Assignment_5

Weijie Gao

11/2/2017

```
suppressWarnings(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 1.8-17. For overview type 'help("mgcv-package")'.

## Loading required package: tseries

##
## Attaching package: 'TSA'

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

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

suppressWarnings(library(forecast))

##
## Attaching package: 'forecast'

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

suppressWarnings(library(tseries))
data(beersales)
```

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.

```
# separate data into training and testing data
beersales_train <- beersales[1:(192-12)]
```

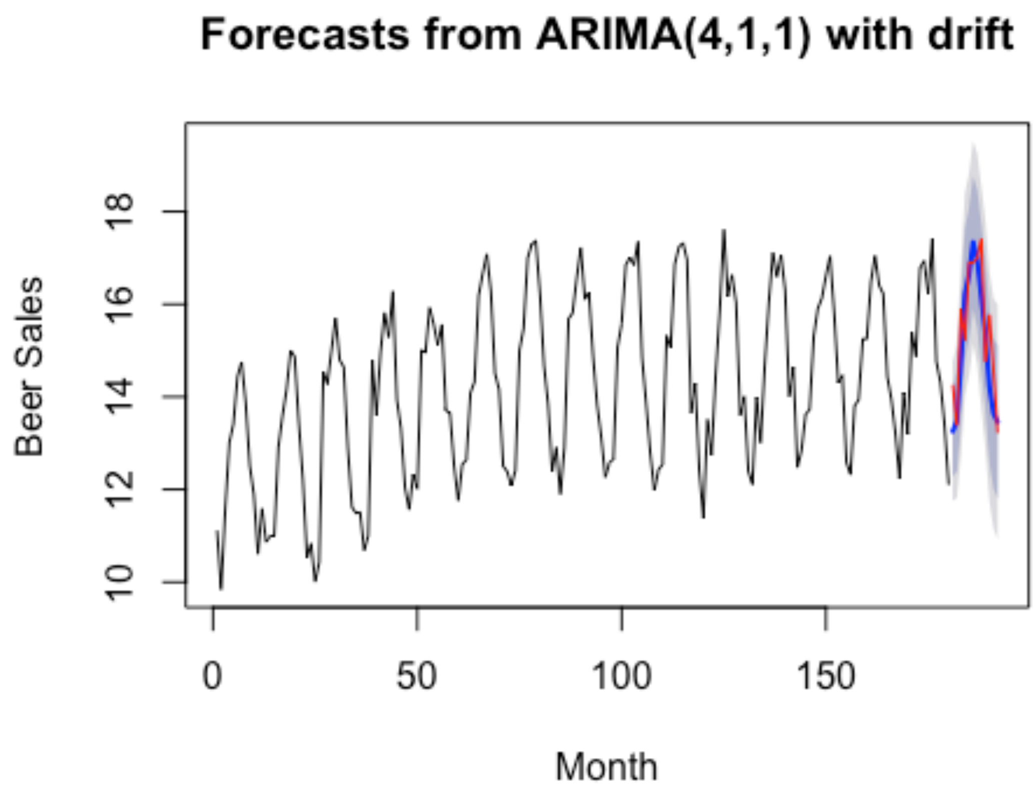
```r
beersales_test <- tail(beersales, 12)
# fit ARIMA(p,d,q) model by h-period to forecast each month of 1990
fit_1a <- auto.arima(beersales_train, stepwise=FALSE, approximation=FALSE)
summary(fit_1a)
```

```
## Series: beersales_train
## ARIMA(4,1,1) with drift
##
## Coefficients:
##          ar1     ar2      ar3     ar4      ma1   drift
##       0.4179  0.4520  -0.0148  -0.667  -0.9237  0.0147
## s.e.  0.0561  0.0651   0.0642   0.056   0.0215  0.0057
##
## sigma^2 estimated as 0.5797:  log likelihood=-204.86
## AIC=423.72   AICc=424.37   BIC=446.03
##
## Training set error measures:
##                       ME     RMSE       MAE        MPE    MAPE      MASE
## Training set 0.008156097 0.746397 0.5785693 -0.1677262 4.13729 0.5876146
##                    ACF1
## Training set -0.1705585
```

```r
# forecast for ARIMA(4,1,1) based on auto.arima() result
(fit_1a_forecast <- forecast(fit_1a, 12)$mean[1:12])
```

```
##  [1] 13.25489 13.47449 14.72775 16.21593 16.64261 17.34046 16.97896
##  [8] 16.15635 15.36622 14.21607 13.64353 13.45672
```

```r
# plot forecast
plot(forecast(fit_1a, 12),xlab = "Month", ylab = "Beer Sales")
lines(x = c(181:192), y = beersales_test, col = "red")
```



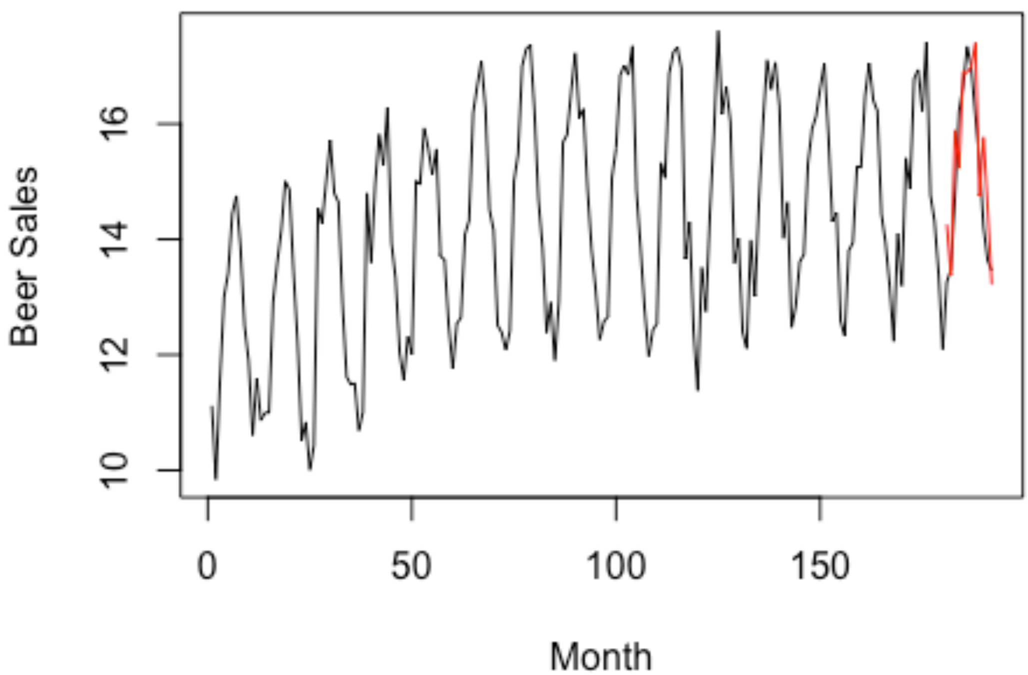
Forecast results for each month of 1990 are: 13.25489, 13.47449, 14.72775, 16.21593, 16.64261, 17.34046 16.97896, 16.15635, 15.36622, 14.21607, 13.64353, 13.45672

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.

```
fit_1b_forecast<-rep(0,12)
fit_1b_forecast[1] <- forecast(fit_1a, 1)$mean[1]
for (h in 2:12){
beersales_1b <- c(beersales_train, fit_1b_forecast[1:h-1])
fit_1b <- auto.arima(beersales_1b, stepwise = FALSE, approximation = FALSE)
fit_1b_forecast[h]<-forecast(fit_1b, 1)$mean[1]
}
fit_1b_forecast
```

```
##  [1] 13.25489 13.47452 14.72801 16.21675 16.64374 17.34224 16.98067
##  [8] 16.15734 15.36634 14.21431 13.64056 13.45309
```

```
# plot forecast
matplot(x = 1:192, y = c(beersales_train,fit_1b_forecast), xlab = "Month",
ylab = "Beer Sales", type = "l")
lines(x = c(181:192), y = beersales_test, col = "red")
```



Forecast results for each month of 1990 are:
13.25489,13.47452,14.72801,16.21675,16.64374,17.34224
16.98067,16.15734,15.36634,14.21431,13.64056,13.45309

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

```
# compare forecast results for each month of 1990
cbind(forecast_1a = fit_1a_forecast, forecast_1b = fit_1b_forecast)

##       forecast_1a forecast_1b
##  [1,]    13.25489    13.25489
##  [2,]    13.47449    13.47452
##  [3,]    14.72775    14.72801
##  [4,]    16.21593    16.21675
##  [5,]    16.64261    16.64374
##  [6,]    17.34046    17.34224
##  [7,]    16.97896    16.98067
##  [8,]    16.15635    16.15734
##  [9,]    15.36622    15.36634
## [10,]    14.21607    14.21431
## [11,]    13.64353    13.64056
## [12,]    13.45672    13.45309
```

```r
# compare MSE
mse_1a <- mean((fit_1a_forecast - beersales_test)^2)
mse_1b <- mean((fit_1b_forecast - beersales_test)^2)
cbind(mse_1a = mse_1a, mse_1b=mse_1b)
```

```
##         mse_1a    mse_1b
## [1,] 0.7351533 0.7358846
```

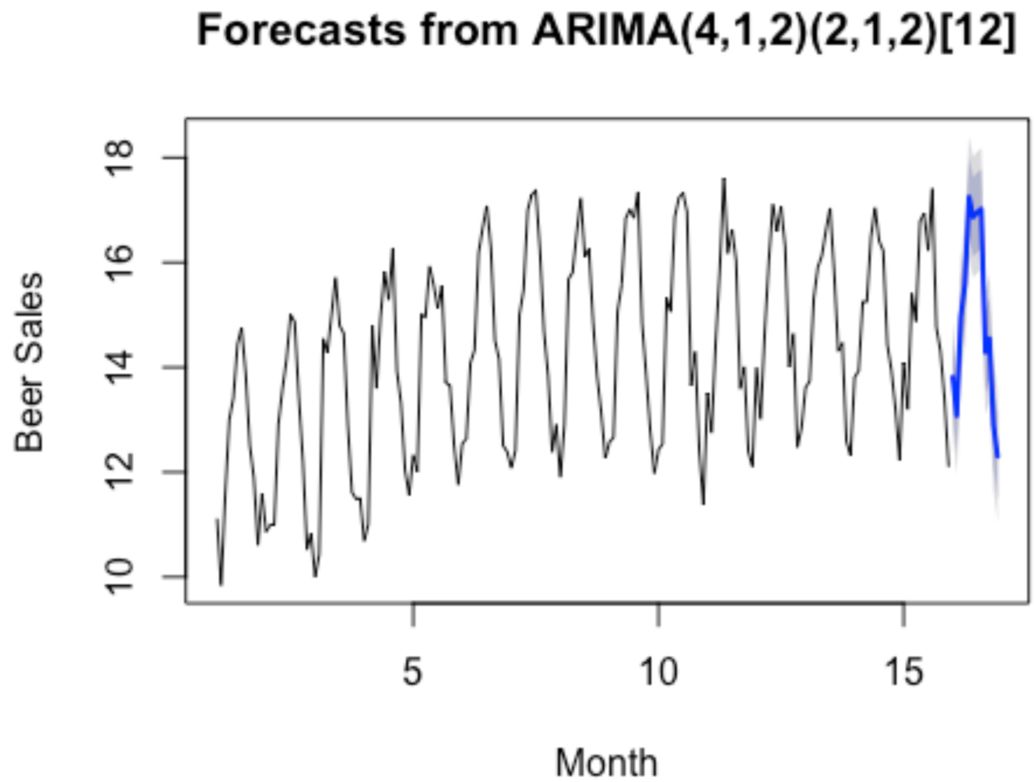
Comparing the Measn Squared Error in two approaches, we see that results are pretty close. MSE of 0.7351533 from Part 1A approach is slightly smaller than that from Part 1B approach. In terms of MSE 1990, I would say the approach in Part 1A ields better results.

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.

```r
# fit ARIMA(p,d,q)(P,Q,D)s model
fit_2 <- auto.arima(ts(beersales_train, frequency = 12))
# forecast beer sales for all the months of 1990
(fit_2_forecast <- forecast(fit_2, 12)$mean[1:12])
```

```
##  [1] 13.81601 13.07707 14.96181 15.58503 17.24847 16.86360 16.95571
##  [8] 17.02231 14.28619 14.55136 12.89695 12.30127
```

```r
# plot forecast
plot(forecast(fit_2,12),xlab = "Month", ylab = "Beer Sales")
```



Forecast results for each month of 1990 are: 13.81601, 13.07707, 14.96181, 15.58503, 17.24847, 16.86360 16.95571, 17.02231, 14.28619, 14.55136, 12.89695, 12.30127

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

```
# compare forecast
cbind(forecast_1a = fit_1a_forecast, forecast_1b = fit_1b_forecast,
forecase_2=fit_2_forecast)

##       forecast_1a forecast_1b forecase_2
##  [1,]    13.25489    13.25489   13.81601
##  [2,]    13.47449    13.47452   13.07707
##  [3,]    14.72775    14.72801   14.96181
##  [4,]    16.21593    16.21675   15.58503
##  [5,]    16.64261    16.64374   17.24847
##  [6,]    17.34046    17.34224   16.86360
##  [7,]    16.97896    16.98067   16.95571
##  [8,]    16.15635    16.15734   17.02231
##  [9,]    15.36622    15.36634   14.28619
## [10,]    14.21607    14.21431   14.55136
## [11,]    13.64353    13.64056   12.89695
## [12,]    13.45672    13.45309   12.30127
```

```r
#---- compare MSE ----#
mse_1a <- mean((fit_1a_forecast - beersales_test)^2)
mse_1b <- mean((fit_1b_forecast - beersales_test)^2)
mse_2 <- mean((fit_2_forecast - beersales_test)^2)
cbind(mse_1a = mse_1a, mse_1b=mse_1b, mse_2=mse_2)
```

```
##         mse_1a    mse_1b     mse_2
## [1,] 0.7351533 0.7358846 0.5650021
```

Comparing forecast results for each month of 1990, results look pretty close from Part 1 and Part 2. In terms of Mean Squared Error, we can see that approach from Part 2 yields a significantly smaller MSE than Part 2. MSE from Part 2 is only 0.565 while MSE from Part 1 are around 0.735. So model in Part 2 is better to forecast beer sales for each month of 1990.