|  |
| --- |
|  |
| ARIMA TIME SERIES FORECASTING |
| Group 9   * Ananya Das M * [Biren Indrajit Sampat](https://olympus.greatlearning.in/groups/8970/users/6065) * Jaya Garg * [Karan Kumar Pathak](https://olympus.greatlearning.in/groups/8970/users/4788) * [Sai Venkatesh](https://olympus.greatlearning.in/groups/8970/users/6272) |
| |  |  |  | | --- | --- | --- | | Great Learning, Bangalore | 3/13/18 | PGP-BABI | |  |  |  | |

# Problem Statement

**Quarterly beer sales data has been provided in the beer.csv files.**

Part A)

Using the Winter-Holts methods and model the data and predict for the next 2 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 2 years. Your submissions should contain the complete modelling steps with explanations. Include pictures and R-code where applicable.

# WINTER-HOLTS METHOD

This method is a triple exponential smoothing. Using this method first we will smooth the level then the trend and then seasonal factor. After doing so we will get the stationary data,

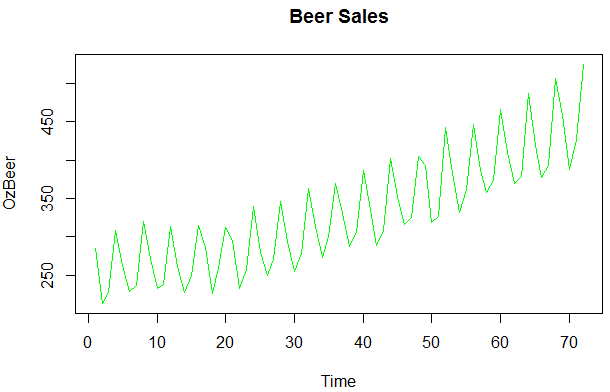
First we will plot the time series for the provided data “beer.csv” to check its properties.

### data= read.csv("beer.csv")

### attach(data)

### beer= ts(beer)

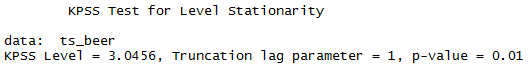
### plot.ts(OzBeer, col="green", main="Beer Sales")



From the above graph we notice that there are level, trend and seasonality in the plot. And we observe that the seasonal span is increasing very slightly with the level of time series. Hence we need to check whether we need to consider multiplicative or additive Holt-Winters Method. This we will do modelling both additive as well as multiplicative Holt-Winters method and then select the model with least AIC value of additive and multiplicative models.

Lets check for stationarity in the data

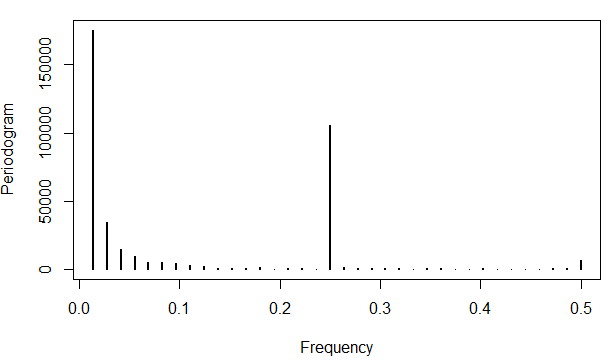
### kpss.test(ts\_beer)



Null hypothesis for the KPSS test states that the time series data is stationary. Since the p value is less than 0.05 the data is not stationary.

To find the time period,

### p=periodogram(OzBeer)



This is the peridogram showing the strongest seasonality signal at 0.0 and 0.25.

### dd=data.frame(freq=p$freq, spec=p$spec)

### order=dd[order(-dd$spec),]

### top2 = head(order,2)

### top2



This will provide the top 2 strongest seasonality signal.

### time= 1/top2$freq

### time



Then on dividing the frequency by 1 we get the seasonal number. Here the 72 is the total number of records provided, hence ignoring it. 4 is the season i.e., the seasonality is quarterly.

Now lets set the frequency,

### ts\_beer=ts(beer,frequency=4)

## Modeling Holt-Winters Multiplicative Model Vs Additive Model

Below graph shows the output of the data when we try to fit the data using multiplicative Vs Additive holt-winters method on top of the actual given data. (Given data is drawn with green color and fitted data in red color).

### plot.ts(OzBeer,col="green",main="Beer Sales")

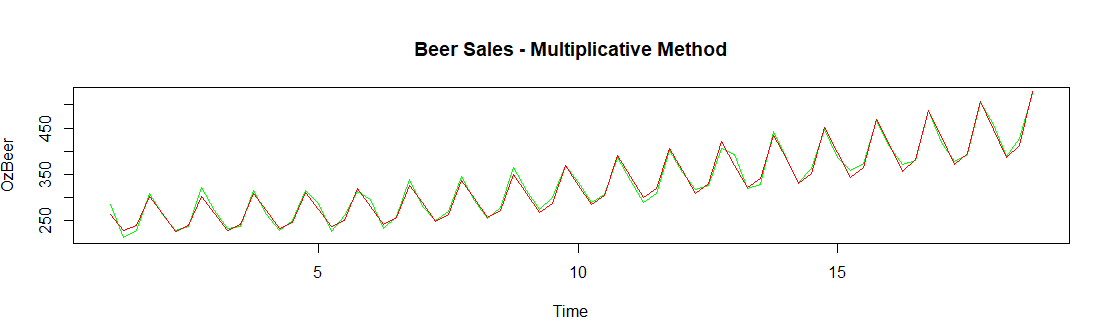
### fit = hw(beer,seasonal="multiplicative")

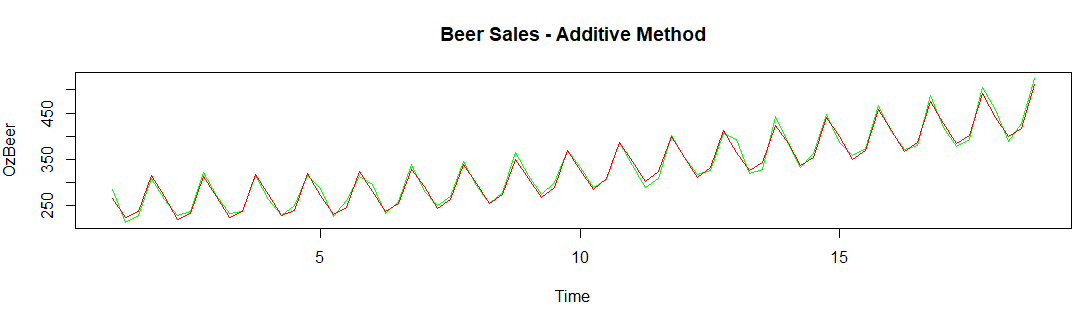
### lines(fitted(fit),col="red")

### plot.ts(OzBeer,col="green",main="Beer Sales")

### fit = hw(beer,seasonal="additive")

### lines(fitted(fit),col="red")



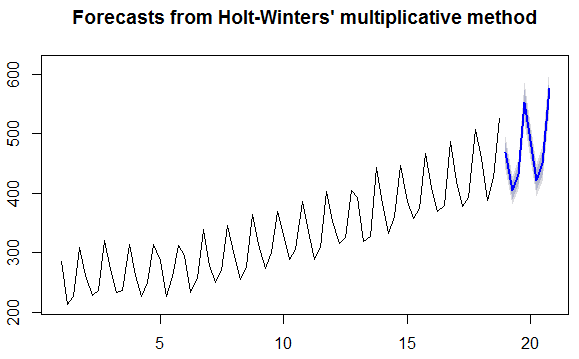


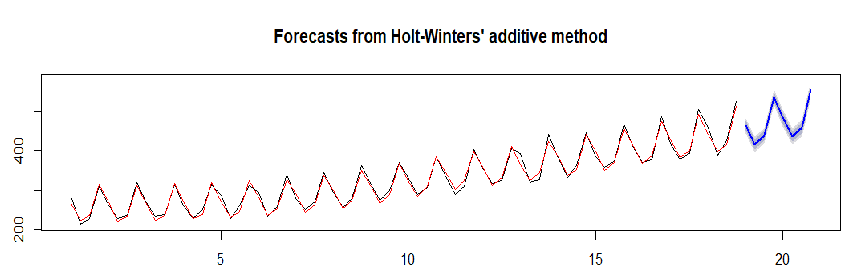
The fitted red line data is given below

On forecasting with Multiplicative Vs Additive method for 2 years, we get the below blue line as the forecast.

### forecast(fit,8)

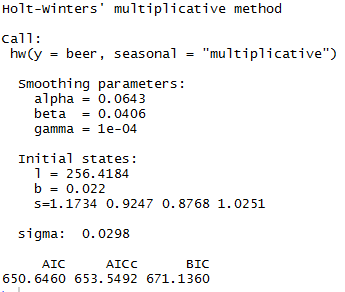
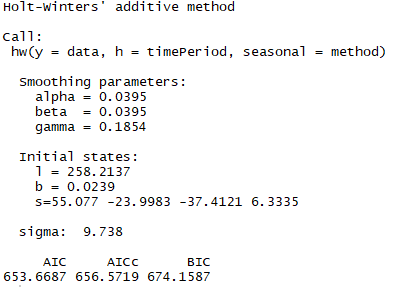
### plot(forecast(fit,8))



****

As seen below figure, the AIC, AICc and BIC values of the multiplicative method is lower than the additive method. So we go with multiplicative method.

### fit$model

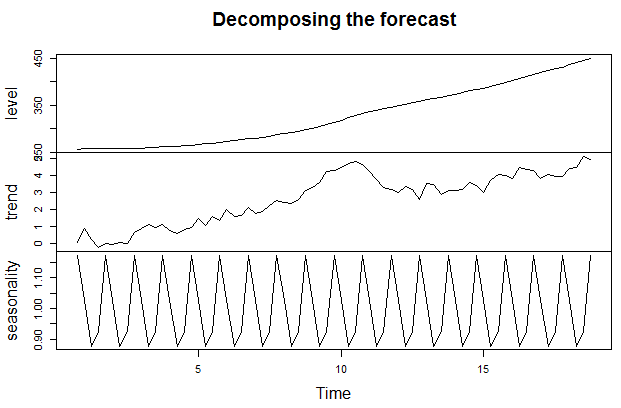
 

From the above output for multiplicative model, we can see the sigma value to be 0.0298, hence the error is square of it which is 0.00089. The level, trend and seasonality from the initial state has been remove. The obtained stationary data with error of 0.00089. AIC, AICc, BIC are the ranks to the model accordingly.

### states = fit$model$states[,1:3]

### colnames(states)=cbind("level", "trend", "seasonality")

### plot(states,cpl="blue", main="Decomposing the forecast")

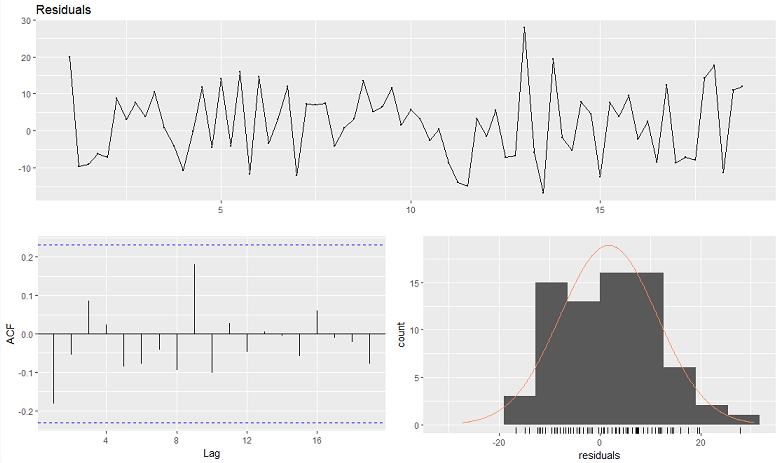


This is the decomposition of the level, trend and seasonality present in the data.

Also the residual plot for the multiplicative model is shown below.

### residuals\_m = fit$residuals

### checkresiduals(residuals(fit))

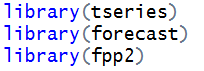


# ARIMA Model:

ARIMA: Auto regressive integrated moving average.

It is based on the assumption that over a period of time the current values are related or correlated with their immediate previous OR n previous values.

# Load Packages



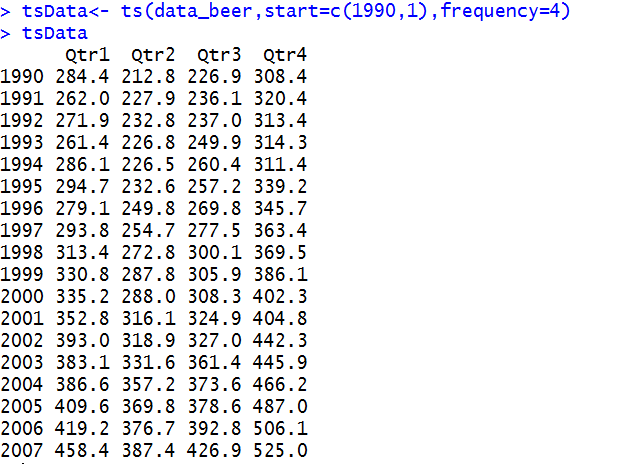
# Get Data



# Exploratory Analysis

The data is a Univariate Time Series data. Now, we want to get a feel for our data in order to decide which models may be appropriate for our forecast. To do this, we will plot our data and diagnose for trend, seasonality, heteroskedasticity, and stationarity.

## Creating time-series data object: As mentioned in the above section we will find the frequency using the periodicity method.



1. Trend: The below two diagrams show that there is a trend component which grows the sales year by year.

## Plot a Time Series and Visualize:



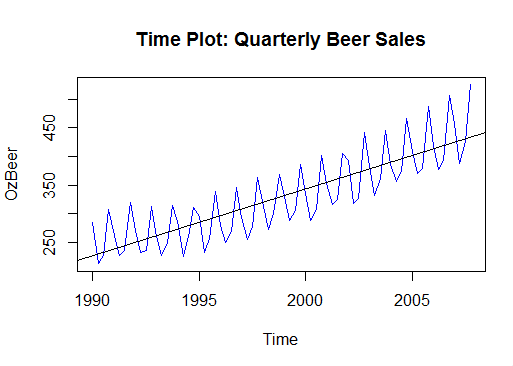


Figure 1



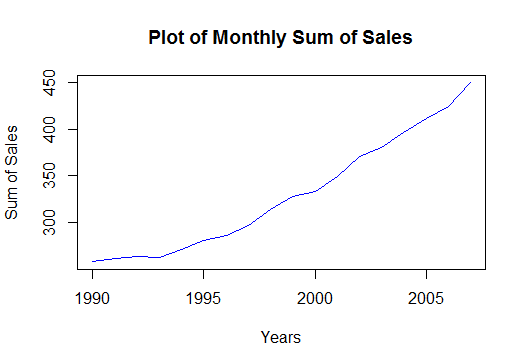
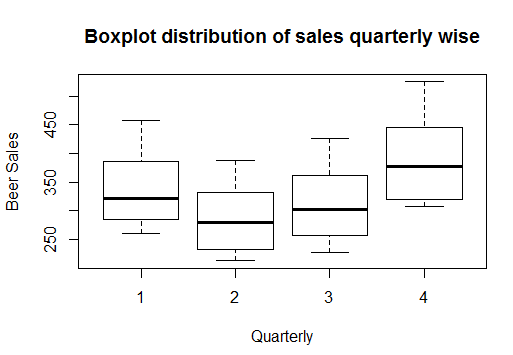


Figure 2

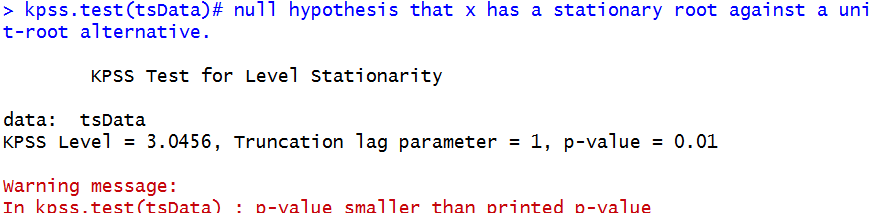
1. Heteroskedasticity: The variance is also increasing year after year from the above diagram(Figure 1). So we will take the log of the series to remove the unequal variances.
2. Seasonality: Below diagram show that there is a seasonal component in the given data.





1. Mean value across quarter is comparatively different and not constant.
2. Stationarity:

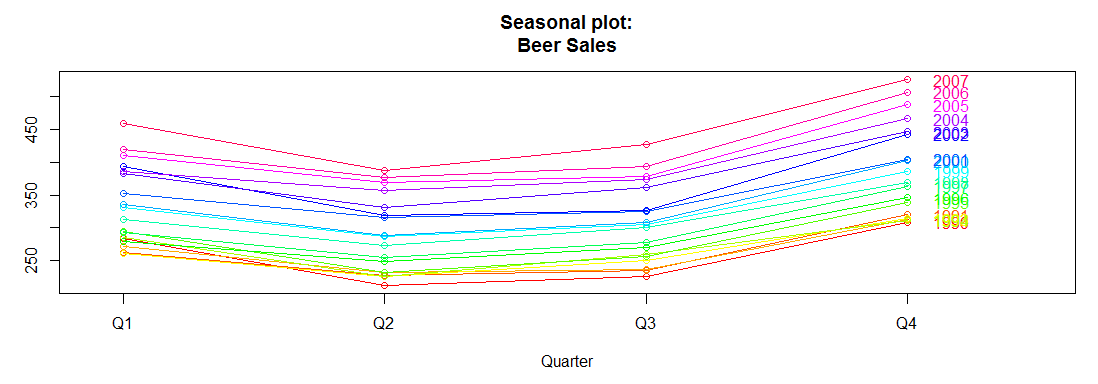
**Test for Stationarity: Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for stationarity,** The null hypothesis states that the time series is stationary. (We will be using *0.05* as our alpha value). Hence, small p-values suggest differencing is required.



The above test validates that the given data is not stationary, hence differencing has to be performed on the logged series to remove trend and seasonality to fit the ARIMA model.

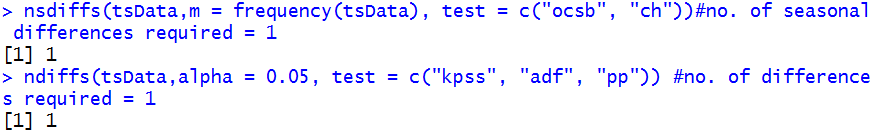
**Test for Seasonality:**





The above diagram confirms the seasonality exists.

### Generating Stationary Timeseries after taking logarithm due to variance in the seasonal data



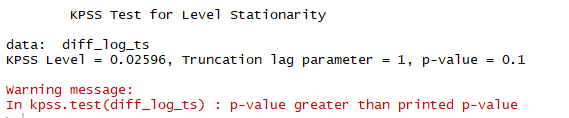
Hence, no. of seasonal differences required = 1 and no. of differences required = 1

### Differencing:

We are differencing once and check for the stationary of data.

diff\_log\_ts = diff(log(ts\_beer))

kpss.test(diff\_log\_ts)

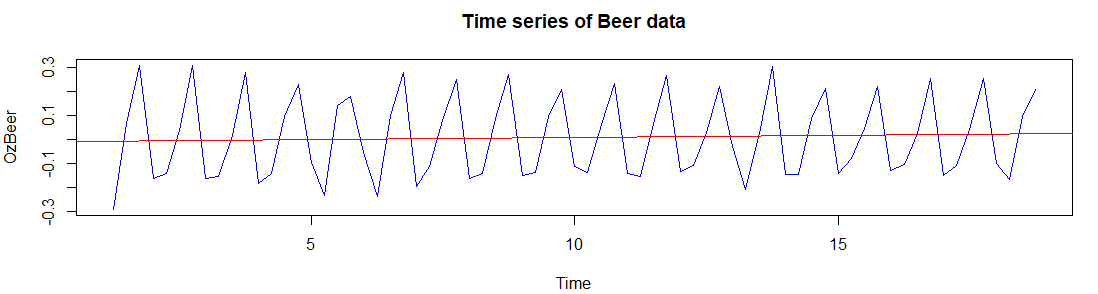


After the first differencing itself, the time series has become stationary as shown by the p-value (0.1). Hence further differencing is not required.

We plot the differenced log data to verify the same.

plot(log\_ts,main="Time series of Beer data", col="Blue")

abline(reg=lm(log\_ts~time(log\_ts)),col="red")



# Model Estimation

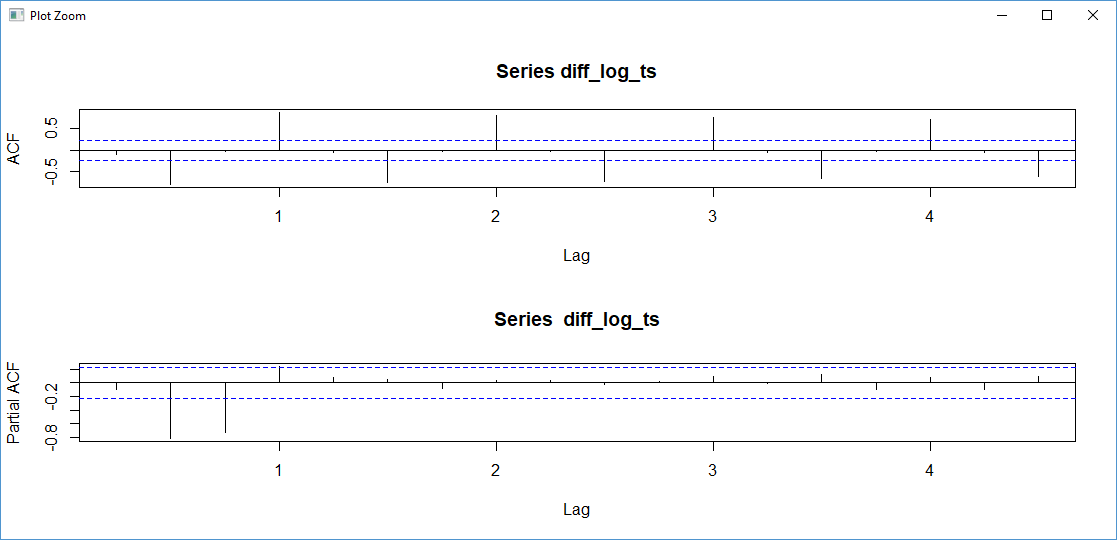
## Diagnosing the ACF and PACF Plots of the differenced Log Time-Series Object

ACF stands for "autocorrelation function" and PACF stands for "partial autocorrelation function." The ACF and PACF diagnosis is employed over a time-series to determine the order in which we are going to create our model using ARIMA modelling. A time series is stationary when its mean, variance, and autocorrelation remain constant over time.

These functions help us understand the correlation component of different data points at different time lags. Lag refers to the time difference between one observation and a previous observation in a dataset.

acf(diff\_log\_ts) # Possible values of q (from ACF Graph) = 2,4,6,8,10

pacf(diff\_log\_ts) # Possible values of p (from PACF Graph) = 2,3,4



ACF plot shows multiple spikes above/below the blue lines and PACF plot shows that lag shuts off after 3 and it is approximated to be 0 beyond that point. Hence from the above ACF graph, the possible values of q are 2,4,6,8,10 and from the above PACF graph, possible values of p are 2, 3 and 4. Now since the possible values of both p and q are non zero, we will try to find the optimal value of p and q by modeling different ARIMA models with different values of p and q. Already we know that the seasonality frequency is 4.

# Arriving at an optimal Model

As mentioned above we will compare the AIC of the different models to arrive at the optimal model. In addition we will have to check if Ljung test for uniform independent distribution of residuals is satisfied and the acf and pacf of residuals are within the limits. The table below compares the different possible models.

modelfit <- function(data,p,d,q,P,D,Q,M){

fit <- arima(log(data),c(p,d,q),seasonal=list(order=c(P,D,Q),period=M))

# Check All the points are within the limits

acf(residuals(fit))

# Check All the points are within the limits

pacf(residuals(fit))

# Test if the residuals are uniformly independently distributed

# Null Hypothesis - the residuals are uniformly independently distributed

# Dont reject the null hypothesis as p-value >0.05

LjungOut <- Box.test(residuals(fit),lag=M,type="Ljung")

cat(sprintf("\"%s\" \"%s\" \"%s\" \"%s\" \n","Model ", "Aic ", "LogLikelihood ", "Ljung P Value"))

cat(sprintf("\"%s\" \"%.2f\" \"%.2f\" \"%s\" \"%.2f\" \n", paste("ARIMA(",p,d,q,P,D,Q,M,")"),round(fit$aic,digits=2), logLik(fit), " ", LjungOut$p.value))

fit

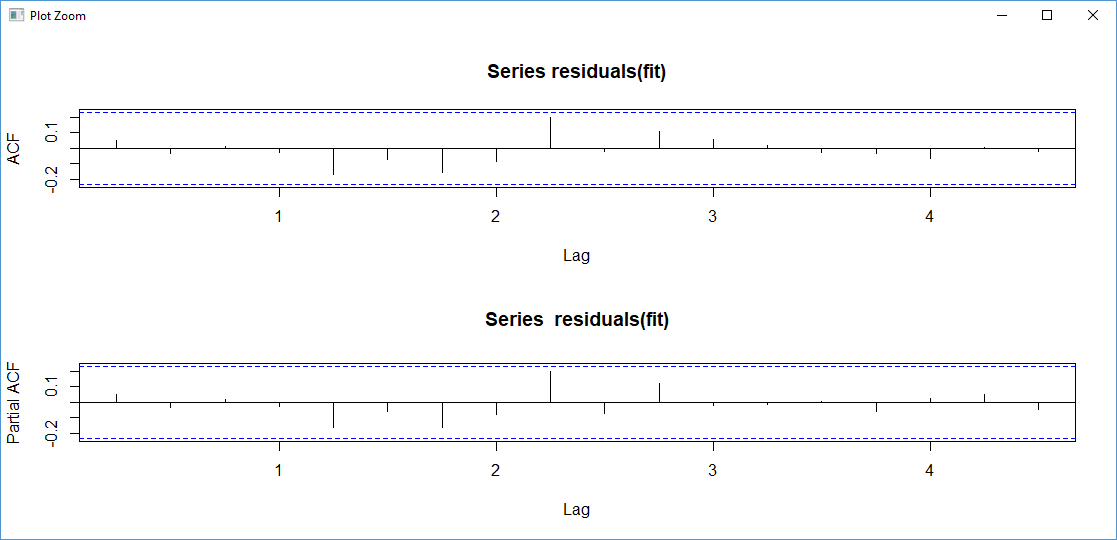
}

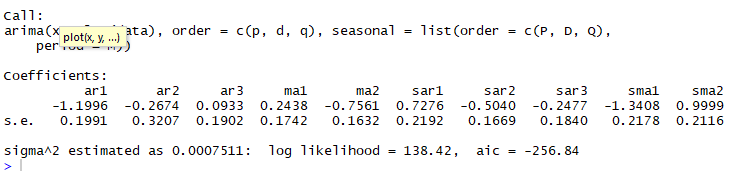
# Check for Model ARIMA (2,2,1)(2,2,1)4

fit = modelfit(ts\_beer,2,2,1,2,2,1,4)

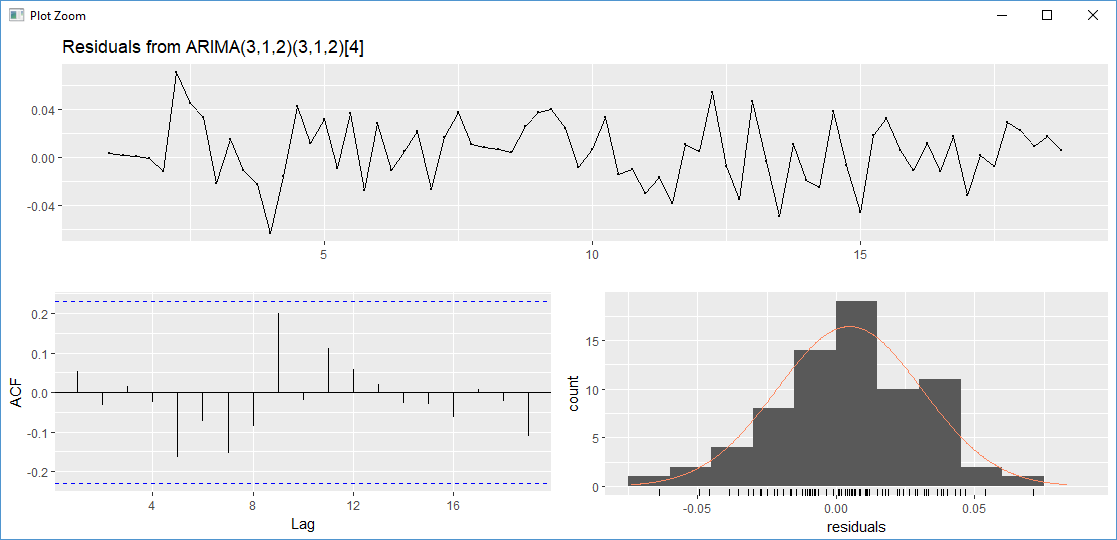
|  |  |  |  |
| --- | --- | --- | --- |
| Model | Ljung | AIC | ACF and PACF Residuals within limits |
| ARIMA(2,1,2)(2,1,2)[4] | 0.95 | -255.02 | Within Limits except one point |
| ARIMA(2,1,4)(2,1,4)[4] | 0.93 | -253.17 | All are within Limits |
| ARIMA(2,1,6)(2,1,6)[4] | 0.92 | -249.8 | All points are within limits |
| ARIMA(2,1,8)(2,1,8)[4] | 0.92 | -249.05 | All points are within limits |
| ARIMA(3,1,2)(3,1,2)[4] | 0.99 | -256.84 | All points within Limits |
| ARIMA(3,1,4)(3,1,4)[4] | 0.91 | -247.79 | All points are with in Limits |
| ARIMA(4,1,2)(4,1,2)[4] | 0.86 | -251.48 | Couple of points are outside limits |

So the optimized model we found is ARIMA(3,1,2)(3,1,2)[4] . The ACF and PACF of the residuals of the selected model ARIMA(3,1,2)(3,1,2)[4] is shown below.





Residuals, ACF and Histogram to test the model:



The above model gives us the coefficients and values of:

1. ma – moving average.
2. sma – seasonal moving average.
3. AIC – Akaike’s Information Criterion in fitting the model we aim to minimize this as much as possible.
4. Sigma^2 - value of the noise component
5. Log-likelihood – this is the log of the likelihood function to determine the coefficients

# Prediction based on the selected Model

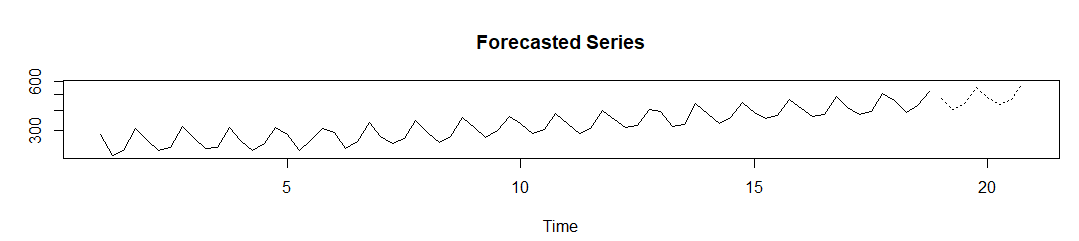
Below is the prediction plot based on the selected model ARIMA(3,1,2)(3,1,2)[4]. The predicted values for the two years are given below

pred <- predict(fit,n.ahead=duration)

ts.plot(data,2.718^pred$pred,log="y",lty=c(1,3),ylabel="Beer Data",main="Forecasted Series")

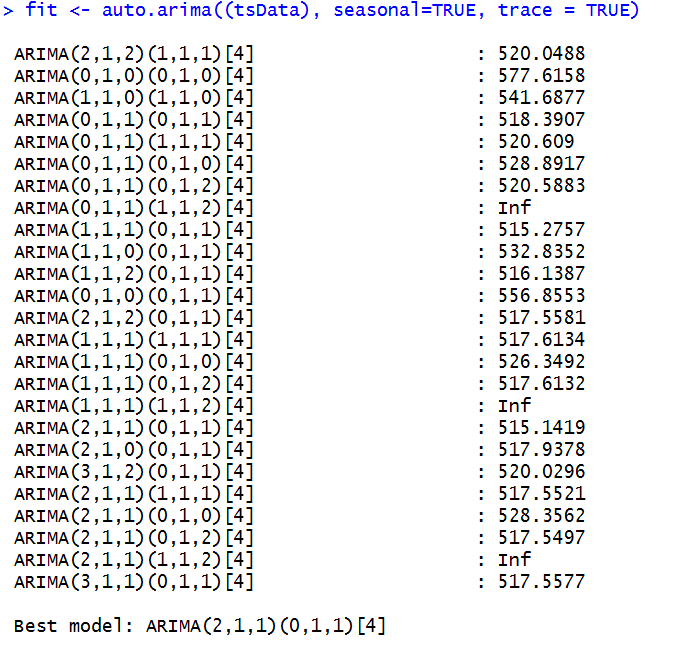
print(2.718^pred$pred)





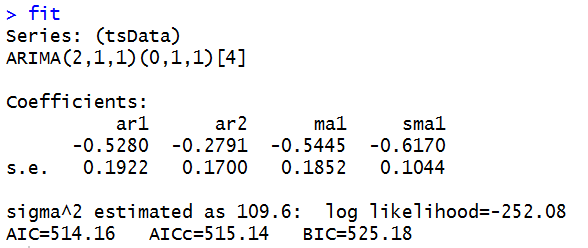
# Alternate Method to find the Optimized Model

The auto.arima() method, found within the forecast package, yields the best model for a time series based on Akaike-Information-Criterion (AIC).

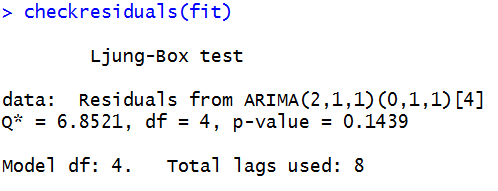


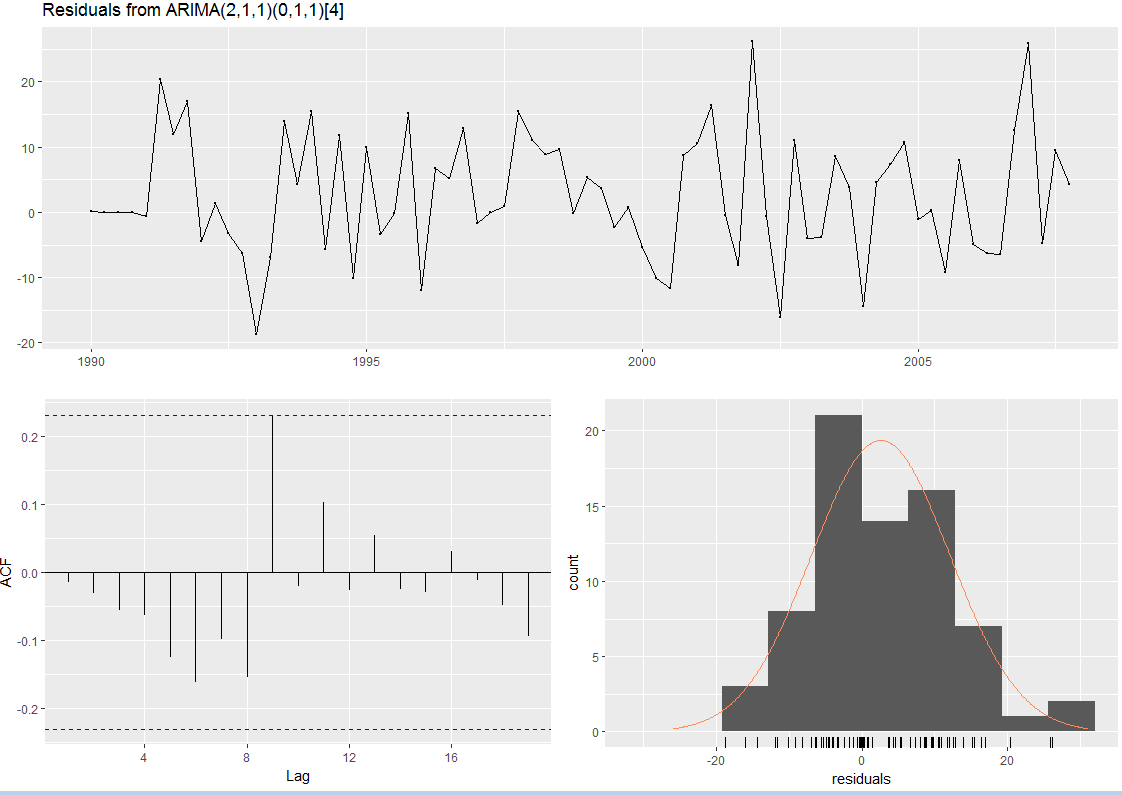
So, with all the above combinations, the best model that comes up is:

ARIMA(2, 1,1)(0,1,1)[4]



Residuals, ACF and Histogram to test the model: ARIMA(2, 1,1)(0,1,1)[4]

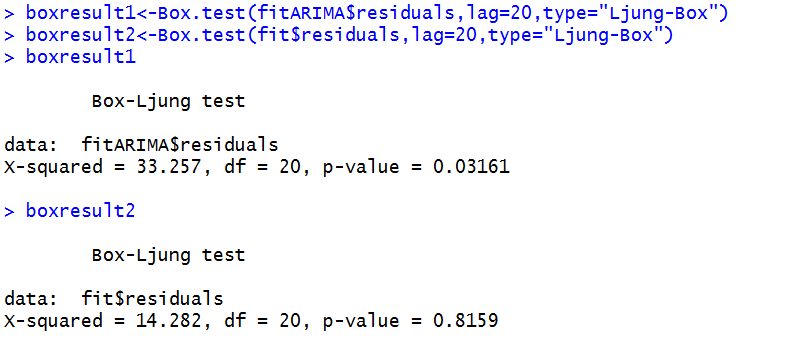




### Residual Diagnostics for both ARIMA models

**Ljung-Box test** is carried out on residuals to see that after fitting the model what remains is actually the residuals.

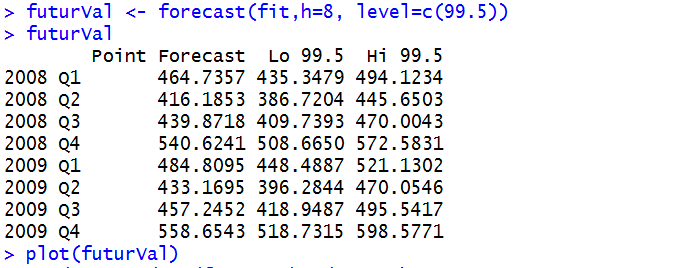
Null Hypothesis=data is independently distributed

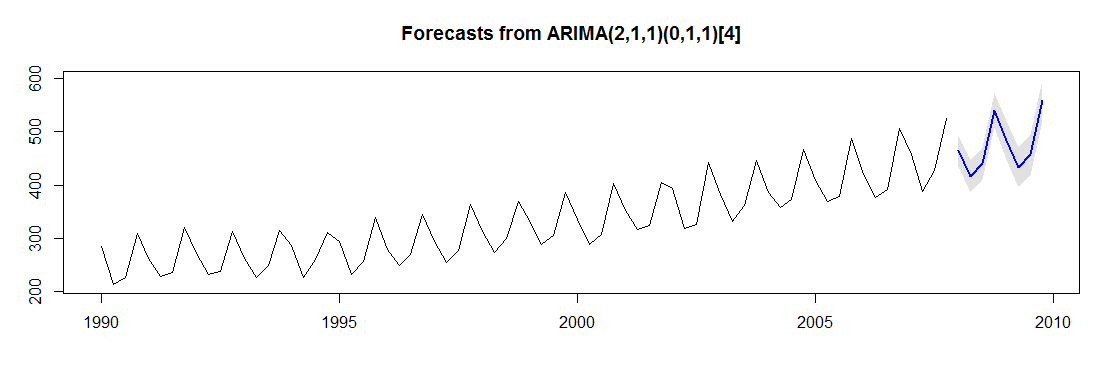


p-value and acf pacf of fit shows that the p-value of fit is all above 0.05 for lag 20 and we cant reject the null hypothesis.

## Forecasting

forecast future values for Model: fit:ARIMA(2,1,1)(0,1,1)[4]





## Conclusion:

The forecasts are shown as a blue line, with the 80% prediction intervals as a dark shaded area, and the 95% prediction intervals as a light shaded area.

This is the overall process by which we can analyze time series data and forecast values from existing series using ARIMA.