Oz Beer Sales – Time series Analysis

|  |
| --- |
|  |

# 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.

Dataset: [beer.csv](https://olympus.greatlearning.in/courses/1735/files/139210/download?wrap=1)

**Info on Data:**

There are 72 observations and one variable names OzBeer. It is quarterly data. So, 72 observations  
means 72/4=18 years data. OzBeer variable contain the sales information. Here the data period is  
not known. Assume the data is for past 18 years.

Objective:

1. The objective is to develop a model that will predict the future sales of beer based historical sales data
2. To perform time series analysis using Winter-Holts model
3. Predict the future sales based on Winter-Holts model
4. To perform time series analysis using ARIMA model
5. Predict the future sales based on ARIMA model
6. Evaluation of accuracy of models

Following are steps involved,

Exploratory data analysis

Time series modeling – HoltWinters method

Forecast HoltWinters model

Identify parameters for ARIMA

Time Series Model -ARIMA

Forecast – ARIMA Model

Compare the model results

Data Preparation

Data Import, cleanse and Split

First step is to import the data into R. File is imported into a dataset

library("forecast")

library("stats")  
library("data.table")

setwd("C:/Users/girish\_ps/Desktop/Work in progress/Time Series Assignment/")  
data = read.csv("beer.csv")

Validating the data that has been imported,

head(data)

## OzBeer  
## 1 284.4  
## 2 212.8  
## 3 226.9  
## 4 308.4  
## 5 262.0  
## 6 227.9

summary(data)

## OzBeer   
## Min. :212.8   
## 1st Qu.:272.6   
## Median :317.5   
## Mean :329.9   
## 3rd Qu.:379.7   
## Max. :525.0

# Checking for Null Values

summary(is.na(data))

[1] 0

Above code reads the files into a dataset. There are no Null (NA) values in the dataset

Converting the data to time series; Frequency quarterly

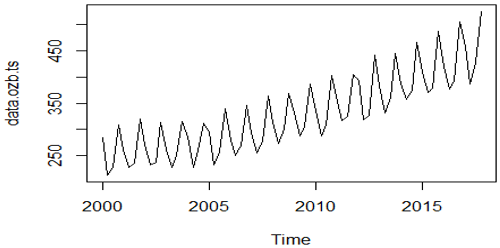
data.ozb.ts = ts(data$OzBeer,start = c(2000),frequency = 4)  
data.ozb.ts

## Qtr1 Qtr2 Qtr3 Qtr4  
## 2000 284.4 212.8 226.9 308.4  
## 2001 262.0 227.9 236.1 320.4  
## 2002 271.9 232.8 237.0 313.4  
## 2003 261.4 226.8 249.9 314.3  
## 2004 286.1 226.5 260.4 311.4  
## 2005 294.7 232.6 257.2 339.2  
## 2006 279.1 249.8 269.8 345.7  
## 2007 293.8 254.7 277.5 363.4  
## 2008 313.4 272.8 300.1 369.5  
## 2009 330.8 287.8 305.9 386.1  
## 2010 335.2 288.0 308.3 402.3  
## 2011 352.8 316.1 324.9 404.8  
## 2012 393.0 318.9 327.0 442.3  
## 2013 383.1 331.6 361.4 445.9  
## 2014 386.6 357.2 373.6 466.2  
## 2015 409.6 369.8 378.6 487.0  
## 2016 419.2 376.7 392.8 506.1  
## 2017 458.4 387.4 426.9 525.0

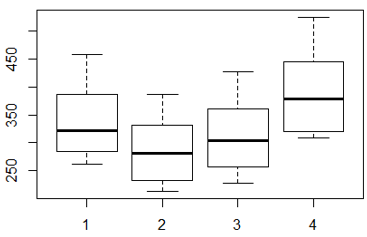
The data has been converted into time series format.

Exploratory data analysis

Plotting the data in a time series graph,  
plot(data.ozb.ts)

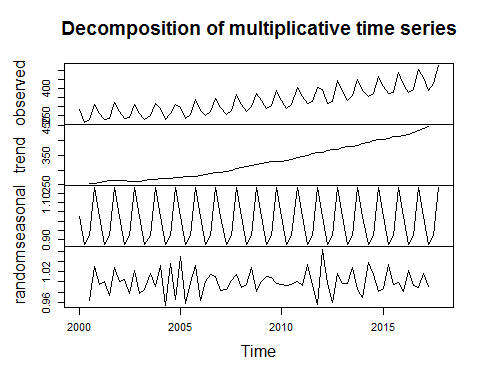


boxplot(data.ozb.ts~cycle(data.ozb.ts))



data.ozb.ts.comp <- decompose(data.ozb.ts, type = "multiplicative")  
plot(data.ozb.ts.comp)

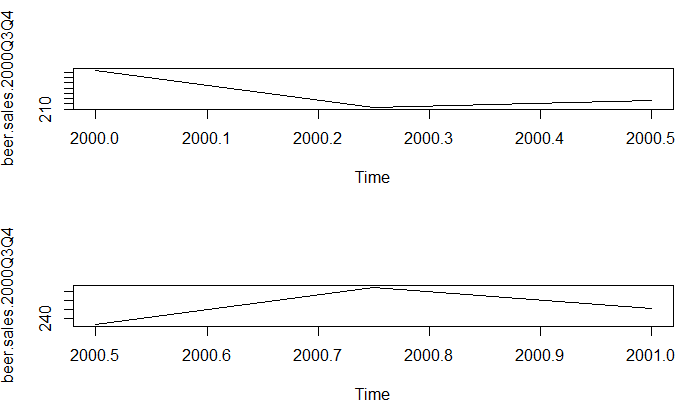
Decomposing the time series data to validate different components,



Decomposition indicates strong upward trend, seasonality and random noise that need to be considered during prediction. Seasonality can be considered as Additive as the seasonal values are fairly constant

**Short term trends -Quarterly year sales**

|  |
| --- |
| > layout(1:2)  > beer.sales.2000Q3Q4 = window(beer.ts, start = c(2000,1), end = c(2000,3))  > plot(beer.sales.2000Q3Q4)  > beer.sales.2000Q3Q4 = window(beer.ts, start = c(2000,3), end = c(2001,1))  > plot(beer.sales.2000Q3Q4) |

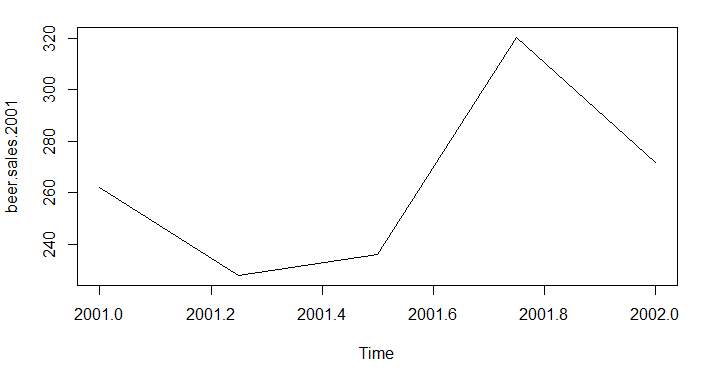
  
Above shows there is a seasonality in the yearly sales.

First two quarter sales shows that the sales reaches the lowest point

Second two quarter sales shows that the sales reaches, the highest point of sales.

**Yearly sales of 2001:**

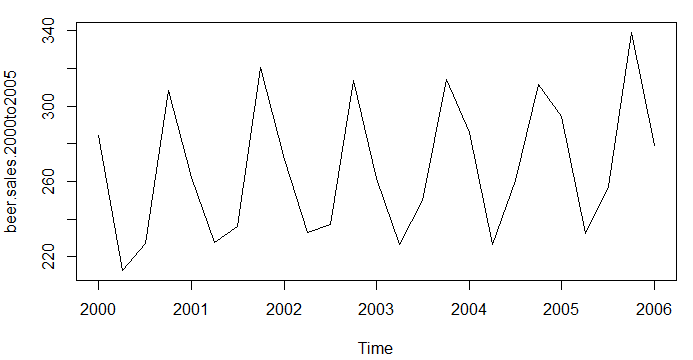
|  |  |
| --- | --- |
| |  | | --- | | > beer.sales.2001 = window(beer.ts, start = c(2001,1), end = c(2002,1))  > plot(beer.sales.2001) | |



Yearly sales of 2001, with peaks and dips in sales

**Six year sales 2000-2006:**

|  |  |
| --- | --- |
| |  | | --- | | > # Short term trends -six years sales  > beer.sales.2000to2005 = window(beer.ts, start = c(2000,1), end = c(2006,1))  > plot(beer.sales.2000to2005) | |



Forecast using HoltWinters method

Holt-Winters. Holt-Winter is used for exponential smoothing to make forecasts by using “additive” or “multiplicative” models with increasing or decreasing trend and seasonality. Smoothing is measured by beta and gamma parameters in Holt's model. As the seasonality is fairly constant, Additive model has been considered.

Applying HoltWinters model for model building and forecasting,

ozbforecasthv <- HoltWinters(data.ozb.ts)  
ozbforecasthv

## Holt-Winters exponential smoothing with trend and additive seasonal component.  
##   
## Call:  
## HoltWinters(x = data.ozb.ts)  
##   
## Smoothing parameters:  
## alpha: 0.1051635  
## beta : 0.3157796  
## gamma: 0.3294624  
##   
## Coefficients:  
## [,1]  
## a 445.766146  
## b 4.736944  
## s1 14.484341  
## s2 -40.981378  
## s3 -21.496644  
## s4 74.632951

ozbforecasthv$fitted

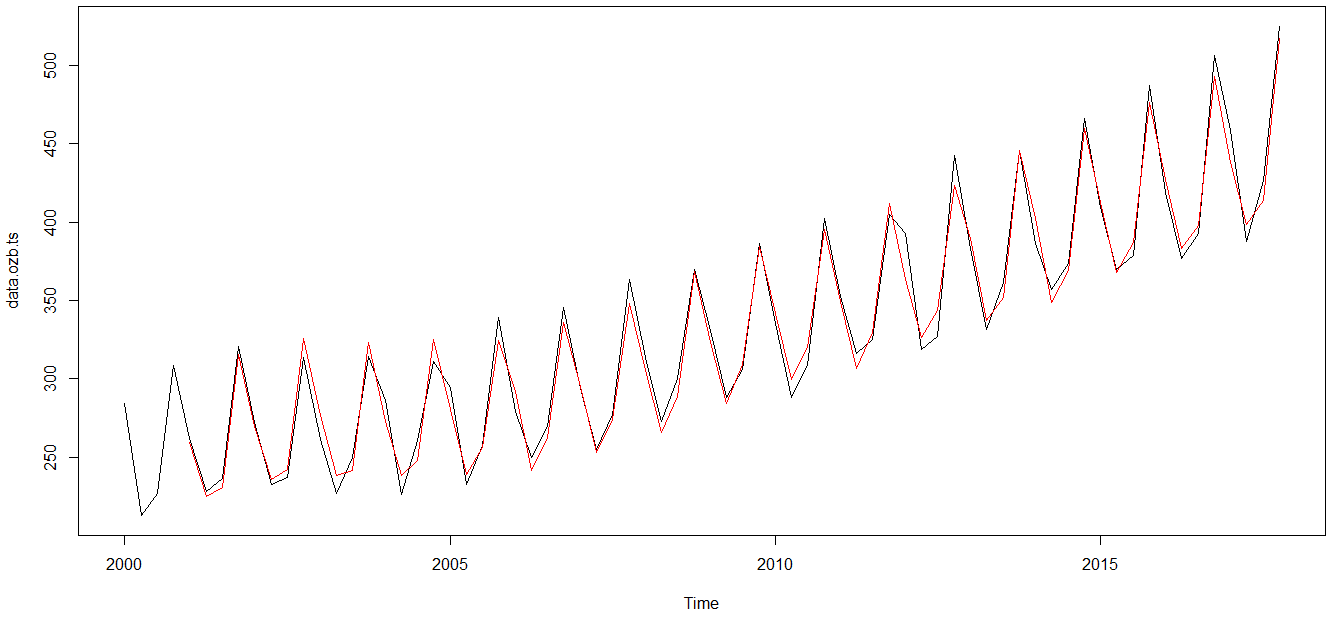
## xhat level trend season  
## 2001 Q1 259.2894 252.4812 1.7362500 5.071875  
## 2001 Q2 224.6507 254.5026 1.8262657 -31.678125  
## 2001 Q3 230.7016 256.6705 1.9341701 -27.903125  
## 2001 Q4 315.7952 259.1724 2.1134435 54.509375  
## 2002 Q1 269.9075 261.7701 2.2663606 5.871008  
…  
## 2016 Q3 397.9172 418.3153 3.4809730 -23.879029  
## 2016 Q4 493.1126 421.2581 3.3110379 68.543461  
## 2017 Q1 438.2090 425.9349 3.7423296 8.531734  
## 2017 Q2 398.5064 431.8006 4.4128418 -37.707035  
## 2017 Q3 413.7018 435.0455 4.0440140 -25.387663  
## 2017 Q4 517.3321 440.4775 4.4823052 72.372340

ozbforecasthv$SSE

## [1] 6737.801

As no parameters have been passed, default smoothing parameters alpha: 0.1052, beta: 0.3158 and gamma: 0.3295 is taken in the model.

Plotting the fitted values to original values,



Based on the above model, forecasting for next 24 months,

ozbforecasthvf <-forecast(ozbforecasthv, h=8)  
plot(ozbforecasthvf)

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

2018 Q1 464.9874 452.2508 477.7240 445.5085 484.4664

2018 Q2 414.2587 401.4007 427.1166 394.5941 433.9232

2018 Q3 438.4803 425.4380 451.5227 418.5338 458.4269

2018 Q4 539.3469 526.0463 552.6475 519.0054 559.6884

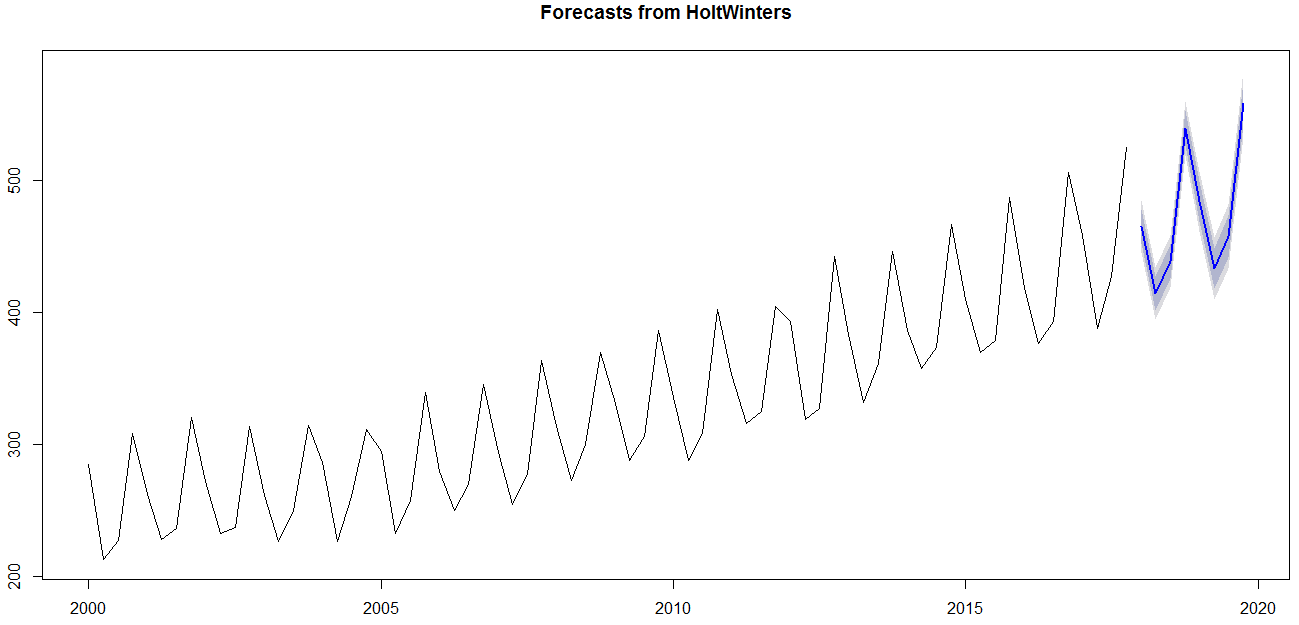
2019 Q1 483.9352 469.0034 498.8670 461.0990 506.7714

2019 Q2 433.2064 417.8803 448.5326 409.7671 456.6457

2019 Q3 457.4281 441.6192 473.2371 433.2504 481.6058

2019 Q4 558.2947 541.9113 574.6780 533.2385 583.3508

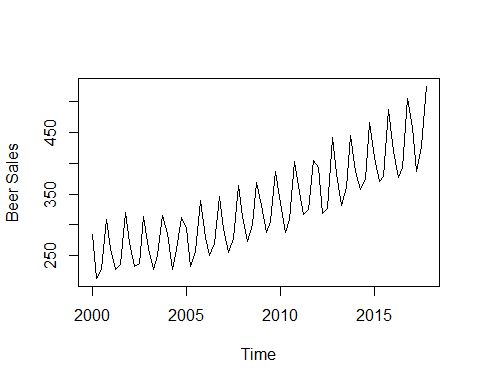
Plotting the forecasted values in the model,



Forecast using ARIMA model – Option analysis

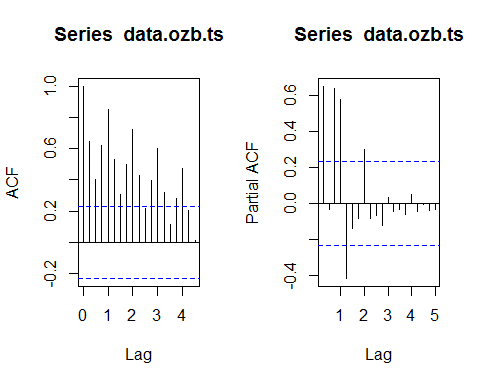
Auto Regressive Integrated Moving Average (ARIMA) is a class of model that captures a suite of different standard temporal structures in time series data. Following steps involves in identifying the options

**Step1:** Plotting the default data to identify different patterns. This is equivalent to plot time series (in terms of ARIMA, it is an ARIMA(0,0,0))  
par(mfrow = c(1, 1))  
plot(data.ozb.ts,ylab="Beer Sales")



Above plot shows a clear trend and seasonality in data.

**Step 2:** Validating the Auto Correlation using Auto Correlation function (ACF) and Partial Auto Correlation function (PACF). Plotting ACF and PACF to get preliminary understanding of the process  
par(mfrow = c(1, 2))  
acf(data.ozb.ts) # Correlogram  
pacf(data.ozb.ts, lag.max = 20)

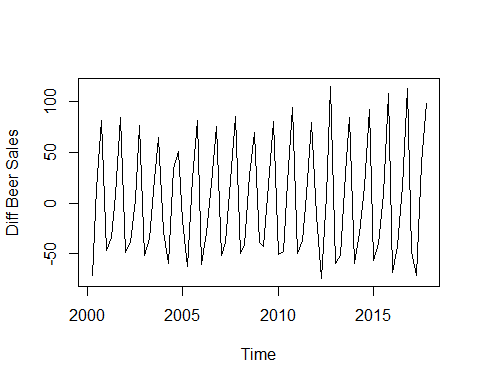


Spikes in ACF and PACF graph shows non stationary data and strong seasonality.

**Step 3:** Perform a non-seasonal difference to validate the model. This is equivalent ARIMA(0,1,0) model  
  
par(mfrow = c(1, 1))  
data.ozb.tsdiff1 <- diff(data.ozb.ts, differences = 1)  
data.ozb.tsdiff1

## Qtr1 Qtr2 Qtr3 Qtr4  
## 2000 -71.6 14.1 81.5  
## 2001 -46.4 -34.1 8.2 84.3  
## 2002 -48.5 -39.1 4.2 76.4  
## 2003 -52.0 -34.6 23.1 64.4  
## 2004 -28.2 -59.6 33.9 51.0  
## 2005 -16.7 -62.1 24.6 82.0  
## 2006 -60.1 -29.3 20.0 75.9  
## 2007 -51.9 -39.1 22.8 85.9  
## 2008 -50.0 -40.6 27.3 69.4  
## 2009 -38.7 -43.0 18.1 80.2  
## 2010 -50.9 -47.2 20.3 94.0  
## 2011 -49.5 -36.7 8.8 79.9  
## 2012 -11.8 -74.1 8.1 115.3  
## 2013 -59.2 -51.5 29.8 84.5  
## 2014 -59.3 -29.4 16.4 92.6  
## 2015 -56.6 -39.8 8.8 108.4  
## 2016 -67.8 -42.5 16.1 113.3  
## 2017 -47.7 -71.0 39.5 98.1

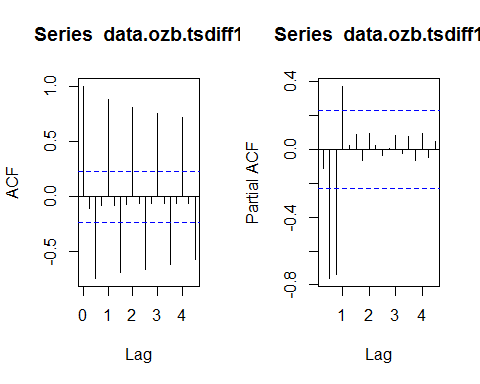
plot(data.ozb.tsdiff1,ylab="Diff Beer Sales")



After one seasonal difference data this looks stationery.

**# Step 4:** Check ACF and PACF to explore remaining dependencies

par(mfrow = c(1, 2))  
acf(data.ozb.tsdiff1)  
pacf(data.ozb.tsdiff1)

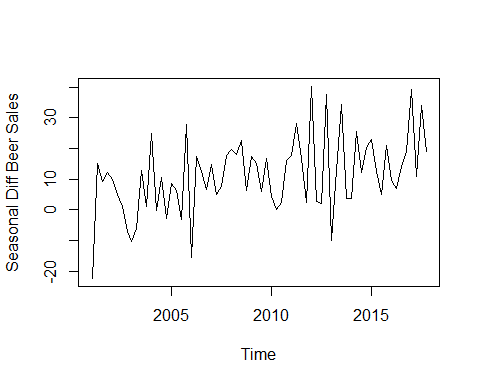


The multiple peaks in ACF shows a strong seasonal element in the data. This has to be removed by one seasonal difference.

**Step 5:** The differenced series looks stationary but has strong seasonal lag. Hence Performing a seasonal differencing on the original time series; Equivalent to the model (ARIMA(0,0,0)(0,1,0)4)  
  
par(mfrow = c(1, 1))  
data.ozb.ts.Seasdiff1 <- diff(data.ozb.ts, lag = 4, differences=1)  
data.ozb.ts.Seasdiff1

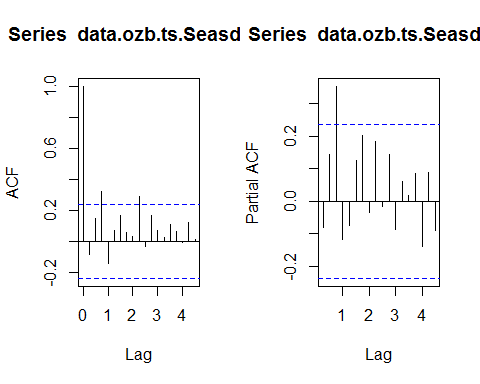
## Qtr1 Qtr2 Qtr3 Qtr4  
## 2001 -22.4 15.1 9.2 12.0  
## 2002 9.9 4.9 0.9 -7.0  
## 2003 -10.5 -6.0 12.9 0.9  
## 2004 24.7 -0.3 10.5 -2.9  
## 2005 8.6 6.1 -3.2 27.8  
## 2006 -15.6 17.2 12.6 6.5  
…  
## 2011 17.6 28.1 16.6 2.5  
## 2012 40.2 2.8 2.1 37.5  
## 2013 -9.9 12.7 34.4 3.6  
## 2014 3.5 25.6 12.2 20.3  
## 2015 23.0 12.6 5.0 20.8  
## 2016 9.6 6.9 14.2 19.1  
## 2017 39.2 10.7 34.1 18.9

plot(data.ozb.ts.Seasdiff1,ylab="Seasonal Diff Beer Sales")



Seasonality looks reduced in the graph.

**Step 6:** Checking ACF and PACF for seasonally differenced data to explore remaining dependencies  
par(mfrow = c(1, 2))  
acf(data.ozb.ts.Seasdiff1)  
pacf(data.ozb.ts.Seasdiff1)

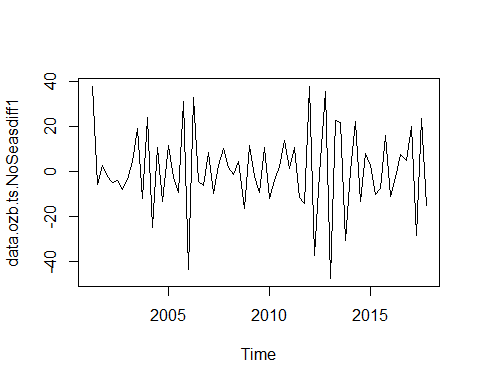


On Seasonally differenced data, ACF-cuts off after (Q) lags and PACF Tails off, so the we need MA(Q) in Model.

**Step 7**: Strong positive autocorrelation indicates need for either an AR component or a non-seasonal differencing. Perform a non-seasonal differencing. This is equivalent to ARIMA(0,1,0)(0,1,0)4  
par(mfrow = c(1, 1))  
data.ozb.ts.NoSeasdiff1 <- diff(data.ozb.ts.Seasdiff1, differences=1)  
data.ozb.ts.NoSeasdiff1

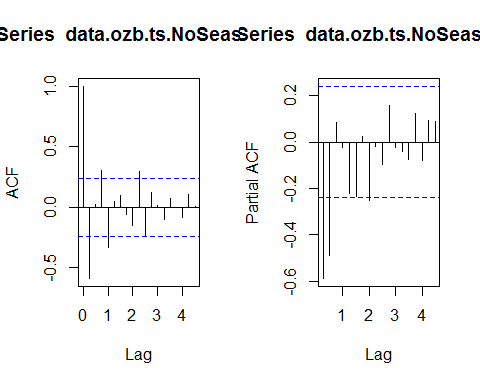
## Qtr1 Qtr2 Qtr3 Qtr4  
## 2001 37.5 -5.9 2.8  
## 2002 -2.1 -5.0 -4.0 -7.9  
## 2003 -3.5 4.5 18.9 -12.0  
## 2004 23.8 -25.0 10.8 -13.4  
## 2005 11.5 -2.5 -9.3 31.0  
## 2006 -43.4 32.8 -4.6 -6.1  
## 2007 8.2 -9.8 2.8 10.0  
## 2008 1.9 -1.5 4.5 -16.5  
## 2009 11.3 -2.4 -9.2 10.8  
## 2010 -12.2 -4.2 2.2 13.8  
## 2011 1.4 10.5 -11.5 -14.1  
## 2012 37.7 -37.4 -0.7 35.4  
## 2013 -47.4 22.6 21.7 -30.8  
## 2014 -0.1 22.1 -13.4 8.1  
## 2015 2.7 -10.4 -7.6 15.8  
## 2016 -11.2 -2.7 7.3 4.9  
## 2017 20.1 -28.5 23.4 -15.2

plot(data.ozb.ts.NoSeasdiff1)



**Step 8:** Checking ACF and PACF to explore remaining dependencies

par(mfrow = c(1, 2))  
acf(data.ozb.ts.NoSeasdiff1)  
pacf(data.ozb.ts.NoSeasdiff1)



**Step 9:** ACF and PACF show significant lag-2, which then cutoff, requiring an AR (2) and an MA (1) term. Also, the significant lag at the seasonal. Period is negative, requiring a Seasonal MA (1) term. Applying the above in ARIMA Model,

ozbArima1 = Arima(data.ozb.ts, order = c(2,1,1), seasonal = c(0,1,1), include.drift = FALSE)  
ozbArima1

## Series: data.ozb.ts   
## ARIMA(2,1,1)(0,1,1)[4]   
##   
## Coefficients:  
## ar1 ar2 ma1 sma1  
## -0.5280 -0.2791 -0.5445 -0.6170  
## s.e. 0.1922 0.1700 0.1852 0.1044  
##   
## sigma^2 estimated as 109.6: log likelihood=-252.08  
## AIC=514.16 AICc=515.14 BIC=525.18

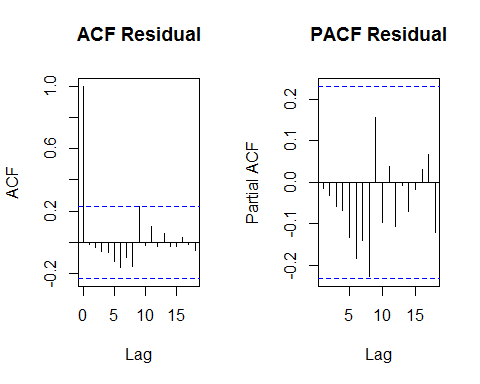
Trying Autopilot ARIMA models for cross checking,

ozbArima2 = auto.arima(data.ozb.ts)  
ozbArima2

## Series: data.ozb.ts   
## ARIMA(2,1,1)(0,1,1)[4]   
##   
## Coefficients:  
## ar1 ar2 ma1 sma1  
## -0.5280 -0.2791 -0.5445 -0.6170  
## s.e. 0.1922 0.1700 0.1852 0.1044  
##   
## sigma^2 estimated as 109.6: log likelihood=-252.08  
## AIC=514.16 AICc=515.14 BIC=525.18

Auto ARIMA takes the same values. Hence no difference.

**Step 10:** Check residuals to ensure they are white noise  
par(mfrow=c(1,2))  
acf(ts(ozbArima1$residuals),main="ACF Residual")  
pacf(ts(ozbArima1$residuals),main="PACF Residual")



**Interpretation on Residual analysis:**

As there are no significant lags in ACF and PACF of residuals, we can conclude that model fits properly

**Performing Box Ljung test,**

Box.test(ozbArima1$residuals, lag=24, type="Ljung-Box")

##   
## Box-Ljung test  
##   
## data: ozbArima1$residuals  
## X-squared = 16.389, df = 24, p-value = 0.8735

|  |
| --- |
| **Interpretation of Ljung-Box test :**  Ljung-Box test to ensure that there is no pattern seen in the residuals of the model  H0: The residuals are independently distributed without any pattern  HA: The residuals has a pattern/correlation  As the P-value is 0.08735, we shall not reject the null hypothesis, and hence the residuals are independent and does not have any pattern. |

**Step 11:** Forecasting the Beer Sales based on above model,

par(mfrow = c(1, 1))  
#BeerSalesForecasts <- forecast.Arima(LogBeerSalesARIMA, h=36)  
ozbForecasts <- forecast(ozbArima1, h=8)  
> ozbForecasts

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

2018 Q1 464.7357 451.3187 478.1527 444.2162 485.2552

2018 Q2 416.1853 402.7331 429.6375 395.6120 436.7587

2018 Q3 439.8718 426.1148 453.6288 418.8323 460.9113

2018 Q4 540.6241 526.0332 555.2149 518.3092 562.9389

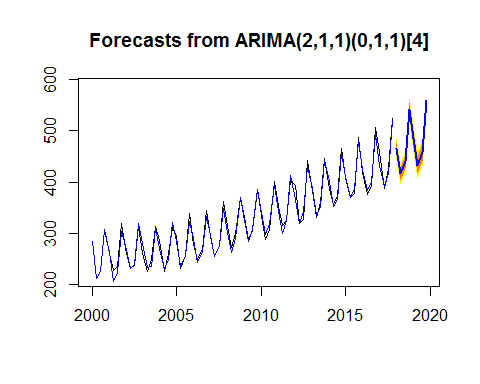
2019 Q1 484.8095 468.2272 501.3917 459.4491 510.1698

2019 Q2 433.1695 416.3296 450.0094 407.4151 458.9239

2019 Q3 457.2452 439.7609 474.7295 430.5053 483.9851

2019 Q4 558.6543 540.4276 576.8811 530.7789 586.5297

#plot.forecast(BeerSalesForecasts, shadecols = "oldstyle")  
plot(ozbForecasts, shadecols = "oldstyle")  
lines(ozbForecasts$fitted,col="blue")



Comparing HoltWinters and ARIMA results,

# Validating the accuracy of ARIMA Model  
accuracy(ozbArima1)

## ME RMSE MAE MPE MAPE MASE  
## Training set 2.5829 9.793161 7.673424 0.7301573 2.367513 0.5666733  
## ACF1  
## Training set -0.01539237

# Validating the accuracy of HoltWinters Model  
accuracy(ozbforecasthvf)

## ME RMSE MAE MPE MAPE MASE  
## Training set 1.328813 9.95416 8.409191 0.2677412 2.568834 0.6210089  
## ACF1  
## Training set -0.2046324

**Validation of accuracy:**

**On comparing the results of accuracy between HoltWinters model and ARIMA model, errors on both models are comparable.**

**Considering the simplest measure MAPE, which is useful in comparing the two models, MAPE of ARIMA is 2.36% which is slightly lesser than the HoltWinters model value of 2.56%**

|  |
| --- |
| **Validating ARIMA with Train data and Test data:** |
| |  | | --- | | #########ARIMA with Train and testdata  > #Dividing the data into Training and test data  > #Taking first 15 years as Training data  > trainbeer<- beernew[1:60,]  > View(trainbeer)  > #Taking last 3 years as Test data  > testbeer <- beernew[61:72,]  > View(testbeer)  > #Converting training and testdata as timeseries object  > trainbeer.ts = ts(trainbeer, start = c(2000,1), frequency = 4)  > testbeer.ts = ts(testbeer, start = c(2015,1), frequency = 4)  > summary(trainbeer.ts)  Min. 1st Qu. Median Mean 3rd Qu. Max.  212.8 261.9 308.4 310.2 347.5 466.2  > summary(testbeer.ts)  Min. 1st Qu. Median Mean 3rd Qu. Max.  369.8 385.2 414.4 428.1 465.6 525.0  > #Building the Model with only Training data  > trainbeerArima1 <- Arima(trainbeer.ts, order = c(2,1,1),  + seasonal = c(0,1,1), include.drift = FALSE)  > trainbeerArima1  Series: trainbeer.ts  ARIMA(2,1,1)(0,1,1)[4]  Coefficients:  ar1 ar2 ma1 sma1  -0.5704 -0.3084 -0.5028 -0.6484  s.e. 0.2129 0.1828 0.2135 0.1204  sigma^2 estimated as 110.9: log likelihood=-207.33  AIC=424.65 AICc=425.87 BIC=434.69  > #Forecating with the built Model for next 3 years  > trainbeertimeseriesforecastsArima1 <- forecast(trainbeerArima1, h=12)  > #plotting the forecast  > plot(trainbeertimeseriesforecastsArima1)    ***Forecasting the Beer sales for the Test data period:***  Forecasting with the built Model for next 3 years 2015-2017  > trainbeertimeseriesforecastsArima1 <- forecast (trainbeerArima1, h=12)  > trainbeertimeseriesforecastsArima1  Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  2015 Q1 411.1334 397.6350 424.6318 390.4894 431.7774  2015 Q2 369.5564 356.0219 383.0909 348.8571 390.2556  2015 Q3 388.2088 374.3212 402.0965 366.9695 409.4481  2015 Q4 478.3353 463.4917 493.1789 455.6340 501.0366  2016 Q1 427.6764 411.0421 444.3108 402.2364 453.1165  2016 Q2 384.3481 367.4141 401.2821 358.4497 410.2464  2016 Q3 402.6401 384.9986 420.2816 375.6598 429.6205  2016 Q4 493.5123 475.1001 511.9245 465.3533 521.6713  2017 Q1 442.5393 421.9186 463.1599 411.0027 474.0758  2017 Q2 399.1601 378.0524 420.2679 366.8786 431.4416  2017 Q3 417.5780 395.6354 439.5206 384.0197 451.1363  2017 Q4 508.3941 485.4746 531.3135 473.3418 543.4464  **> #Checking the accuracy with Test data**  > accuracy(trainbeertimeseriesforecastsArima1,testbeer.ts)  ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  Training set 2.683575 9.710779 7.628459 0.80776977 2.476602 0.6044909 -0.01045413 NA  Test set 1.201472 10.440144 9.345977 0.05279216 2.137038 0.7405897 -0.06628840 0.1550357  **Forecasting the Beer sales for 2018-2020:**  As our model fits successfully, now we can forecast our data for the required period 2018-2020  Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  2018 Q1 457.4142 432.1827 482.6457 418.8260 496.0024  2018 Q2 414.0563 388.1824 439.9301 374.4856 453.6269  2018 Q3 432.4642 405.5891 459.3393 391.3623 473.5661  2018 Q4 523.2794 495.2736 551.2851 480.4483 566.1105  2019 Q1 472.3031 441.9125 502.6937 425.8247 518.7815  2019 Q2 428.9434 397.7596 460.1271 381.2520 476.6348  2019 Q3 447.3512 415.0236 479.6788 397.9104 496.7921  2019 Q4 538.1670 504.5743 571.7597 486.7914 589.5426  2020 Q1 487.1904 451.1492 523.2316 432.0701 542.3107  2020 Q2 443.8307 406.8586 480.8028 387.2867 500.3747  2020 Q3 462.2386 423.9951 500.4822 403.7501 520.7271  2020 Q4 553.0544 513.4251 592.6837 492.4466 613.6622 | |
|  |

|  |
| --- |
| **Conclusion:**  On testing the accuracy of our model with test data, we can see the MAPE of test data is 2.13% , which shows there is no overfitting and our model performs very Well.  ARIMA model performs very well statistically by having lesser error ratios compared to HoltWinters.  ARIMA considers Stationarity and Seasonality very well into the model, whereas HoltWinters is more of a mathematical model embedded on a data.  **Hence we can unanimously say, for this dataset with stationarity and Seasonality issues, ARIMA model works very well.** |