"

Hence, now the dataset we are interested in is a timeseries data.

*#Now we will first obtain the time series plot of the data to understand the nature of the time series data.*  
**ts.plot**(production, gpars = **list**(main="Milk Production: Time series plot [Figure 1]", xlab="Time",ylab="Milk production",lty=**c**(1**:**20)))



***Interpretation:*** From the above time series plot we observe that there exists a trend component in the dataset since there is observed a increase pattern for a longer period of time. Also there exist a seasonal component in the dataset since it is seen from the plot that there’s is repeatative movement over a period of time. Also we observe that there is some kind of irregularity in the dataset hence we can say that there also exists a error component in the dataset.

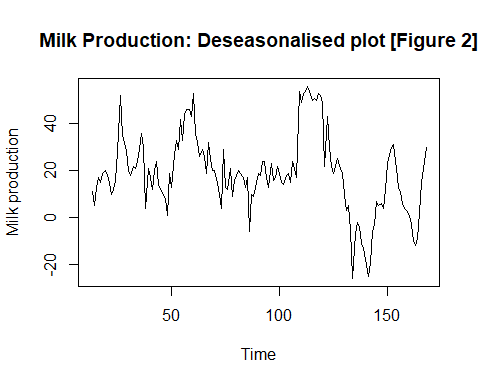
1. **Is the model additive or multiplicative in nature?**

Since the magnitude is same therefore the model is additive in nature.

1. **Perform seasonal differencing and moving average smoothing to extract the stationary component from your data set.**

2.1 Seasonal Differencing

*#Now, we proceed to perform the deseasonalising of data.*  
sd=**diff**(production,lag = 12)  
  
*#Obtaining the timeseries plot.*  
**ts.plot**(sd,gpars = **list**(main="Milk Production: Deseasonalised plot [Figure 2]", xlab="Time",ylab="Milk production",lty=**c**(1**:**20)))



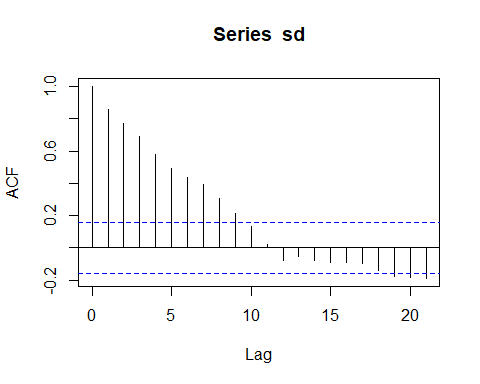
*#Obtaining the acf plot for the above time series after first differencing to check for stationarity.*  
**acf**(sd)  
  
*#loading the package 'tseries'*  
**library**(tseries)

## Warning: package 'tseries' was built under R version 4.0.5

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo



**MILK PRODUCTION: ACF PLOT [FIGURE 3]**



*#Now we want to validate for the stationarity of the new time series dataset using Augmented Fuller(ADF test) test. Dickey*  
**adf.test**(sd)

##   
## Augmented Dickey-Fuller Test  
##   
## data: sd  
## Dickey-Fuller = -2.8615, Lag order = 5, p-value = 0.2172  
## alternative hypothesis: stationary

***Interpretation:*** From the acf plot Figure 3 the series seems to be non stationary because most of the lag values are beyond the threshold line also from the above statistical test Augmented Dickey-Fuller Test it is observed tha pvalue = 0.2172 > 0.05, thus we accept the null hypothesis and also since seasonality is removed there might be a trend component still existing in the dataset thus we conclude that the time series is a non-stationary series.

2.2 Moving average smoothing

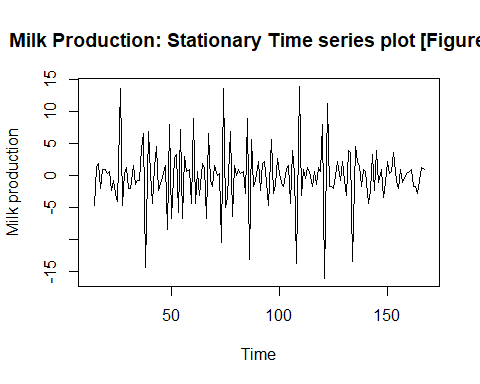
*#Loading the package 'forecast'*  
**library**(forecast)

## Warning: package 'forecast' was built under R version 4.0.5

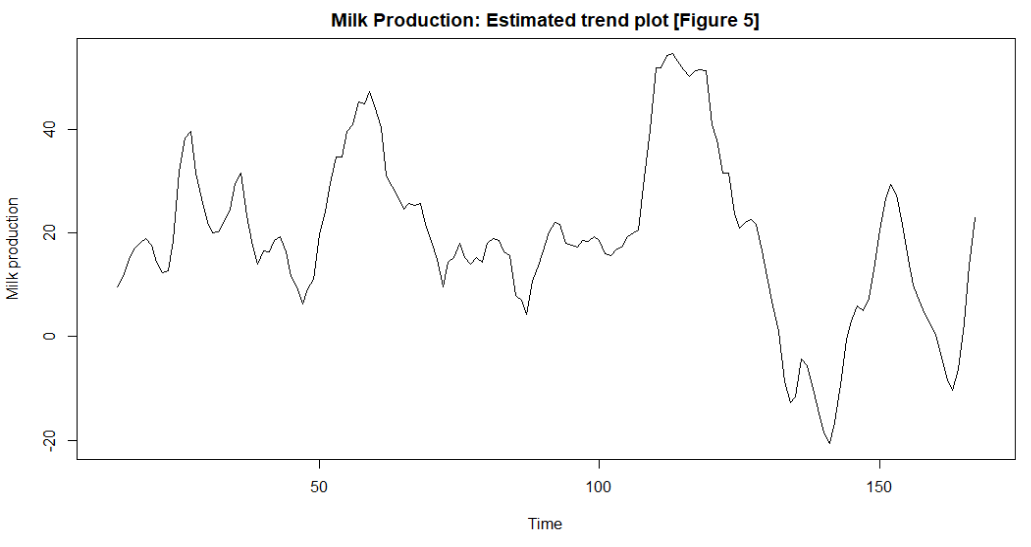
*#Here we are estimating the trend values using method of moving average smoothing of order 1.*  
ma1=**ma**(sd,order=3)  
ma1

## Time Series:  
## Start = 13   
## End = 168   
## Frequency = 1   
## [1] NA 9.6666667 11.6666667 15.0000000 17.0000000 18.0000000  
## [7] 19.0000000 17.6666667 14.3333333 12.3333333 12.6666667 18.6666667  
## [13] 32.0000000 38.3333333 39.6666667 31.6666667 26.6666667 22.0000000  
## [19] 20.0000000 20.3333333 22.3333333 24.6666667 29.6666667 31.6666667  
## [25] 23.3333333 18.3333333 14.0000000 16.6666667 16.3333333 18.6666667  
## [31] 19.3333333 16.3333333 11.6666667 9.6666667 6.3333333 9.3333333  
## [37] 11.0000000 19.6666667 24.3333333 29.6666667 34.6666667 34.6666667  
## [43] 39.6666667 41.0000000 45.3333333 45.0000000 47.3333333 44.0000000  
## [49] 40.3333333 31.3333333 29.0000000 27.0000000 24.6666667 25.6666667  
## [55] 25.3333333 25.6666667 21.6666667 18.3333333 15.0000000 9.6666667  
## [61] 14.3333333 15.3333333 18.0000000 15.3333333 14.0000000 15.3333333  
## [67] 14.3333333 18.0000000 19.0000000 18.6666667 16.3333333 15.6666667  
## [73] 8.0000000 7.0000000 4.3333333 10.6666667 13.6666667 16.6666667  
## [79] 20.3333333 22.0000000 21.6666667 18.0000000 17.6666667 17.3333333  
## [85] 18.6666667 18.3333333 19.3333333 18.6666667 16.0000000 15.6666667  
## [91] 17.0000000 17.3333333 19.3333333 20.0000000 20.6666667 30.6666667  
## [97] 40.0000000 52.0000000 52.0000000 54.3333333 54.6666667 53.3333333  
## [103] 51.6666667 50.3333333 51.3333333 51.6666667 51.3333333 41.0000000  
## [109] 38.0000000 31.6666667 31.6666667 23.6666667 21.0000000 22.0000000  
## [115] 22.6666667 21.6666667 16.6666667 10.6666667 6.0000000 1.0000000  
## [121] -8.6666667 -12.6666667 -11.6666667 -4.3333333 -5.6666667 -9.3333333  
## [127] -14.0000000 -18.6666667 -20.6666667 -16.6666667 -9.3333333 -0.6666667  
## [133] 3.0000000 6.0000000 5.0000000 7.3333333 13.0000000 20.6666667  
## [139] 26.6666667 29.3333333 27.3333333 21.6666667 15.0000000 10.0000000  
## [145] 7.0000000 4.3333333 2.6666667 0.3333333 -4.0000000 -8.3333333  
## [151] -10.3333333 -6.3333333 2.6666667 13.6666667 23.0000000 NA

*#Now, we are eliminating the estimated trend from the time series by subtracting the estimated trend values from the corresponding time series values.*  
res1=sd**-**ma1  
  
*#Obtaining the time series plot for the detrended series by the method of moving average smoothing.*  
**ts.plot**(res1, gpars = **list**(main="Milk Production: Stationary Time series plot [Figure 4]", xlab="Time",ylab="Milk production",lty=**c**(1**:**20)))

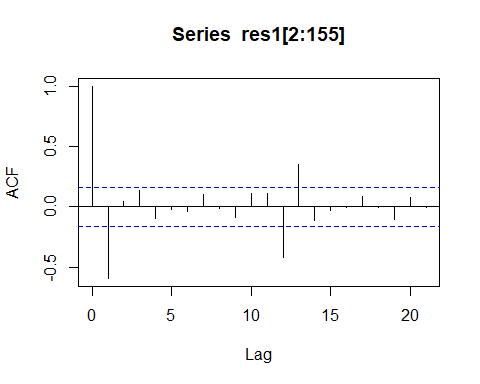


*#Obtaining the plot for estimated trends using moving average smoothing.*  
**ts.plot**(ma1, gpars = **list**(main="Milk Production: Estimated trend plot [Figure 4]", xlab="Time",ylab="Milk production",lty=**c**(1**:**20)))



*#Now, we obtain the acf plot for the detrended dataset.*  
**acf**(res1[2**:**155])

**MILK PRODUCTION: ACF PLOT**





*#Now, we check for the stationarity of the detrended dataset using Augmented Dickey-Fuller Test.*  
**adf.test**(res1[2**:**155])

## Warning in adf.test(res1[2:155]): p-value smaller than printed p-value

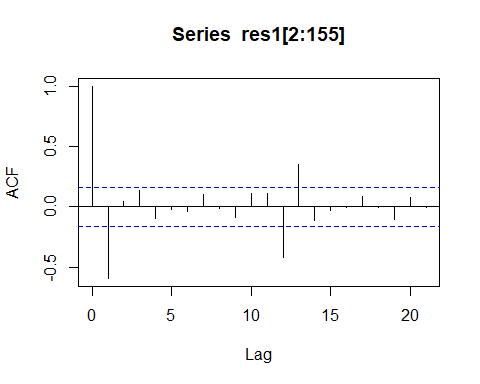
##   
## Augmented Dickey-Fuller Test  
##   
## data: res1[2:155]  
## Dickey-Fuller = -9.4081, Lag order = 5, p-value = 0.01  
## alternative hypothesis: stationary

Interpretation: From the acf plot the series seems to be non stationary because few of the lag values are beyond the threshold line but from the above statistical test i.e. Augmented Dickey-Fuller Test it is observed tha pvalue <0.01 which is < 0.05, thus we accept the alternative hypothesis and conclude that the timeseries is a stationary series.

1. **Plot acf and pacf for the stationary component and comment about its behaviour.**

*#Now, we obtain the acf plot for the stationary time series dataset.*  
**acf**(res1[2**:**155])

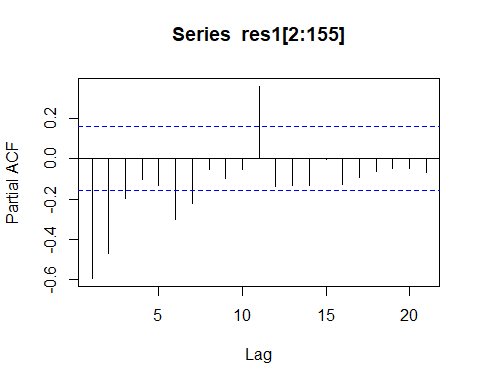
**MILK PRODUCTION: PACF PLOT**





*#Now, we obtain the pacf plot for the stationary time series dataset.*  
**pacf**(res1[2**:**155])

**MILK PRODUCTION: ACF PLOT**





***Interpretation:*** From the acf plot figure 7 it is observed that the grapgh is oscilatory which indicates that the data is stationary and also from the pacf plot figure 8 it is observed that it is a autoregressive process of order 2 since plot has consecutive significant value till lag 2.

1. **Predict the next 2 values using a suitable forecasting technique for the chosen data set.**

Since, we have removed the components from our time series data and hence, extracted the stationary component therefore we use simple exponential smoothing technique to predict the next two values for the chosen dataset.

*#Forecasting using Holt's exponential smoothing.*  
milkproduction\_forecast<-**HoltWinters**(res1[2**:**155],beta=FALSE, gamma = FALSE)  
milkproduction\_forecast

## Holt-Winters exponential smoothing without trend and without seasonal component.  
##   
## Call:  
## HoltWinters(x = res1[2:155], beta = FALSE, gamma = FALSE)  
##   
## Smoothing parameters:  
## alpha: 0.05576878  
## beta : FALSE  
## gamma: FALSE  
##   
## Coefficients:  
## [,1]  
## a -0.09429867

***Interpretation:*** Thus, we observe from the above model that the alpha value = 0.05576878 which is farther from 1 which indicates that the weightage is given to all the observations that means forecast is not just on the basis of recent past observation it is based on all the observations.

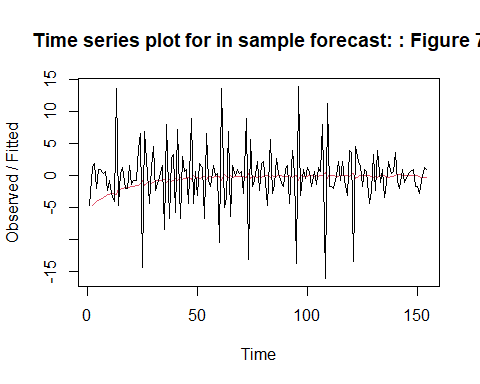
Obtaining the in sample and out of sample forecast for next 2 data points.

*#Performing in-sample forecast.*  
milkproduction\_forecast**$**fitted

## Time Series:  
## Start = 2   
## End = 154   
## Frequency = 1   
## xhat level  
## 2 -4.666666667 -4.666666667  
## 3 -4.332053981 -4.332053981  
## 4 -3.978923050 -3.978923050  
## 5 -3.868560924 -3.868560924  
## 6 -3.597047216 -3.597047216  
## 7 -3.340675497 -3.340675497  
## 8 -3.135780503 -3.135780503  
## 9 -2.923722660 -2.923722660  
## 10 -2.890797367 -2.890797367  
## 11 -2.766760309 -2.766760309  
## 12 -2.761178209 -2.761178209  
## 13 -2.830265790 -2.830265790  
## 14 -1.910251978 -1.910251978  
## 15 -2.063973865 -2.063973865  
## 16 -1.930278965 -1.930278965  
## 17 -1.748271285 -1.748271285  
## 18 -1.762309889 -1.762309889  
## 19 -1.775565577 -1.775565577  
## 20 -1.583596481 -1.583596481  
## 21 -1.569639610 -1.569639610  
## 22 -1.519281910 -1.519281910  
## 23 -1.471732597 -1.471732597  
## 24 -1.147991147 -1.147991147  
## 25 -0.712177207 -0.712177207  
## 26 -1.471812479 -1.471812479  
## 27 -0.999349825 -0.999349825  
## 28 -0.925027710 -0.925027710  
## 29 -1.115104760 -1.115104760  
## 30 -0.978558352 -0.978558352  
## 31 -0.663731035 -0.663731035  
## 32 -0.756842719 -0.756842719  
## 33 -0.751813711 -0.751813711  
## 34 -0.691296383 -0.691296383  
## 35 -0.559795658 -0.559795658  
## 36 -0.993316378 -0.993316378  
## 37 -0.491770087 -0.491770087  
## 38 -0.836136542 -0.836136542  
## 39 -0.640789477 -0.640789477  
## 40 -0.419157492 -0.419157492  
## 41 -0.711804682 -0.711804682  
## 42 -0.263137142 -0.263137142  
## 43 -0.620254178 -0.620254178  
## 44 -0.418357016 -0.418357016  
## 45 -0.357846568 -0.357846568  
## 46 -0.282121120 -0.282121120  
## 47 -0.508052286 -0.508052286  
## 48 0.022200199 0.022200199  
## 49 -0.220702597 -0.220702597  
## 50 -0.171215095 -0.171215095  
## 51 -0.328972980 -0.328972980  
## 52 -0.199088996 -0.199088996  
## 53 -0.113627671 -0.113627671  
## 54 -0.479082667 -0.479082667  
## 55 -0.080572938 -0.080572938  
## 56 -0.113258671 -0.113258671  
## 57 -0.199890341 -0.199890341  
## 58 -0.095794732 -0.095794732  
## 59 -0.090452377 -0.090452377  
## 60 -0.066818364 -0.066818364  
## 61 -0.639369389 -0.639369389  
## 62 0.158460802 0.158460802  
## 63 -0.129220268 -0.129220268  
## 64 -0.307909748 -0.307909748  
## 65 0.099643470 0.099643470  
## 66 -0.259115804 -0.259115804  
## 67 -0.151717263 -0.151717263  
## 68 -0.143256177 -0.143256177  
## 69 -0.079498173 -0.079498173  
## 70 -0.056475063 -0.056475063  
## 71 -0.016146331 -0.016146331  
## 72 -0.163962619 -0.163962619  
## 73 0.347100405 0.347100405  
## 74 -0.397251114 -0.397251114  
## 75 -0.059073811 -0.059073811  
## 76 -0.148727305 -0.148727305  
## 77 -0.177612152 -0.177612152  
## 78 -0.037579783 -0.037579783  
## 79 -0.165611160 -0.165611160  
## 80 -0.044837666 -0.044837666  
## 81 0.087790032 0.087790032  
## 82 0.027125308 0.027125308  
## 83 -0.234641748 -0.234641748  
## 84 0.094467028 0.094467028  
## 85 -0.059518032 -0.059518032  
## 86 -0.130557159 -0.130557159  
## 87 0.025440604 0.025440604  
## 88 0.042611406 0.042611406  
## 89 -0.015533761 -0.015533761  
## 90 -0.107615430 -0.107615430  
## 91 -0.045845068 -0.045845068  
## 92 0.049659624 0.049659624  
## 93 -0.194774550 -0.194774550  
## 94 0.039162913 0.039162913  
## 95 0.055568438 0.055568438  
## 96 -0.709703885 -0.709703885  
## 97 0.110638368 0.110638368  
## 98 -0.062838141 -0.062838141  
## 99 -0.003564954 -0.003564954  
## 100 -0.021955734 -0.021955734  
## 101 0.053627085 0.053627085  
## 102 0.087815555 0.087815555  
## 103 -0.010029780 -0.010029780  
## 104 0.027708756 0.027708756  
## 105 -0.048194902 -0.048194902  
## 106 0.028851244 0.028851244  
## 107 0.064421432 0.064421432  
## 108 0.506978975 0.506978975  
## 109 -0.413595119 -0.413595119  
## 110 0.241516760 0.241516760  
## 111 0.135099697 0.135099697  
## 112 0.034617383 0.034617383  
## 113 -0.078850748 -0.078850748  
## 114 -0.074453338 -0.074453338  
## 115 0.059825989 0.059825989  
## 116 0.019310380 0.019310380  
## 117 0.148360619 0.148360619  
## 118 0.102907541 0.102907541  
## 119 -0.070137830 -0.070137830  
## 120 0.156848795 0.156848795  
## 121 0.352587059 0.352587059  
## 122 -0.410660037 -0.410660037  
## 123 -0.127503717 -0.127503717  
## 124 0.009734166 0.009734166  
## 125 0.102139271 0.102139271  
## 126 0.003495120 0.003495120  
## 127 0.059068983 0.059068983  
## 128 0.092953965 0.092953965  
## 129 -0.153894682 -0.153894682  
## 130 -0.275439319 -0.275439319  
## 131 -0.074182467 -0.074182467  
## 132 -0.200172557 -0.200172557  
## 133 0.034065946 0.034065946  
## 134 -0.023602651 -0.023602651  
## 135 0.033482421 0.033482421  
## 136 -0.154280789 -0.154280789  
## 137 -0.201445519 -0.201445519  
## 138 -0.060083992 -0.060083992  
## 139 -0.038143587 -0.038143587  
## 140 0.001162821 0.001162821  
## 141 0.205583502 0.205583502  
## 142 0.156939174 0.156939174  
## 143 0.036649305 0.036649305  
## 144 0.090374199 0.090374199  
## 145 0.029565359 0.029565359  
## 146 0.009326942 0.009326942  
## 147 0.027396383 0.027396383  
## 148 0.063047708 0.063047708  
## 149 0.115300395 0.115300395  
## 150 0.015922264 0.015922264  
## 151 -0.077913669 -0.077913669  
## 152 -0.222285268 -0.222285268  
## 153 -0.247067877 -0.247067877  
## 154 -0.158930828 -0.158930828

*#Visualizing the in-sample forecast.*  
**plot**(milkproduction\_forecast, main = "Time series plot for in sample forecast")

**TIME SERIES PLOT FOR IN SAMPLE FORECAST**





Thus, the insample forecasted values are obtained in the above table and are plotted in figure 9.

*#Loading the library 'forecast'.*  
**library**(forecast)

*#Forecasting the last 2 observation to compare with actual out-sample forecast result if the model obtained is good or not.*  
milkproduction\_forecast1<-**HoltWinters**(res1[2**:**153], beta=FALSE, gamma = FALSE)  
milkproduction\_forecast1

## Holt-Winters exponential smoothing without trend and without seasonal component.  
##   
## Call:  
## HoltWinters(x = res1[2:153], beta = FALSE, gamma = FALSE)  
##   
## Smoothing parameters:  
## alpha: 0.05590065  
## beta : FALSE  
## gamma: FALSE  
##   
## Coefficients:  
## [,1]  
## a -0.2476273

forecast1<-**forecast**(milkproduction\_forecast1,h=2)  
forecast1

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 153 -0.2476273 -6.563053 6.067798 -9.906237 9.410982  
## 154 -0.2476273 -6.572912 6.077658 -9.921316 9.426062

Thus, from the above for table it is observed that the forecasted values are somewhat closer to the the actual values in the dataset.

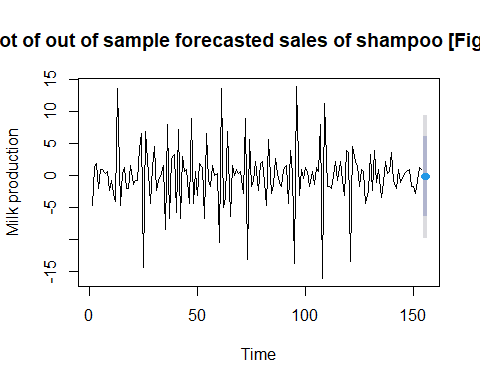
*#Performing out-sample 2 step ahead forecast for milk production on 13/1/1975 and 14/1/1975.*  
forecast2<-**forecast**(milkproduction\_forecast, h=2)  
forecast2

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 155 -0.09429867 -6.369150 6.180553 -9.690856 9.502259  
## 156 -0.09429867 -6.378901 6.190303 -9.705768 9.517171

Thus, the out of sample forecast for next 2 data points, i.e. milk production on 13/1/1975 and 14/1/1975 are obtained.

*#Plotting the forecasted value.*  
**plot**(forecast2, main = "Plot of out of sample forecasted value of milk production[Figure 10]", xlab ="Time", ylab="Milk production")

**PLOT OF OUT OF SAMPLE VALUE OF MILK PRODUCTION**





Thus, the forecasted values are plotted in the graph.

1. **Obtaining the accuracy measures and comment about the findings.**

Root mean square value

*#Loading the package 'Metrics'.*  
**library**(Metrics)

## Warning: package 'Metrics' was built under R version 4.0.5

##   
## Attaching package: 'Metrics'

## The following object is masked from 'package:forecast':  
##   
## accuracy

*#Obtaining the accuracy measure for the forecasting model obtained by removing the last value.*  
ma<-**rmse**(res1[2**:**155],milkproduction\_forecast**$**fitted)

## Warning in `-.default`(actual, predicted): longer object length is not a  
## multiple of shorter object length

ma

## [1] 4.718382

***Interpretation:*** It is observed from the above calculations that the values obtained using forecasting model are almost similar the actual values also the root mean square for the above model is 4.718382 which is evident to the fact that the above model is good in forecasting the future values.

**CONCLUSION**

Thus, from the above analysis we interpret that the forecasting model obtained for the milk production data is good in predicting future values that is the model is pretty accurate. Thus, the forecasted value for milk production from 13/1/1975 and 14/1/1975 is correct.