Time Series Modeling –

ARIMA Model

Contents

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.**

- ARIMA models **rely heavily on the autocorrelation pattern** in the data.

- ARIMA models can also be developed in the presence of seasonality in the time series.

- ARIMA models thus **essentially ignore domain theory (by ignoring “traditional” explanatory variables)**

When to Use ARIMA Models

- Little or nothing is known about the dependent variable being forecasted
- The independent variables known to be important cannot be forecasted effectively
- Objective is to obtain short term forecasts

Basic ARIMA Models

1. Autoregressive model of order p (AR(p)):

$$\boxed{y_t = \delta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t}$$

Where y_t depends on its *p* previous values

2. Moving Average model of order q (MA(q))

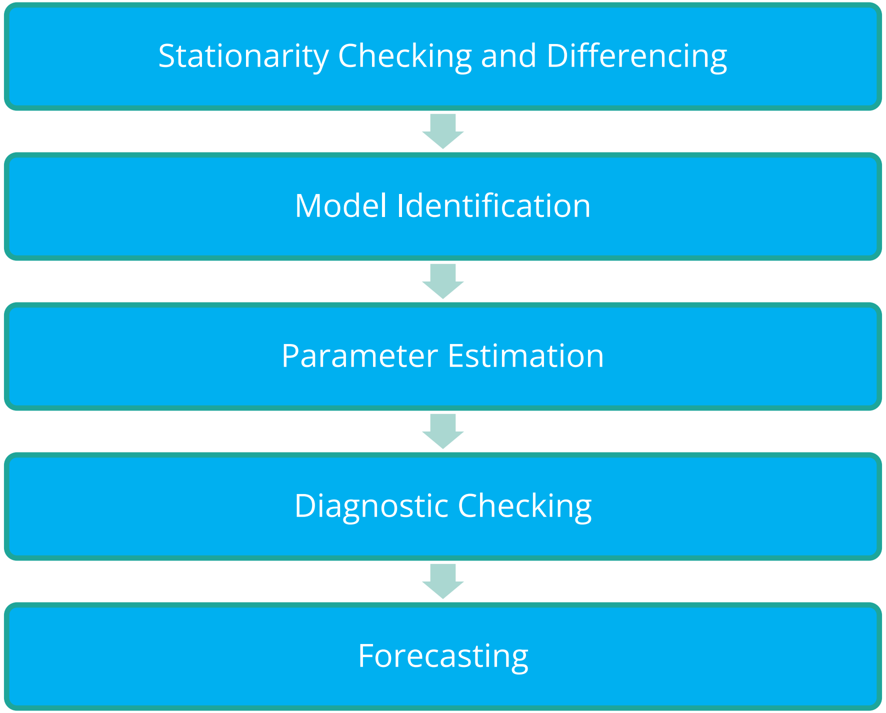
$$\boxed{y_t = \delta + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}}$$

y_t depends on *q* previous random error terms

3. Autoregressive-Moving Average model of order p and q (ARMA(p,q))

$$\boxed{\begin{gathered} y_t = \delta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} \\ + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \end{gathered}}$$

Five-Step Iterative Procedure



Step 1: Stationarity Checking

Differencing

- Differencing continues until stationarity is achieved

$$\boxed{\Delta y_t = y_t - y_{t-1}} \quad \boxed{\Delta^2 y_t = \Delta(\Delta y_t) = \Delta(y_t - y_{t-1}) = y_t - 2y_{t-1} + y_{t-2}}$$

- The differenced series has n-1 values after taking the first-difference, n-2 values after taking the second difference, and so on

- The number of times that the original series must be differenced in order to achieve stationarity is called the order of integration, denoted by d ?

?

How many times should a series be differenced?
In practice, it is not required to go beyond second difference.

Case Study

Background

- Annual Sales for a specific company from year 1961 to 2017

Objective

- To develop time series model and forecast sales for next 3 years

Available Information

- Number of cases: 57
- Variables: Year, sales(in 10’s GBP)

Data Snapshot

turnover_annual data

Variables

ons on Discrete Time Scale

Year	sales
1961	224786
1962	230034
1963	236562
1964	250960
1965	261615
1966	268316
1967	283589
1968	280160
1969	301422
1970	308018
1971	[illegible]
1974	364834
1975	392503

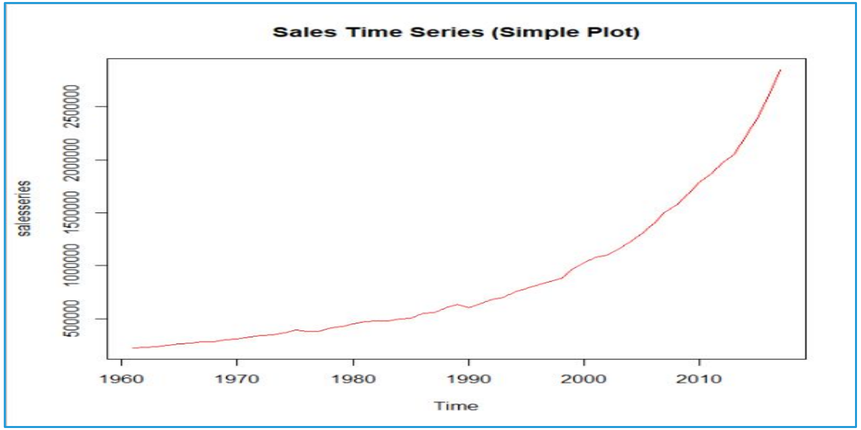
Columns	Description	Type	Measurement	Possible values
Year	Financial Year	Numeric	-	-
sales	sales(in 10's GBP)	Numeric	In British Pound	Positive values

Creating and Plotting Time Series in R

```
#Importing the Data
turnover_annual <- read.csv("turnover_annual.csv",header=TRUE)
#Creating and Plotting a Time Series Object
salesseries<-ts(turnover_annual$sales,start=1961,end=2017)
```

ts() converts a column from a data frame to time series object. The **start=** and **end=** arguments specify start and end period

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



Interpretation :

- The time-series clearly shows a positive trend.

Dickey Fuller Test For Checking Stationarity

```
# Install & load package "urca" for executing Dickey Fuller Test
install.packages("urca")
library(urca)
df<-ur.df(salesseries, lag=0)
summary(df)
```

ur.df() performs a Dickey Fuller unit root test on time series data.

```
# 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
-73514 -19075  -8490    3412   76065

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
z.lag.1 0.064334   0.003338   19.27   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25950 on 55 degrees of freedom
Multiple R-squared:  0.871,     Adjusted R-squared:  0.8687
F-statistic: 371.5 on 1 and 55 DF,  p-value: < 2.2e-16


Value of test-statistic is: 19.2745

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.6 -1.95 -1.61
```

Interpretation :

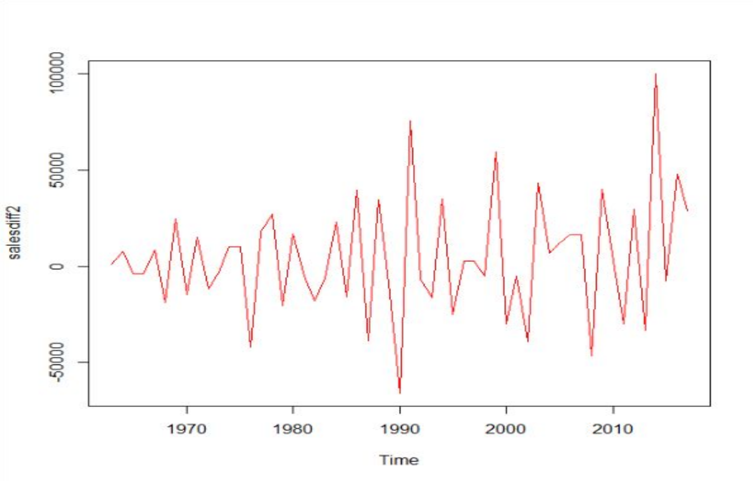
- Time series is non-stationary as value of test statistic is greater than 5% critical value.

Determining Order of Differencing

```
# Install & load package "forecast"
install.packages("forecast")
library(forecast)
ndiffs(salesseries)
[1] 2

salesdiff2<-diff(salesseries,differences=2)
plot(salesdiff2,col="red")
```

- **ndiffs()** by default uses a Unit Root test to estimate the number of differences required.

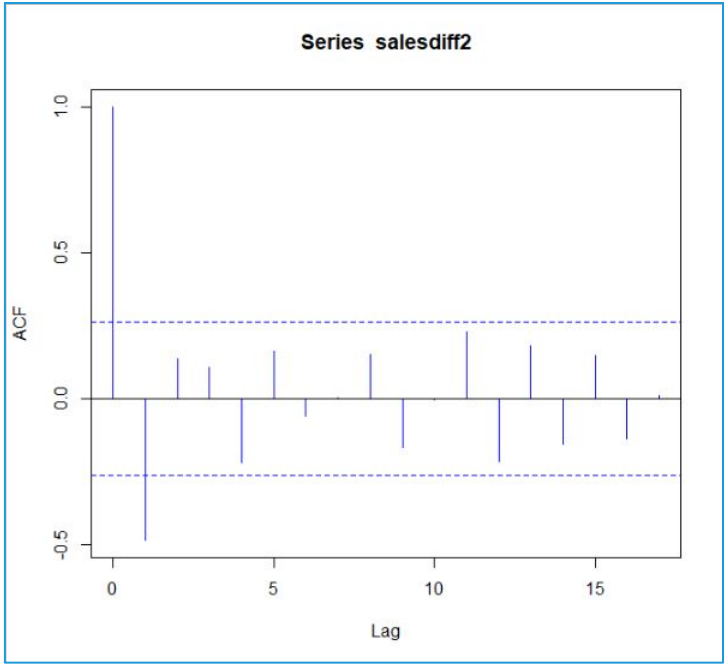


Interpretation :

- After 2nd order differencing, the series looks **stationary**.

Correlogram for 2nd Order Differenced Time Series

```
# ACF Plot
acf(salesdiff2,col="blue")
# Output
```



Interpretation :

- Stationarity is achieved with 2nd order difference.

Step 2: Model Identification

Model Identification

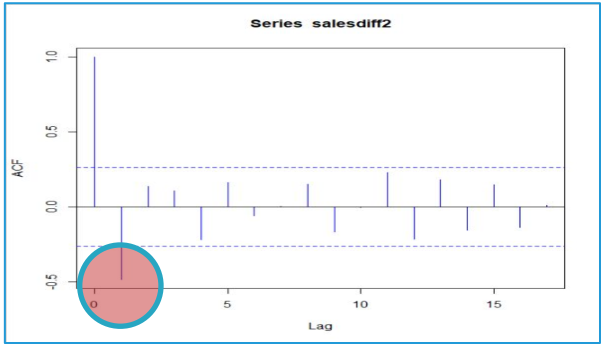
- When the data are confirmed stationary, proceed to tentative identification of models through **visual inspection of correlogram and partial correlogram**

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

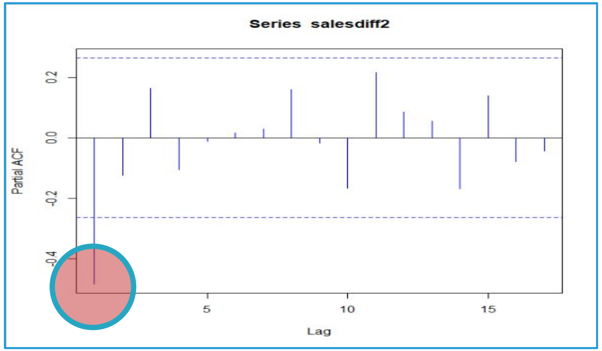
Model Identification

- ARIMA model is expressed as arima (p,d,q) where
  - p = no. of autoregressive terms
  - d = order of differencing
  - q = no. of moving average terms

ACF Plot



PACF Plot



- ACF and PACF correlograms will help in determining the MA and AR values respectively.

Indicative Model :
arima(2,2,2)

Step 3: Parameter Estimation

Parameter Estimation in R

```
#Estimation using arima function
salesmodel<-arima(salesseries,order=c(2,2,2))
```

- **arima()** fits a model to a univariate time series.
- **order=** argument specifies values of p,d and q

```
coef(salesmodel)
AIC(salesmodel)
```

coef() and **AIC()** return the model coefficients and AIC value.

```
# Output
```

```
> coef(gdpmodel)
       ar1        ar2        ma1        ma2
-1.2355735 -0.6701038  0.7851143  0.3295140
> AIC(gdpmodel)
[1] 1285.984
```

There are two AR coefficients and two MA coefficients.
Smaller the AIC value, better is the model. We need to try out various combinations of AR and MA terms to arrive at final model.

Automatic Estimation of Model Parameters

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

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

auto.arima(salesseries,d=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 models within the order constraints provided.
- **trace=** TRUE returns the list of all models considered.
- **ic=** specifies the information criterion. We have specified it as **"aic"**.

Automatic Estimation of Model Parameters

```
# Output
 ARIMA(2,2,2)                    : 1285.984
 ARIMA(0,2,0)                    : 1294.497
 ARIMA(1,2,0)                    : 1283.644
 ARIMA(0,2,1)                    : 1285.212
 ARIMA(2,2,0)                    : 1285.114
 ARIMA(1,2,1)                    : 1285.361
 ARIMA(2,2,1)                    : 1283.568
 ARIMA(1,2,2)                    : 1286.479

 Best model: ARIMA(2,2,1)

Series: gdpseries
ARIMA(2,2,1)

Coefficients:
          ar1      ar2     ma1
      -1.3949  -0.5100  0.9727
s.e.   0.1171   0.1168  0.1019

sigma^2 estimated as 714703483:  log likelihood=-637.78
AIC=1283.57   AICc=1284.37   BIC=1291.6
```

Interpretation :

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

arima Function Using BEST Order

```
# Run arima() for cross checking parameters based on model suggested
# by auto.arima
```

```r
salesmodel<-arima(salesseries,order=c(2,2,1))
coef(salesmodel)
AIC(salesmodel)
```

```
# Output
```

```
> coef(gdpmodel)
       ar1        ar2        ma1
-1.3948728 -0.5100167  0.9726749
> AIC(gdpmodel)
[1] 1283.568
```

Model Selection Criteria

Akaike Information Criterion (AIC)

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

where L = Likelihood function

k = Number of parameters to be estimated

<u>Ideally, AIC should be as small as possible</u>

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 $\mathbf{e_t}$ **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)
plot(resi,col="red")
```

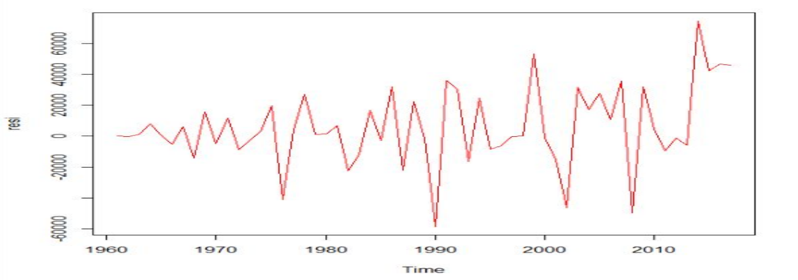
residuals() calculates residual values.

- **Box.test()** runs can run a variety of diagnostic tests. It computes the Box Pierce statistic by default.
- **Box.test()** also has the provision for calculating Ljung–Box test statistic, by adding the **type = “Ljung-Box”** argument.

```
# Output
```

```
        Box-Pierce test

data:  resi
X-squared = 0.17544, df = 1, p-value = 0.6753
```



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 generate next 3 years sales forecasts

```
# Output
```

```
$pred
Time Series:
Start = 2018
End = 2020
Frequency = 1
[1] 3072045 3308799 3537673

$se
Time Series:
Start = 2018
End = 2020
Frequency = 1
[1] 26017.21 48562.55 75729.34
```

Next 3 years 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