Beer Sales - Forecasting

Georgios Skouras

5/16/2017

#Beer Sales - Forecasting

Load data from TSA package (the package is written by authors Jonathan Cryer and Kung-Sik Chan).

library("TSA")

data(beersales)

The data is the monthly beer sales in millions of barrels, 01/1975 - 12/1990.

```
library(tseries)
library(TSA)
```

```
## Loading required package: leaps

## Loading required package: locfit

## locfit 1.5-9.1    2013-03-22

## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-22. For overview type 'help("mgcv-package")'.

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```
library(forecast)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2018i.
## 1.0/zoneinfo/America/Los_Angeles'

##
## Attaching package: 'forecast'

## The following object is masked from 'package:nlme':
##
##     getResponse
```

Loading data

```
data("beersales")
```

```
head(beersales)
```

```
##           Jan     Feb     Mar     Apr     May     Jun
## 1975 11.1179  9.8413 11.5732 13.0097 13.4182 14.4418
```

```r
tail(beersales)
```

```
##        Jul   Aug   Sep   Oct   Nov   Dec
## 1990 17.00 17.40 14.75 15.77 14.54 13.22
```

#Part 1

Use ARIMA(p,d,q) model to forecast beer sales for all months of 1990.

1A - Use the h-period in forecast() to forecast each month of 1990.

First we need to split our data into train (1975-1989) and test (1990)

```r
beerdata.train<-beersales[c(1:180)]
beerdata.test<-beersales[c(181:192)]
```

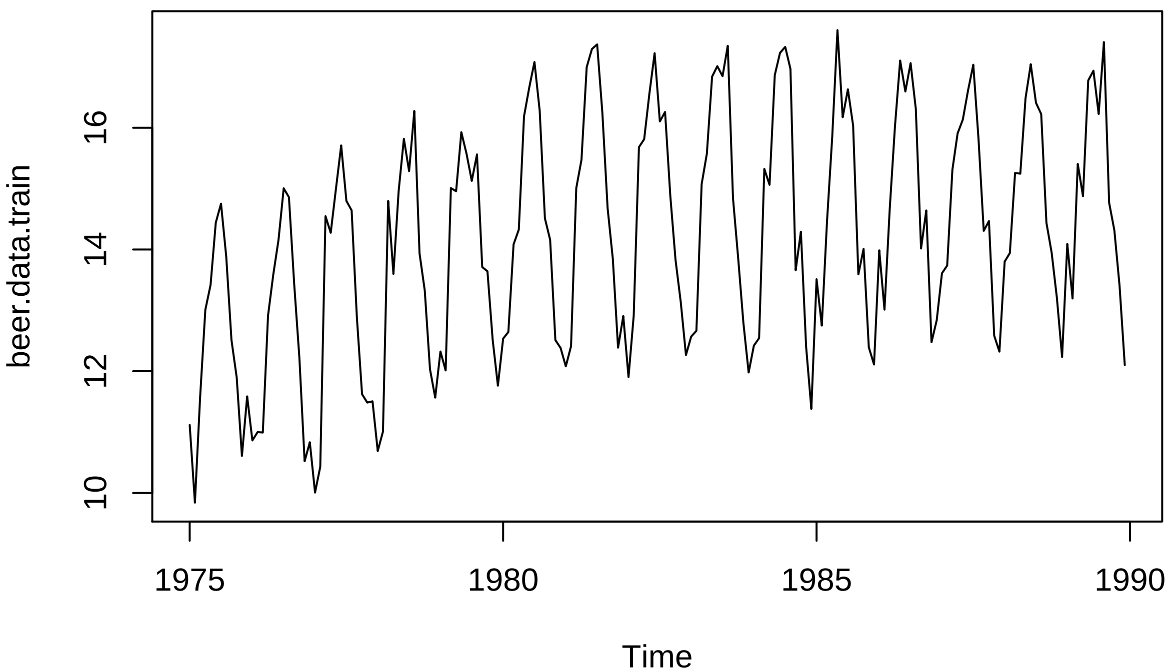
```r
beer.data.train <- ts(beerdata.train,start=c(1975, 1),end=c(1989,12), frequency = 12)
head(beer.data.train)
```

```
##           Jan     Feb     Mar     Apr     May     Jun
## 1975 11.1179  9.8413 11.5732 13.0097 13.4182 14.4418
```

```r
tail(beer.data.train)
```

```
##           Jul     Aug     Sep     Oct     Nov     Dec
## 1989 16.2259 17.4078 14.7684 14.3167 13.4048 12.0999
```

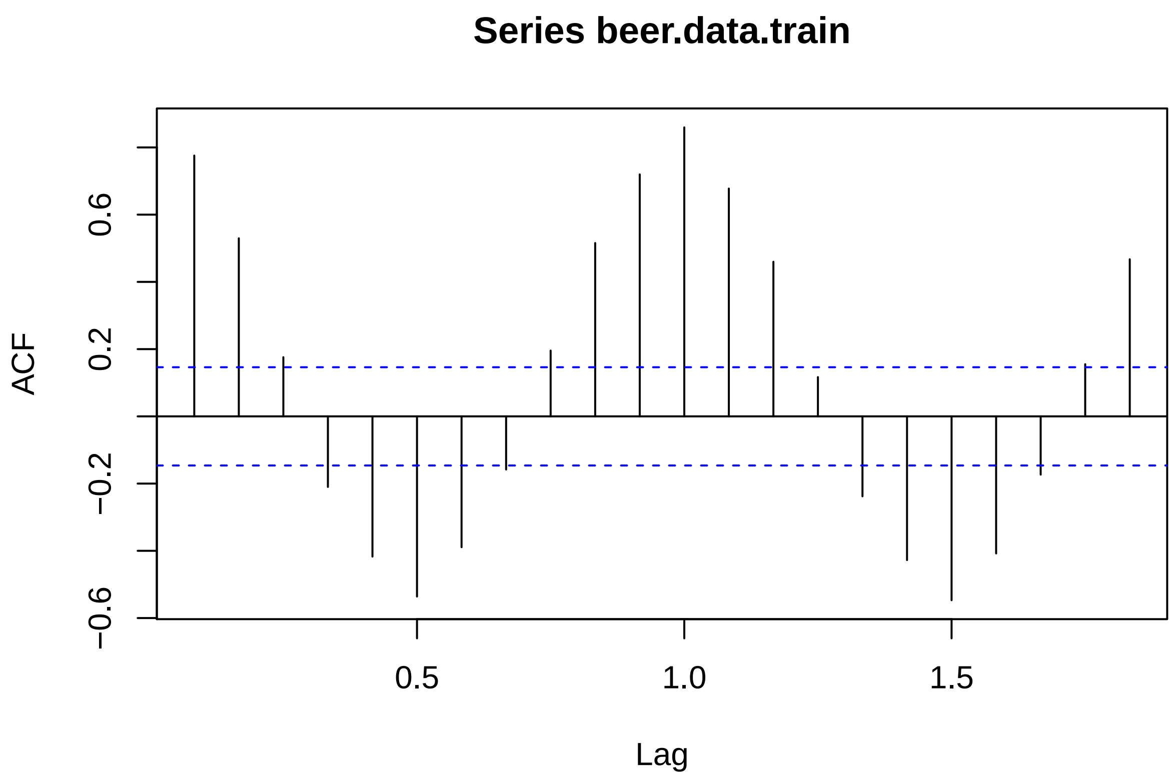
```r
plot(beer.data.train)
```



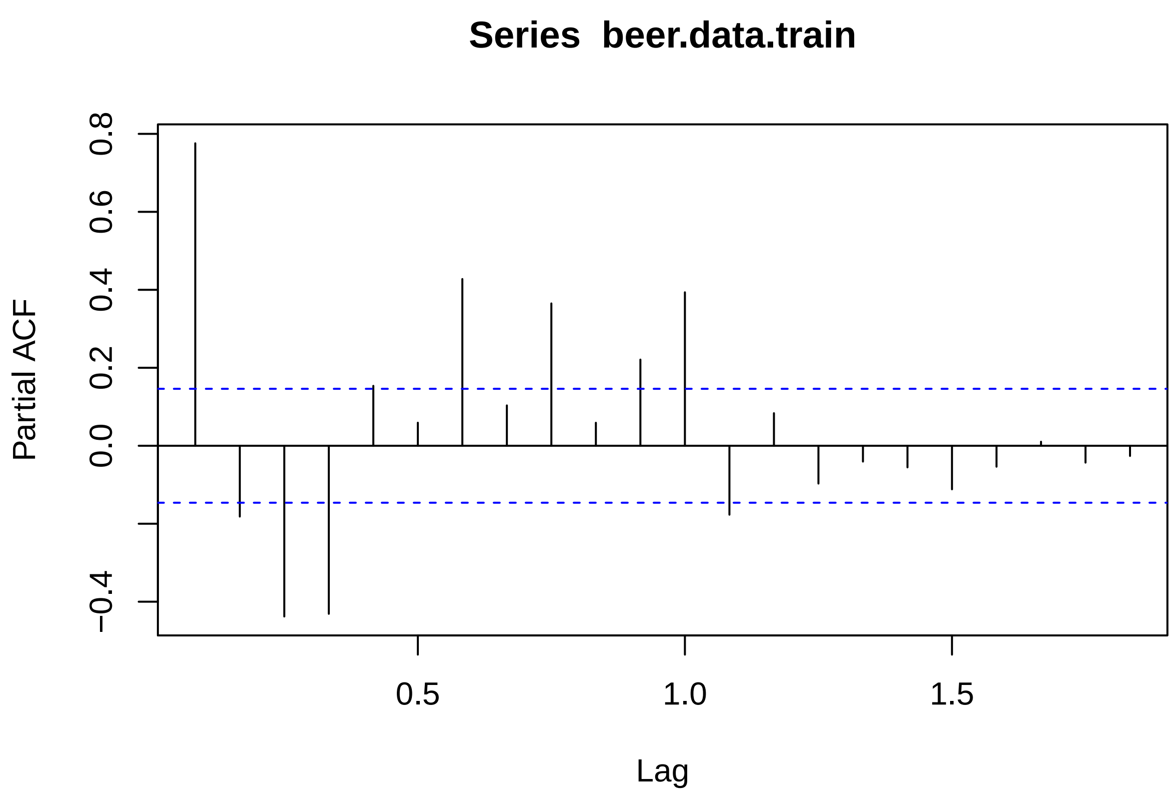
Based on the plot, we can tell that there is seasonality in our ts as well as an upward trend over time.

Next we will check acf and pacf of the data

```r
acf(beer.data.train)
```




```r
pacf(beer.data.train)
```



We see many significant lags in the ACF and less in the PACF.

Next we will test stationarity of our ts.

```r
adf.test(beer.data.train)
```

```
## Warning in adf.test(beer.data.train): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  beer.data.train
## Dickey-Fuller = -9.1654, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Surprisingly and despite the seasonality and the upward trend it seems that our ts is stationary.

Next we will check auto.arima for suggestions regarding the model we need to use for our prediction

```r
auto.arima(beer.data.train, seasonal = FALSE)
```

```
## Series: beer.data.train
## ARIMA(1,1,3)
##
## Coefficients:
##           ar1     ma1     ma2     ma3
##       -0.3636  0.3530  0.3702  0.6659
## s.e.   0.1142  0.0856  0.0563  0.0626
##
## sigma^2 estimated as 1.111:  log likelihood=-262.29
## AIC=534.58   AICc=534.93   BIC=550.52
```

Auto.arima suggestion is to use p = 1 and q = 3 and d = 1 (although original time series turned out to be stationary).

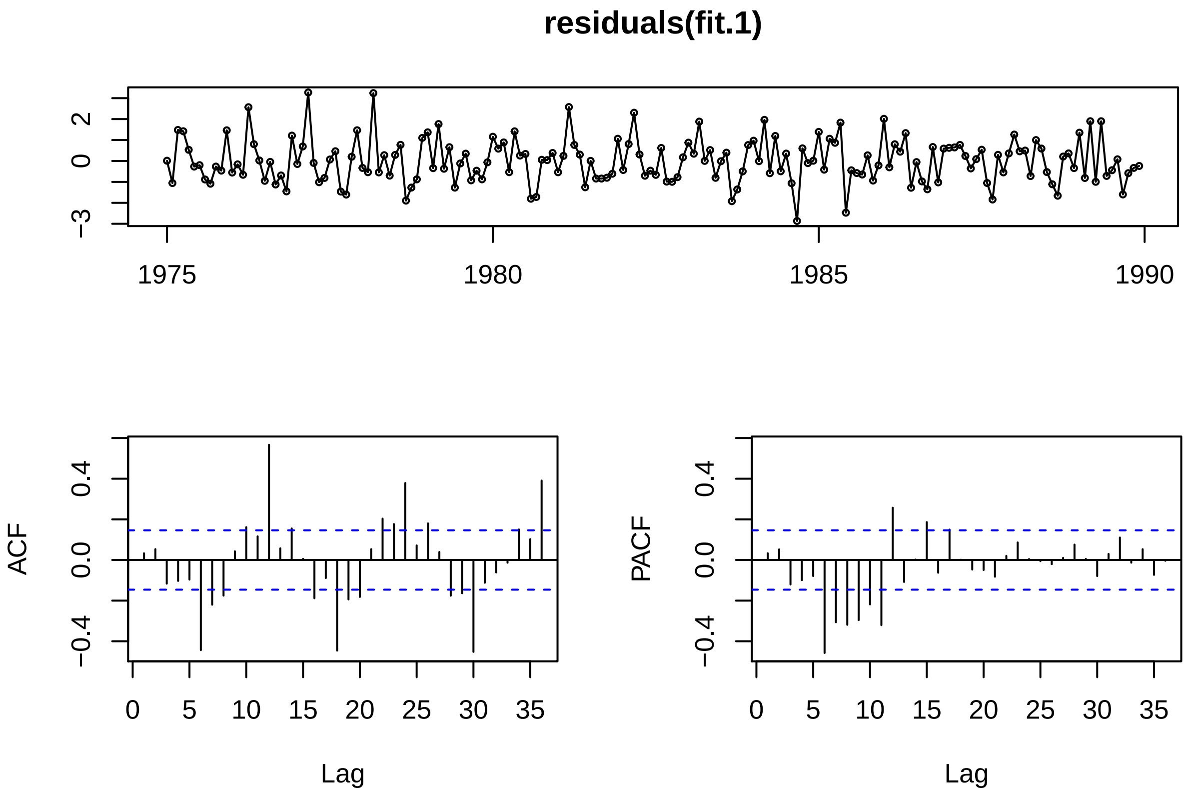
We will use the suggested parameters to fit our model

```r
fit.1 <- Arima(beer.data.train, order = c(1, 1, 3)); fit.1
```

```
## Series: beer.data.train
## ARIMA(1,1,3)
##
## Coefficients:
##           ar1     ma1     ma2     ma3
##       -0.3636  0.3530  0.3702  0.6659
## s.e.   0.1142  0.0856  0.0563  0.0626
##
## sigma^2 estimated as 1.111:  log likelihood=-262.29
## AIC=534.58   AICc=534.93   BIC=550.52
```

Next we will check our residuals

```r
tsdisplay(residuals(fit.1))
```




```r
Box.test(residuals(fit.1), lag = 12, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(fit.1)
## X-squared = 129.83, df = 12, p-value < 2.2e-16
```

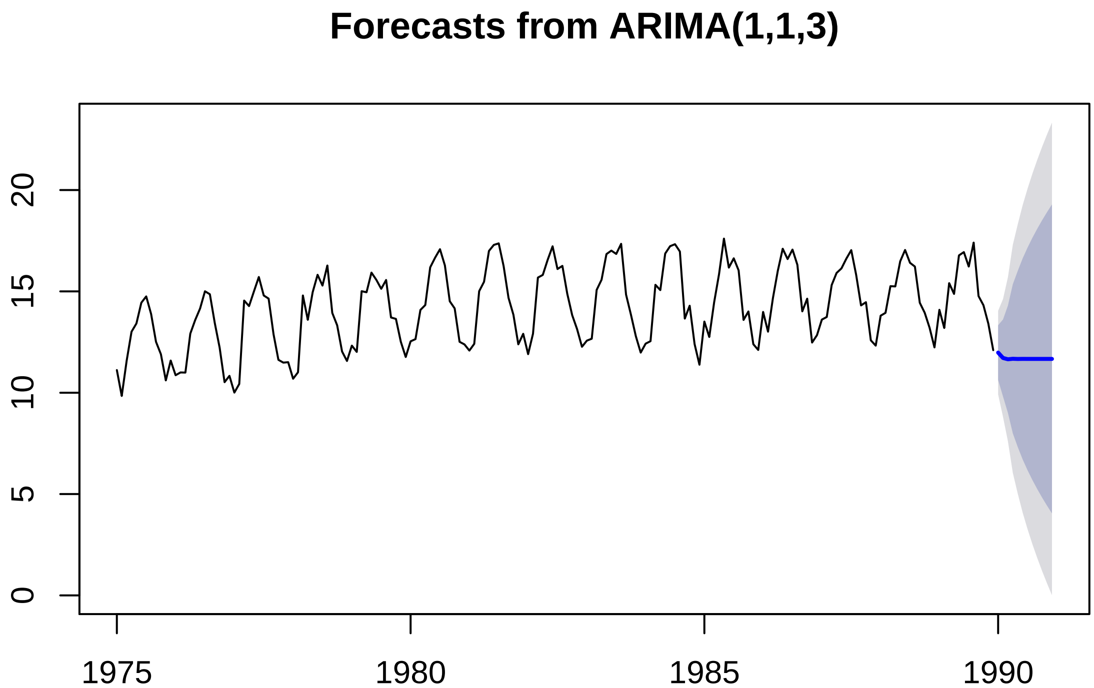
We are noticing significant auto-correlations at certain lags, thus, our residuals are not similar to white noise. This might have to do with the fact that additional seasonal terms were not included in the model. Moreover, the Ljung-Box test indicates we should reject the null hypothesis (at 90% confidence level) saying there is no auto correlation.

We will use the model to predict beersales for all 12 months of 1990.

```r
beer.forecast.fit.1 <- forecast(fit.1, h = 12)
beer.forecast.fit.1
```

```
##          Point Forecast     Lo 80    Hi 80      Lo 95    Hi 95
## Jan 1990       11.97945 10.628648 13.33025 9.91357746 14.04532
## Feb 1990       11.71412  9.813929 13.61431 8.80802859 14.62021
## Mar 1990       11.65137  9.005129 14.29761 7.60429365 15.69844
## Apr 1990       11.67419  7.994068 15.35431 6.04593128 17.30244
## May 1990       11.66589  7.327641 16.00414 5.03111134 18.30067
## Jun 1990       11.66891  6.714988 16.62283 4.09254115 19.24527
## Jul 1990       11.66781  6.181574 17.15405 3.27733590 20.05828
## Aug 1990       11.66821  5.691971 17.64445 2.52834221 20.80808
## Sep 1990       11.66806  5.240740 18.09539 1.83831970 21.49781
## Oct 1990       11.66812  4.818780 18.51745 1.19296061 22.14327
## Nov 1990       11.66810  4.421479 18.91472 0.58535163 22.75084
## Dec 1990       11.66810  4.044813 19.29140 0.00928638 23.32692
```

```r
plot(beer.forecast.fit.1)
```



The predictions shows no seasonality, an indication that this model is not the best to use for predicting the beer sales.

Next we will calculate the errors and plot them

```r
cbind(beerdata.test, as.vector(beer.forecast.fit.1$mean))
```

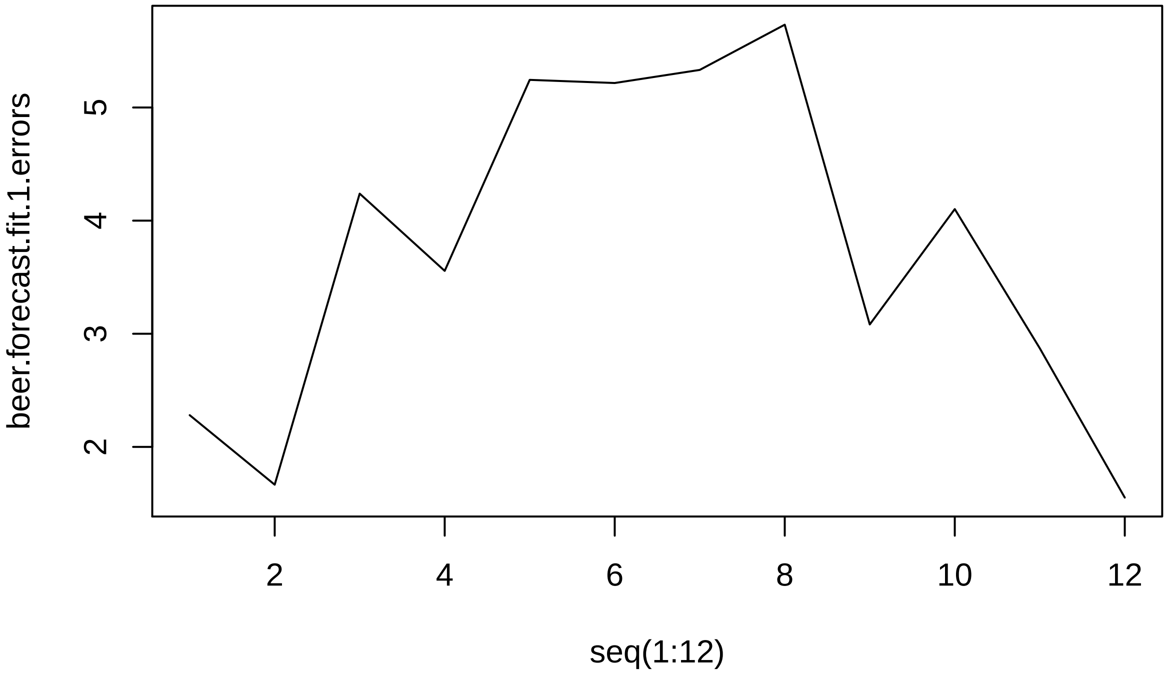
```
##       beerdata.test
##  [1,]       14.2600 11.97945
##  [2,]       13.3800 11.71412
##  [3,]       15.8900 11.65137
##  [4,]       15.2300 11.67419
##  [5,]       16.9100 11.66589
##  [6,]       16.8854 11.66891
##  [7,]       17.0000 11.66781
##  [8,]       17.4000 11.66821
##  [9,]       14.7500 11.66806
## [10,]       15.7700 11.66812
## [11,]       14.5400 11.66810
## [12,]       13.2200 11.66810
```

```r
beer.forecast.fit.1.errors <- beerdata.test - as.vector(beer.forecast.fit.1$mean)

beer.forecast.fit.1.errors
```

```
##  [1] 2.280553 1.665881 4.238631 3.555813 5.244110 5.216493 5.332190
##  [8] 5.731791 3.081936 4.101884 2.871903 1.551896
```

```r
plot(seq(1:12), beer.forecast.fit.1.errors, type = "l")
```



The errors show seasonality

Lastly we will calculate the sum of squared errors

```r
sum(beer.forecast.fit.1.errors^2)
```

```
## [1] 191.564
```

#1B

Use the monthly data as a continuous time series. Forecast for 1990 Jan, Plug forecast into the time series to forecast for 1990 Feb. And so on and so forth. In other words, h=1 in all the forecasts.

```r
h <- 1
n <- length(beerdata.test) - h + 1
fit.2 <- auto.arima(beer.data.train)
fc <- ts(numeric(n), start=1990+(h-1)/12, freq=12)
```

```r
for(i in 1:n)
{
  x <- window(beersales, end=1989 + (i-1)/12)
  refit <- Arima(x, model=fit.2)
  fc[i] <- forecast(refit, h=h)$mean[h]
}
```

```r
plot(fc)
```

In the new prediction we observe seasonality

Lastly we will calculate the sum of squared errors

```
beer.forecast.fit.2.errors <- beerdata.test - as.vector(fc)
beer.forecast.fit.2.errors
```

```
##  [1]  0.75275366 -1.35660359  0.27802641 -1.42168014  0.02848341
##  [6]  0.31588539  0.43896477  2.90267937  0.47235477  2.56664564
## [11]  2.37707852 -0.59601001
```

```
sum(beer.forecast.fit.2.errors^2)
```

```
## [1] 26.04084
```

#1C Which of the two above approaches yield the better results in terms of Mean Squared Error 1990?

The second approach yield better results

#Part 2 Use month of the year seasonal ARIMA(p,d,q)(P,Q,D)s model to forecast beer sales for all the months of 1990.

First we will use auto.arima to determine our parameters

```
auto.arima(beer.data.train, seasonal = TRUE)
```

```
## Series: beer.data.train
## ARIMA(4,1,2)(2,1,2)[12]
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ma1     ma2    sar1    sar2
##       0.5103  -0.1662  0.1032  -0.3966  -1.1757  0.3125  0.6838  -0.592
## s.e.  0.1453   0.0986  0.0863   0.0789   0.1493  0.1421  0.1451   0.165
##          sma1    sma2
##       -1.1967  0.5849
## s.e.   0.1394  0.2087
##
```

```
## sigma^2 estimated as 0.2837:  log likelihood=-134.55
## AIC=291.1   AICc=292.81   BIC=325.4
```

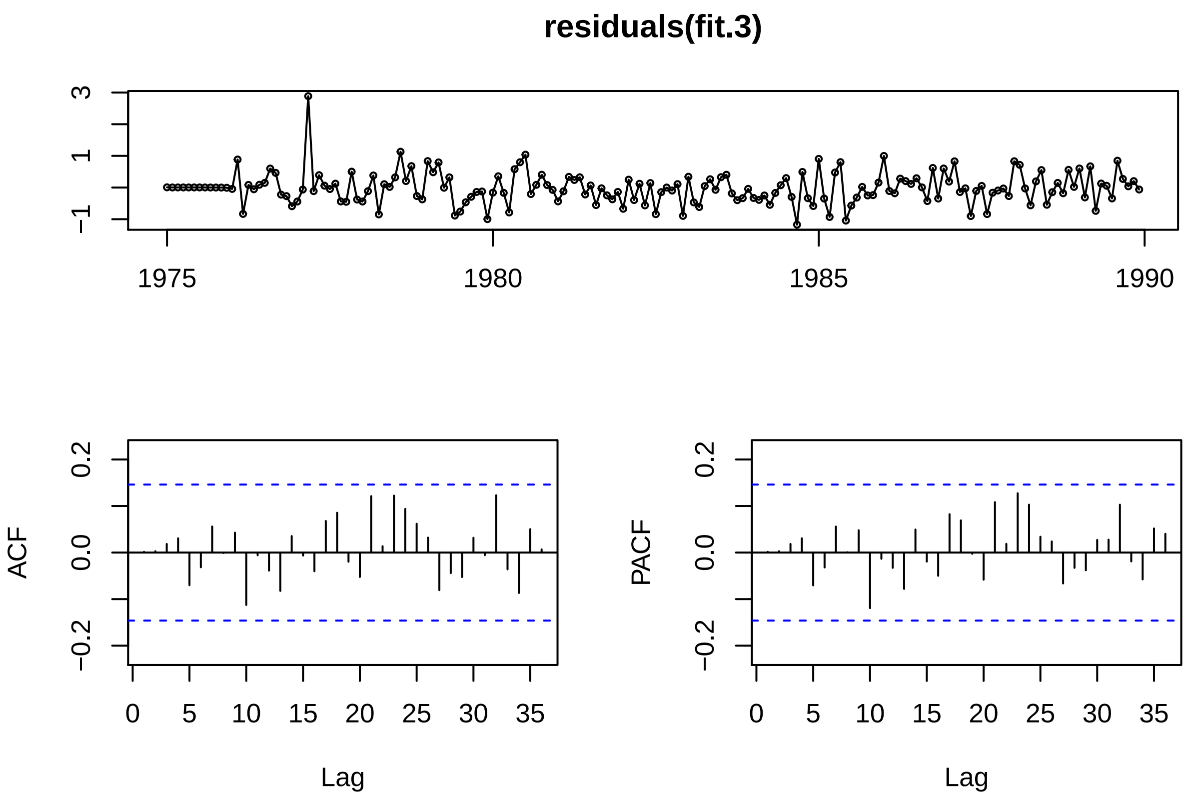
Next we will use suggested parameters to fit our model

```r
fit.3 <- Arima(beer.data.train, order = c(4, 1, 2), seasonal = c(2,1,2)); fit.3
```

```
## Series: beer.data.train
## ARIMA(4,1,2)(2,1,2)[12]
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ma1     ma2    sar1    sar2
##       0.5103  -0.1662  0.1032  -0.3966  -1.1757  0.3125  0.6838  -0.592
## s.e.  0.1453   0.0986  0.0863   0.0789   0.1493  0.1421  0.1451   0.165
##          sma1    sma2
##       -1.1967  0.5849
## s.e.   0.1394  0.2087
##
## sigma^2 estimated as 0.2837:  log likelihood=-134.55
## AIC=291.1   AICc=292.81   BIC=325.4
```

Next we will check our residuals

```r
tsdisplay(residuals(fit.3))
```



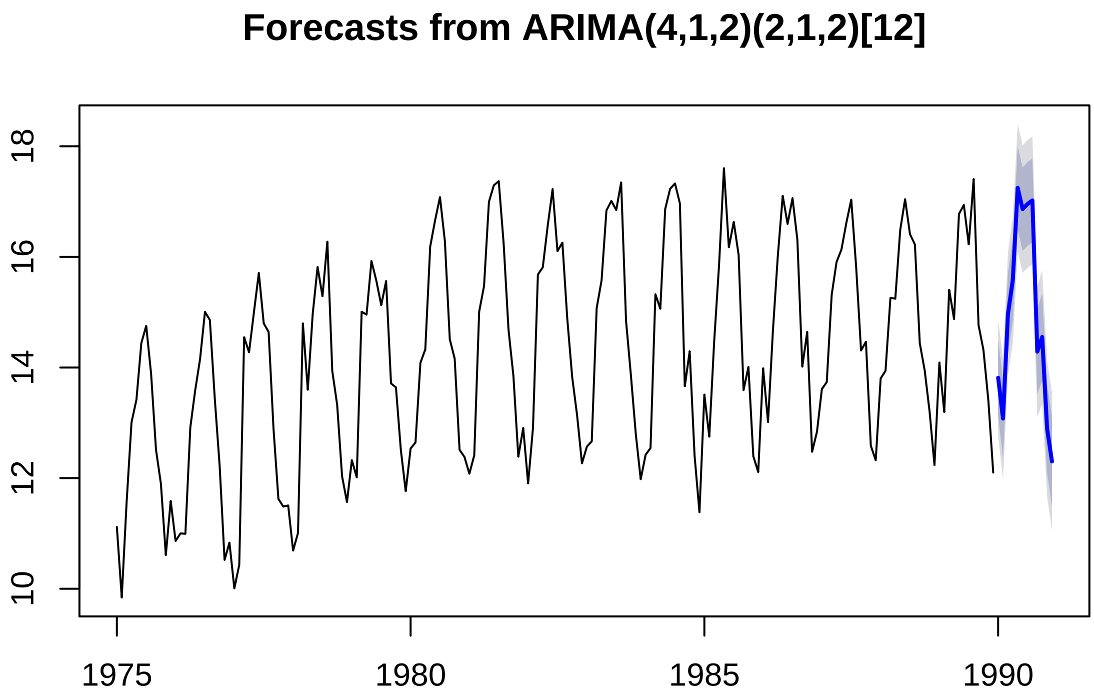

```r
Box.test(residuals(fit.3), lag = 12, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(fit.3)
## X-squared = 5.0383, df = 12, p-value = 0.9567
```

Residuals are not similar to white noise with no autocorrelation. Hence, after including the seasonal parameters into our model we got a better model.

Next we will use our SARIMA model to forecast beer sales for 1990

```
beer.forecast.fit.3 <- forecast(fit.3, h = 12)
plot(beer.forecast.fit.3)
```



Prediction now shows a similar seasonal patern.

Next we will calculate the errors and plot them

```
cbind(beerdata.test, as.vector(beer.forecast.fit.1$mean))
```

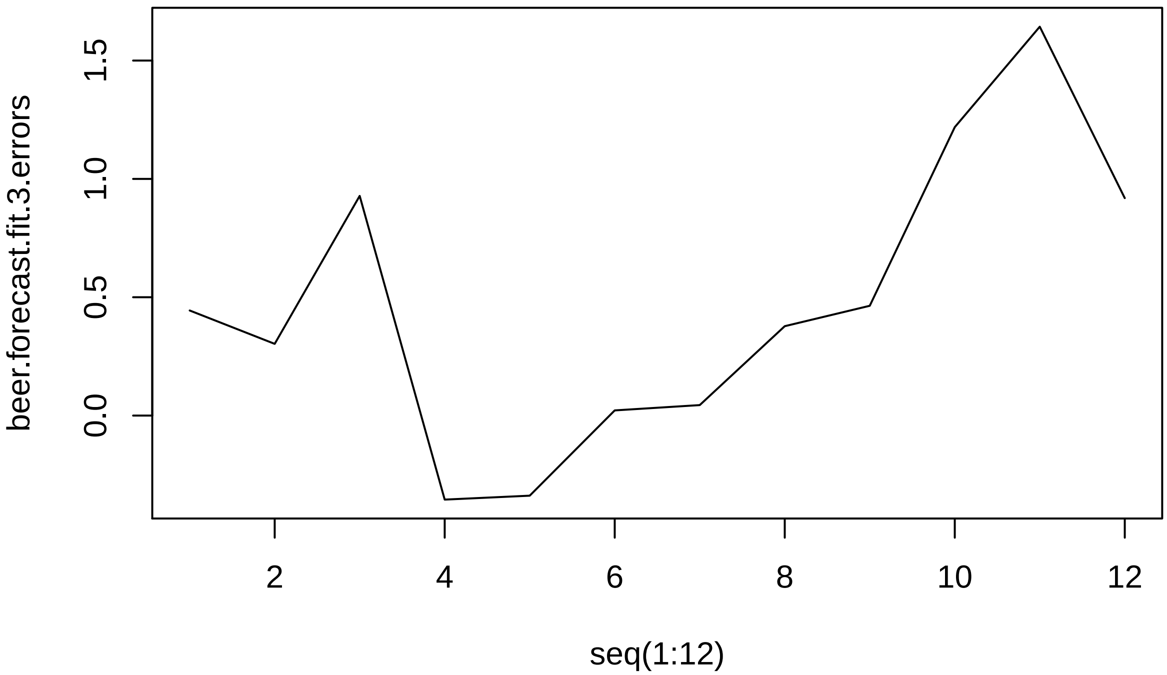
```
##       beerdata.test
##  [1,]       14.2600 11.97945
##  [2,]       13.3800 11.71412
##  [3,]       15.8900 11.65137
##  [4,]       15.2300 11.67419
##  [5,]       16.9100 11.66589
##  [6,]       16.8854 11.66891
##  [7,]       17.0000 11.66781
##  [8,]       17.4000 11.66821
##  [9,]       14.7500 11.66806
## [10,]       15.7700 11.66812
## [11,]       14.5400 11.66810
## [12,]       13.2200 11.66810
```

```
beer.forecast.fit.3.errors <- beerdata.test - as.vector(beer.forecast.fit.3$mean)

beer.forecast.fit.3.errors
```

```
##  [1]  0.44398999  0.30293424  0.92818610 -0.35502719 -0.33847037
##  [6]  0.02180024  0.04428818  0.37768567  0.46381099  1.21864480
## [11]  1.64305375  0.91873134
```

```r
plot(seq(1:12), beer.forecast.fit.3.errors, type = "l")
```



The errors show seasonality

Lastly we will calculate the sum of squared errors

```r
sum(beer.forecast.fit.3.errors^2)
```

```
## [1] 6.780024
```

#Part 3

Which model (Part 1 or Part 2) is better to forecast beer sales for each month of 1990 (Jan, Feb, . . . , Dec) ?

In terms of Mean Squared Error the last model (Part 2) is better to forecast beer sales.