greatlearning

Submitted to

GREAT LAKES
INSTITUTE OF MANAGEMENT
Global Mindset - Indian Roots

ASSIGNMENT SUBMITTED BY

MOHAMED YUSUF S

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 modelling 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 modelling 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

Understanding the data

Let's first read the given text file data in R which is assumed to be a time series data with a single column containing the sales details for the period of January 1987 to December 1993. We need to load the required libraries to run certain functions smoothly

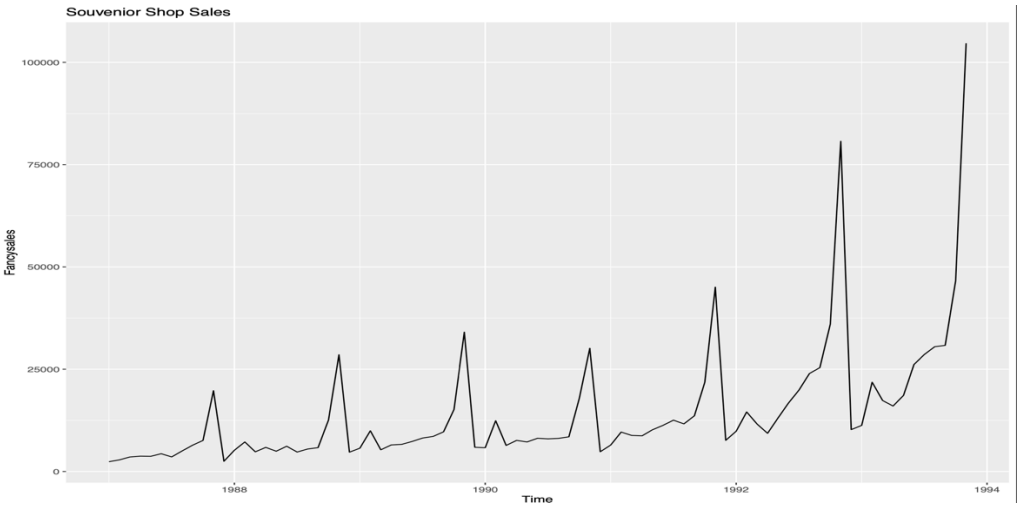
```
library(forecast)
library(ggplot2)
library(tseries)
```

Once we read the data, we shall convert and store them to a time series data Object in R which will helps us to use multiple R functions suitable for analysis and visualization. We shall use ts() function in R which would help us to store as a Time series Object for given period using Start and End parameters with an interval of 12 months a year.

```
Fancysales <- ts(fancy[,1], start=c(1987,1), end=c(1993,11), frequency=12)
```

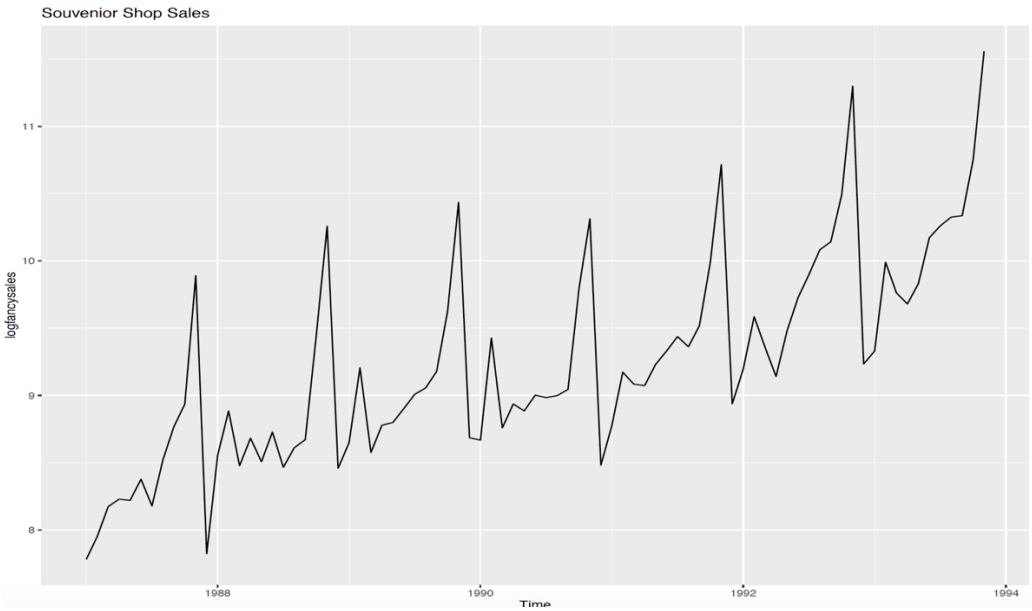
Visualize the data for better understanding:

Since we read the data and convert it into to a time series object, lets plot the data using plot.ts() function in R



Based on above plot we could determine this time series data can be explained with additive model because the size of the seasonality and random fluctuation increase with the level of time series data points. Hence we may have to convert the original data into a natural logarithmic data for an additive model.

```
logfancysales<-log(Fancysales)
plot.ts(logfancysales)
```

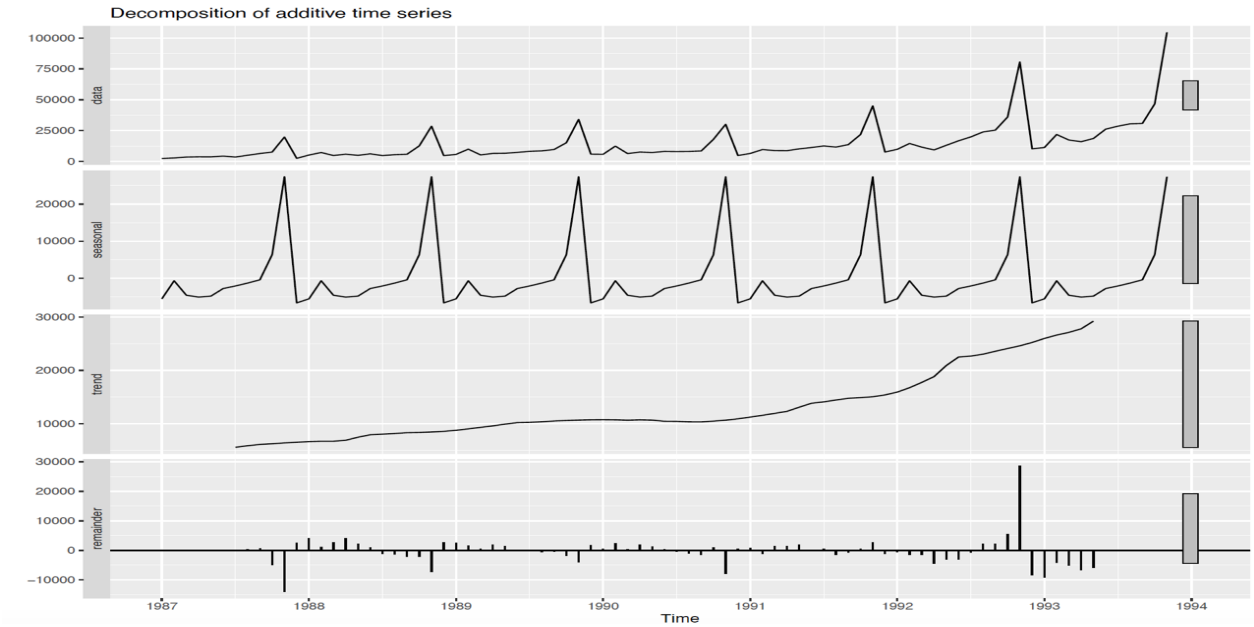


Now the size of seasonal and random fluctuation is constant over time series compared the original data and will not depend on the level of time series. Hence the natural log transformed time series is explained using additive model.

Decomposing the time series data

Let's decompose the Time series data to understand the seasonality, trend and irregularity using decompose() function in R. This returns a list of object as its result where the estimates of all three components are stored in a list objects called Seasonal, trend and remainder as seen below. We can compare these with original data.

```
autoplot(decompose(Fancysales))
```



Check for Stationarity in Times Series:

Let's test for stationarity in means and variances of time series data using adf.test() function in R. Since the p value is > 0.05 lets accept the null hypothesis that data is non-stationary and reject the alternative hypothesis

```
adf.test(Fancysales)
```

```
Augmented Dickey-Fuller Test

data:  Fancysales
Dickey-Fuller = -2.0694, Lag order = 4, p-value = 0.5474
alternative hypothesis: stationary
```

Decomposing data for better MAPE value:

Let's try to decompose the Time series data to have a better MAPE value so we use stl() function on train data.

```
Ftrain.stl<-stl(Ftrain[,1],s.window = "p")
```

Let's recheck the accuracy by determining the MAPE value

```
accuracy(fcst.Ftrain.stl,Ftest)
```

	ME	RMSE	MAE	MPE	MAPE
Training set	-4.823734e-05	0.1466281	0.1199308	-0.01914438	1.352216
Test set	8.806653e-02	0.1835912	0.1539033	0.81871065	1.514941

	MASE	ACF1	Theil's U
Training set	0.4165049	-0.03259571	NA
Test set	0.5344870	0.58387412	0.2771195

Now we could see the slighter improvement in MAPE value which is 1.35 now compared to 1.51 earlier

Splitting data into Test/Train or Dev/holdout:

Let's divide the Time series data into Train data and Test dataset in a proportion on 80:20 ratio correspondingly ie the sales data is split into Train data for a period of January 1987 to July 1992 and the balance data into test data or holdout sample data. We shall use window() function in R to split the data as mentioned below

```
Ftrain <- window(Saleslog, start=c(1987,1), end=c(1992,7))

Ftest<- window(Saleslog, start=c(1992,8), end=c(1993,11))
```

Model Creation using Holt Winder Method:

We need to forecast the time series data using simple exponential smoothing in R. We need Holter Winter model function 'HoltWinters()'

```
Fancysales.hw <- HoltWinters(Ftrain)
Fancysales.hw

Holt-Winters exponential smoothing with trend and additive seasonal component.

Smoothing parameters:
 alpha: 0.3016185
 beta : 0
 gamma: 0.8098382
```

The output of HoltWinters() returns an alpha of 0.30 which is not close to zero and it tells that the forecasts are based on faraway observations.

Predicting the values for test dataset using HW model:

We shall predict the values of trained HoltWinters model in holdout/Test sample dataset using the forecast() function in R for next 5 years or 60 months period

Fancysales.f1 <- forecast(Fancysales.hw , h = 16)

		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Aug	1992	9.662701	9.435789	9.889612	9.315669	10.009732
Sep	1992	9.795297	9.558289	10.032306	9.432824	10.157771
Oct	1992	10.342185	10.095492	10.588877	9.964901	10.719468
Nov	1992	11.052065	10.796055	11.308076	10.660531	11.4436
Dec	1992	9.270967	9.005966	9.535968	8.865683	9.676251
Jan	1993	9.527255	9.253558	9.800951	9.108672	9.945837
Feb	1993	9.976941	9.694817	10.259065	9.54547	10.408412
Mar	1993	9.754908	9.464602	10.045215	9.310922	10.198894
Apr	1993	9.658002	9.359736	9.956267	9.201844	10.114159
May	1993	9.934257	9.62824	10.240273	9.466244	10.402269
Jun	1993	10.103316	9.789739	10.416893	9.623742	10.582891
Jul	1993	10.187207	9.866249	10.508166	9.696343	10.678071
Aug	1993	9.987029	9.610551	10.363507	9.411256	10.562802
Sep	1993	10.119626	9.736977	10.502274	9.534416	10.704836
Oct	1993	10.666513	10.277792	11.055234	10.072016	11.26101
Nov	1993	11.376394	10.981694	11.771094	10.772752	11.980035

Validating against the actual values using MAPE:

> accuracy(Fancysales.f1,Ftest)

	ME	RMSE	MAE	MPE	MAPE
Training set	-0.009702272	0.1757113	0.1376660.	-0.1194736	1.517937
Test set	0.114488064	0.2005902	0.1566735	1.0743067	1.519923

	MASE	ACF1	Theil's U
Training set	0.4780968	-0.02951184	NA
Test set	0.5441075	0.46916557	0.2866784

From the above result we get a MAPE (Mean Absolute Percentage Error) value of 1.51 which is used to check the accuracy of forecast. Lower MAPE value means lesser error in forecast on prediction against the actuals.

Forecasting decomposed Train data on Forecasting Period of 5 years:

We shall forecast the decomposed Train data on test data using Exponential Smoothening Method for next 60 months

```
fcst.Ftrain.stl <- forecast(Ftrain.stl, method="ets", h=60)

fcst.Ftrain.stl
```

		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Aug	1992	9.742462	9.548677	9.936247	9.446094	10.038831
Sep	1992	9.870987	9.667132	10.074842	9.559218	10.182756
Oct	1992	10.393234	10.179779	10.60669	10.066782	10.719687
Nov	1992	11.159823	10.937174	11.382471	10.819311	11.500334
Dec	1992	9.31576	9.084278	9.547242	8.961739	9.669781
Jan	1993	9.597282	9.357286	9.837277	9.23024	9.964323
Feb	1993	10.03095	9.782727	10.279172	9.651326	10.410573
Mar	1993	9.730508	9.474319	9.986698	9.3387	10.122317
Apr	1993	9.800603	9.536682	10.064524	9.39697	10.204236

May	1993	9.848197	9.576759	10.119634	9.433069	10.263324
Jun	1993	10.00683	9.728075	10.285585	9.580511	10.433149
Jul	1993	9.992647	9.706757	10.278537	9.555416	10.429878
Aug	1993	10.037867	9.745011	10.330722	9.589983	10.485751
Sep	1993	10.166392	9.866728	10.466055	9.708096	10.624687
Oct	1993	10.688639	10.382315	10.994962	10.220158	11.15712
Nov	1993	11.455227	11.142381	11.768073	10.976771	11.933684
Dec	1993	9.611165	9.291926	9.930404	9.122931	10.099399
Jan	1994	9.892686	9.567176	10.218197	9.394861	10.390511
Feb	1994	10.326354	9.994687	10.658021	9.819113	10.833595
Mar	1994	10.025913	9.688198	10.363628	9.509422	10.542403
Apr	1994	10.096007	9.752347	10.439667	9.570425	10.62159
May	1994	10.143601	9.794093	10.493109	9.609075	10.678127
Jun	1994	10.302235	9.946972	10.657497	9.758908	10.845562
Jul	1994	10.288051	9.927122	10.64898	9.736058	10.840044
Aug	1994	10.333271	9.96676	10.699782	9.772741	10.893802
Sep	1994	10.461796	10.089783	10.833809	9.892851	11.030741
Oct	1994	10.984043	10.606605	11.361481	10.406802	11.561285
Nov	1994	11.750632	11.367843	12.133421	11.165206	12.336057
Dec	1994	9.906569	9.518499	10.294639	9.313068	10.50007
Jan	1995	10.188091	9.794808	10.581373	9.586617	10.789564
Feb	1995	10.621758	10.223328	11.020189	10.012412	11.231105
Mar	1995	10.321317	9.917802	10.724833	9.704194	10.938441
Apr	1995	10.391412	9.982871	10.799952	9.766603	11.01622
May	1995	10.439005	10.025498	10.852513	9.806601	11.07141
Jun	1995	10.597639	10.179221	11.016057	9.957724	11.237554

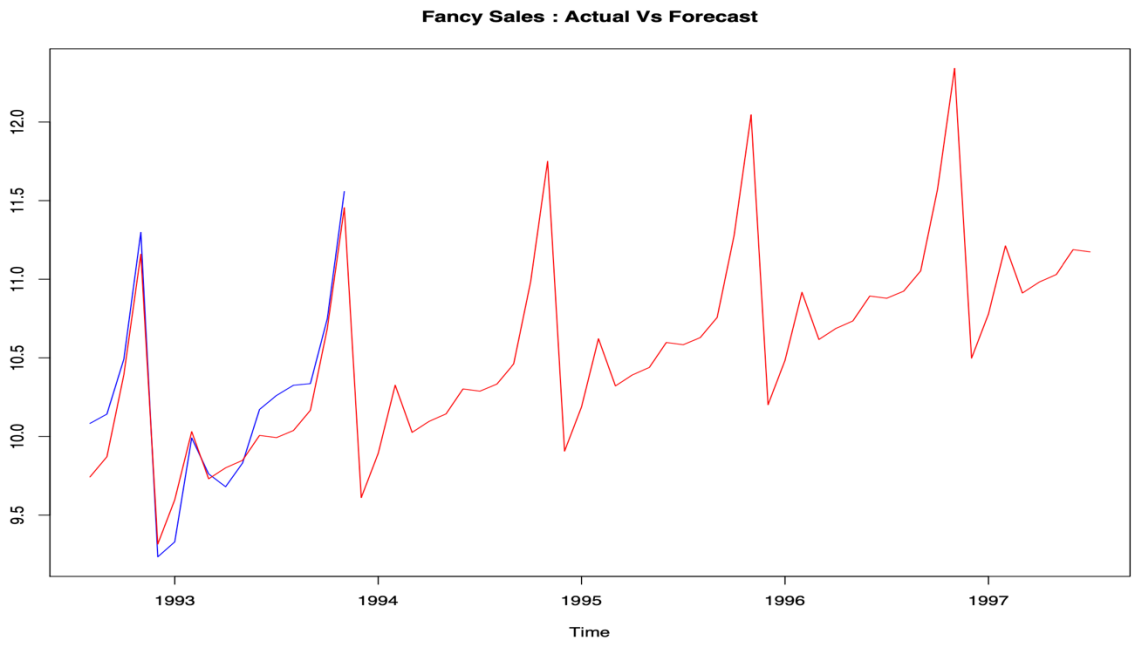
Jul	1995	10.583456	10.160181	11.006731	9.936112	11.230799
Aug	1995	10.628676	10.200596	11.056755	9.973984	11.283367
Sep	1995	10.7572	10.324367	11.190034	10.095238	11.419163
Oct	1995	11.279448	10.841909	11.716987	10.610289	11.948606
Nov	1995	12.046036	11.603839	12.488233	11.369754	12.722318
Dec	1995	10.201973	9.755164	10.648783	9.518637	10.88531
Jan	1996	10.483495	10.032118	10.934872	9.793173	11.173817
Feb	1996	10.917163	10.461261	11.373065	10.219921	11.614405
Mar	1996	10.616722	10.156336	11.077107	9.912623	11.32082
Apr	1996	10.686816	10.221988	11.151644	9.975923	11.397709
May	1996	10.73441	10.265179	11.203641	10.016783	11.452037
Jun	1996	10.893043	10.419448	11.366639	10.168741	11.617346
Jul	1996	10.87886	10.400937	11.356783	10.147939	11.609781
Aug	1996	10.92408	10.441866	11.406295	10.186597	11.661564
Sep	1996	11.052605	10.566134	11.539075	10.308612	11.796597
Oct	1996	11.574852	11.08416	12.065544	10.824403	12.325301
Nov	1996	12.34144	11.84656	12.836321	11.584586	13.098295
Dec	1996	10.497378	9.998342	10.996414	9.734168	11.260587
Jan	1997	10.778899	10.27574	11.282059	10.009383	11.548415
Feb	1997	11.212567	10.705315	11.719819	10.436792	11.988342
Mar	1997	10.912126	10.400812	11.423441	10.130139	11.694114
Apr	1997	10.982221	10.466874	11.497568	10.194066	11.770376
May	1997	11.029814	10.510464	11.549165	10.235536	11.824093
Jun	1997	11.188448	10.665122	11.711774	10.38809	11.988806
Jul	1997	11.174265	10.646991	11.701539	10.367869	11.980661

Predicting the values of forecast for next 5 years:

We shall combine the train data with the test data period with can function called cbind() in R and plot the timeseries data of actual vs forecast.

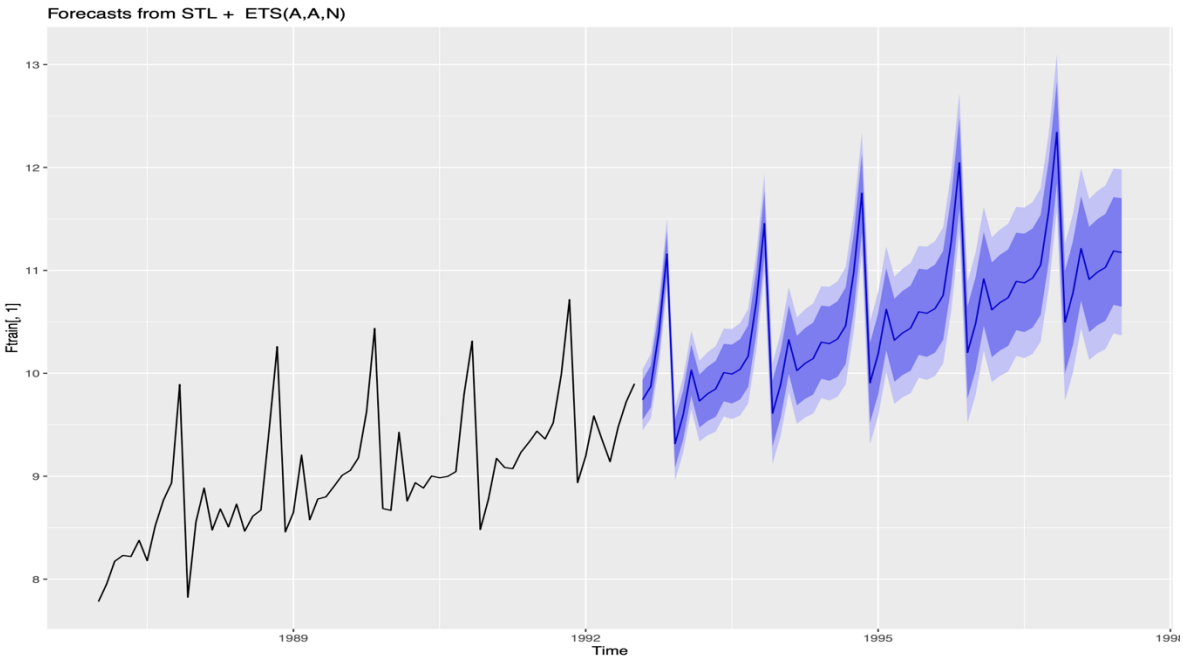
```
vec<-cbind(Ftest,fcst.Ftrain.stl$mean)

ts.plot(vec,col=c("blue","red"), main="Fancy Sales : Actual Vs Forecast")
```



Red coloured graph line represent the forecast data and Blue represent the actual data which aligns most of the periods except for few.

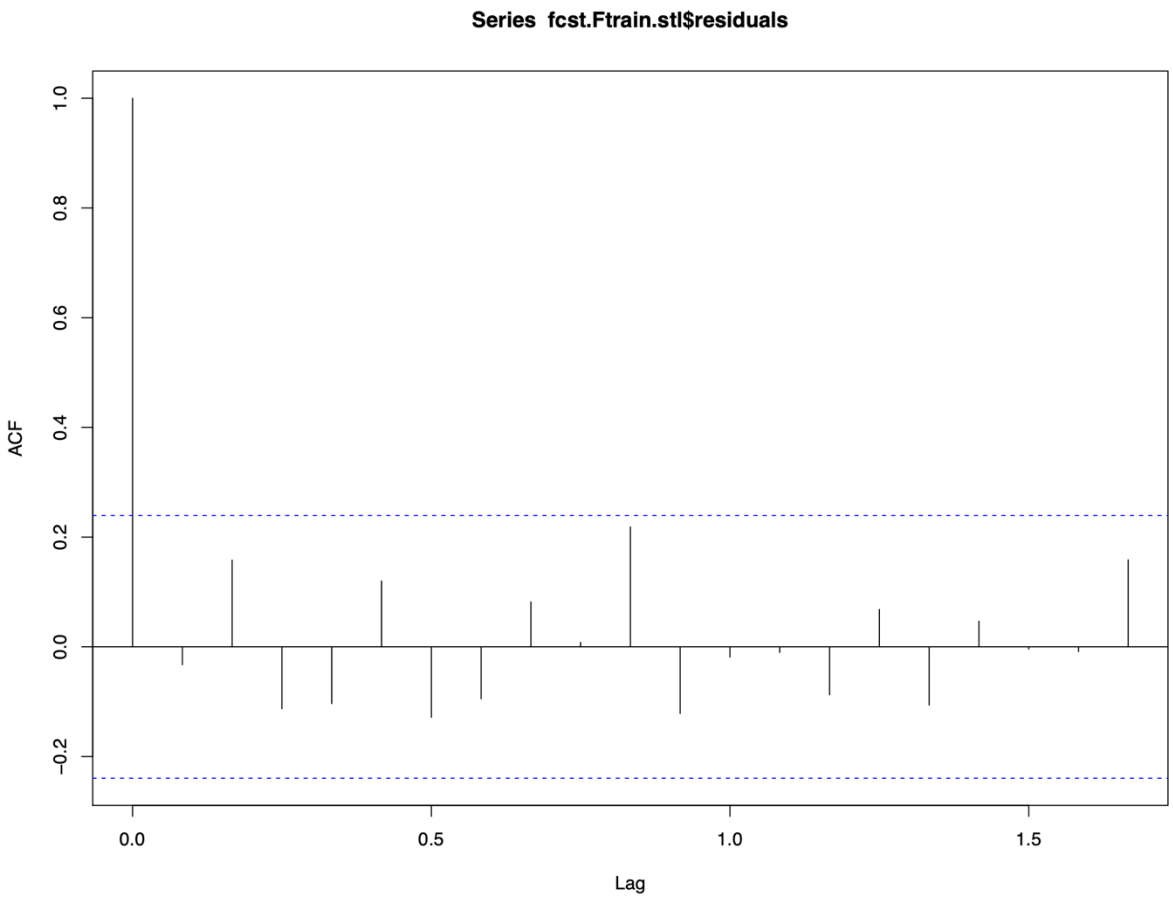
```
autoplot(fcst.Ftrain.stl,h=60)
```



The above function gives you the forecast for a year with 80% prediction interval for forecast in darker shades and a 95% prediction interval for the forecast on lighter shades.

Correlation check:

The 'forecast errors' are measured as the observed value minus the predicted value for each point. One measure for accuracy of predict model is the SSE(Sum of Square Errors) for the in sample/Train data forecast error. The in-sample errors are stored in an element called 'residuals' or 'remainders' which was returned by a function fcst.Ftrain.stl. If there is a correlation between forecast errors for successive predictions it is likely that simple smoothening forecasts could be improved by another forecasting technique. We can calculate a correlogram of the forecast errors using the acf() function in R. To specify the maximum lag that we want to look at, we use the 'lag.max' parameter in acf().In lag 10 the correlation is close to significant bound. We can carry out Ljung-Box test to check the correlation between lag 1 - 20.



Box-Ljung test

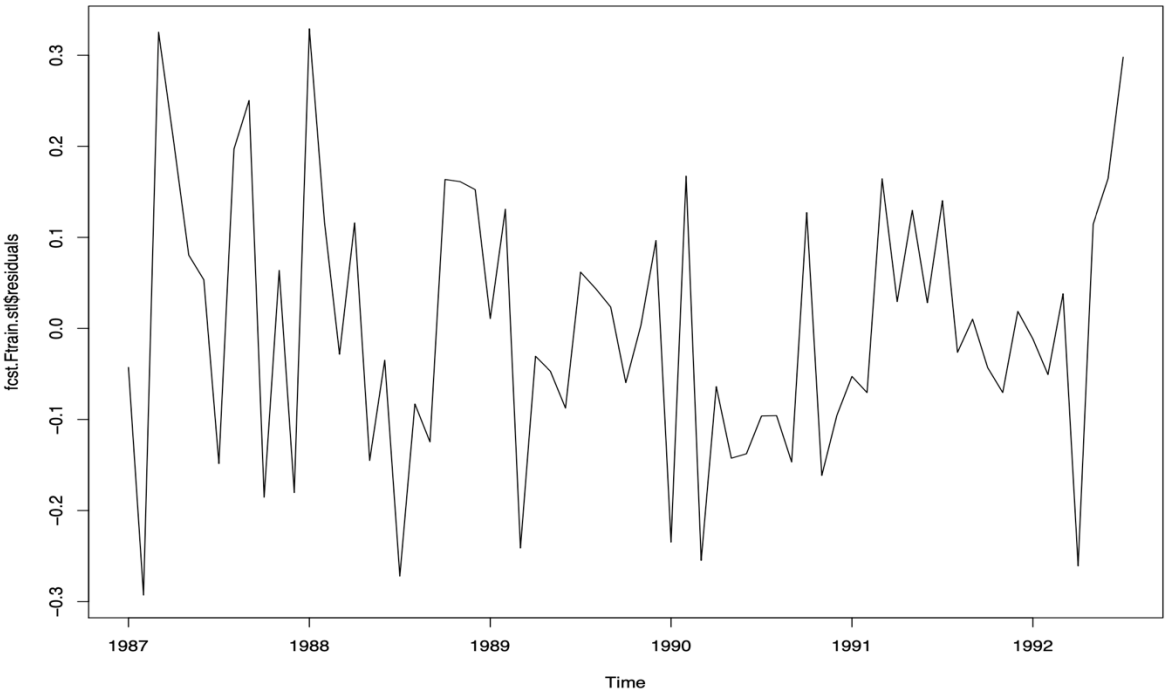
data: fcst.Ftrain.stl$residuals

X-squared = 17.035, df = 20, p-value = 0.6507

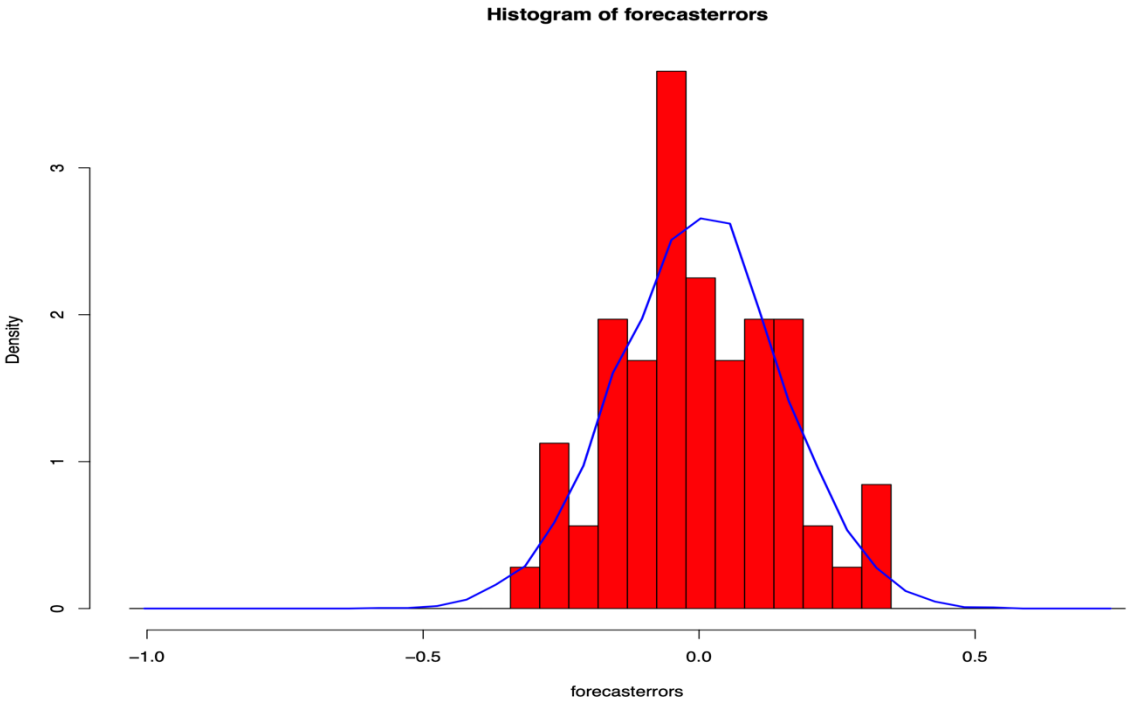
As the P value is greater than 0.05 and test statistics is 17.03 which means there is a little evidence of non-zero auto correlation between lad 1 -20.

In order to confirm the predictive model could not be improved further let's check the forecast errors are normally distributed with mean zero and constant variance. we can plot and find the in-sample forecast errors

plot.ts(fcst.Ftrain.stl$residuals)



We could see from above plot that there is a higher variance over time. We can also check whether the forecast errors are normally distributed. We can plot histogram to check if the curve has mean zero and same standard deviations ad forecast errors.



ARIMA model

With the same set of in-sample and holdout sample in the ratio of 80:20 we shall build a ARIMA model in addition to the HoltWinters Model.Auto Regressive Integrated Moving Average (ARIMA) models include an explicit statistical model for the irregular component of a time series data, that allows for non-zero autocorrelation in the irregular component.

```
Fancysales.ar <- Arima(Ftrain)
```

Predicting the values of trained ARIMA model in Holdout Samples:

We shall predict the ARIMA model trained on in-sample and test it in holdout sample to predict the accuracy of the model

```
Fancysales.f2 <- forecast(Fancysales.ar , h = 16)
```

Validating against the actual values using MAPE:

Similar to HW model, we shall also check the accuracy of forecast of ARIMA model by determining the MAPE value

```
accuracy(Fancysales.f2,Ftest)
```

	ME	RMSE	MAE	MPE	MAPE
Training set	-1.073742e-15	0.6032955	0.466278	-0.4388217	5.153139
Test set	1.190726e+00	1.3349532	1.190726	11.3693404	11.369340

	MASE	ACF1	Theil's U
Training set	1.619326	0.4234299	NA
Test set	4.135243	0.2817817	2.162683

We could see the MAPE value is 5.15 which is good. We can decompose the timeseries data for better MAPE value

```
Ftrain.stl1<-stl(Ftrain[,1],s.window = "p")
```

Decomposing data for better MAPE value:

Let's try to decompose the Time series data to have a better MAPE value so we use stl() function on train data. The MAPE value has come down to 1.30 after decomposition of data

	ME	RMSE	MAE	MPE	MAPE
Training set	-0.001448914.	0.1464175	0.1160561	-0.03297644	1.307567
Test set	-0.079922818	0.1830573	0.1462688	-0.82918067	1.481701

	MASE	ACF1	Theil's U
Training set	0.4030485	-0.03830384	NA
Test set	0.5079733	0.55570518	0.303695

We shall forecast the trained model using ARIMA method for test period

		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Aug	1992	9.842764	9.650775	10.034753	9.549142	10.136386
Sep	1992	10.035202	9.826348	10.244056	9.715787	10.354617
Oct	1992	10.527442	10.273199	10.781684	10.138611	10.916272
Nov	1992	11.317721	11.041059	11.594383	10.894603	11.740839

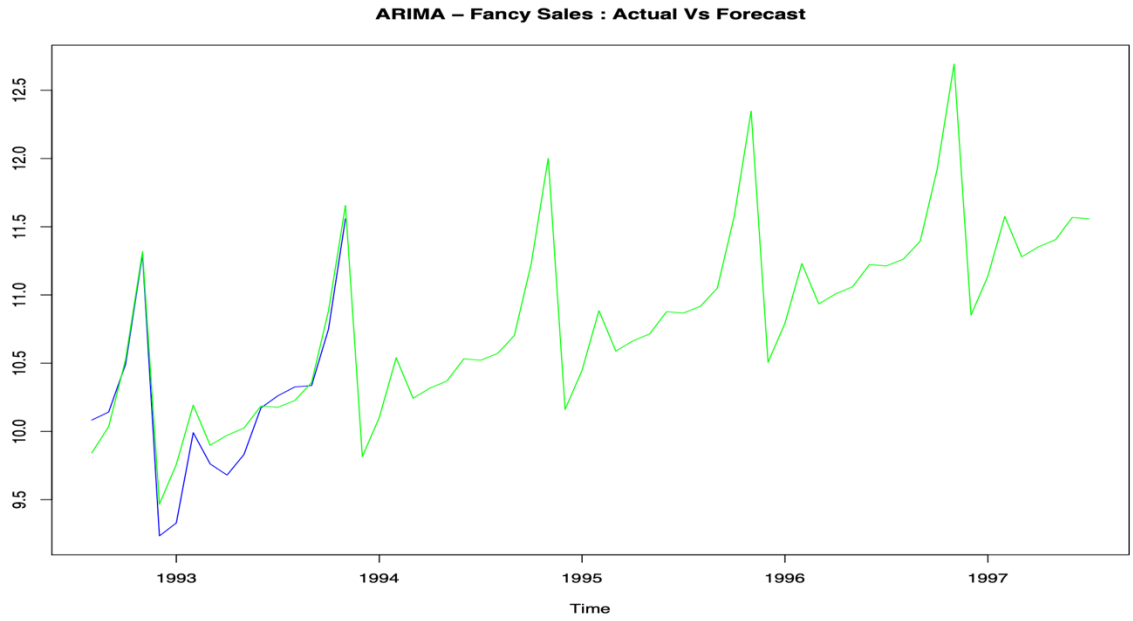
Dec	1992	9.466648	9.16113	9.772166	8.999398	9.933897
Jan	1993	9.758712	9.43124	10.086184	9.257886	10.259537
Feb	1993	10.192886	9.842514	10.543259	9.657038	10.728734
Mar	1993	9.89869	9.528092	10.269288	9.33191	10.46547
Apr	1993	9.971748	9.581289	10.362208	9.374592	10.568905
May	1993	10.024182	9.615197	10.433167	9.398694	10.64967
Jun	1993	10.186583	9.759672	10.613494	9.533678	10.839487
Jul	1993	10.17678	9.732778	10.620781	9.497738	10.855822
Aug	1993	10.22603	9.765509	10.68655	9.521724	10.930335
Sep	1993	10.358785	9.882352	10.835217	9.630144	11.087425
Oct	1993	10.885148	10.393298	11.376997	10.132929	11.637366
Nov	1993	11.655917	11.149131	12.162704	10.880854	12.43098
Dec	1993	9.815998	9.294697	10.3373	9.018736	10.613261
Jan	1994	10.101685	9.566265	10.637106	9.28283	10.92054
Feb	1994	10.539506	9.990328	11.088684	9.699611	11.379401
Mar	1994	10.243225	9.680626	10.805823	9.382805	11.103645
Apr	1994	10.317475	9.741769	10.893182	9.437008	11.197942
May	1994	10.369227	9.780705	10.957749	9.46916	11.269294
Jun	1994	10.532017	9.930952	11.133083	9.612767	11.451268
Jul	1994	10.521992	9.90864	11.135344	9.583952	11.460032
Aug	1994	10.571369	9.945972	11.196766	9.614907	11.527831
Sep	1994	10.704051	10.066837	11.341265	9.729516	11.678586
Oct	1994	11.230456	10.58164	11.879272	10.238177	12.222735
Nov	1994	12.001202	11.340987	12.661416	10.991491	13.010913
Dec	1994	10.161297	9.489877	10.832716	9.134449	11.188144
Jan	1995	10.446975	9.764535	11.129416	9.403273	11.490678

Feb	1995	10.884801	10.191514	11.578087	9.824511	11.94509
Mar	1995	10.588517	9.884552	11.292482	9.511896	11.665138
Apr	1995	10.662769	9.948285	11.377253	9.57006	11.755478
May	1995	10.71452	9.989669	11.43937	9.605957	11.823083
Jun	1995	10.877311	10.14224	11.612382	9.753117	12.001504
Jul	1995	10.867285	10.122134	11.612436	9.727675	12.006895
Aug	1995	10.916662	10.161566	11.671759	9.761842	12.071483
Sep	1995	11.049344	10.284432	11.814257	9.879512	12.219177
Oct	1995	11.575749	10.801145	12.350354	10.391094	12.760404
Nov	1995	12.346495	11.562318	13.130671	11.1472	13.545789
Dec	1995	10.50659	9.712956	11.300223	9.292832	11.720347
Jan	1996	10.792269	9.98929	11.595247	9.564219	12.020318
Feb	1996	11.230094	10.417878	12.04231	9.987916	12.472271
Mar	1996	10.93381	10.11246	11.75516	9.677664	12.189957
Apr	1996	11.008062	10.177678	11.838445	9.7381	12.278024
May	1996	11.059813	10.220493	11.899133	9.776184	12.343442
Jun	1996	11.222604	10.374442	12.070766	9.925452	12.519755
Jul	1996	11.212578	10.355666	12.069491	9.902043	12.523113
Aug	1996	11.261955	10.396381	12.12753	9.938173	12.585738
Sep	1996	11.394638	10.520486	12.268789	10.057738	12.731537
Oct	1996	11.921042	11.038398	12.803687	10.571154	13.270931
Nov	1996	12.691788	11.800731	13.582845	11.329034	14.054542
Dec	1996	10.851883	9.952492	11.751273	9.476384	12.227382
Jan	1997	11.137562	10.229914	12.045209	9.749435	12.525689
Feb	1997	11.575387	10.659557	12.491217	10.174746	12.976028
Mar	1997	11.279103	10.355163	12.203044	9.866059	12.692148

Apr	1997	11.353355	10.421375	12.285335	9.928015	12.778695
May	1997	11.405106	10.465156	12.345057	9.967576	12.842637
Jun	1997	11.567897	10.620043	12.515751	10.118279	13.017515
Jul	1997	11.557871	10.602178	12.513564	10.096265	13.019477

Predicting the values of forecast for next 5 years:

We shall combine the train data with the test data period with can function called cbind() in R and plot the timeseries data of actual vs forecast.



Green coloured graph line represent the forecast data and Blue represent the actual data which aligns most of the periods except for few.

Forecast using ARIMA model with drift for next 5 years:

Auto.arima() function in R helps us to find the best model

tsdisplay(logfancysales)

auto.arima(logfancysales,trace = TRUE, stepwise = F,approximation = F)

Best model: ARIMA(2,0,0)(0,1,1)[12] with drift

Series: logfancysales

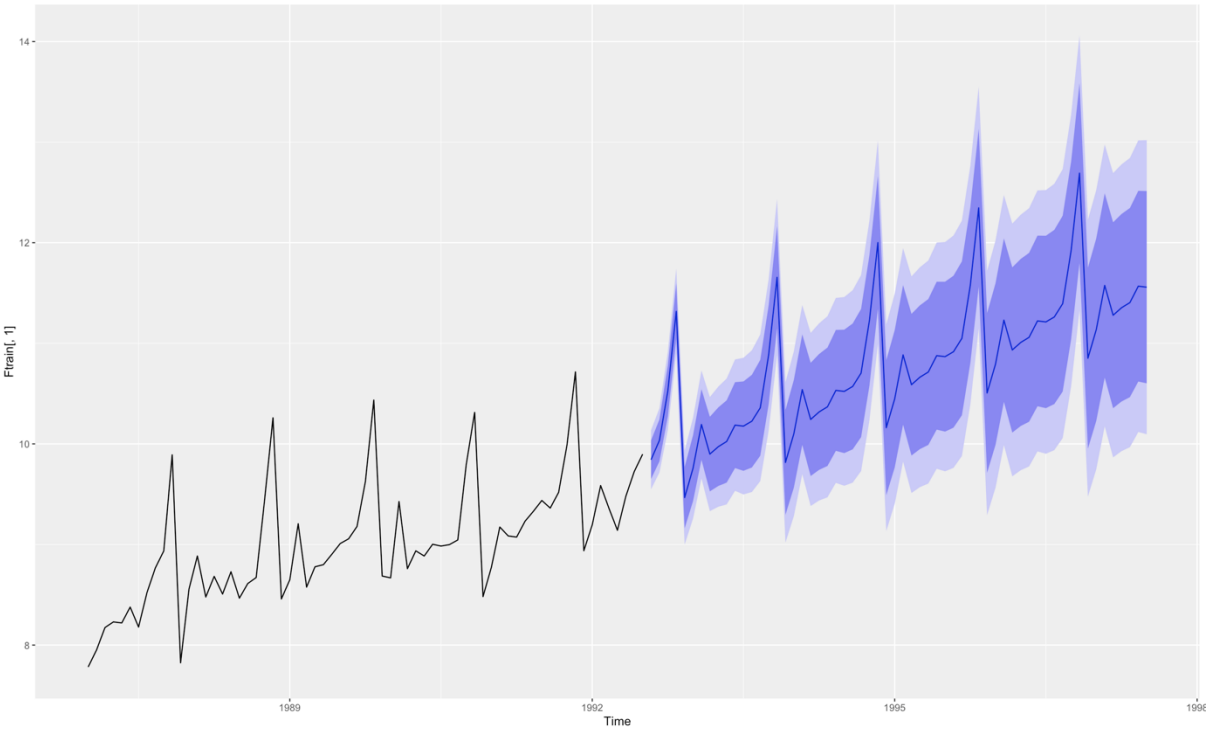
ARIMA(2,0,0)(0,1,1)[12] with drift

Coefficients:

	ar1	ar2	sma1	drift
	0.3478	0.3723	-0.4813	0.0243
s.e.	0.1097	0.1181	0.1837	0.0036

sigma^2 estimated as 0.03009: log likelihood=23.83

We could see that ARIMA(2,0,0) with drift is the best fit model for this time series data.



Model comparison:

Let compare the results of Holtwinter and ARIMA model with forecast function for better understanding