TIME SERIES PRACTICAL MSE

Pranav Gopalkrishna, 1940223

2022-03-29

data = read.csv("~/Documents/Study/computerScience/programming/r/data/monthlyBeerProductionIndia.csv")  
head(data)

## Month Monthly.beer.production  
## 1 1956-01 93.2  
## 2 1956-02 96.0  
## 3 1956-03 95.2  
## 4 1956-04 77.1  
## 5 1956-05 70.9  
## 6 1956-06 64.8

summary(data)

## Month Monthly.beer.production  
## 1956-01: 1 Min. : 64.8   
## 1956-02: 1 1st Qu.:112.9   
## 1956-03: 1 Median :139.2   
## 1956-04: 1 Mean :136.4   
## 1956-05: 1 3rd Qu.:158.8   
## 1956-06: 1 Max. :217.8   
## (Other):470

#========================

DATA HANDLING

# Checking first and last dates  
data$Month[1]

## [1] 1956-01  
## 476 Levels: 1956-01 1956-02 1956-03 1956-04 1956-05 1956-06 1956-07 ... 1995-08

data$Month[length(data$Month)]

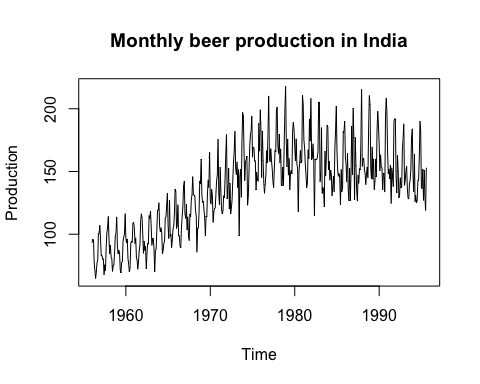
## [1] 1995-08  
## 476 Levels: 1956-01 1956-02 1956-03 1956-04 1956-05 1956-06 1956-07 ... 1995-08

# Converting to time series  
z = ts(data$Monthly.beer.production, start = c(1956, 1, 1), end = c(1995, 8, 1), frequency = 12)  
#========================

BASIC ANALYSIS

Time plot

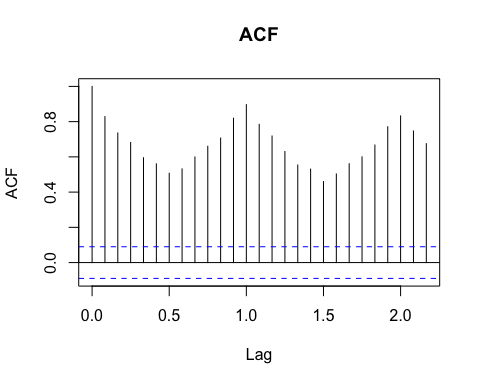
ts.plot(z,  
 main="Monthly beer production in India",  
 xlab="Time",  
 ylab="Production")



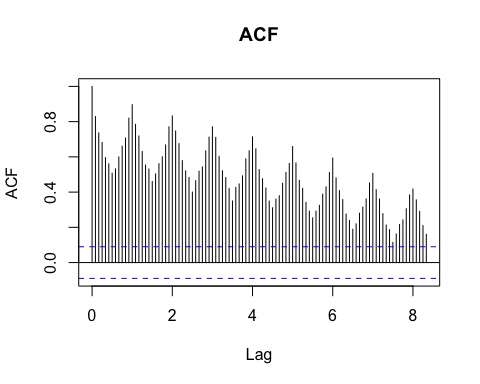
From the data, there can be said to be an overall upward trend, although this is not linear nor constantly increasing, as it stagnates and starts dipping after around 1975. There is no sign of cyclic fluctuation yet, but given a larger dataset, we may observe it. Irregular fluctuations seem to be limited, and the main source of fluctuation seems to be periodic.

ACF plot

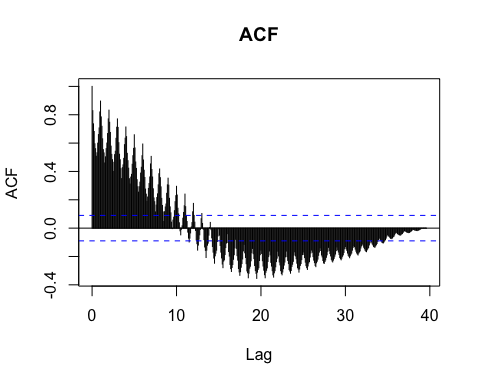
acf(z, main="ACF")



acf(z, main="ACF", 100)



acf(z, main="ACF", 1000)



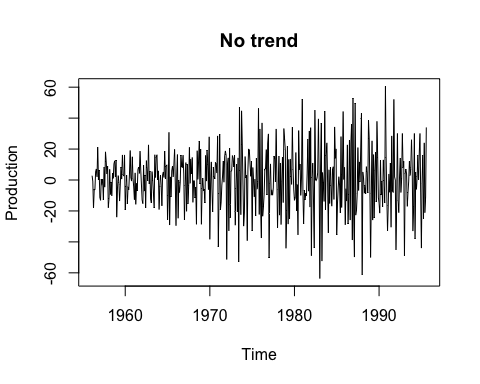
Hence, from the ACF, due to the ACF being a periodic function of the lag 1, we observe autoregression.

#========================

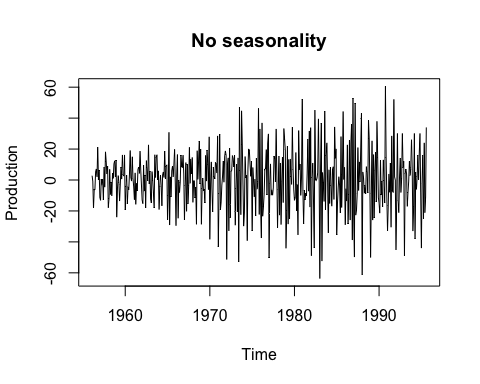
MAKING DATA STATIONARY

Differencing

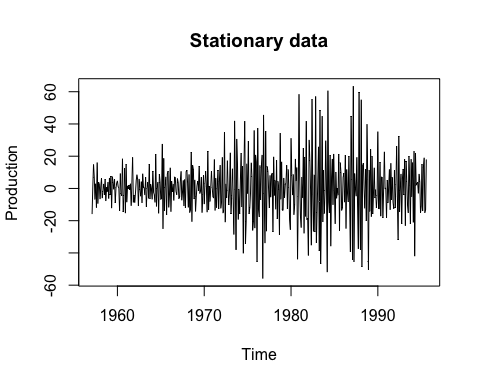
# Making time series stationary using differencing  
z\_notrend = diff(z, 1)  
z\_noseasonality = diff(z, 1)  
z\_stationary = diff(diff(z, 12), 1)  
ts.plot(z\_notrend,  
 main="No trend",  
 xlab="Time",  
 ylab="Production")



ts.plot(z\_noseasonality,  
 main="No seasonality",  
 xlab="Time",  
 ylab="Production")

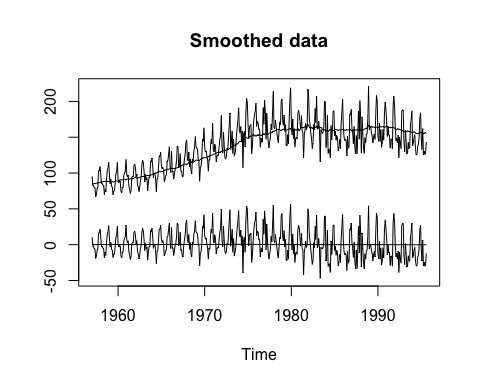


ts.plot(z\_stationary,  
 main="Stationary data",  
 xlab="Time",  
 ylab="Production")



Exponential smoothing

# Making time series stationary using Winter's exponential smoothing  
smoothed = HoltWinters(z, gamma = TRUE)  
ts.plot(smoothed$fitted,  
 main="Smoothed data")



#========================

IDENTIFYING AND FITTING SUITABLE STATIONARY TIME SERIES MODEL

Model shows trend and seasonality. Seasonality implies a strong degree of autocorrelation in the data. From the ACF of the data, we observe significant autocorrelation for observations between the every 1 lag. Hence, an AR(1) model will be most suitable.

Trying to fit ARMA(1, 1)

library(forecast)

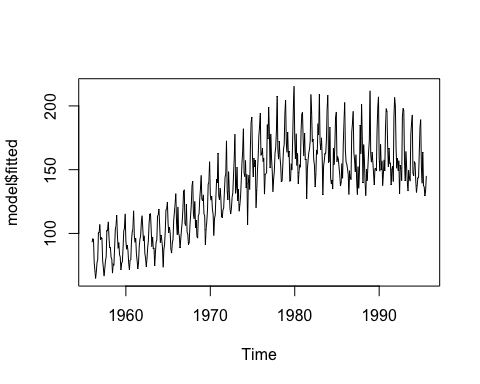
## Warning: package 'forecast' was built under R version 3.6.2

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

model = auto.arima(z,  
 max.d = 0,  
 start.p = 1,  
 max.p = 1,  
 start.q = 1,  
 max.q = 1,  
 seasonal = TRUE)  
summary(model)

## Series: z   
## ARIMA(0,0,0)(0,1,2)[12] with drift   
##   
## Coefficients:  
## sma1 sma2 drift  
## -0.3444 -0.0570 0.1391  
## s.e. 0.0458 0.0408 0.0277  
##   
## sigma^2 = 138: log likelihood = -1800.92  
## AIC=3609.85 AICc=3609.94 BIC=3626.41  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -0.0139961 11.56128 8.46091 -0.2045497 6.0675 0.9040558 0.02305584

plot(model$fitted)



# Order of integration i.e. order of differencing = 0  
# Order oving average = 1  
# Order of autoregression = 1

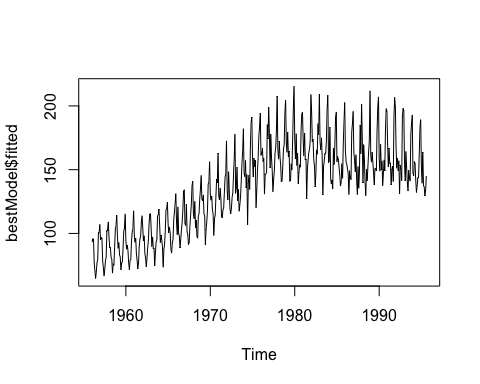
This model seems to fit the data closely. ARMA(1, 1) is fairly close to the suggested model of AR(1), hence this may not be surprising.

Identifying the best model

bestModel = auto.arima(z,  
 max.d = 0,  
 seasonal = TRUE)  
summary(bestModel)

## Series: z   
## ARIMA(0,0,0)(0,1,2)[12] with drift   
##   
## Coefficients:  
## sma1 sma2 drift  
## -0.3444 -0.0570 0.1391  
## s.e. 0.0458 0.0408 0.0277  
##   
## sigma^2 = 138: log likelihood = -1800.92  
## AIC=3609.85 AICc=3609.94 BIC=3626.41  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -0.0139961 11.56128 8.46091 -0.2045497 6.0675 0.9040558 0.02305584

plot(bestModel$fitted)



We can see that the information criteria are similar to the model fitted for ARMA(1, 1). Hence, we can say that ARMA(1, 1) is suitable model for our time series.

Forecasting next 20 points

forecasted = forecast(bestModel, h = 20)  
  
print("Next 20 data points forecasted:")

## [1] "Next 20 data points forecasted:"

forecasted

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Sep 1995 144.9671 129.9116 160.0226 121.9417 167.9925  
## Oct 1995 164.3372 149.2817 179.3927 141.3118 187.3626  
## Nov 1995 189.8000 174.7445 204.8555 166.7746 212.8254  
## Dec 1995 186.6652 171.6097 201.7207 163.6398 209.6906  
## Jan 1996 144.4652 129.4097 159.5207 121.4398 167.4906  
## Feb 1996 139.4628 124.4073 154.5183 116.4374 162.4882  
## Mar 1996 157.3100 142.2545 172.3655 134.2846 180.3354  
## Apr 1996 133.9444 118.8889 148.9999 110.9190 156.9698  
## May 1996 148.2481 133.1926 163.3036 125.2227 171.2735  
## Jun 1996 131.8715 116.8160 146.9269 108.8461 154.8968  
## Jul 1996 126.2738 111.2183 141.3293 103.2484 149.2992  
## Aug 1996 151.9990 136.9435 167.0545 128.9736 175.0244  
## Sep 1996 146.7041 128.7020 164.7063 119.1722 174.2361  
## Oct 1996 166.1939 148.1918 184.1961 138.6620 193.7259  
## Nov 1996 191.0738 173.0716 209.0760 163.5419 218.6057  
## Dec 1996 188.7458 170.7437 206.7480 161.2139 216.2778  
## Jan 1997 146.9599 128.9577 164.9620 119.4279 174.4918  
## Feb 1997 141.3244 123.3222 159.3266 113.7925 168.8563  
## Mar 1997 159.6541 141.6520 177.6563 132.1222 187.1860  
## Apr 1997 136.2136 118.2115 154.2158 108.6817 163.7455

plot(forecasted,  
 main = "Past data + forecasts for the next 20 monthts",  
 xlab = "Year",  
 ylab = "Price")

