Foundation Course

on

Business Forecasting

using

Python

CONTENTS

INTRODUCTION
to
BUSINESS FORECASTING

INTRODUCTION

Forecast

A prediction of some future event or events

Example:

The number of 2 wheeler sales in Bangalore during next month

The average volume of an airline passengers in the next quarter

Field of applications

Business and industry	Medicine
Government	Social science
Economics	Politics
Environmental science	Finance

INTRODUCTION

Methodology

Based on identifying, modeling and extrapolating the patterns found in historical data

Historical data usually exhibit inertia and do not change dramatically very quickly

Involves use of statistical methods and time series data

Time series

A time oriented or chronological sequence of observations on a variable or metric of interest

A collection of observations or data made sequentially in time

A dataset consisting of observations arranged in chronological order

A sequence of observations over time

EXPLORATION *of* TIME SERIES

TIME SERIES EXPLORATION

Time Series Plot:

The graphical representation of time series data by taking time on x axis & data on y axis.

A plot of data over time

Reveals patterns such as random, trends, level shifts, periods or cycles, seasonal, unusual observations or combination of patterns

TIME SERIES EXPLORATION

Time Series Plot

1. Trend: A long term increase or decrease in the data
2. Cyclic: The time series data exhibiting rises and falls
3. Seasonal Pattern: The time series data exhibiting rises and falls influenced by seasonal factors
4. Unusual observation: A data point which is unusually high or low compared to other data points

TIME SERIES EXPLORATION

Time Series Plot: Example

The data on weekly sales of pharmaceutical products is given in the file pharmaceutical_Product file. Draw the time series plot and identify the underline pattern?

TIME SERIES EXPLORATION

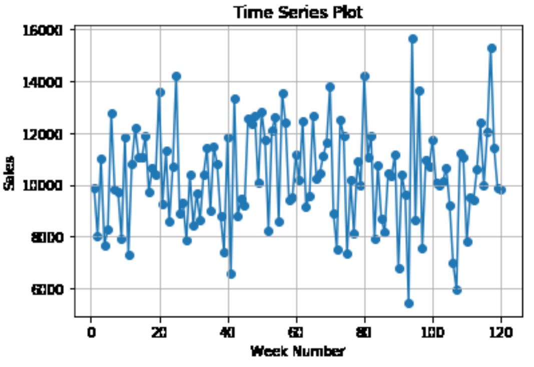
Time Series Plot: Example

Python Code

```
import pandas as mypd
import matplotlib.pyplot as myplot

mydata = mypd.read_csv("D:/LKQ_India/ModuleIII_Dataset/Pharmaceutical_Product.csv")
mydata.head()
sales =  mydata.Sales
week = mydata.Week

myplot.scatter(week, sales)
myplot.plot(week, sales)
myplot.title("Time Series Plot")
myplot.xlabel("Week Number")
myplot.ylabel("Sales")
myplot.grid()
myplot.show()
```

TIME SERIES EXPLORATION

Time Series Plot: Example



Conclusion: More or less random pattern

TIME SERIES EXPLORATION

Time Series Plot: Exercise 1

The data on annual production of diary products from 1960 to 1999 is given in the file Diary_Products file. Draw the time series plot and identify the underline pattern?

Conclusion: ?

TIME SERIES EXPLORATION

Time Series Plot: Exercise 2

The data on monthly sales of an aircraft component from April 2010 to October 2013 is given in the file Aircraft_Component file. Draw the time series plot and identify the underline pattern?

Conclusion: ?

TIME SERIES EXPLORATION

Time Series Plot: Exercise 3

The data on monthly sales of a branded jacket from January 2002 to December 2005 is given in the file Branded_Jackets file. Draw the time series plot and identify the underline pattern?

Conclusion: ?

TIME SERIES EXPLORATION

Time Series Plot: Exercise 4

The data on viscosity readings in a chemical process is given in the file Chemical_Process file. Draw the time series plot and identify the underline pattern?

Conclusion: ?

Indian Statistical Institute

TIME SERIES EXPLORATION

Stationary Series:

A series free from trend and seasonal patterns

A series exhibits only random fluctuations around mean

A stationary time series exhibits similar statistical behavior in time and this is often characterized by a constant probability distribution in time

The mean of the stationary time series does not depend on time and auto covariance function for any lag k is only the function of k and not time

TIME SERIES EXPLORATION

Test for Stationary: Unit root test

Augmented Dickey Fuller Test (ADF) :

Checks whether any specific patterns exists in the series

H0: |ρ| = 1 (series is non stationary)

H1: |ρ| < 1 (series is stationary)

A small p-value suggest data is stationary

Kwiatkowski-Phillips-Schmidt-Shin Test (KPSS) :

Checks especially the existence of trend in the data set

H0: series is trend stationary

H1: series is not trend stationary

A large p-value suggest data is stationary

Note: To consider a time series to be stationary it has to pass both the tests

TIME SERIES EXPLORATION

Stationary Series: A series free from trend and seasonal patterns.

A series exhibits only random fluctuations around mean

Example : The data on daily shipments is given in shipment.csv. Check whether the data is stationary

Day	Shipments	Day	Shipments
1	99	13	101
2	103	14	111
3	92	15	94
4	100	16	101
5	99	17	104
6	99	18	99
7	103	19	94
8	101	20	110
9	100	21	108
10	100	22	102
11	102	23	100
12	101	24	98

TIME SERIES EXPLORATION

Stationary Series: A series free from trend and seasonal patterns.

A series exhibits only random fluctuations around mean

Example : The data on daily shipments is given in shipment.csv. Check whether the data is stationary

Python code

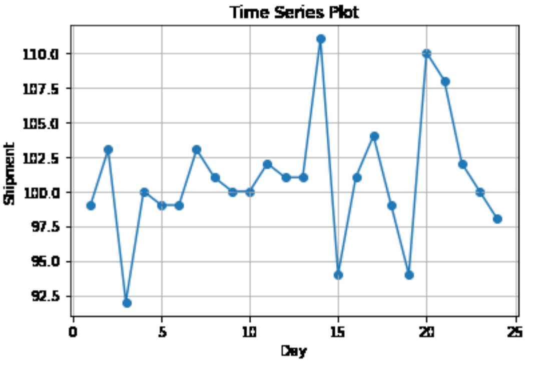
```
import pandas as mypd
import matplotlib.pyplot as myplot
from statsmodels.tsa.stattools import adfuller, kpss

mydata = mypd.read_csv("D:/LKQ_India/ModuleIII_Dataset/Shipment.csv")
Shipment =  mydata.Shipments
Day = mydata.Day

myplot.scatter(Day,Shipment)
myplot.plot(Day,Shipment)
myplot.title("Time Series Plot")
myplot.xlabel("Day")
myplot.ylabel("Shipment")
myplot.grid()
myplot.show()
```


TIME SERIES EXPLORATION

Stationary Series: A series free from trend and seasonal patterns.

A series exhibits only random fluctuations around mean

Example : The data on daily shipments is given in shipment.csv. Check whether the data is stationary



TIME SERIES EXPLORATION

Test for checking series is Stationary: Unit root test in R

ADF Test

Python Code

```
mytest = adfuller(Shipment)
test_statistics = mytest[0]
p_value = mytest[1]
test_statistics
p_value
```

Statistic	Value
Dickey-Fuller	-2.74
P value	0.068

Since p value = 0.068 < 0.1, the data is stationary at 10% significant level

TIME SERIES EXPLORATION

Test for checking series is Stationary : Unit root test in R

KPSS test

Python Code

```
mytest = kpss(Shipment)
test_statistics = mytest[0]
p_value = mytest[1]
test_statistics
p_value
```

Statistic	Value
KPSS Level	0.3
P value	0.1

Since p value = 0.1 >= 0.05, the data is stationary

TIME SERIES EXPLORATION

Exercise 1: The annual GDP values from 1993 to 2005 is given in the file GDP file : Check whether the series is stationary?

Exercise 2: The data on weekly sales of pharmaceutical products is given in the file pharmaceutical_Product file : Check whether the series is stationary?

Exercise 3: The data on annual production of diary products from 1960 to 1999 is given in the file Diary_Products file. Check whether the series is stationary?

Indian Statistical Institute

TIME SERIES EXPLORATION

Differencing: A method for making series stationary

A differenced series is the series of difference between each observation y_t and the previous observation y_{t-1}

$$y_t' = y_t - y_{t-1}$$

A series with trend can be made stationary with 1st differencing

A series with seasonality can be made stationary with seasonal differencing

TIME SERIES EXPLORATION

Differencing: A method for making series stationary

Example: Is it possible to make the GDP data (1993 to 2005) given in GDP.csv stationary?

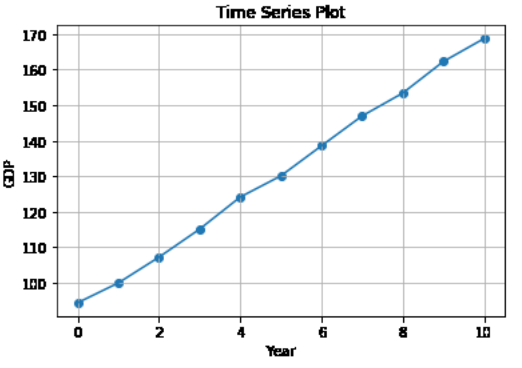
Python Code

```
import pandas as mypd
import matplotlib.pyplot as myplot
from statsmodels.tsa.stattools import adfuller, kpss

mydata = mypd.read_csv("D:/LKQ_India/ModuleIII_Dataset/GDP.csv")
gdp=  mydata.GDP
Year = mydata.index

myplot.scatter(Year,gdp)
myplot.plot(Year,gdp)
myplot.title("Time Series Plot")
myplot.xlabel("Year")
myplot.ylabel("GDP")
myplot.grid()
myplot.show()
```


TIME SERIES EXPLORATION

Differencing: A method for making series stationary

Example: Is it possible to make the GDP data (1993 to 2005) given in GDP.csv stationary?



TIME SERIES EXPLORATION

Differencing: A method for making series stationary

Example: Is it possible to make the GDP data (1993 to 2005) given in GDP.csv stationary?

Python Code

```
mytest = adfuller(gdp)
test_statistics = mytest[0]
p_value = mytest[1]

mytest = kpss(gdp)
test_statistics = mytest[0]
p_value = mytest[1]
```

Test	Statistic	p-value
ADF	-0.75	0.833
KPSS	0.39	0.082

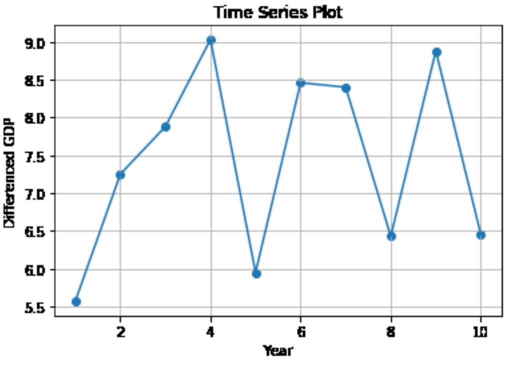
TIME SERIES EXPLORATION

Differencing: A method for making series stationary

Example: Is it possible to make the GDP data (1993 to 2005) given in GDP.csv stationary?

Python Code

```
diff_gdp = gdp.diff()
year = diff_gdp.index

myplot.scatter(year,diff_gdp)
myplot.plot(year,diff_gdp)
myplot.title("Time Series Plot")
myplot.xlabel("Year")
myplot.ylabel("Differenced GDP")
myplot.grid()
myplot.show()
```



TIME SERIES EXPLORATION

Differencing: A method for making data stationary

Example: Is it possible to make the GDP data (1993 to 2005) given in GDP.csv stationary?

Differencing required is 1

$y_t' = y_t - y_{t-1}$

Python Code

```
diff_gdp = diff_gdp.dropna()
mytest = adfuller(diff_gdp)
test_statistics = mytest[0]
p_value = mytest[1]

mytest = kpss(diff_gdp)
test_statistics = mytest[0]
p_value = mytest[1]
```

Test	Statistic	p-value
ADF	-23.73	0.00
KPSS	0.35	0.097

TIME SERIES EXPLORATION

Differencing: A method for making series stationary

Exercise 1: The data on annual production of diary products from 1960 to 1999 is given in the file Diary_Products file. Check whether the series is stationary? If not can it be made stationary by differencing?

EXPONENTIAL SMOOTHING

EXPONENTIAL SMOOTHING

Used to make short term forecasts of time series data

Single Exponential Smoothing:

- Used for time series with no trend or seasonality
- Smoothing is controlled by the parameter alpha
- Value of alpha lies between 0 and 1
- Alpha is estimated by minimizing the MSE
- Give more weight to recent values compared to the old values

EXPONENTIAL SMOOTHING

Single Exponential Smoothing: Methodology

Let y_1, y_2, - - - y_t be the time series, then

y_{t+1} estimate = $S_{t+1} = \alpha\, y_t + (1- \alpha)\, S_t$

where $0 \le \alpha \le 1$ and $S_1 = y_1$

EXPONENTIAL SMOOTHING

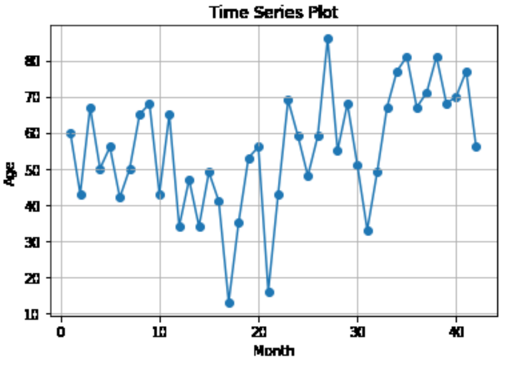
Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?

Python Code

```
import pandas as mypd
import matplotlib.pyplot as myplot
from statsmodels.tsa.stattools import adfuller, kpss
from statsmodels.tsa.holtwinters import SimpleExpSmoothing

mydata = mypd.read_csv("D:/LKQ_India/ModuleIII_Dataset/rulers.csv")
age= mydata.Age
month = mydata.Month

myplot.scatter(month, age)
myplot.plot(month, age)
myplot.title("Time Series Plot")
myplot.xlabel("Month")
myplot.ylabel("Age")
myplot.grid()
myplot.show()
```

EXPONENTIAL SMOOTHING

Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?



EXPONENTIAL SMOOTHING

Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?

Python code - Checking whether series is stationary

```
mytest = adfuller(age)
test_statistics = mytest[0]
p_value = mytest[1]

mytest = kpss(age)
test_statistics = mytest[0]
p_value = mytest[1]
```

Test	Statistic	P-value
ADF	-4.09	0.001
KPSS	0.3	0.1

EXPONENTIAL SMOOTHING

Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?

R code

Fitting Single Exponential Model

```
mymodel = SimpleExpSmoothing(age)
mymodel = mymodel.fit()
mymodel.summary()
```

Statistic	Value
Optimum α	0.2561
AIC	231.432
BIC	234.907
AICC	232.513

MODEL VALIDATION

Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?

Model diagnostics

Residual = Actual – Predicted

Mean Absolute Error: MAE

Root Mean Square Error: RMSE

Mean Absolute Percentage Error: MAPE

EXPONENTIAL SMOOTHING

Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?

R code

Computing predicted values and residuals (errors)

```
pred = mymodel.predict(0, 41)
res = age –pred
abs_res = res.abs()
mae = abs_res.mean()

res_sq = res**2
mse = res_sq.mean()

import math as mymath
rmse = mymath.sqrt(mse)

pae = abs_res/age
mape = pae.mean()
```


Indian Statistical Institute

MODEL VALIDATION

Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?

Model diagnostics

Statistic	Description	Value
MAE	Average of absolute residuals	12.0257
MSE	Average of residual squares	224.762
RMSE	Square root of MSE	14.9921
MAPE	Average of absolute % error	29.68%

Criteria

MAPE < 10% is reasonably good

MAPE < 5 % is very good

MODEL VALIDATION

Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?

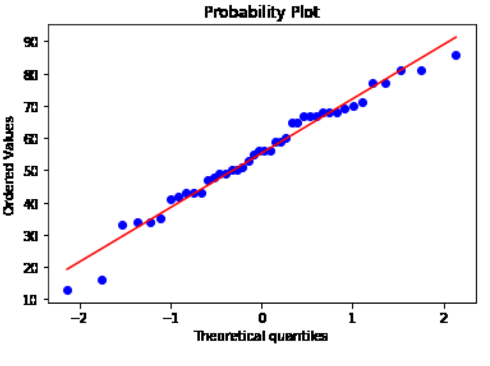
Model diagnostics - Normality of Errors with zero

Python Code

```
from scipy import stats
import matplotlib.pyplot as myplot

stats.probplot(age, plot=myplot)
myplot.show()

normality_test = stats.mstats.normaltest(age)
test_statististic = normality_test.statistic
p_value = normality_test.pvalue
```


MODEL VALIDATION

Example: The age of the rulers of an European country is given in file Rulers. Forecast the age of the future ruler using single exponential smoothing method with best value of α?



Statistic (w)	P value
1.8654	0.3935

MODEL DEPLOYMENT

Forecast and Prediction Interval

Prediction interval : Predicted value $\pm$ z $\sqrt{MSE}$

where z = width of prediction interval

Prediction Interval	Z
90%	1.645
95%	1.960
99%	2.576

Forecasted value $S_{t+1} = \alpha y_t + (1 - \alpha)S_t$
Forecasted value $S_{43} = \alpha y_{42} + (1 - \alpha)S_{42}$
Forecasted value S_{43} = 0.2560892 x 56 + (1-0.2560892) x 71.901387= 67.829

MODEL DEPLOYMENT

Forecast

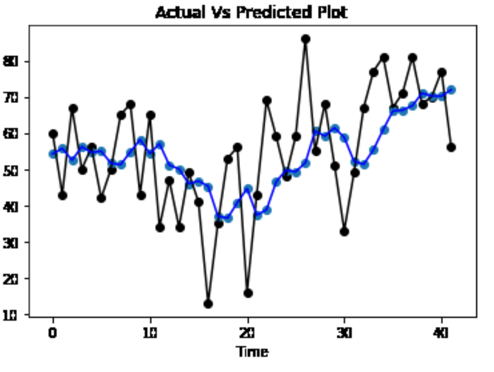
Python Code

```
S43 = mymodel.predict()
S43
mymodel.predict(42, 45)
```

Period	Forecast
43	67.829

MODEL DEPLOYMENT

```
import matplotlib.pyplot as myplot
myplot.scatter(x, age, color = "black")
myplot.scatter(x, pred)
myplot.plot(x, age, color = "black")
myplot.plot(x, pred, color = "blue")
myplot.title("Actual Vs Predicted Plot")
myplot.xlabel("Time")
myplot.show()
```



MULTIPLE REGRESSION ANALYSIS

CORRELATION & REGRESSION

Regression

Regression helps

- To identify the exact form of the relationship
- To model output in terms of input or process variables

Examples:

Expected (Yield) = 5 + 3 x Time - 2 x Temperature

REGRESSION ANALYSIS

Exercise : The effect of temperature and reaction time affects the % yield. The data collected in given in the Mult-Reg_Yield file. Develop a model for % yield in terms of temperature and time?

Step 1: Read packages

```
# importing the packages
import pandas as mypd
import matplotlib.pyplot as myplot
from scipy import stats
import math as mymath
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
import seaborn as mysb
```

REGRESSION ANALYSIS

Exercise : The effect of temperature and reaction time affects the % yield. The data collected in given in the Mult-Reg_Yield file. Develop a model for % yield in terms of temperature and time?

Step 2: Read data

```
#importing the dataset
mydata = mypd.read_csv("E:/hp/hp_2020/Module1/Dataset/Mult_Reg_Yield.csv")
mydata.head()

# Seperating x and y
x = mydata.iloc[:, 0:2]
y = mydata.Yield
```


REGRESSION ANALYSIS

Exercise : The effect of temperature and reaction time affects the % yield. The data collected in given in the Mult-Reg_Yield file. Develop a model for % yield in terms of temperature and time?

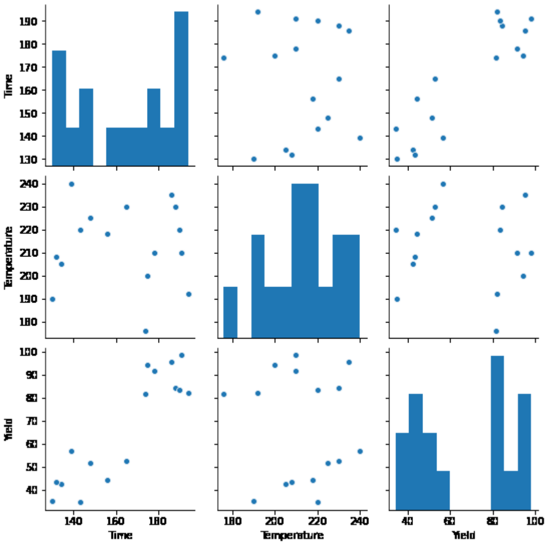
Step 3: Correlation Analysis

```
# Scatter plot
mysb.pairplot(mydata)
myplot.show()
```

Correlation between xs & y should be high

Correlation between xs should be low

REGRESSION ANALYSIS



REGRESSION ANALYSIS

Step 4: Regression Modeling

```
# fitting the model
mymodel = LinearRegression()
mymodel = mymodel.fit(x,y)
mymodel.intercept_
mymodel.coef_
```

	Coefficient
Intercept	-67.8845
Time	0.9061
Temperature	-0.0642

Yield = -67.8845 + 0.9061 x Time – 0.0642 x Temperature

REGRESSION ANALYSIS

Step 4: Regression Modeling

```
# Model accuracy
rsq = mymodel.score(x,y)
pred = mymodel.predict(x)

# Model Adequacy
mse = mean_squared_error(y, pred)
rmse = mymath.sqrt(mse)
```

Statistic	Coefficient
R^2	0.8064
MSE	102.0051
RMSE	10.0998

REGRESSION ANALYSIS

Step 4: Residual Analysis

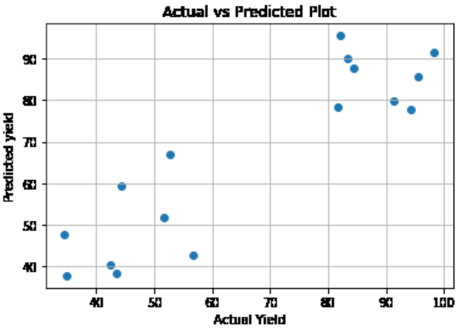
```
# Residual Analysis
res = y – pred
myresult = [y, pred, res]
myresult = mypd.DataFrame(myresult)
myresult =myresult.transpose()
```

SL No	Yield	Predicted	Residuals
0	35	37.71	-2.71
1	81.7	78.48	3.22
2	42.5	40.37	2.13
3	98.3	91.70	6.60
4	52.7	66.86	-14.16
5	82	95.57	-13.57
6	34.5	47.56	-13.06
7	95.4	85.56	9.84
8	56.7	42.66	14.04
9	84.4	87.70	-3.30
10	94.3	77.84	16.46
11	44.3	59.47	-15.17
12	83.3	90.15	-6.85
13	91.4	79.92	11.48
14	43.5	38.37	5.13
15	51.7	51.77	-0.07

REGRESSION ANALYSIS

Step 4: Residual Analysis – Actual Vs Predicted Plot

```
myplot.scatter(y, pred)
myplot.title("Actual vs Predicted Plot")
myplot.xlabel("Actual Yield")
myplot.ylabel("Predicted yield"
myplot.grid()
myplot.show()
```



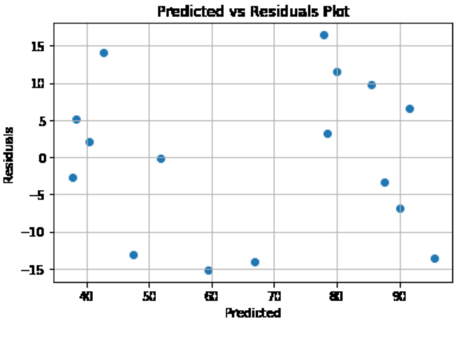
Note: There need to be strong positive correlation between actual and fitted response

Indian Statistical Institute

REGRESSION ANALYSIS

Step 4: Residual Analysis – Predicted Vs Residuals Plot

```
myplot.scatter(pred, res)
myplot.title("Predicted vs Residuals Plot")
myplot.xlabel("Predicted")
myplot.ylabel("Residuals")
myplot.grid()
myplot.show()
```



Note: There need to be strong positive correlation between actual and fitted response

REGRESSION ANALYSIS

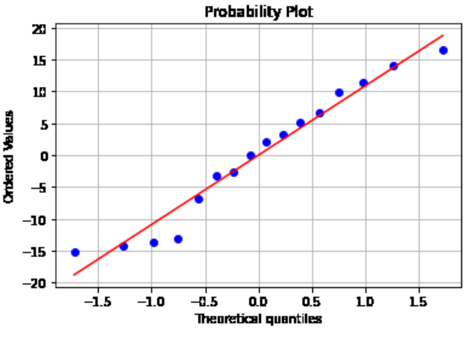
Step 4: Residual Analysis: Normality test

```
norm_test = stats.normaltest(res)
w = norm_test[0]
p_value = norm_test[1]

stats.probplot(res, plot= myplot)
myplot.grid()
myplot.show()
```

Normality Test: Yield data	
w	p value
1.9835	0.3709

REGRESSION ANALYSIS

Step 4: Residual Analysis: Normality test



REGRESSION ANALYSIS

Step 5: Cross Validation

```
# Cross Validation
myscore = cross_val_score(mymodel, x, y, scoring='neg_mean_squared_error', cv = 4)
cv_mse = -1*myscore.mean()
rmse = mymath.sqrt(cv_mse)
```

Statistic	Training	Test
MSE	102.0051	122.5726
RMSE	10.0998	11.0713

AUTO REGRESSIVE INTEGRATED MOVING AVERAGE MODEL (ARIMA)

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Exponential Smoothing: Assumption

A time series can be represented as the sum of two distinct components:

- Deterministic
- Stochastic (random)

Deterministic component: A function of time

Stochastic component: Assumed to be random noise and generates stochastic behavior of the time series

Random noise assumed to be generated through independent shocks in the process

But often the successive observations show serial dependence

Moreover for exponential smoothing, the forecast error need to be uncorrelated and are normally distributed with mean zero and constant variance

Then exponential smoothing methods may be inefficient and sometimes inappropriate.

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

General class of model

Takes into account the correlations in the data

Includes an explicit statistical model for irregular component of the time series that allows for non zero autocorrelations in the irregular component

Often defined for stationary time series

Non stationary time series need to be made stationary by differencing or decomposition

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Auto covariance: Measure of association between the values of a variable and its values at another time period

Auto covariance (of lag 1): Measure of association between the consecutive values of a variable

Autocorrelation (1) = Autocovariance(1)/ Variance

Indian Statistical Institute

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Partial correlation :

Let x, y and z be three random variables

Consider simple linear regression of x on z: $\hat{x} = a_1 + b_1 z$

And residuals: $x^* = x - \hat{x}$

Consider simple linear regression of y on z: $\hat{y} = a_2 + b_2 z$

And residuals: $y^* = y - \hat{y}$

Partial correlation between x and y (after adjusting for z) is the correlation between x* and y*

In general, partial correlation can be seen as the correlation between two variables after adjusting for a common factor that may be affecting them

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Partial correlation : Example

The data on three random variables namely yield, time and temperature are given below: Compute the correlation between yield and time (after adjusting for temperature)

Time	Temperature	Yield
130	190	35
174	176	81.7
134	205	42.5
191	210	98.3
165	230	52.7
194	192	82
143	220	34.5
186	235	95.4
139	240	56.7
188	230	84.4
175	200	94.3
156	218	44.3
190	220	83.3
178	210	91.4
132	208	43.5
148	225	51.7

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Partial correlation : Example

The data on three random variables namely yield, time and temperature are given below: Compute the correlation between yield and time (after adjusting for temperature)

Models

Yield = 82.5967 - 0.073 x Temperature

Time = 166.0776 - 0.01004 x Temperature

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Partial correlation : Example

The data on three random variables namely yield, time and temperature are given below: Compute the correlation between yield and time (after adjusting for temperature)

Residuals(Yield*)	Residuals(Time*)
-33.6715	-34.1692
12.0024	9.6902
-25.0722	-30.0185
31.0943	27.0317
-13.0399	1.2326
13.4751	29.8509
-31.9728	-20.8678
30.0266	22.2829
-8.3070	-24.6669
18.6601	24.2326
26.3614	10.9313
-22.3194	-7.8879
16.8272	26.1322
24.1943	14.0317
-23.8523	-31.9884
-14.4063	-15.8176

Partial correlation between yield & time
= Correlation between yield * & time *
= 0.8976

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Partial auto correlation :

Let y_1, y_2, - - -, y_t be a time series

Partial auto correlation between y_t and y_{t-k} is the autocorrelation between y_t and y_{t-k} after adjusting for y_{t-1}, y_{t-2}, - - - , y_{t-k-1}

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Auto Regressive Integrated Moving Average Models (ARIMA (p,d,q))

Widely used and very effective modeling approach

Proposed by George Box and Gwilym Jenkins

Also known as Box – Jenkins model or ARIMA(p,d,q)

where

p: number of auto regressive (AR) terms

q: number of moving average (MA) terms

d: level of differencing

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

General Form

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + - - - + \theta_1 e_{t-1} + \theta_2 e_{t-2} - - - -$$

Where

c: constant

$\phi_1, \phi_2, \theta_1, \theta_2$, - - - are model parameters

$e_{t-1} = y_{t-1} - s_{t-1}$, e_t are called errors or residuals

s_{t-1} : predicted value for the t-1th observation (y_{t-1})

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 1:

Draw time series plot and check for trend, seasonality, etc

Step 2:

Draw Auto Correlation Function (ACF) and Partially Auto Correlation Function (PACF) graphs to identify auto correlation structure of the series

Step 3:

Check whether the series is stationary using unit root test (ADF test, KPSS test)

If series is non stationary do differencing or transform or decompose the series

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 4:

Identify the model using ACF and PACF or automatically

The best model is one which minimizes AIC or BIC or both

Step 5:

Estimate the model parameters using maximum likelihood method (MLE)

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 6:

Do model diagnostic checks

The errors or residuals should be white noise and should not be auto correlated

Do Portmanteau and Ljung & Box tests. If p value > 0.05, then there is no autocorrelation in residuals and residuals are purely white noise.

The model is a good fit

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Example: The age of the rulers of an European country is given in file Rulers. Fit Forecasting model using ARIMA?

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

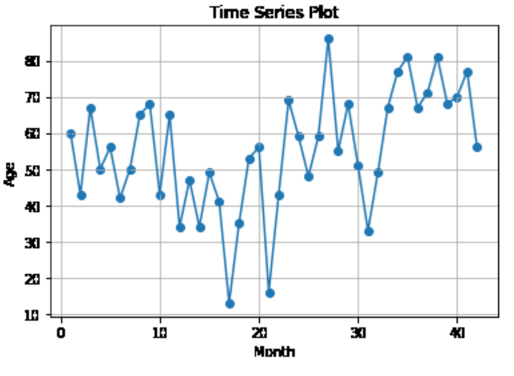
Step 1: Read and plot the series

```
import pandas as mypd
import matplotlib.pyplot as myplot
from statsmodels.tsa.stattools import adfuller, kpss
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima_model import ARIMA
from pmdarima import auto_arima

mydata = mypd.read_csv("D:/LKQ_India/ModuleIII_Dataset/rulers.csv")
age =  mydata.Age
month = mydata.Month
```

Step 2: Time series plot

```
myplot.scatter(month, age)
myplot.plot(month, age)
myplot.title("Time Series Plot")
myplot.xlabel("Month")
myplot.ylabel("Age")
myplot.grid()
myplot.show()
```


AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL



AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 3: Check whether the series is stationary

```
mytest = adfuller(age)
test_statistics = mytest[0]
p_value = mytest[1]

mytest = kpss(age)
test_statistics = mytest[0]
p_value = mytest[1]
```

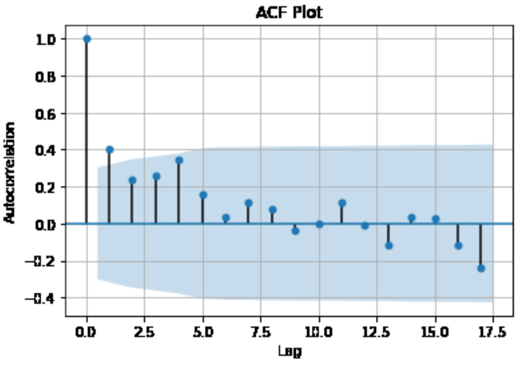
Test	Statistic	P value
ADF	-4.09	0.001
KPSS	0.30	0.1

Since p value of ADF test > 0.05 and that of KPSS test < 0.05, the series is stationary

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 4: Draw ACF & PACF Graphs

```
plot_acf(age)
myplot.xlabel("Lag")
myplot.ylabel("Autocorrelation")
myplot.title("ACF Plot")
myplot.grid()
myplot.show()
```



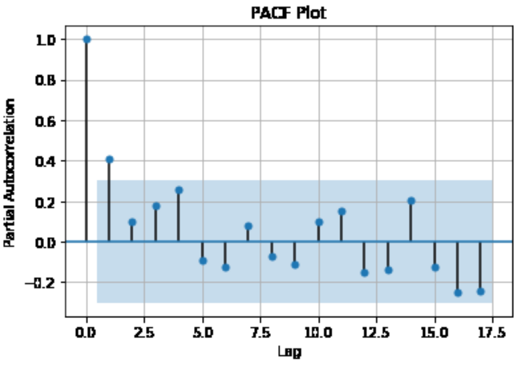
Remark

Since ACF is exponentially decaying, MA terms may be significant

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 4: Draw ACF & PACF Graphs

```
plot_pacf(age)
myplot.xlabel("Lag")
myplot.ylabel("Partial Autocorrelation")
myplot.title("PACF Plot")
myplot.grid()
myplot.show()
```



Remark

Since PACF is exponentially decaying, AR terms may not be significant

Indian Statistical Institute

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 4: Identifying the arima model

```
mymodel = ARIMA(age, order= (1,0,1))
mymodel = mymodel.fit()
mymodel.summary()
```

Model	Log Likelihood	AIC	BIC
arima(1, 0, 1)	-172.626	353.252	360.202

Terms	Coefficient	Std Error	z	p-value
Constant	56.0866	5.48	10.235	0.000
ar1	0.8341	0.17	4.914	0.000
ma1	-0.574	0.253	-2.272	0.029

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 4: Identifying the optimum arima model

Model	AIC	BIC
arima(1, 0, 1)	353.252	360.202
arima(0, 0, 1)	354.242	359.455
arima(1, 0, 0)	352.823	358.036
arima(1, 1, 1)	-	-
arima(0, 1, 1)	345.814	350.954
arima(1, 1, 0)	354.035	359.176
arima(0, 0, 2)	355.328	362.279
arima(0, 1, 2)	347.179	354.034
arima(2, 0, 0)	354.468	361.419
arima(2, 1, 0)	351.153	358.007

Remark: Model with minimum aic, bic is th optimum model

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 5: Identification of optimum model automatically

```
mymodel = auto_arima(age, trace= True, error_action= 'ignore', suppress_warnings = True,
                 seasonal= False)
mymodel = mymodel.fit(age)
mymodel.summary()
```

Model	Log likelihood	AIC	BIC
Arima (0,1,1)	-169.906	345.813	350.953

Terms	Coefficient	Std Error	z	p-value
Constant	0.3882	0.636	0.610	0.542
ma1	-0.7463	0.140	-5.335	0.000

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 5: Identification of optimum model automatically

Model	Log likelihood	AIC	BIC
Arima (0,1,1)	-169.906	345.813	350.953

Terms	Coefficient	Std Error	z	p-value
Constant	0.3882	0.636	0.610	0.542
ma1	-0.7463	0.140	-5.335	0.000

Forecast
$\nabla y_t = 0.3882 - 0.7463 \text{ x } (y_{t-1} - s_{t-1})$

where
$\nabla y_t = y_t - y_{t-1}$
s_{t-1} : y_{t-1} predicted

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 6: Checking the residuals are white noise

Test	Test Statistic	P-value
Ljung-Box	22.54	0.99

Remark: p-value > 0.05, residuals are white noise

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 7: Model diagnostic statistics

```
pred = mymodel.predict_in_sample()
pred = pred[1:42]
Age = age[1:42]
res = age –pred
abs_res = res.abs()
mae = abs_res.mean()
res_sq = res**2
mse = res_sq.mean()
import math as mymath
rmse = mymath.sqrt(mse)
pae = abs_res/age
mape = pae.mean()
```

Statistic	Description	Value
MAE	Average of absolute residuals	12.3012
MSE	Average of residual squares	232.1439
RMSE	Square root of MSE	15.2363
MAPE	Average of absolute error / actual	31.24

Indian Statistical Institute

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 8: Normality check on residuals

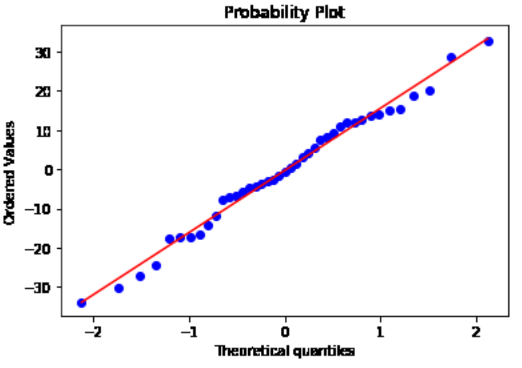
```
from scipy import stats
import matplotlib.pyplot as myplot
stats.probplot(res, plot=myplot)
myplot.show()

normality_test = stats.mstats.normaltest(res)
test_statistic = normality_test.statistic
p_value = normality_test.pvalue
```

Test Statistic	p-value
0.2614	0.8775

Indian Statistical Institute

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 8: Normality check on residuals



87

Indian Statistical Institute

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 8: Model diagnostics

```
myresult = [age, pred, res]
myresult = mypd.DataFrame(myresult)
myresult = myresult.transpose()
```

Period	Age	Predicted	Residuals
0	60	0.388205	59.611795
1	43	60.378055	-17.378055
41	56	73.386046	-17.386046

Step 9: Forecast Calculation

∇y_{42} = 0.3882 - 0.7463 x -17.386046 = 13.3634

$y_{42} = y_{42} + \nabla y_{42}$ = 56 + 13.3634 = 69.3634

88

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 9: Forecast using python

y_{42} = mymodel.predict(n_periods=1)

y_{42} = 69.363

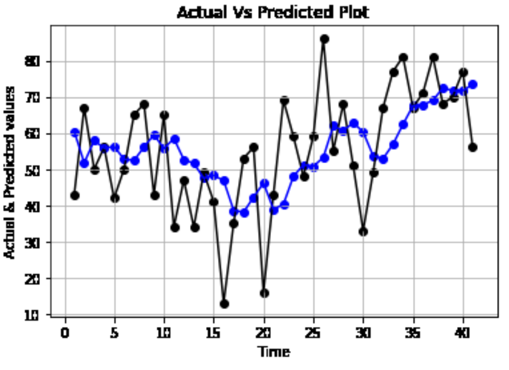
AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 10: Actual vs Predicted plot

```
import matplotlib.pyplot as myplot
x = x[1:42]

myplot.scatter(x, age, color = "black", label = "Actual")
myplot.scatter(x, pred, label = "Predicted")
myplot.plot(x, age, color = "black", label = "Actual")
myplot.plot(x, pred, color = "blue", label = "Predicted")
myplot.title("Actual Vs Predicted Plot")
myplot.xlabel("Time")
myplot.show()
```

AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Step 10: Actual vs Predicted plot



AUTO REGRESIVE INTEGRATED MOVING AVERAGE MODEL

Exercise1:The weekly production of an industrial product is given Industrial_Prduction file. Develop a model to predict the weekly production of the product?

Exercise 2:The monthly sales of an industrial product is given Industrial_Sales file. Develop a model to predict the monthly sales of the product?

Exercise 3: The annual production values of diary products from 1960 to 1999 is given in Diary_Products file. Develop a forecasting methodology using ARIMA?

INTERVENTION MODELS

INTERVENTION MODELS

Many variables may be correlated in time

A time series can not only have serial correlation with past values but also can be correlated with other variables.

Forecasting can be made more accurate by incorporating other external variables into the model

Intervention Models

In some times y_t can be affected by a known event that happens at a specific time such as fiscal policy changes, introduction of new regulatory laws, or switches suppliers, etc.

Such interventions can be modelled using indicator variables

INTERVENTION MODELS

Example

The weekly sales of laptop computers in a computer and accessories shop in Hutchins read street shop is collected for 51 weeks and is given in Laptop_Sales file. On week 40, the Government has declared lockdown to control the Covid pandemic and most of the educational institutions switched over to online mode of teaching. Fit a model to forecast the weekly laptop sales

INTERVENTION MODELS

Reading the packages

```
import pandas as mypd
import matplotlib.pyplot as myplot
from statsmodels.tsa.stattools import adfuller, kpss
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import numpy as mynp
from pmdarima import auto_arima
```

Reading the data

```
mydata = mypd.read_excel("D:/ISI/BF-06-Online//Laptop_Sales.xlsx")
```

Explore the data

```
mydata.head()
```

INTERVENTION MODELS

Copying the variables

```
sales = mydata.Sales
status = mydata.Status
week = mydata.Week
```

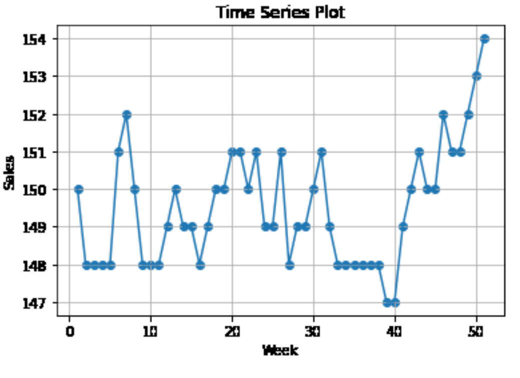
Reshaping the status variable to required 2 dimensional array

```
status = status.array.reshape(-1,1)
```

Time Sries plot of Sales

```
myplot.scatter(week, sales)
myplot.plot(week, sales)
myplot.title("Time Series Plot")
myplot.xlabel("Week")
myplot.ylabel("Sales")
myplot.grid()
myplot.show()
```


INTERVENTION MODELS



INTERVENTION MODELS

Check for Stationary series – ADF test

```
mytest = adfuller(sales)
adf = mytest[0]
round(adf,4)

p_value = mytest[1]
round(p_value,4)
```

Check for Stationary series – KPSS test

```
mytest = kpss(sales)
kpss = mytest[0]
round(kpss,4)

p_value = mytest[1]
round(p_value,4)
```


INTERVENTION MODELS

Test	Statistic	P value
ADF	-0.6986	0.8471
KPSS	0.2063	0.1

INTERVENTION MODELS

Autocorrelation Plot

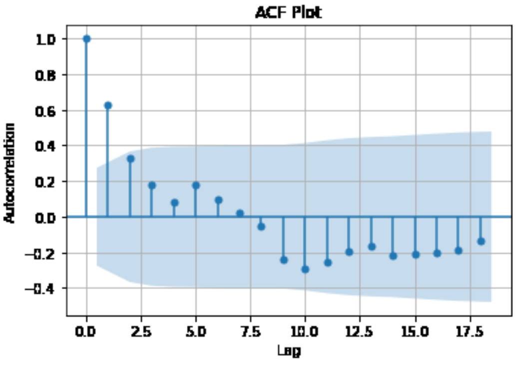
```
plot_acf(sales)
myplot.title("ACF Plot")
myplot.xlabel("Lag")
myplot.ylabel("Autocorrelation")
myplot.grid()
myplot.show()
```

Partial Autocorrelation Plot

```
plot_pacf(sales)
myplot.title("PACF Plot")
myplot.xlabel("Lag")
myplot.ylabel("Partial Autocorrelation")
myplot.grid()
myplot.show()
```

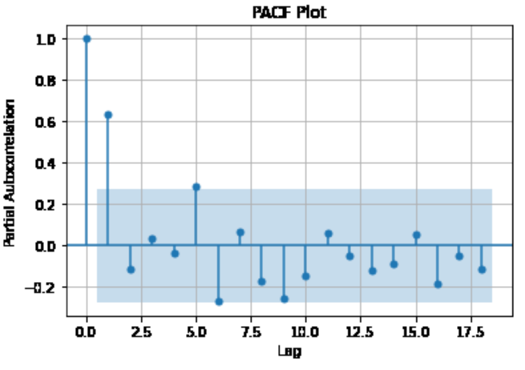

INTERVENTION MODELS

Autocorrerlation Plot



INTERVENTION MODELS

Partial Autocorrerlation Plot



INTERVENTION MODELS

Develop intervention model

```
mymodel = auto_arima(sales, X= status, trace= True, error_action= 'ignore',
          suppress_warnings= True)
mymodel = mymodel.fit(sales, X = status)
mymodel.summary()
```

Statistics	Value
Model	Arima(0,0,1)
No. of observations	51
Log Likelihood	-74.177
AIC	156.354
BIC	164.081

INTERVENTION MODELS

Model Coefficient Table

	Coefficient	Std error	z	p value
intercept	149.1384	0.287	519.018	0.000
x	2.0037	0.537	3.733	0.000
ma1	0.7629	0.096	7.962	0.000

Model

$y_t = 149.1384 + 0.7629e_{t-1} + 2.0037\ status_t$

INTERVENTION MODELS

Residual Analysis – Ljung-Box test

Statistic	Value
Ljung-Box	0.00
p value	0.97

INTERVENTION MODELS

Residuals and Predicted values

```
ypred = mymodel.predict_in_sample(X = status)
myres = sales – ypred
```

Model Accuracy Measures

```
abs_res =abs(myres)
mae = abs_res.mean()

res_sq = myres**2
mse = res_sq.mean()

pae = abs_res/sales
mape = pae.mean()
```


INTERVENTION MODELS

Model Accuracy Measures

Statistic	Value
Mean Absolute Error	0.8897
Mean Square Error	1.0722
Root Mean Square Error	1.0355
Mean Absolute Percent Error	0.59

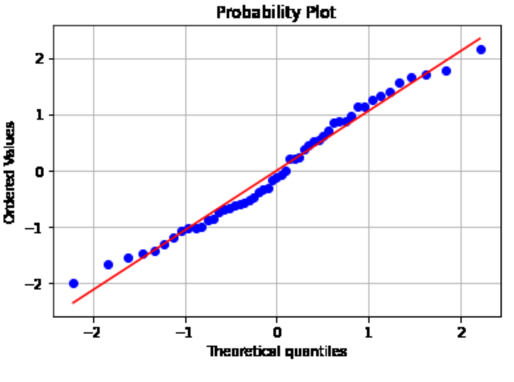
INTERVENTION MODELS

Normality Test - Residuals

```
stats.probplot(myres, plot= myplot)
myplot.grid()
myplot.show()

mytest = stats.normaltest(myres)
w = mytest[0]
p_value -= myest[1]
```


INTERVENTION MODELS

Normal Probability Plot - Residuals



INTERVENTION MODELS

Normality Test: Residuals

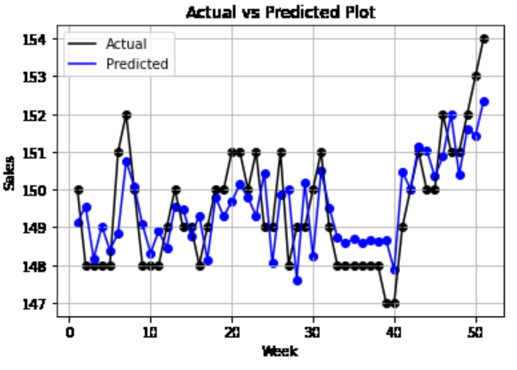
Statistic	Value
w	4.7597
p value	0.0926

INTERVENTION MODELS

Actual versus Predicted Plot

```
myplot.scatter(week, sales, color = 'black')
myplot.scatter(week, ypred, color = 'blue')
myplot.plot(week,sales, color = 'black', label = "Actual")
myplot.plot(week, ypred, color = 'blue', label = "Predicted")
myplot.title("Actual vs Predicted Plot")
myplot.xlabel("Week")
myplot.ylabel("Sales")
myplot.grid()
myplot.legend()
myplot.show()
```

INTERVENTION MODELS



Indian Statistical Institute

Foundation Course

on

Business Forecasting

Thank You

boby@isibang.ac.in