Group assignment

Jai Kushwaha

21/03/2020

# Problem Statement:

Sales of souvenir data have been provided in the fancy.txt file. Part A) Using the Winter-Holts methods and model the data and predict for the next 5 years. Your submission should contain the complete modeling steps with explanations. Include pictures and R-code where applicable. Part B) Using the ARIMA method model the data and predict for the next 5 years. Your submissions should contain the complete modeling steps with explanations. Include pictures and R-code where applicable.

# Dataset: fancy

Info on Data: contains monthly sales for a souvenir shop at a beach resort town in Queensland, Australia, for January 1987-December 1993

fancy = file.choose()  
library(timeSeries)

## Warning: package 'timeSeries' was built under R version 3.6.3

## Loading required package: timeDate

library(xts)

## Warning: package 'xts' was built under R version 3.6.2

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following object is masked from 'package:timeSeries':  
##   
## time<-

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

## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

library(tseries)

## Warning: package 'tseries' was built under R version 3.6.2

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

library(forecast)

## Warning: package 'forecast' was built under R version 3.6.2

## Registered S3 methods overwritten by 'forecast':  
## method from   
## fitted.fracdiff fracdiff  
## residuals.fracdiff fracdiff

library(quantmod)

## Warning: package 'quantmod' was built under R version 3.6.2

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 3.6.2

## Version 0.4-0 included new data defaults. See ?getSymbols.

library(ggplot2)  
class(fancy)

## [1] "character"

head(fancy)

## [1] "D:\\Study\\Great Lakes\\time series\\Group Assignment\\fancy.dat.txt"

fancy

## [1] "D:\\Study\\Great Lakes\\time series\\Group Assignment\\fancy.dat.txt"

read.table(fancy,header=F)

## V1  
## 1 1664.81  
## 2 2397.53  
## 3 2840.71  
## 4 3547.29  
## 5 3752.96  
## 6 3714.74  
## 7 4349.61  
## 8 3566.34  
## 9 5021.82  
## 10 6423.48  
## 11 7600.60  
## 12 19756.21  
## 13 2499.81  
## 14 5198.24  
## 15 7225.14  
## 16 4806.03  
## 17 5900.88  
## 18 4951.34  
## 19 6179.12  
## 20 4752.15  
## 21 5496.43  
## 22 5835.10  
## 23 12600.08  
## 24 28541.72  
## 25 4717.02  
## 26 5702.63  
## 27 9957.58  
## 28 5304.78  
## 29 6492.43  
## 30 6630.80  
## 31 7349.62  
## 32 8176.62  
## 33 8573.17  
## 34 9690.50  
## 35 15151.84  
## 36 34061.01  
## 37 5921.10  
## 38 5814.58  
## 39 12421.25  
## 40 6369.77  
## 41 7609.12  
## 42 7224.75  
## 43 8121.22  
## 44 7979.25  
## 45 8093.06  
## 46 8476.70  
## 47 17914.66  
## 48 30114.41  
## 49 4826.64  
## 50 6470.23  
## 51 9638.77  
## 52 8821.17  
## 53 8722.37  
## 54 10209.48  
## 55 11276.55  
## 56 12552.22  
## 57 11637.39  
## 58 13606.89  
## 59 21822.11  
## 60 45060.69  
## 61 7615.03  
## 62 9849.69  
## 63 14558.40  
## 64 11587.33  
## 65 9332.56  
## 66 13082.09  
## 67 16732.78  
## 68 19888.61  
## 69 23933.38  
## 70 25391.35  
## 71 36024.80  
## 72 80721.71  
## 73 10243.24  
## 74 11266.88  
## 75 21826.84  
## 76 17357.33  
## 77 15997.79  
## 78 18601.53  
## 79 26155.15  
## 80 28586.52  
## 81 30505.41  
## 82 30821.33  
## 83 46634.38  
## 84 104660.67

Observation: 84 data points.

readLines(fancy,n=10)

## [1] "1664.81" "2397.53" "2840.71" "3547.29" "3752.96" "3714.74" "4349.61"  
## [8] "3566.34" "5021.82" "6423.48"

fan = ts(as.vector(read.table(fancy,header=F)), start=1987, end=c(1993,12), frequency=12)   
class(fan)

## [1] "ts"

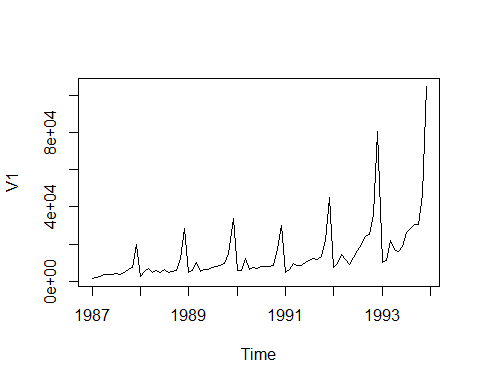
head(fan)

## Jan Feb Mar Apr May Jun  
## 1987 1664.81 2397.53 2840.71 3547.29 3752.96 3714.74

fan

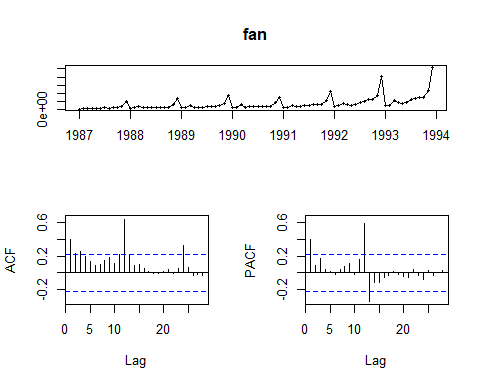
## Jan Feb Mar Apr May Jun Jul  
## 1987 1664.81 2397.53 2840.71 3547.29 3752.96 3714.74 4349.61  
## 1988 2499.81 5198.24 7225.14 4806.03 5900.88 4951.34 6179.12  
## 1989 4717.02 5702.63 9957.58 5304.78 6492.43 6630.80 7349.62  
## 1990 5921.10 5814.58 12421.25 6369.77 7609.12 7224.75 8121.22  
## 1991 4826.64 6470.23 9638.77 8821.17 8722.37 10209.48 11276.55  
## 1992 7615.03 9849.69 14558.40 11587.33 9332.56 13082.09 16732.78  
## 1993 10243.24 11266.88 21826.84 17357.33 15997.79 18601.53 26155.15  
## Aug Sep Oct Nov Dec  
## 1987 3566.34 5021.82 6423.48 7600.60 19756.21  
## 1988 4752.15 5496.43 5835.10 12600.08 28541.72  
## 1989 8176.62 8573.17 9690.50 15151.84 34061.01  
## 1990 7979.25 8093.06 8476.70 17914.66 30114.41  
## 1991 12552.22 11637.39 13606.89 21822.11 45060.69  
## 1992 19888.61 23933.38 25391.35 36024.80 80721.71  
## 1993 28586.52 30505.41 30821.33 46634.38 104660.67

plot(fan)

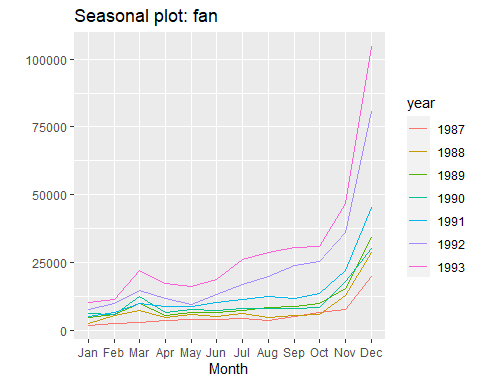


Observation: 1. Yearly spikes are there which indicates there is seasonality. 2. Slight increasing trend can be observed.

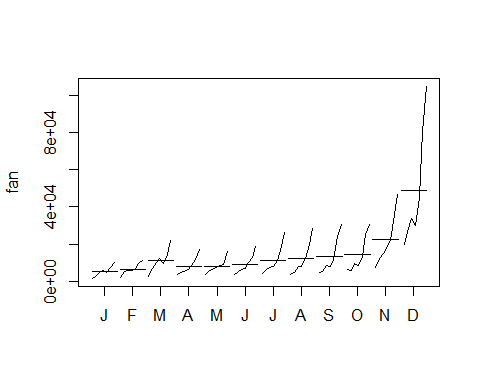
tsdisplay(fan)



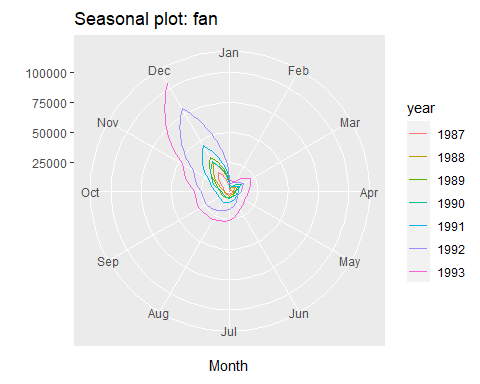
ggseasonplot(fan)

 Observation: 1. From the season plot we can see each year highest sales can be observed in the month of december. 2. Spikes can be in the month of March.

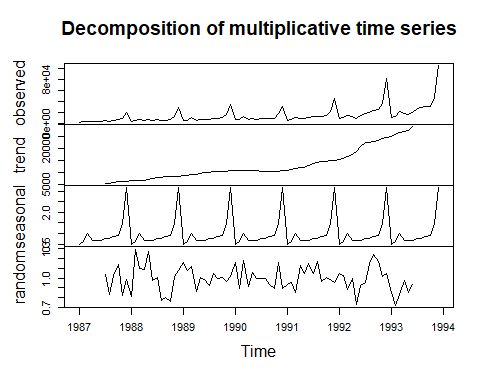
monthplot(fan)



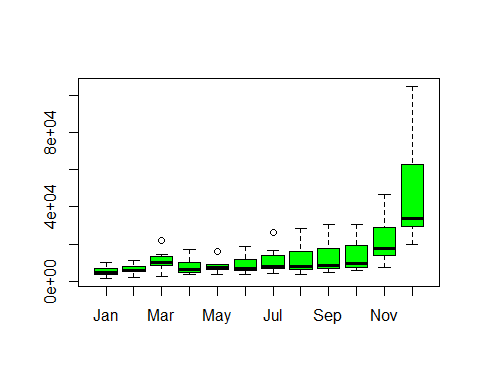
ggseasonplot(fan, polar=TRUE)



TSDecmpose<-decompose(fan, type = "multiplicative")   
plot(TSDecmpose)

 Observation: 1. From the above graph we can say there is trend and seasonality.

boxplot(split(fan, cycle(fan)), names = month.abb, col = "green")

 # Lets Check for stationarity of the series ###ADF Test Hypothesis: The null hypothesis is that a unit root is present in a time series sample. The alternative hypothesis is usually stationarity or trend-stationarity. ###kpss Test Hypothesis: The null hypothesis for the test is that the data is stationary. The alternate hypothesis for the test is that the data is not stationary.

kpss.test(fan)

## Warning in kpss.test(fan): p-value smaller than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: fan  
## KPSS Level = 1.3089, Truncation lag parameter = 3, p-value = 0.01

Observation: 1. Series is not stationary as pvalue<.05

kpss.test(log(fan))

## Warning in kpss.test(log(fan)): p-value smaller than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: log(fan)  
## KPSS Level = 1.7909, Truncation lag parameter = 3, p-value = 0.01

Observation: 1. Series still not stationary.

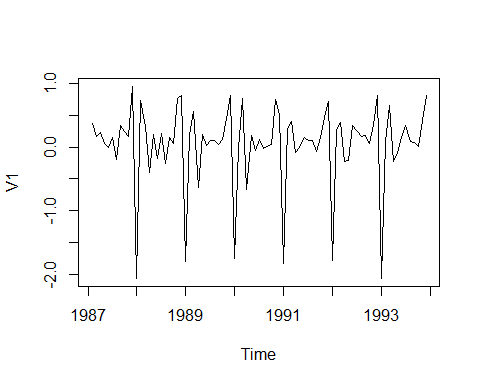
kpss.test(diff(log(fan)))

## Warning in kpss.test(diff(log(fan))): p-value greater than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: diff(log(fan))  
## KPSS Level = 0.062948, Truncation lag parameter = 3, p-value = 0.1

Observation: 1. Series now stationary.

plot(diff(log(fan)))

 Onservation: 1. Still the seasonality is there. Lets Try with sine function

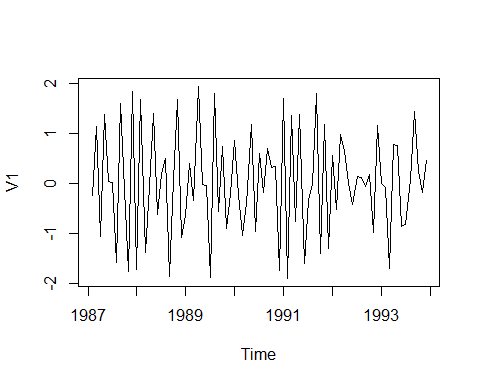
kpss.test(diff(sin(fan)))

## Warning in kpss.test(diff(sin(fan))): p-value greater than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: diff(sin(fan))  
## KPSS Level = 0.037499, Truncation lag parameter = 3, p-value = 0.1

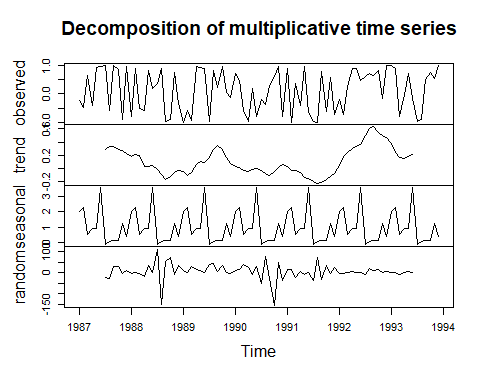
Observation: Series still stationary

plot(diff(sin(fan)))

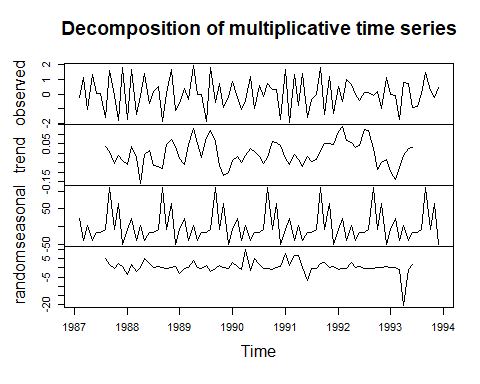


Onservation 1. Seanality removed through sine function.

TSDecmposeSin<-decompose(sin(fan), type = "multiplicative")   
plot(TSDecmposeSin)



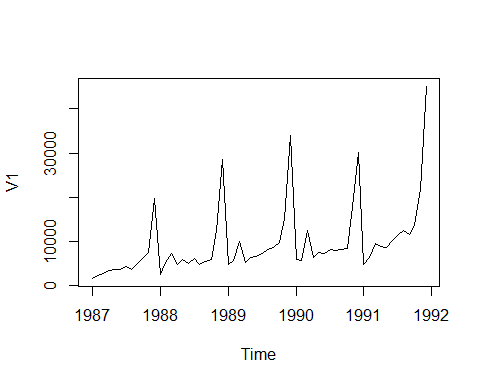
TSDecmposeSin<-decompose(diff(sin(fan)), type = "multiplicative")   
plot(TSDecmposeSin)

 Observation: 1. Some seasonality still there.

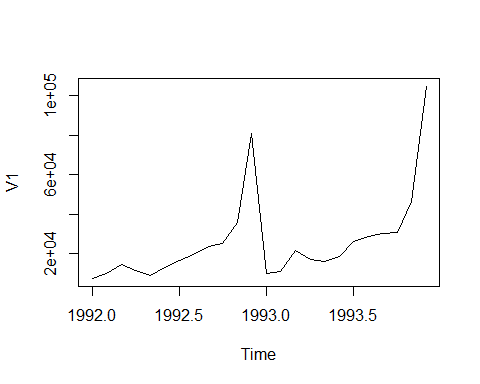
# Dividing the data into training( 5 years)and testing(2 years data).

fan\_train = window(fan, start=1987, end=c(1991,12), frequency=12)   
fan\_test = window(fan, start = c(1992,1), end = c(1993,12), frequency = 12)

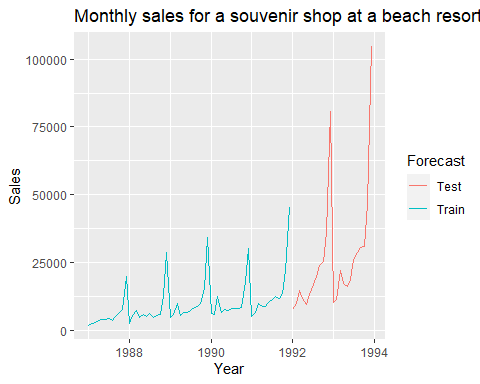
plot(fan\_train)



plot(fan\_test)



autoplot(fan\_train, series="Train") +   
 autolayer(fan\_test, series="Test") +   
 ggtitle("Monthly sales for a souvenir shop at a beach resort town in Queensland, Australia : Traning and Test data") +   
 xlab("Year") +   
 ylab("Sales") +   
 guides(colour=guide\_legend(title="Forecast"))

 # Stationarity Test for test and training

adf.test(fan\_train)

##   
## Augmented Dickey-Fuller Test  
##   
## data: fan\_train  
## Dickey-Fuller = -3.1882, Lag order = 3, p-value = 0.09786  
## alternative hypothesis: stationary

Observation 1. Series not stationary as the p value >.05 .

kpss.test(fan\_train)

## Warning in kpss.test(fan\_train): p-value smaller than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: fan\_train  
## KPSS Level = 0.78358, Truncation lag parameter = 3, p-value = 0.01

Observation: 1. Kpss test also suggesting the series is not stationary.

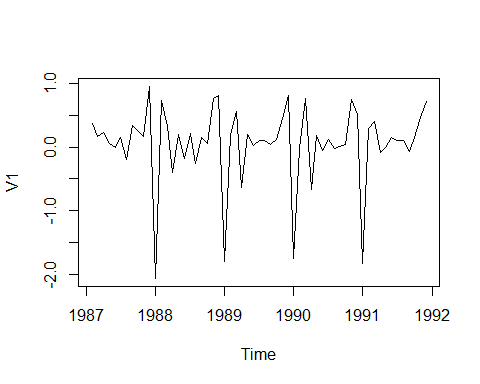
Making the series stationary by adding the transformation funtion.

kpss.test(diff(log(fan\_train)))

## Warning in kpss.test(diff(log(fan\_train))): p-value greater than printed p-  
## value

##   
## KPSS Test for Level Stationarity  
##   
## data: diff(log(fan\_train))  
## KPSS Level = 0.073614, Truncation lag parameter = 3, p-value = 0.1

plot(diff(log(fan\_train)))

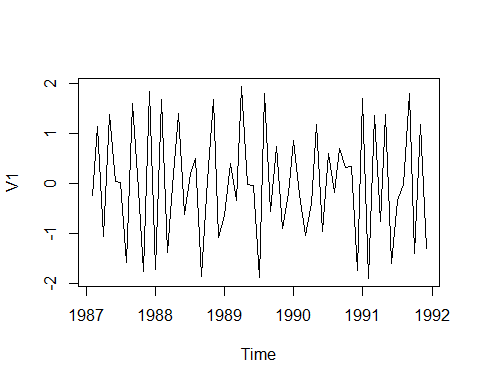
 OBservation: 1. Series is stationary but seasonality factor still there.

kpss.test(diff(sin(fan\_train)))

## Warning in kpss.test(diff(sin(fan\_train))): p-value greater than printed p-  
## value

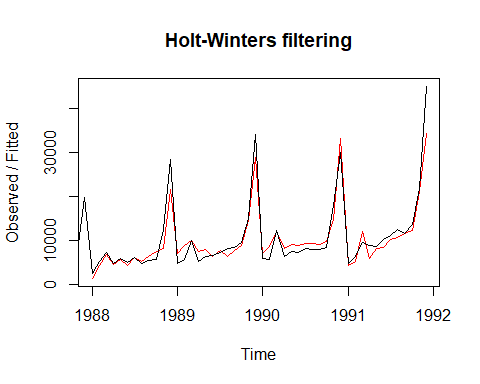
##   
## KPSS Test for Level Stationarity  
##   
## data: diff(sin(fan\_train))  
## KPSS Level = 0.05439, Truncation lag parameter = 3, p-value = 0.1

plot(diff(sin(fan\_train)))

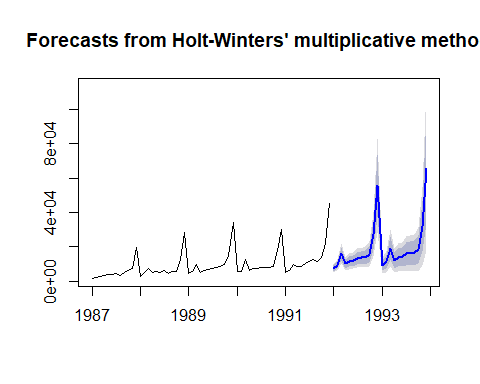
 Observation 1. Mean of the series is stable.

# Buliding Holts Winter Model

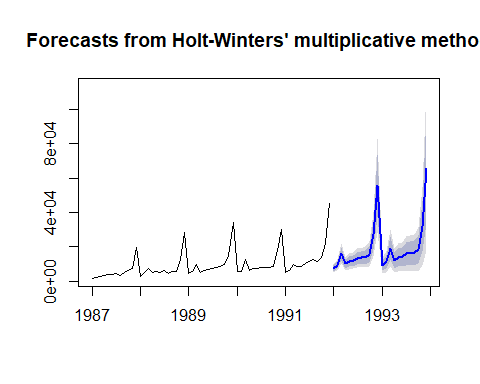
hwmodel=HoltWinters(fan\_train)  
plot(hwmodel)



library(forecast)  
hwmodel1 =hw(fan\_train, seasonal ="multiplicative")  
plot(hwmodel1)



hwforecast = forecast(hwmodel1, h =24)  
plot(hwforecast)

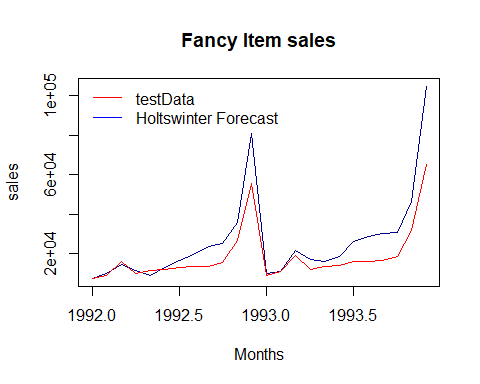


summary(hwmodel1)

##   
## Forecast method: Holt-Winters' multiplicative method  
##   
## Model Information:  
## Holt-Winters' multiplicative method   
##   
## Call:  
## hw(y = fan\_train, seasonal = "multiplicative")   
##   
## Smoothing parameters:  
## alpha = 0.2178   
## beta = 0.0198   
## gamma = 1e-04   
##   
## Initial states:  
## l = 4208.3295   
## b = 195.0312   
## s = 3.0591 1.5073 0.8884 0.7983 0.7929 0.7956  
## 0.7177 0.7055 0.6299 1.0177 0.6002 0.4875  
##   
## sigma: 0.1746  
##   
## AIC AICc BIC   
## 1124.164 1138.735 1159.767   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 59.63395 1206.105 934.3848 -1.783939 12.38556 0.3718488  
## ACF1  
## Training set 0.05860524  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 1992 7398.234 5743.181 9053.286 4867.049 9929.418  
## Feb 1992 9270.708 7139.314 11402.103 6011.022 12530.395  
## Mar 1992 15993.626 12207.158 19780.095 10202.723 21784.530  
## Apr 1992 10070.057 7610.864 12529.249 6309.046 13831.067  
## May 1992 11468.651 8575.632 14361.670 7044.161 15893.141  
## Jun 1992 11860.745 8766.828 14954.662 7129.008 16592.482  
## Jul 1992 13363.384 9755.654 16971.115 7845.837 18880.932  
## Aug 1992 13532.931 9749.475 17316.386 7746.634 19319.227  
## Sep 1992 13840.576 9831.948 17849.203 7709.909 19971.243  
## Oct 1992 15642.251 10947.952 20336.550 8462.941 22821.561  
## Nov 1992 26945.902 18566.544 35325.260 14130.779 39761.025  
## Dec 1992 55516.695 37629.287 73404.102 28160.266 82873.123  
## Jan 1993 8978.435 5981.692 11975.178 4395.312 13561.558  
## Feb 1993 11216.226 7339.318 15093.134 5287.007 17145.446  
## Mar 1993 19292.305 12389.036 26195.574 8734.666 29849.944  
## Apr 1993 12111.908 7627.214 16596.603 5253.160 18970.657  
## May 1993 13755.448 8487.473 19023.423 5698.776 21812.120  
## Jun 1993 14187.070 8570.187 19803.952 5596.789 22777.350  
## Jul 1993 15942.281 9420.563 22463.998 5968.175 25916.386  
## Aug 1993 16103.212 9300.198 22906.226 5698.900 26507.524  
## Sep 1993 16428.331 9264.850 23591.812 5472.732 27383.930  
## Oct 1993 18521.995 10190.497 26853.494 5780.068 31263.922  
## Nov 1993 31831.697 17069.103 46594.291 9254.258 54409.137  
## Dec 1993 65433.075 34162.769 96703.382 17609.268 113256.882

# Ploting only the Test data and Predicted values.

hw.mean <-forecast(hwmodel1,h=24)$mean  
  
plot(fan\_test, main="Fancy Item sales", ylab="sales", xlab="Months", col="darkblue")   
lines(hw.mean, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("testData","Holtswinter Forecast"))

 Observation: 1. Forecasted values are somewhat more than actual values of test data.

accuracy(hwmodel1, fan\_test)

## ME RMSE MAE MPE MAPE MASE  
## Training set 59.63395 1206.105 934.3848 -1.783939 12.38556 0.3718488  
## Test set 7611.21070 11910.654 7908.8205 21.577547 24.30646 3.1474027  
## ACF1 Theil's U  
## Training set 0.05860524 NA  
## Test set 0.34742262 0.7454845

Observation: 1. There is a significant difference between Mape values of Forecast values and actual test data.

shapiro.test(hwforecast$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: hwforecast$residuals  
## W = 0.98618, p-value = 0.731

Observation: 1. As p value>.05 Null hypothesis being normally distributed. Errors are normally distributed.

# Let’s check for any autocorrelation between errors using box test

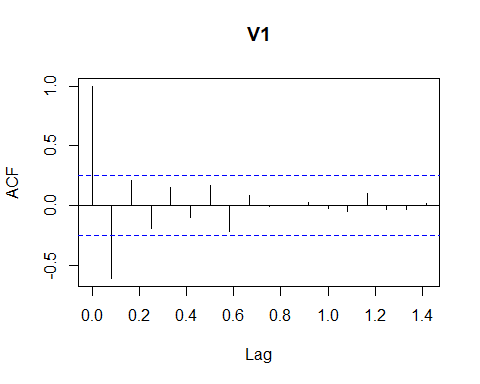
Box.test(hwforecast$residuals, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: hwforecast$residuals  
## X-squared = 0.037058, df = 1, p-value = 0.8473

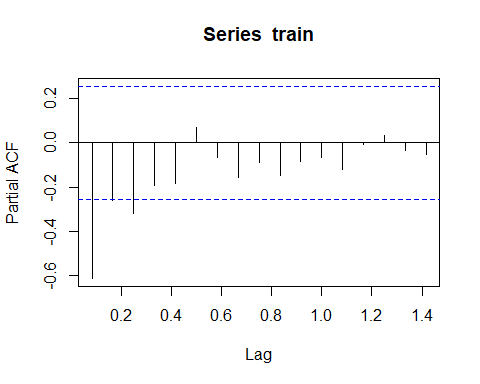
Observation: 1. Null hypothesis of box test is the model is fit or there is not autocorrelation. 2. As p value>.05 ie..84 there is no null hypothesis being true there is no auto correlation.

# Building Arima Model

train = diff(sin(fan\_train)) # d value is 1.  
acf(train)

 Observation: 1. q value from the above graph is 1.

pacf(train)

 Observation: 1. p value from the above graph is 3.

fitARIMA <- arima(log(fan\_train), order=c(3,1,1),seasonal = list(order = c(3,1,1), period = 12),method="ML")

#Prediction

pred = predict(fitARIMA,n.ahead = 2\*12)  
pred

## $pred  
## Jan Feb Mar Apr May Jun Jul  
## 1992 8.770482 9.245897 9.556392 9.470977 9.567486 9.499398 9.681456  
## 1993 9.183975 9.682655 9.976473 9.767158 9.906732 9.868952 10.032247  
## Aug Sep Oct Nov Dec  
## 1992 9.471473 9.647562 9.780020 10.339614 11.128624  
## 1993 9.920867 10.095881 10.250324 10.683609 11.584041  
##   
## $se  
## Jan Feb Mar Apr May Jun Jul  
## 1992 0.1360193 0.1421169 0.1528011 0.1561830 0.1616457 0.1658383 0.1707107  
## 1993 0.2061122 0.2104621 0.2151088 0.2190705 0.2231892 0.2271046 0.2310670  
## Aug Sep Oct Nov Dec  
## 1992 0.1750572 0.1795967 0.1836850 0.1883040 0.1924801  
## 1993 0.2348358 0.2386933 0.2422146 0.2463528 0.2500361

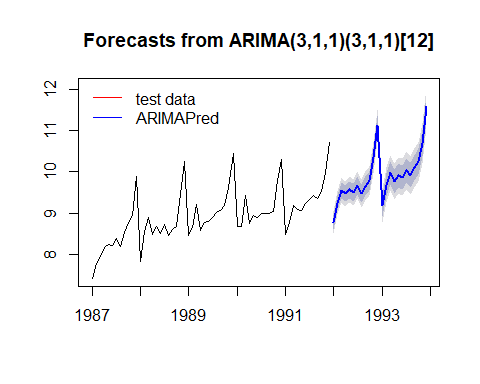
predf<-2.718^pred$pred  
predf

## Jan Feb Mar Apr May Jun  
## 1992 6435.421 10352.034 14120.754 12964.824 14278.269 13338.538  
## 1993 9730.520 16020.919 21492.063 17433.440 20044.386 19301.314  
## Jul Aug Sep Oct Nov Dec  
## 1992 16001.732 12971.257 15468.506 17659.098 30900.935 68014.060  
## 1993 22724.681 20329.699 24217.595 28261.657 43586.450 107241.693

sd\_forecast<-forecast(fitARIMA, 2\*12)  
sd\_forecast

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 1992 8.770482 8.596166 8.944798 8.503889 9.037075  
## Feb 1992 9.245897 9.063767 9.428027 8.967353 9.524441  
## Mar 1992 9.556392 9.360569 9.752214 9.256907 9.855876  
## Apr 1992 9.470977 9.270820 9.671134 9.164864 9.777090  
## May 1992 9.567486 9.360329 9.774643 9.250666 9.884306  
## Jun 1992 9.499398 9.286867 9.711928 9.174361 9.824435  
## Jul 1992 9.681456 9.462682 9.900231 9.346869 10.016043  
## Aug 1992 9.471473 9.247128 9.695818 9.128367 9.814579  
## Sep 1992 9.647562 9.417399 9.877724 9.295559 9.999565  
## Oct 1992 9.780020 9.544619 10.015422 9.420004 10.140037  
## Nov 1992 10.339614 10.098292 10.580935 9.970545 10.708683  
## Dec 1992 11.128624 10.881950 11.375297 10.751370 11.505878  
## Jan 1993 9.183975 8.919831 9.448118 8.780002 9.587947  
## Feb 1993 9.682655 9.412936 9.952373 9.270156 10.095153  
## Mar 1993 9.976473 9.700800 10.252146 9.554868 10.398079  
## Apr 1993 9.767158 9.486408 10.047908 9.337788 10.196528  
## May 1993 9.906732 9.620703 10.192760 9.469289 10.344174  
## Jun 1993 9.868952 9.577905 10.159998 9.423835 10.314068  
## Jul 1993 10.032247 9.736123 10.328371 9.579364 10.485130  
## Aug 1993 9.920867 9.619913 10.221821 9.460597 10.381136  
## Sep 1993 10.095881 9.789984 10.401779 9.628051 10.563712  
## Oct 1993 10.250324 9.939914 10.560735 9.775592 10.725056  
## Nov 1993 10.683609 10.367895 10.999323 10.200767 11.166452  
## Dec 1993 11.584041 11.263607 11.904476 11.093980 12.074103

sd\_forecast<-forecast(fitARIMA, 2\*12)  
plot(sd\_forecast)  
lines(fan\_test, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("test data","ARIMAPred"))



# Checking for accuracy

accuracy(sd\_forecast)

## ME RMSE MAE MPE MAPE  
## Training set -0.01695894 0.0969944 0.07363563 -0.1999694 0.8228154  
## MASE ACF1  
## Training set 0.2615879 -0.03844031

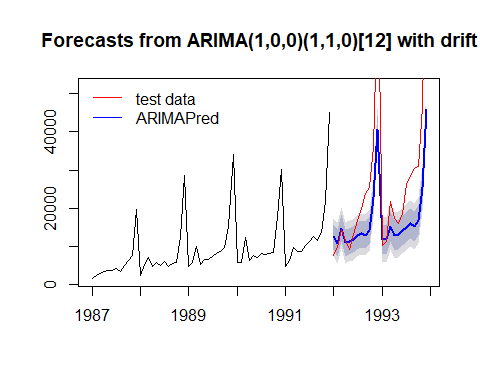
Observation: 1. MAPE value comparitive lower than Holtwinter model . Model is quite better.

summary(fitARIMA)

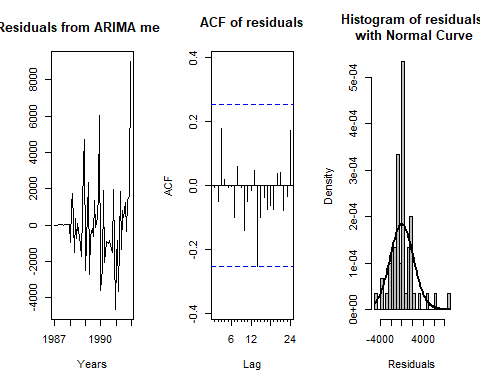
##   
## Call:  
## arima(x = log(fan\_train), order = c(3, 1, 1), seasonal = list(order = c(3, 1,   
## 1), period = 12), method = "ML")  
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 sar1 sar2 sar3 sma1  
## 0.0412 0.1466 -0.0876 -0.7245 -0.1749 -0.2563 -0.6032 -0.9999  
## s.e. 0.2865 0.2458 0.1838 0.2544 0.2432 0.2150 0.1805 3.9249  
##   
## sigma^2 estimated as 0.01199: log likelihood = 14.21, aic = -10.42  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE  
## Training set -0.01695894 0.0969944 0.07363563 -0.1999694 0.8228154  
## MASE ACF1  
## Training set 0.1854667 -0.03844031

arimaMod = auto.arima(fan\_train, stepwise=FALSE, approximation=FALSE)

arimaMod.Fr = forecast(arimaMod,h=24)  
  
# plot of the prediction and of the test set  
  
plot(arimaMod.Fr)  
lines(fan\_test, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("test data","ARIMAPred"))



res.fr = residuals(arimaMod.Fr)  
  
par(mfrow=c(1,3))  
  
plot(res.fr, main="Residuals from ARIMA method",  
 ylab="", xlab="Years")  
  
Acf(res.fr, main="ACF of residuals")  
  
u = residuals(arimaMod)  
  
m = mean(u)  
std = sqrt(var(u))  
hist(u, breaks=20, col="gray", prob=TRUE,   
xlab="Residuals", main="Histogram of residuals\n with Normal Curve")  
curve(dnorm(x, mean=m, sd=std),   
 col="black", lwd=2, add=TRUE)



accuracy(arimaMod)

## ME RMSE MAE MPE MAPE MASE  
## Training set 30.64675 2155.88 1420.407 -5.875649 15.12105 0.5652665  
## ACF1  
## Training set -0.006201752

Observation: 1. MAPE value increased while comparing with ARIMA model.

arimaMod$aic

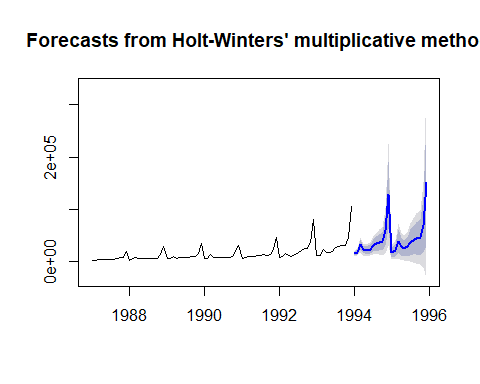
## [1] 895.4454

AIC value also increased.

# Predicting for 5 years with Holt Winters model .

# Using Holtswinter MOdel

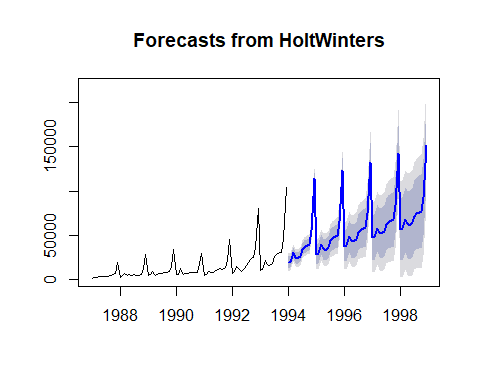
hwmodel1 =hw(fan, seasonal ="multiplicative") # Only 2 years data predicted  
hwforecast5 = forecast(hwmodel1,60)  
plot(hwforecast5)

 # Observation 1. Only 2 years forecasting done To resolve this we use Holtswinter funtion as below

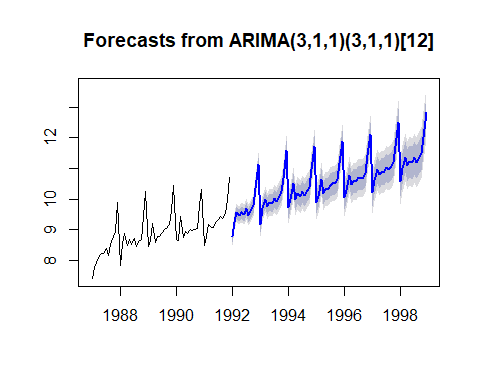
hwmodel = HoltWinters(fan)  
hwforecast5 = forecast(hwmodel,60)  
hwforecast5

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 1994 19336.94 12850.36 25823.52 9416.573 29257.30  
## Feb 1994 20039.00 13546.21 26531.79 10109.138 29968.86  
## Mar 1994 30337.70 23830.96 36844.43 20386.507 40288.89  
## Apr 1994 25705.19 19173.73 32236.65 15716.189 35694.20  
## May 1994 24266.11 17696.21 30836.01 14218.313 34313.91  
## Jun 1994 26903.89 20279.02 33528.76 16772.026 37035.75  
## Jul 1994 34522.35 27823.39 41221.31 24277.173 44767.53  
## Aug 1994 37068.94 30274.42 43863.45 26677.616 47460.26  
## Sep 1994 39199.81 32286.24 46113.37 28626.424 49773.19  
## Oct 1994 39844.30 32786.54 46902.07 29050.384 50638.23  
## Nov 1994 55978.11 48749.71 63206.52 44923.226 67033.00  
## Dec 1994 114004.40 106578.04 121430.77 102646.754 125362.05  
## Jan 1995 28680.67 17625.15 39736.19 11772.709 45588.63  
## Feb 1995 29382.73 18150.04 40615.42 12203.814 46561.65  
## Mar 1995 39681.43 28248.74 51114.12 22196.645 57166.22  
## Apr 1995 35048.93 23392.86 46705.00 17222.508 52875.35  
## May 1995 33609.85 21706.64 45513.05 15405.460 51814.23  
## Jun 1995 36247.62 24073.31 48421.94 17628.618 54866.63  
## Jul 1995 43866.09 31396.65 56335.52 24795.725 62936.45  
## Aug 1995 46412.67 33624.17 59201.16 26854.355 65970.99  
## Sep 1995 48543.54 35412.27 61674.82 28460.994 68626.09  
## Oct 1995 49188.04 35690.58 62685.50 28545.458 69830.62  
## Nov 1995 65321.85 51435.20 79208.50 44084.052 86559.64  
## Dec 1995 123348.14 109049.76 137646.51 101480.660 145215.62  
## Jan 1996 38024.41 20637.13 55411.68 11432.861 64615.95  
## Feb 1996 38726.47 20951.86 56501.07 11542.553 65910.38  
## Mar 1996 49025.17 30842.11 67208.23 21216.580 76833.76  
## Apr 1996 44392.66 25780.34 63004.99 15927.566 72857.76  
## May 1996 42953.58 23891.52 62015.64 13800.676 72106.48  
## Jun 1996 45591.36 26059.48 65123.23 15719.931 75462.79  
## Jul 1996 53209.82 33188.45 73231.19 22589.777 83829.86  
## Aug 1996 55756.40 35226.29 76286.52 24358.306 87154.50  
## Sep 1996 57887.28 36829.61 78944.94 25682.354 90092.20  
## Oct 1996 58531.77 36928.18 80135.36 25491.936 91571.61  
## Nov 1996 74665.58 52498.16 96833.01 40763.429 108567.74  
## Dec 1996 132691.87 109943.14 155440.61 97900.682 167483.06  
## Jan 1997 47368.14 21834.21 72902.07 8317.371 86418.91  
## Feb 1997 48070.20 21972.82 74167.58 8157.701 87982.70  
## Mar 1997 58368.90 31690.89 85046.92 17568.398 99169.41  
## Apr 1997 53736.40 26460.92 81011.87 12022.162 95450.63  
## May 1997 52297.31 24407.92 80186.70 9644.173 94950.45  
## Jun 1997 54935.09 26415.68 83454.51 11318.413 98551.77  
## Jul 1997 62553.55 33388.36 91718.74 17949.244 107157.86  
## Aug 1997 65100.14 35273.77 94926.51 19484.638 110715.64  
## Sep 1997 67231.01 36728.39 97733.63 20581.283 113880.74  
## Oct 1997 67875.51 36681.91 99069.10 20169.023 115581.99  
## Nov 1997 84009.32 52110.34 115908.29 35224.048 132794.58  
## Dec 1997 142035.61 109417.17 174654.05 92150.008 191921.20  
## Jan 1998 56711.87 21486.26 91937.49 2838.951 110584.80  
## Feb 1998 57413.93 21480.52 93347.35 2458.513 112369.36  
## Mar 1998 67712.64 31057.32 104367.95 11653.164 123772.11  
## Apr 1998 63080.13 25689.10 100471.16 5895.482 120264.78  
## May 1998 61641.05 23500.76 99781.33 3310.513 119971.58  
## Jun 1998 64278.83 25376.01 103181.65 4782.097 123775.56  
## Jul 1998 71897.29 32218.91 111575.66 11214.450 132580.12  
## Aug 1998 74443.87 33977.17 114910.57 12555.398 136332.35  
## Sep 1998 76574.75 35307.20 117842.29 13461.477 139688.01  
## Oct 1998 77219.24 35138.55 119299.93 12862.383 141576.10  
## Nov 1998 93353.05 50447.16 136258.94 27734.159 158971.94  
## Dec 1998 151379.34 107636.42 195122.27 84480.309 218278.37

plot(hwforecast5)

 # Using Arima Model

sd\_forecast5<-forecast(fitARIMA, 7\*12)  
plot(sd\_forecast5)



accuracy(fitARIMA)

## ME RMSE MAE MPE MAPE  
## Training set -0.01695894 0.0969944 0.07363563 -0.1999694 0.8228154  
## MASE ACF1  
## Training set 0.1854667 -0.03844031