Time Series Modeling

Seasonal ARIMA Model

Contents

Seasonal Box-Jenkins (ARIMA) Models

- ARIMA (Auto Regressive Integrated Moving Average) models are **Regression models that use lagged values of the dependent variable and/or random disturbance term as explanatory variables.**

- **Seasonal ARIMA** (Often abbreviated as **SARIMA**) Model is **formed by including seasonal terms in the ARIMA model.**

- Several real world time series have a seasonal component. Some examples are: Sales of woolen clothes, demand for fertilizers, electricity consumption, etc.

Seasonal Box-Jenkins (ARIMA) Models

- The **seasonal ARIMA model** incorporates both **non-seasonal and seasonal** factors in a multiplicative model.

- Shorthand notation for the model is,

$$\text{ARIMA}\ (p, d, q) \times (P, D, Q)S,$$

with,
p = non-seasonal AR order,
d = non-seasonal differencing,
q = non-seasonal MA order,
P = seasonal AR order,
D = seasonal differencing,
Q = seasonal MA order, and
S = time span of repeating seasonal pattern.

Five-Step Iterative Procedure

1. Stationarity Checking and Seasonal Differencing
2. Model Identification
3. Parameter Estimation
4. Diagnostic Checking
5. Forecasting

Step 1: Stationarity Checking

Assessing Stationarity of Time Series

- Stationarity of a time series can be assessed using:

Time Series Plot (Time v/s Variable) OR Correlogram OR Dickey-Fuller Test

- If a time series is non-stationary then it can be converted via

Differencing

De-trending

Seasonal Differencing

- Seasonal differencing is denoted as ,

$$\boxed{\Delta_s y_t = y_t - y_{t-s}}$$

Where,

s denotes **frequency of season**

s = 12 if data is monthly; s = 4 if data is quarterly and so on

- First and seasonal span differencing for monthly data is,

$$\boxed{\Delta_1 \Delta_s y_t = \Delta_1 (y_t - y_{t-s}) = y_t - y_{t-1} - y_{t-s} + y_{t-s-1}}$$

Case Study

Background

- Sales Data for 3 Years (2013, 2014, 2015)

Objective

- To develop seasonal ARIMA Model for generating forecasts

Available Information

- Sample size is 36
- Variables: Year, Month, Sales

Data Snapshot

Sales Data for 3 Years

Variables

Monthly Observations

Year	Month	Sales
2013	Jan	123
2013	Feb	142
2013	Mar	164
2013	Apr	173
2013	May	183
2013	Jun	192
2013	Jul	199
2013	Aug	203
2013	Sep	207
2013	Oct	209
2013	Nov	214
2014	Jul	245

Columns	Description	Type	Measurement	Possible values
Year	Year	nemeric	2013, 2014, 2015	3
Month	Month	factor	Jan - Dec	12
Sales	Sales in USD Million	numeric	USD Million	Positive values

Plotting a Time Series in R

Importing the Data

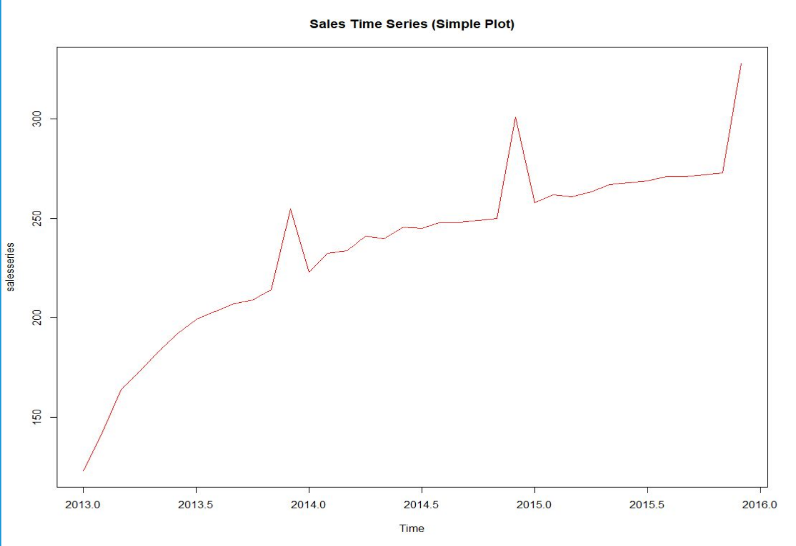
```
salesdata<-read.csv("Sales Data for 3 Years.csv",header=TRUE)
```

#Creating and Plotting a Time Series Object

```
salesseries<-ts(salesdata$Sales,start=c(2013,1),end=c(2015,12),
                frequency=12)

plot(salesseries,col="red",main="Sales Time Series (Simple Plot)")
```

Output



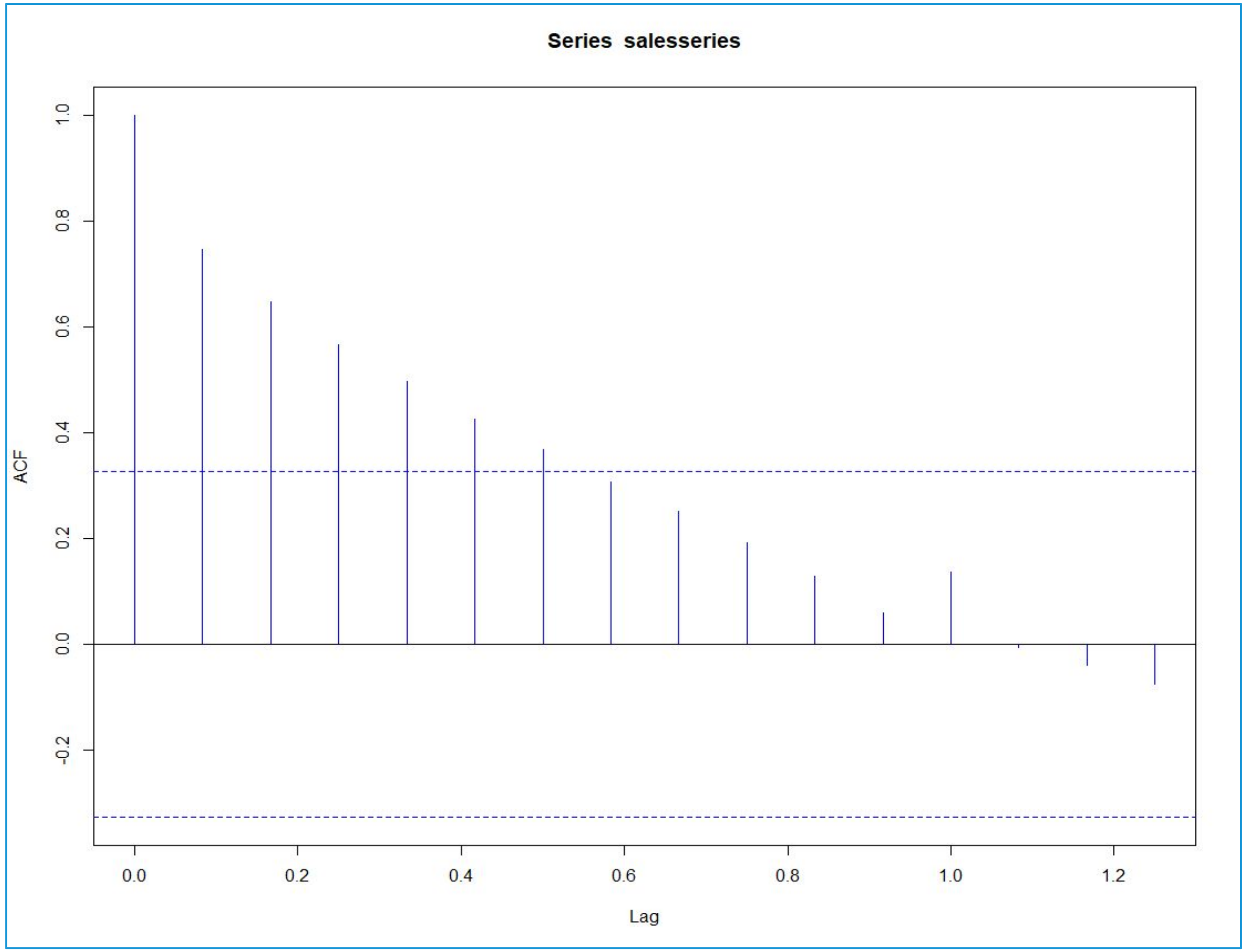
Interpretation :

- The time series shows periodic peaks, indicative of seasonality.

Correlogram

```
acf(salesseries,col="blue")
```

```
# Output
```



Interpretation :

- ACF plot shows a slow decay indicating non-stationarity.

Determining the Order of Differencing and Dickey Fuller Test For Original Series

```
# Better Way of Determining the Order of Differencing
```

```
ndiffs(salesseries)
[1] 1
```

```
# Dickey Fuller Test
```

```
df<-ur.df(salesseries, lag=0)
summary(df)
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-49.216  -4.668  -1.959   3.965  49.363

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
z.lag.1  0.02065    0.01274   1.621    0.114

Residual standard error: 17.82 on 34 degrees of freedom
Multiple R-squared:  0.07176,   Adjusted R-squared:  0.04446
F-statistic: 2.628 on 1 and 34 DF,  p-value: 0.1142
```

Interpretation :

- Time series is non-stationary. Value of test statistic is greater than 5% critical value.

*

nsdiffs() function in forecast package uses seasonal unit root tests to determine the number of seasonal differences required for time series x to be made stationary. However, in most cases seasonal differencing of the first order is enough and hence not much focus is put on checking order of differencing.

Dickey Fuller Test – Differenced Series

```
# Dickey Fuller Test for Difference Series
salesdiff <- diff(salesseries,differences = 1)
summary(ur.df(salesdiff,lags = 0))
```

```
# Output
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-26.753   1.097   2.598   7.343  55.287 

Coefficients:
        Estimate Std. Error t value Pr(>|t|)    
z.lag.1  -1.3186     0.1913  -6.891 7.18e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.75 on 33 degrees of freedom
Multiple R-squared:   0.59,	Adjusted R-squared:  0.5776 
F-statistic: 47.49 on 1 and 33 DF,  p-value: 7.184e-08


Value of test-statistic is: -6.8914 

Critical values for test statistics: 
      1pct  5pct 10pct
tau1 -2.62 -1.95 -1.61
```

Interpretation :

- Time series is stationary. Value of test statistic is less than 5% critical value.

Step 2: Model Identification

Model Identification

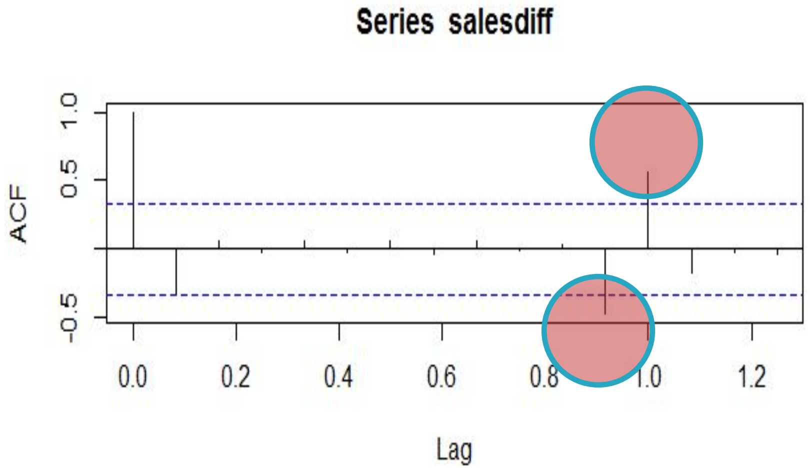
- When the data are confirmed stationary, proceed to tentative identification of models through **visual inspection of correlogram and partial correlogram**

Model	AC	PAC
AR (p) $y_t = \delta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t$	Dies down	Cuts off after lag p
MA (q) $y_t = \delta + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$	Cuts off after lag q	Dies down
ARMA (p,q) $y_t = \delta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$	Dies down	Dies down

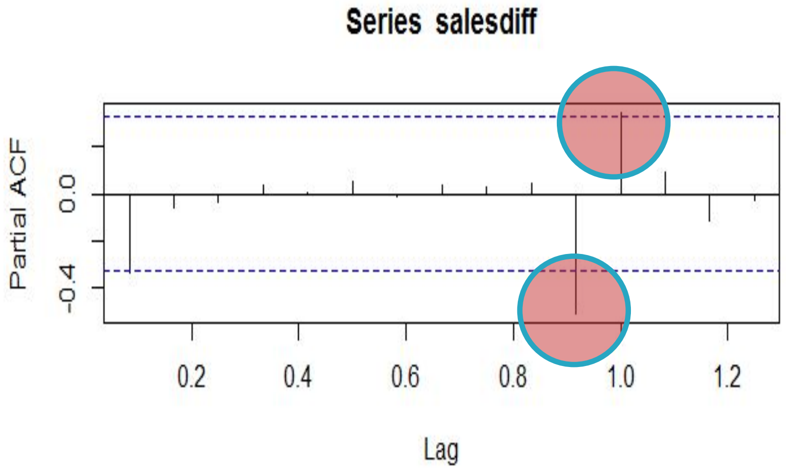
Model Identification

- Seasonal ARIMA model is expressed as arima(p,d,q) (P,D,Q) where
  - p = no. of autoregressive terms
  - d = order of differencing
  - q = no. of moving average terms
  - (P,D,Q) are seasonal equivalents of autoregressive, difference and moving average terms

ACF Plot



PACF Plot



Indicative Model : arima(2,1,2)(2,1,2)

Step 3: Parameter Estimation

Parameter Estimation

- There are two ways in which parameters of arima models can be estimated

1. Ordinary Least Squares
2. Maximum Likelihood Method – when the model involves MA component

- Given n observations y_1, y_2, …, y_n, the likelihood function **L is defined as - the probability of obtaining the data actually observed**
- The maximum likelihood estimators **(MLE) are those values of the parameters for which the data actually observed are most likely**, that is, the values that maximize the likelihood function L.

Parameter Estimation in R

```
# Install and load package "forecast"
# Automatic Model Identification and Parameter Estimation
```

```
install.packages("forecast")
library(forecast)

auto.arima(salesseries,d=1,D=1,max.p=2,max.q=2,max.P=2,max.Q=2,
           trace=TRUE,ic="aic")
```

- **auto.arima()** generates the best order arima model. The function conducts a search over possible model within the order constraints provided.
- Seasonal model requires **max.D,max.P** and **max.Q** arguments as well.
- **trace=** TRUE returns the list of all models considered.
- **ic=** specifies the information criterion. We have specified it as **"aic"**.

Automatic Model Identification

Output

```
> auto.arima(salesseries,d=1,D=1,max.p=2,max.q=2,max.P=2,max.Q=2,
+                trace=TRUE,ic="aic")

 ARIMA(2,1,2)(1,1,1)[12]                    : 157.0397
 ARIMA(0,1,0)(0,1,0)[12]                    : 156.1096
 ARIMA(1,1,0)(1,1,0)[12]                    : 159.1271
 ARIMA(0,1,1)(0,1,1)[12]                    : 159.4057
 ARIMA(0,1,0)(1,1,0)[12]                    : 157.6536
 ARIMA(0,1,0)(0,1,1)[12]                    : 157.6536
 ARIMA(0,1,0)(1,1,1)[12]                    : 159.6536
 ARIMA(1,1,0)(0,1,0)[12]                    : 157.1806
 ARIMA(0,1,1)(0,1,0)[12]                    : 157.6069
 ARIMA(1,1,1)(0,1,0)[12]                    : 154.6016
 ARIMA(1,1,1)(1,1,0)[12]                    : 161.3531
 ARIMA(1,1,1)(0,1,1)[12]                    : 156.6
 ARIMA(1,1,1)(1,1,1)[12]                    : 163.329
 ARIMA(2,1,1)(0,1,0)[12]                    : 152.5563
 ARIMA(2,1,1)(1,1,0)[12]                    : 153.1015
 ARIMA(2,1,1)(0,1,1)[12]                    : Inf
 ARIMA(2,1,1)(1,1,1)[12]                    : 155.0459
 ARIMA(2,1,0)(0,1,0)[12]                    : 151.6156
 ARIMA(2,1,0)(1,1,0)[12]                    : 151.8961
 ARIMA(2,1,0)(0,1,1)[12]                    : Inf
 ARIMA(2,1,0)(1,1,1)[12]                    : 153.864

 Best model: ARIMA(2,1,0)(0,1,0)[12]

Series: salesseries
ARIMA(2,1,0)(0,1,0)[12]

Coefficients:
         ar1     ar2
      0.1583  0.6353
s.e.  0.1545  0.1856

sigma^2 estimated as 34.14:  log likelihood=-72.81
AIC=151.62   AICc=152.88   BIC=155.02
```

Interpretation :

- Model with the lowest AIC value is selected as the best model.

Using BEST order in arima Function

```
# Obtaining Coefficient
```

```
salesmodel<-arima(salesseries,order=c(2,1,0),
seasonal=list(order=c(0,1,0),period=12))

coef(salesmodel)
ar1       ar2
0.1583489 0.6352769
```

Model Selection Criteria

- **Akaike Information Criterion (AIC)**

$$\boxed{AIC = -2\ \ln(L) + 2k}$$

where L = Likelihood function

k = Number of parameters to be estimated

Ideally, AIC should be as small as possible

Step 4: Diagnostic Checking

Residual Analysis

If an ARMA(p,q) model is an adequate representation of the data generating process then the residuals should be 'White Noise'

- White Noise time series has **zero mean, constant variance and zero covariance with lagged time series.**

- **Box-Pierce Test (Q Statistic) is the most recommended method for checking if the residuals are white noise process.**

- Ljung-Box test is also used for the same purpose.

Box Pierce Test

Objective	To test the **null hypothesis** that **e_t is a white noise process**

Test Statistic	$$Q_{BP} = T\sum_{\tau=1}^{m}\hat{\rho}^2(\tau) \sim \chi^2(m)$$ for large T (based on autocorrelations upto lag m and T observations in a time series)
Decision Criteria	Reject the null hypothesis **if p-value < 0.05**

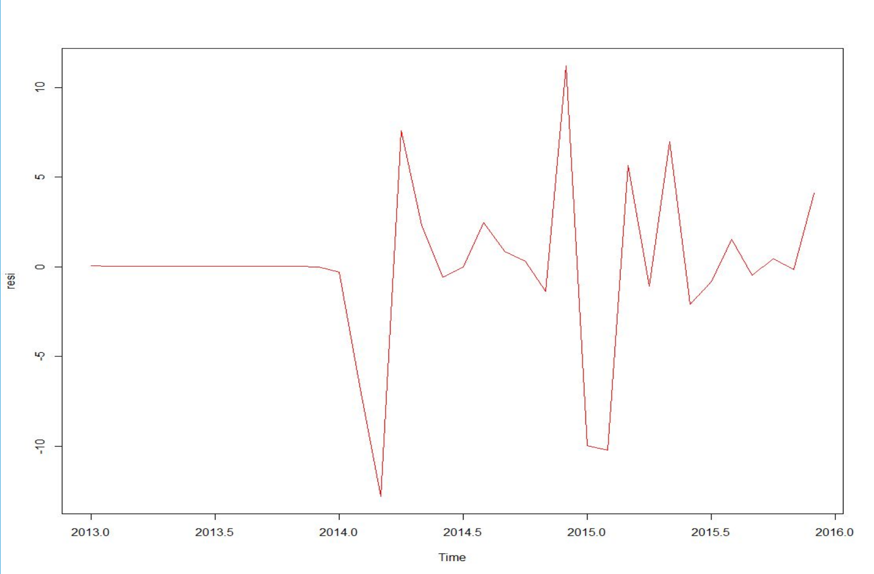
Box-Pierce Test in R

```
# Box Test
resi<-residuals(salesmodel)
Box.test(resi)
```

```
# Output
        Box-Pierce test

data:  resi
X-squared = 0.78552, df = 1, p-value = 0.3755
```

```
plot(resi,col="red")
```



Interpretation :

- Do not reject Ho, as p-value is greater than 0.05.
- Errors follow white noise process.

Step 5: Forecasting

Forecasting

```
# Forecast for next 3years
predict(salesmodel,n.ahead=3)
```

predict() function is used to forecast sales for next 3 periods

```
# Output
> predict(salesmodel,n.ahead=3)
$pred
          Jan      Feb      Mar
2016 285.6334 292.1748 292.0954

$se
           Jan       Feb       Mar
2016  5.582376  8.542627 13.268502
```

Next 3 period sales forecasts

Quick Recap

Stationarity Checking	• Use `ndiffs()` to determine order of differencing • Plot correlogram using `acf()` and validate stationarity using `ur.df()`
Model Identification	• Tentative identification of models through visual inspection of correlogram and partial correlogram
Parameter Estimation	• `auto.arima()` is recommended for obtaining best ARIMA model • It uses AIC as the model selection criteria
Diagnostic Checking	• `Box.test()` performs a Box-Pierce test for checking whether errors follow white noise process
Forecasting	• Use `predict()` to generate forecasts