Week 05 Homework

Brian Ritz

Monday, May 18, 2015

Load data from TSA package (the package is written by textbook 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.

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.

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.

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

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.

Part 3 - Which model (Part 1 or Part 2) is better to forecast beer sales for each month of 1990 (Jan, Feb, …, Dec) ?

Due date: 2015 Feb 19th 11:59pm

```r
# load the dataset
library(forecast, quietly=T)
```

```
## Warning: package 'forecast' was built under R version 3.1.3
```

```
## Warning: package 'zoo' was built under R version 3.1.3
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Warning: package 'timeDate' was built under R version 3.1.3
```

```
## This is forecast 5.9
```

```r
library("TSA", quietly=T)
```

```
## Warning: package 'TSA' was built under R version 3.1.3
```

```
## Warning: package 'leaps' was built under R version 3.1.3
```

```
## Warning: package 'locfit' was built under R version 3.1.3
```

```
## locfit 1.5-9.1    2013-03-22
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:forecast':
##
##     getResponse
##
## This is mgcv 1.8-3. For overview type 'help("mgcv-package")'.
```

```
## Warning: package 'tseries' was built under R version 3.1.3
```

```
##
## Attaching package: 'TSA'
##
## The following object is masked from 'package:forecast':
##
##     fitted.Arima
##
## The following objects are masked from 'package:timeDate':
##
##     kurtosis, skewness
##
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

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

```
data(beersales)
```

Create an ARIMA Model

```
train <- window(beersales, start=c(1975,1), end=c(1989,12))
beersales.1990 <- window(beersales, start=c(1990, 1), end=c(1990, 12))
```

adf test checks for stationarity

```
print(adf.test(train))
```

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

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

We see that it is stationary because the p-value on the adf test is 0.01. Therefore, we can restrict our model to a stationary model.

```
# we have approximation = F to get rid of warnings that the algorithm could not fit a max likelihood model
fit <- auto.arima(train, max.P=0, max.D=0, max.Q=0, stationary = T, seasonal = F, approximation = F)
```

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

```
# the h parameter gives the number of periods forward to forecast
(forecast.1990.arima <- forecast(fit, h = 12))
```

```
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Jan 1990       12.76178 11.77716 13.74640 11.255938 14.26763
## Feb 1990       11.88259 10.69047 13.07470 10.059395 13.70577
## Mar 1990       13.59564 12.19437 14.99691 11.452578 15.73869
## Apr 1990       13.82764 12.16897 15.48631 11.290916 16.36436
## May 1990       14.89512 13.19085 16.59939 12.288663 17.50158
## Jun 1990       15.80365 14.07849 17.52882 13.165239 18.44207
## Jul 1990       15.35292 13.60664 17.09920 12.682220 18.02362
## Aug 1990       16.03725 14.22259 17.85192 13.261962 18.81254
## Sep 1990       14.64734 12.73998 16.55470 11.730284 17.56440
## Oct 1990       14.59559 12.56857 16.62262 11.495531 17.69566
## Nov 1990       13.61000 11.55917 15.66082 10.473530 16.74647
## Dec 1990       13.03485 10.97132 15.09837  9.878957 16.19074
```

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.

```
train.loop <- train
forecasts.loop <- c()
```

```
h<-0
while(h < 12){
  fit.loop <- auto.arima(train.loop, max.P=0, max.D=0, max.Q=0, stationary = T, seasonal = F, approximation
=F)
  one.mo.forecast <- forecast(fit.loop, h = 1)

  # log the forecast we just made
  forecasts.loop <- c(forecasts.loop, one.mo.forecast$mean)

  # add the forecast to the training dataset
  train.loop <- ts(c(train.loop, one.mo.forecast$mean), start=start(train.loop), frequency=frequency(train.
loop))

  # use the length of the forecasts to exit the loop if its time
  h = length(forecasts.loop)
}

(forecasts.loop <- ts(forecasts.loop, start=c(1990, 1), frequency=12))
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 1990 12.76178 11.88229 14.41447 15.42172 16.09008 17.30622 16.54417
##           Aug      Sep      Oct      Nov      Dec
## 1990 16.23178 14.84159 13.55763 12.59556 11.91995
```

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

```
mse <- function(actual, forecast){
  errors <- as.vector(actual) - as.vector(forecast)
```

```
  return (sum(errors**2)/length(errors))
}

forecast.error.yearly <- mse(beersales.1990, forecast.1990.arima$mean)
forecast.error.monthly <- mse(beersales.1990, forecasts.loop)

print(forecast.error.yearly)
```

```
## [1] 1.983871
```

```
print(forecast.error.monthly)
```

```
## [1] 1.624796
```

The monthly continuous estimation gave a lower mean squared error.

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.

```
summary(seasonality.fit <- auto.arima(train, seasonal = T, stationary = T, approximation=F))
```

```
## Series: train
## ARIMA(4,0,4)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3     ar4      ma1      ma2     ma3      ma4
##       1.4349  -0.2208  -1.0526  0.8227  -1.1726  -0.0139  1.1750  -0.9065
```

```
## s.e.  0.0527   0.1031   0.1031  0.0502   0.0461   0.0515  0.0434   0.0508
##         sar1  intercept
##       0.9081    13.6554
## s.e.  0.0275     1.4075
##
## sigma^2 estimated as 0.3156:  log likelihood=-163.75
## AIC=349.51   AICc=351.08   BIC=384.63
##
## Training set error measures:
##                     ME      RMSE       MAE       MPE    MAPE      MASE
## Training set 0.0569171 0.5617764 0.4271958 0.2286961 3.05659 0.4338746
##                    ACF1
## Training set -0.0276574
```

```
(seasonality.forecast <- forecast(seasonality.fit, h=12))
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1990       13.89784 13.17777 14.61790 12.79658 14.99909
## Feb 1990       13.47219 12.72773 14.21666 12.33364 14.61075
## Mar 1990       15.31135 14.55994 16.06277 14.16216 16.46054
## Apr 1990       15.12000 14.34429 15.89571 13.93366 16.30634
## May 1990       16.61369 15.83797 17.38941 15.42732 17.80005
## Jun 1990       16.85835 16.08263 17.63407 15.67198 18.04472
## Jul 1990       15.96027 15.17526 16.74528 14.75971 17.16084
## Aug 1990       17.13487 16.34983 17.91992 15.93425 18.33550
## Sep 1990       14.64374 13.85860 15.42889 13.44297 15.84452
## Oct 1990       14.42170 13.62335 15.22005 13.20073 15.64267
```

```
## Nov 1990       13.57073 12.76420 14.37725 12.33725 14.80420
## Dec 1990       12.49404 11.67969 13.30840 11.24859 13.73949
```

Part 3 - Which model (Part 1 or Part 2) is better to forecast beer sales for each month of 1990 (Jan, Feb, ..., Dec) ?

We will check the mse of part 2, and compare it to part 1:

```
forecast.error.seasonality <- mse(beersales.1990, seasonality.forecast$mean)



print(forecast.error.yearly)
```

```
## [1] 1.983871
```

```
print(forecast.error.monthly)
```

```
## [1] 1.624796
```

```
print(forecast.error.seasonality)
```

```
## [1] 0.4185143
```

Using the seasonality decrease the mean squared error by half! The seasonality model is much better for forecasting beer sales for 1990.

Break it down by forecasts for each month:

```
forecasts.and.actuals <- ts(data.frame(actual = beersales.1990, yearly=forecast.1990.arima$mean, monthly=forecasts.loop, seasonal=seasonality.forecast$mean), start=c(1990, 1), frequency=12)

(all.errors <- apply(forecasts.and.actuals, FUN=function(x) x-forecasts.and.actuals[,"actual"], MARGIN=2))
```

```
##       actual     yearly     monthly    seasonal
##  [1,]      0 -1.4982171 -1.49821706 -0.36216428
##  [2,]      0 -1.4974150 -1.49771271  0.09219411
##  [3,]      0 -2.2943647 -1.47553276 -0.57864913
##  [4,]      0 -1.4023605  0.19172052 -0.10999928
##  [5,]      0 -2.0148800 -0.81991877 -0.29631131
##  [6,]      0 -1.0817468  0.42082340 -0.02705021
##  [7,]      0 -1.6470803 -0.45583339 -1.03972932
##  [8,]      0 -1.3627478 -1.16822040 -0.26512570
##  [9,]      0 -0.1026580  0.09159016 -0.10625590
## [10,]      0 -1.1744063 -2.21236677 -1.34830290
## [11,]      0 -0.9300016 -1.94443760 -0.96927236
## [12,]      0 -0.1851525 -1.30004996 -0.72595675
```

The seasonal forecasts are usually always better than the non-seasonal forecasts, with the exception of May, July, and December.