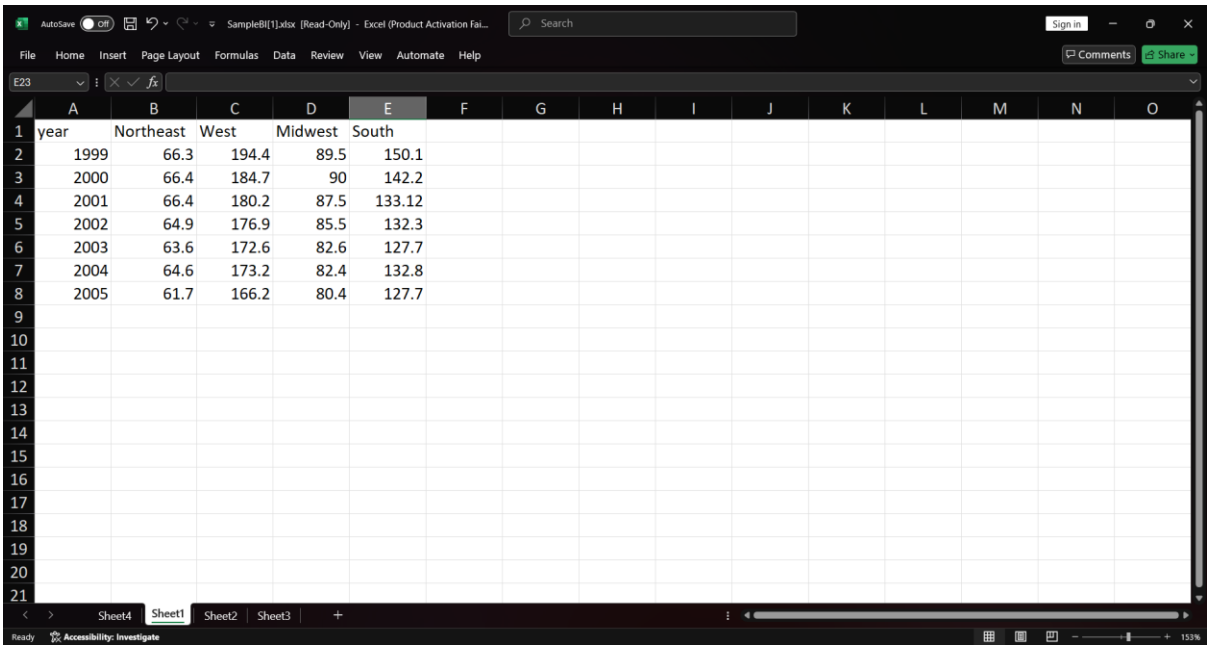
Practical No. 1

Aim: Perform the Analysis for the following.

a. Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart.

Solution:

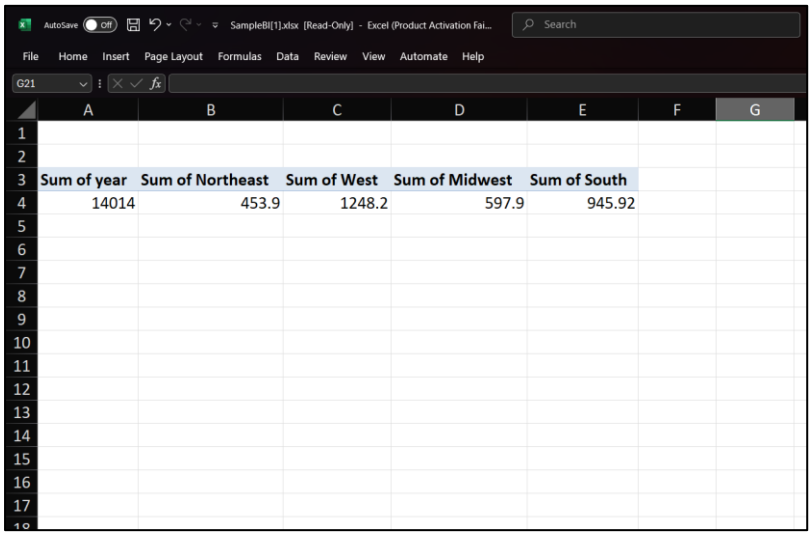
1. Import sample data into Excel.



	A	B	C	D	E
1	year	Northeast	West	Midwest	South
2	1999	66.3	194.4	89.5	150.1
3	2000	66.4	184.7	90	142.2
4	2001	66.4	180.2	87.5	133.12
5	2002	64.9	176.9	85.5	132.3
6	2003	63.6	172.6	82.6	127.7
7	2004	64.6	173.2	82.4	132.8
8	2005	61.7	166.2	80.4	127.7

2. Select the table and choose **Pivot Table** under the **Insert** tab.
3. Keep default settings in the pop-up and click **OK**.
4. A new worksheet will open with Pivot Table fields on the right.
5. To update changes, select any Pivot cell, go to **Analyze > Refresh > Refresh All**.
6. To add a new column, select a Pivot cell, go to **Analyze > Change Data Source**.
7. Choose the updated data range and click **OK**.
8. The new column will appear in the Pivot Table fields.

Pivot chart:

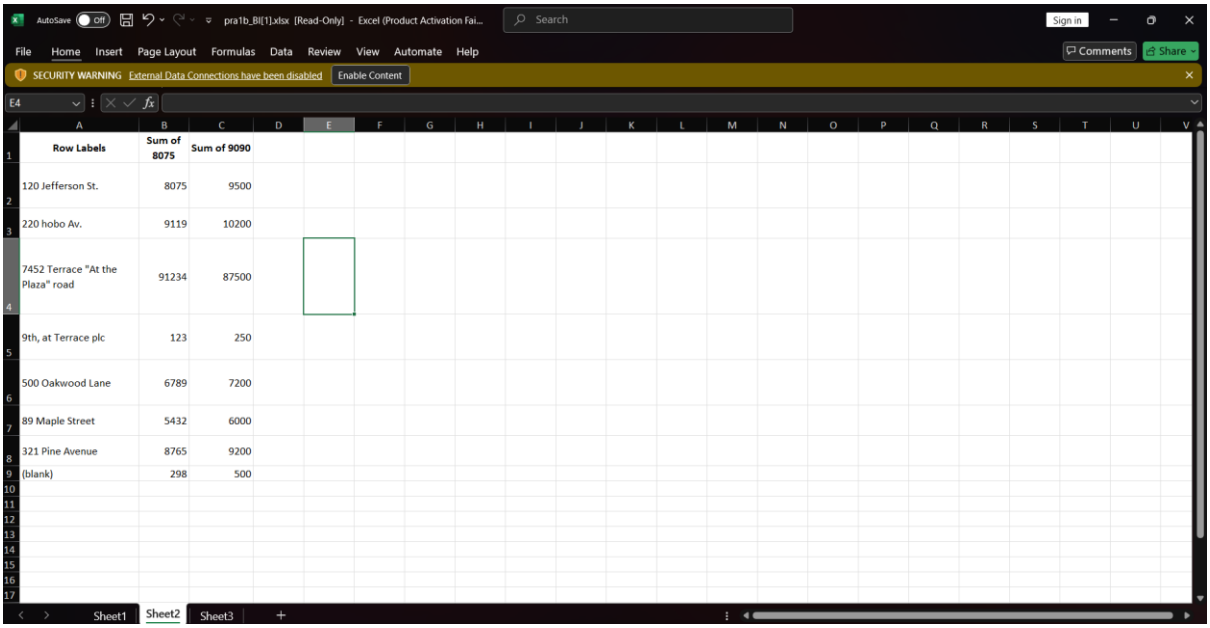


	A	B	C	D	E	F	G
1							
2							
3	Sum of year	Sum of Northeast	Sum of West	Sum of Midwest	Sum of South		
4	14014	453.9	1248.2	597.9	945.92		

b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis.

Solution:

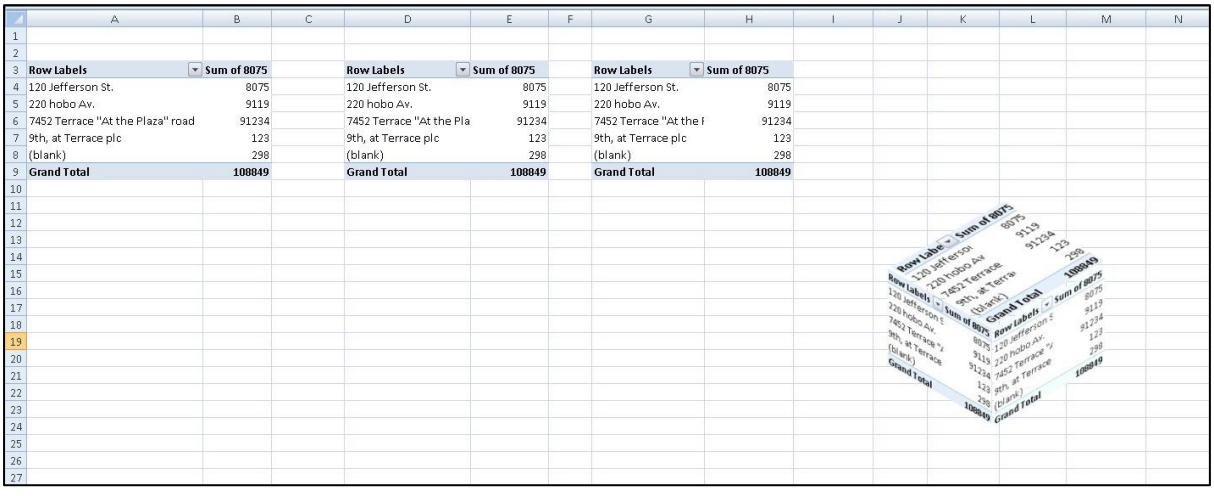
1. Import sample data into Excel.



Row Labels	Sum of 8075	Sum of 9090
120 Jefferson St.	8075	9500
220 hobo Av.	9119	10200
7452 Terrace "At the Plaza" road	91234	87500
9th, at Terrace plc	123	250
500 Oakwood Lane	6789	7200
89 Maple Street	5432	6000
321 Pine Avenue	8765	9200
(blank)	298	500

2. For three sides of the cube, three pivot tables need to be created. Based on the requirement, you can select the fields that you want to display.
3. Initially, you will get a grand total row under every table. This row is **not needed** in the cube. To remove it:
   Click on any cell of the table.
4. Go to the **Design** tab.
5. Select **Off for Rows and Columns** under **Grand Totals** options.
6. To make the table more **presentable**:
7. Select **Show in tabular form** under **Report Layout**.
8. To add a formatted table to the cube:
9. Select the table you want to add.
10. Ensure changes made in formatting are reflected.
11. This needs to be done **individually for all tables**.

Output:



Row Labels	Sum of 8075
120 Jefferson St.	8075
220 hobo Av.	9119
7452 Terrace "At the Plaza" road	91234
9th, at Terrace plc	123
(blank)	298
Grand Total	108849

Row Labels	Sum of 8075
120 Jefferson St.	8075
220 hobo Av.	9119
7452 Terrace "At the Pla	91234
9th, at Terrace plc	123
(blank)	298
Grand Total	108849

Row Labels	Sum of 8075
120 Jefferson St.	8075
220 hobo Av.	9119
7452 Terrace "At the F	91234
9th, at Terrace plc	123
(blank)	298
Grand Total	108849

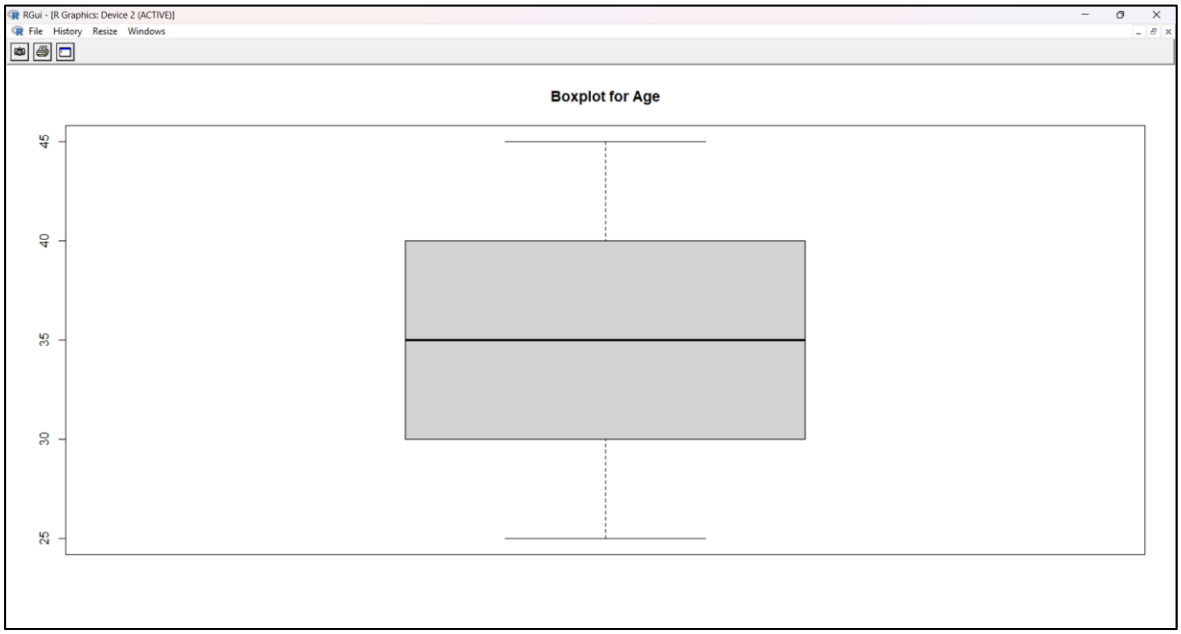
Practical No. 2

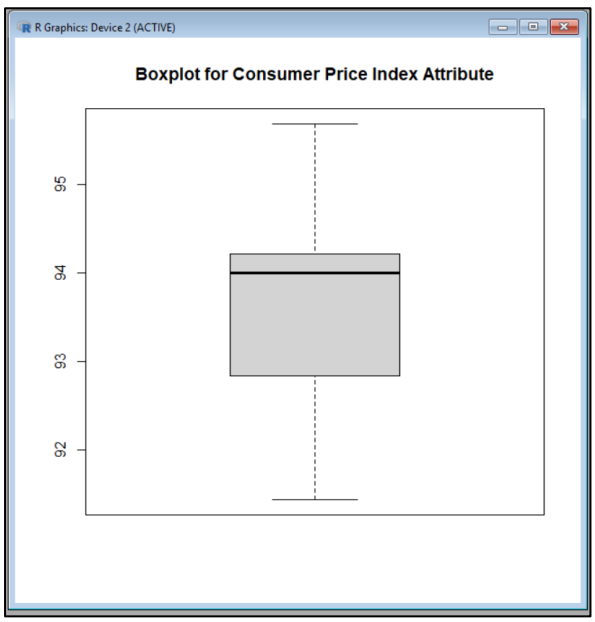
Aim: Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data. Use Excel.

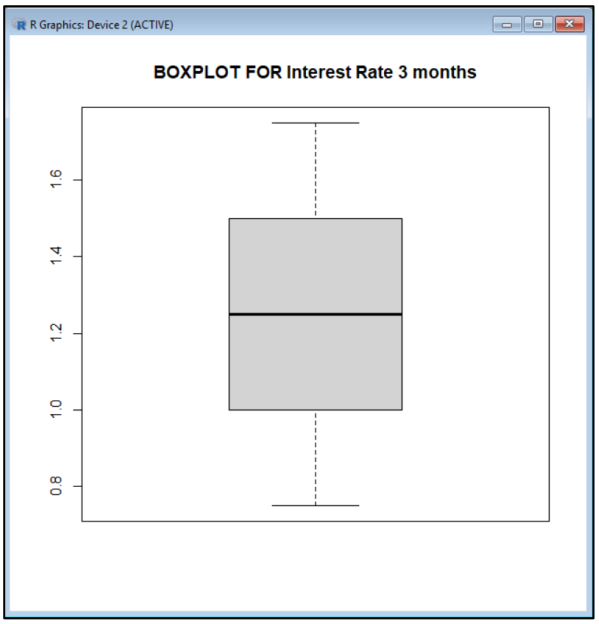
Code:

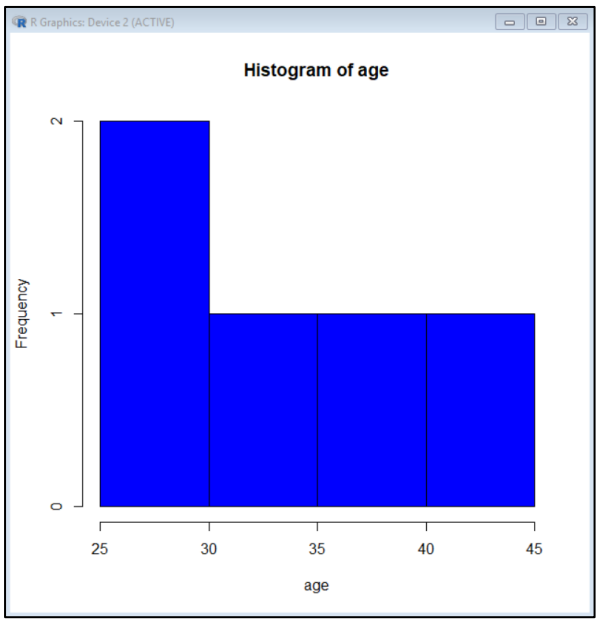
```
op <- read.csv("pra2.csv", header = TRUE)
age <- op$age
job <- op$job
marital <- op$marital  # Corrected from 'martial'
education <- op$education
house <- op$`housing.loan`  # Use backticks if column names contain special characters
car <- op$`car.loan`
cons <- op$cons.price.idx
intr <- op$interest.rate.3months
nr <- op$nr.employed
gr <- op$`grant.loan`
df <- data.frame(
+   age = age,
+   job = job,
+   marital = marital,
+   education = education,
+   house = house,
+   car = car,
+   cons = cons,
+   intr = intr,
+   nr = nr,
+   gr = gr
+ )
boxplot(df$age, main="Boxplot for Age")
summary(df$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    25      30      35      35      40      45
boxplot(df$age, main="Boxplot for Age")
boxplot(df$cons, main="Boxplot for Consumer Price Index Attribute")
boxplot(intr, main="BOXPLOT FOR Interest Rate 3 months")
hist(age, col="blue", border="black")
pie(table(job))
```

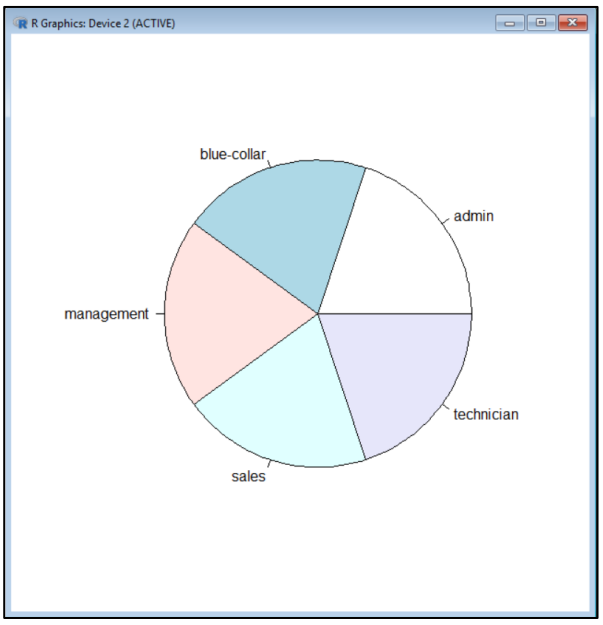
Output:


RGui - [R Graphics: Device 2 (ACTIVE)]
File History Resize Windows
Boxplot for Age
45
40
35
30
25


R Graphics: Device 2 (ACTIVE)
Boxplot for Consumer Price Index Attribute
95
94
93
92


R Graphics: Device 2 (ACTIVE)
BOXPLOT FOR Interest Rate 3 months
1.6
1.4
1.2
1.0
0.8


R Graphics: Device 2 (ACTIVE)
Histogram of age
Frequency
2
1
0
25
30
35
40
45
age


R Graphics: Device 2 (ACTIVE)
blue-collar
admin
management
technician
sales

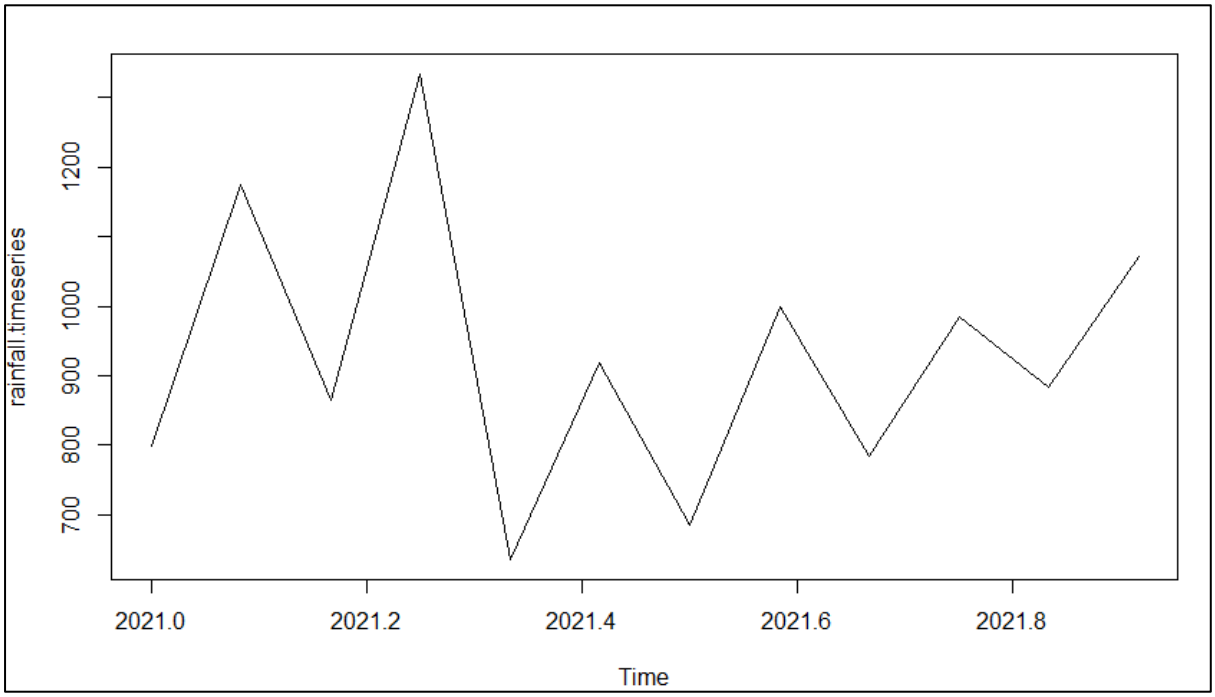
Practical No. 3

Aim: Perform the data classification using classification algorithm using R/Python.

Code:

```
rainfall< c(799,1174.8,865.1,1334.6,635.4,9 18.5,685.5,998.6,784.2,985,882.8,1071)
rainfall.timeseries<-ts(rainfall,start=c(2012, 1),frequency=12)
print(rainfall.timeseries)
png(file="rainfall.png")
 plot(rainfall.timeseries)
dev.off()
```

Output:



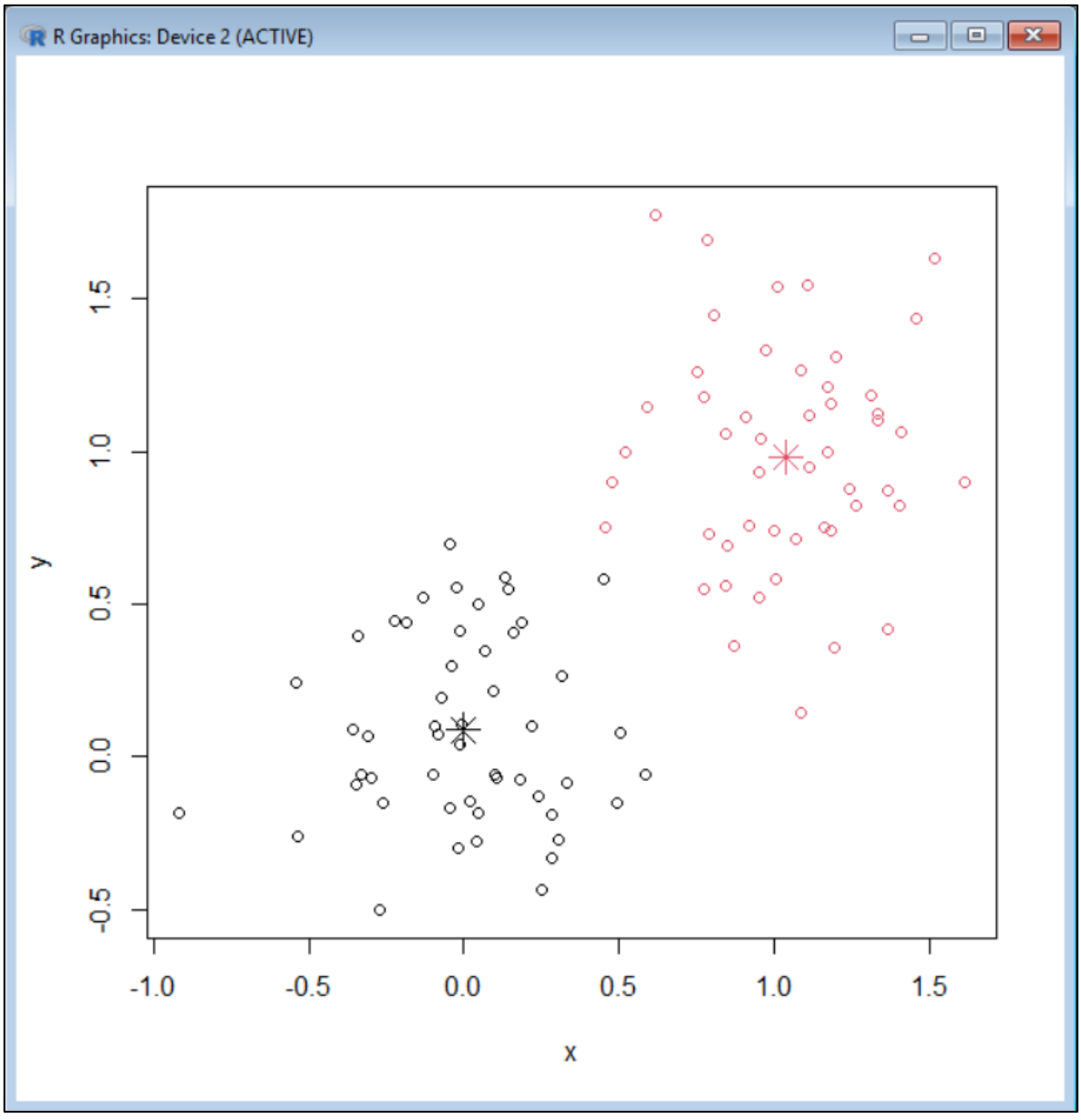
Practical No. 4

Aim: Perform the data clustering using clustering algorithm using R/Python.

Code:

```
require(graphics)
x<-rbind(matrix(rnorm(100,sd=0.3),ncol=2),
+ +
+ + +      matrix(rnorm(100,mean=1,sd=0.3),ncol=2))
colnames(x)<-c("x","y")
(cl<-kmeans(x,2))
plot(x, col = cl$cluster)
points(cl$centers, col = 1:2,pch = 8,cex =2)
```

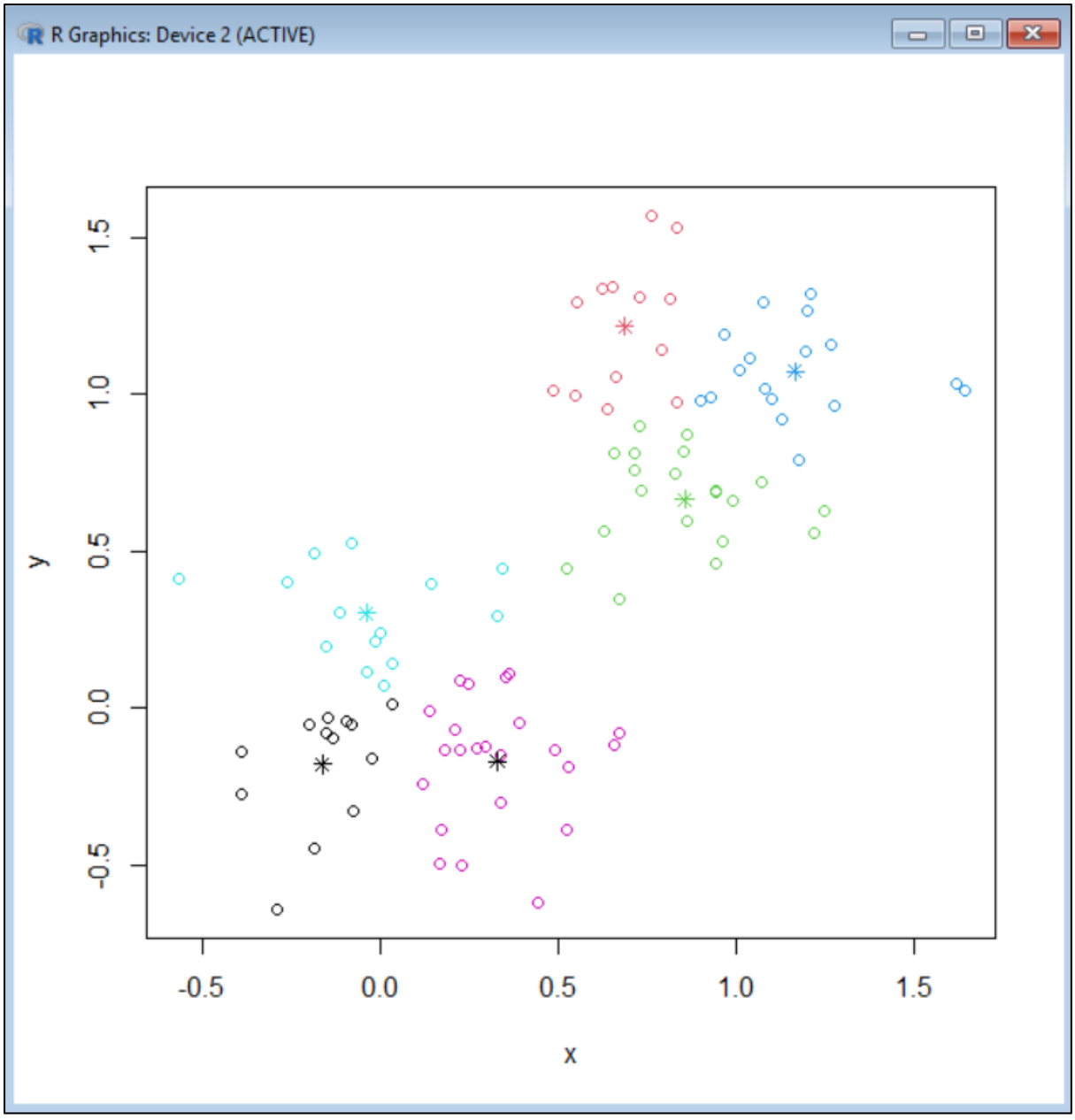
Output:



Code:

```
kmeans(x,1)$withinss
(cl <- kmeans(x,6,nstart=29))
plot(x,col=cl$cluster)
points(cl$centers,col = 1:5,pch=8)
```

Output:



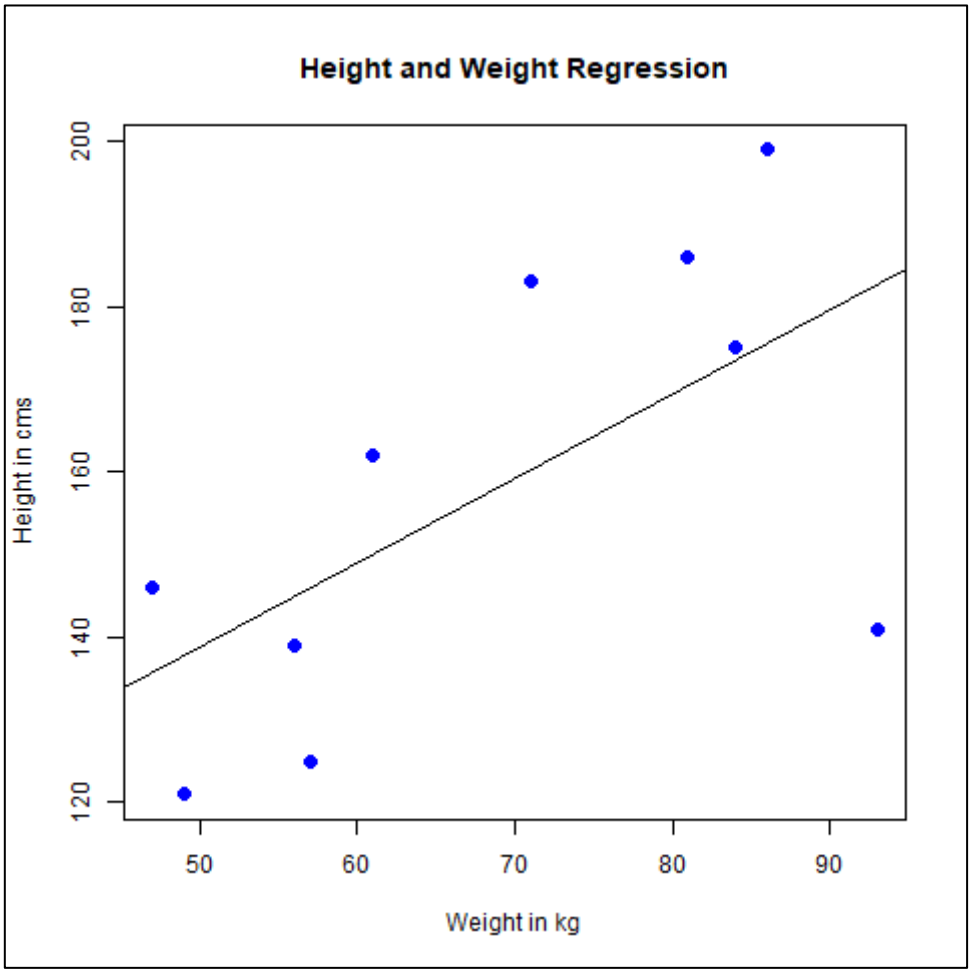
Practical No. 5

Aim: Perform the Linear regression on the given data warehouse data using R/Python.

Code:

```
x <- c(141, 175, 139, 186, 125, 146, 199, 183, 162, 121)
y <- c(93, 84, 56, 81, 57, 47, 86, 71, 61, 49)
relation <- lm(y~x)
print(relation)
print(summary(relation))
a<-data.frame(x=170)
result<-predict(relation,a)
print(result)
png(file = "linearregression.png")
 plot(y,x,col="blue",main="Height and Weight
Regression",abline(lm(x~y)),cex=1.3,pch=16,xlab="Weight in kg",ylab="Height in cms")
```

Output:



Practical No. 6

Aim: Perform the logistic regression on the given data warehouse data using R/Python.

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
df = pd.read_csv('data.csv')
print(df.head())
print(df.isnull().sum())
df = df.dropna()
X = df.iloc[:, :-1]  # Features (all columns except last)
y = df.iloc[:, -1]   # Target variable (last column)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print("Classification Report:\n", classification_report(y_test, y_pred))
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

Output:

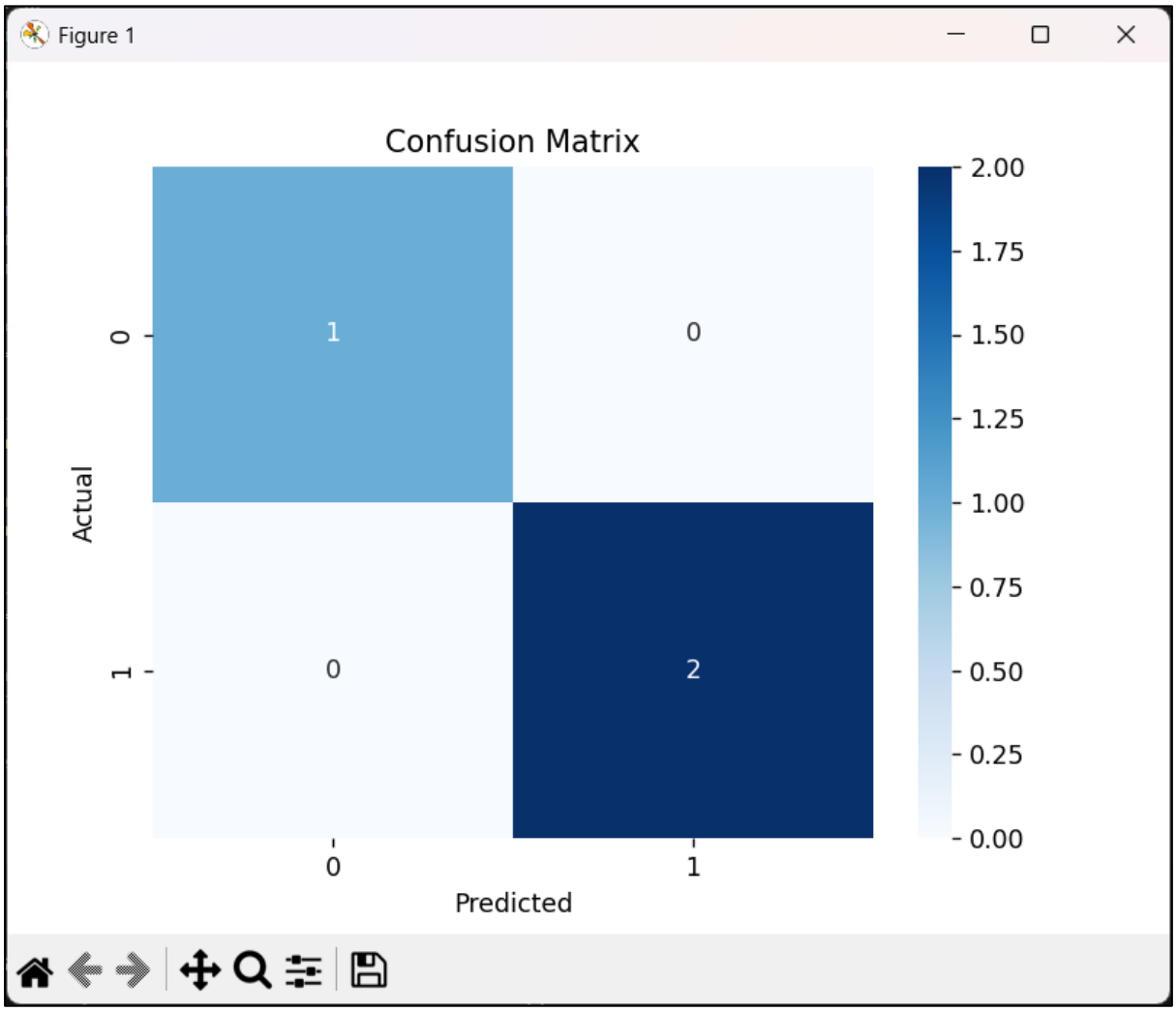


Figure 1
Confusion Matrix
1
0
0
2
Actual
0
1
0
1
Predicted
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

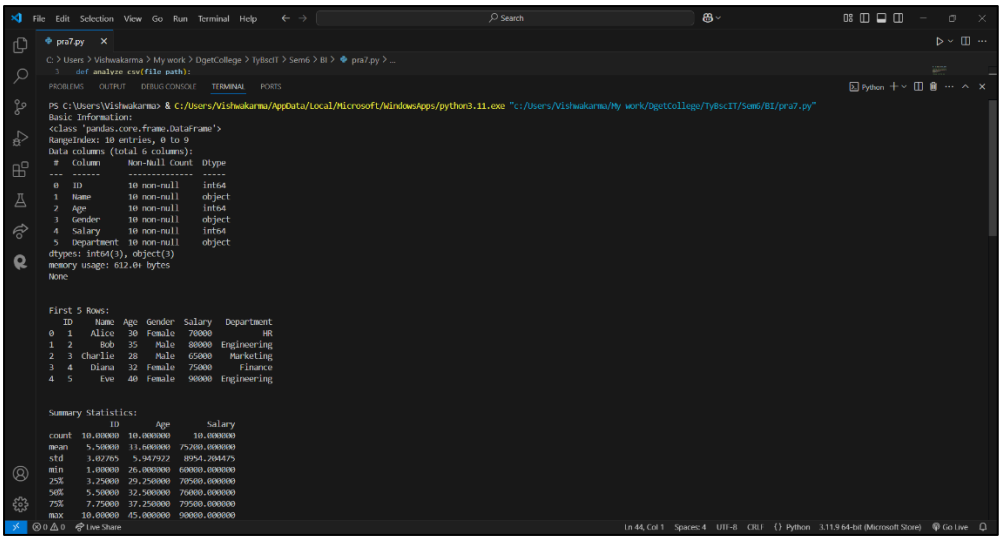
Practical No. 7

Aim: Write a Python program to read data from a CSV file, perform simple data analysis, and generate basic insights. (Use Pandas is a Python library).

Code:

```
import pandas as pd
def analyze_csv(file_path):
    df = pd.read_csv(file_path)
    print("Basic Information:")
    print(df.info())
    print("\n")
    print("First 5 Rows:")
    print(df.head())
    print("\n")
    print("Summary Statistics:")
    print(df.describe())  # This will work correctly as it excludes non-numeric data
    print("\n")
    print("Missing Values:")
    print(df.isnull().sum())
    print("\n")
    categorical_columns = df.select_dtypes(include=['object']).columns
    for col in categorical_columns:
        print(f"Unique values in {col}:")
        print(df[col].value_counts())
        print("\n")
    print("Numerical Data Only:")
    numeric_df = df.select_dtypes(include=['number'])  # Select only numeric columns
    print(numeric_df.corr())
    print("\n")
if __name__ == "__main__":
    file_path = "sample_data.csv"  # Ensure the correct CSV file path
    analyze_csv(file_path)
```

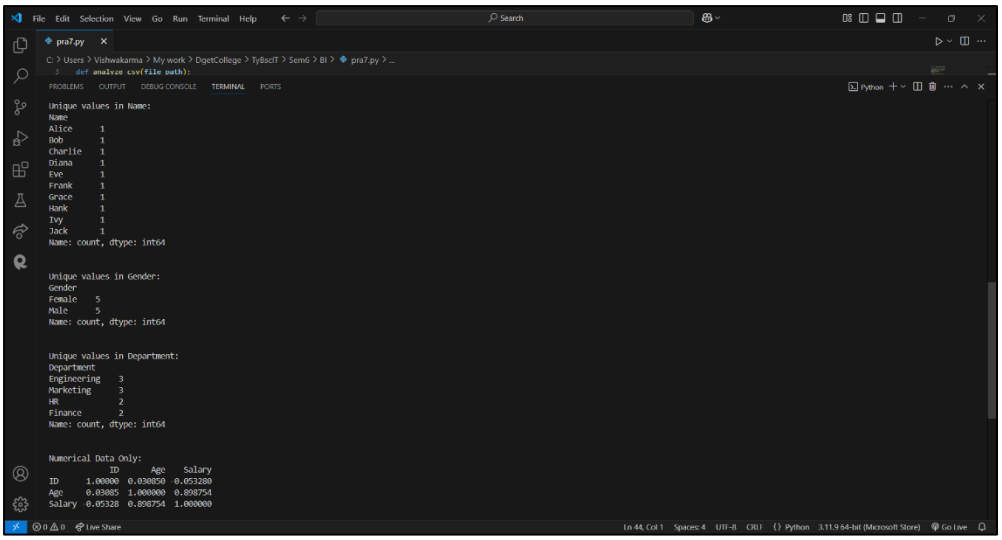
Output:



```
PS C:\Users\Vishwakarma> & C:/Users/Vishwakarma/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Vishwakarma/My work/DgetCollege/TyBscIT/Sem6/BI/pra7.py"
Basic Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   ID          10 non-null     int64
 1   Name        10 non-null     object
 2   Age         10 non-null     int64
 3   Gender      10 non-null     object
 4   Salary      10 non-null     int64
 5   Department  10 non-null     object
dtypes: int64(3), object(3)
memory usage: 612.0+ bytes
None


First 5 Rows:
   ID     Name  Age  Gender  Salary   Department
0   1    Alice   30  Female   70000           HR
1   2      Bob   35    Male   80000  Engineering
2   3  Charlie   28    Male   65000    Marketing
3   4    Diana   32  Female   75000      Finance
4   5      Eve   40  Female   90000  Engineering


Summary Statistics:
              ID        Age        Salary
count  10.000000  10.000000     10.000000
mean    5.500000  33.600000  75200.000000
std     3.02765   5.947922   8954.2044/5
min     1.00000   26.000000  60000.000000
25%     3.25000   29.250000  70500.000000
50%     5.50000   32.500000  76000.000000
75%     7.75000   37.250000  79500.000000
max    10.00000   45.000000  90000.000000
```

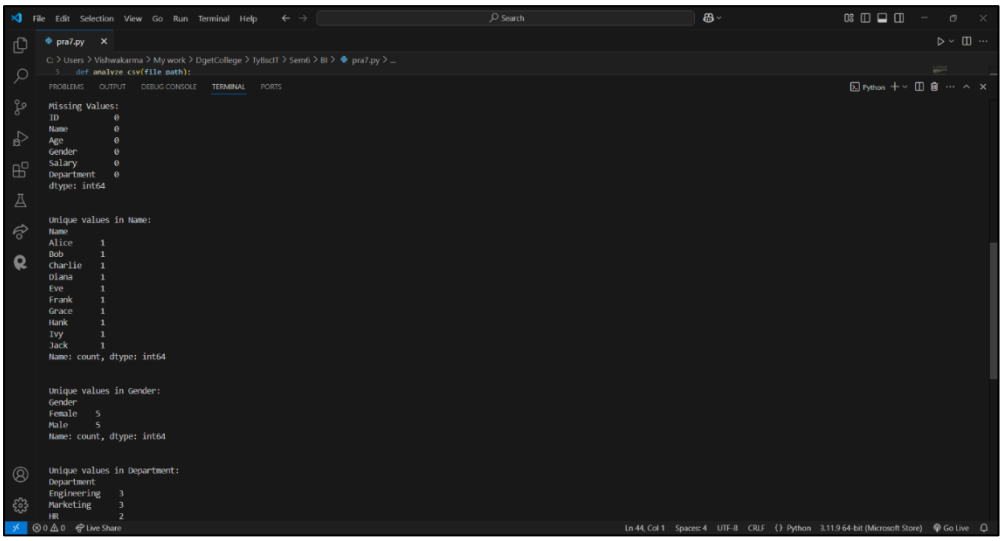


```
Unique values in Name:
Name
Alice      1
Bob        1
Charlie    1
Diana      1
Eve        1
Frank      1
Grace      1
Hank       1
Ivy        1
Jack       1
Name: count, dtype: int64


Unique values in Gender:
Gender
Female    5
Male      5
Name: count, dtype: int64


Unique values in Department:
Department
Engineering    3
Marketing      3
HR             2
Finance        2
Name: count, dtype: int64


Numerical Data Only:
              ID       Age    Salary
ID      1.00000  0.030850  0.053280
Age     0.03085  1.000000  0.898754
Salary  0.05328  0.898754  1.000000
```



```
Missing Values:
ID            0
Name          0
Age           0
Gender        0
Salary        0
Department    0
dtype: int64


Unique values in Name:
Name
Alice      1
Bob        1
Charlie    1
Diana      1
Eve        1
Frank      1
Grace      1
Hank       1
Ivy        1
Jack       1
Name: count, dtype: int64


Unique values in Gender:
Gender
Female    5
Male      5
Name: count, dtype: int64


Unique values in Department:
Department
Engineering    3
Marketing      3
HR             2
```

Practical No.8

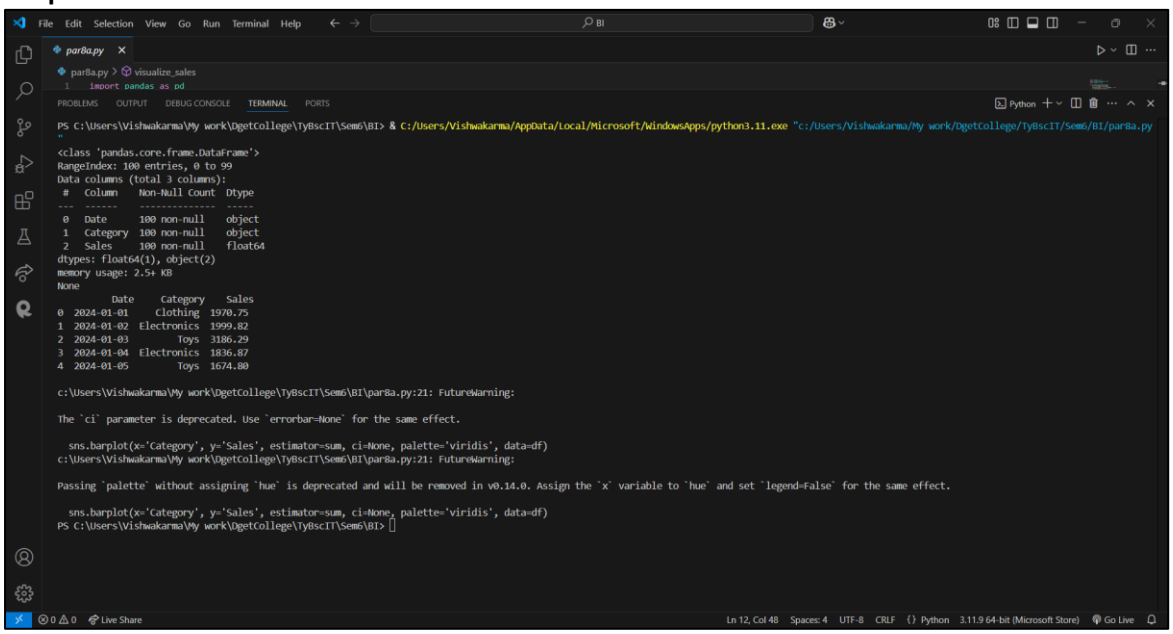
Aim: Perform data visualization

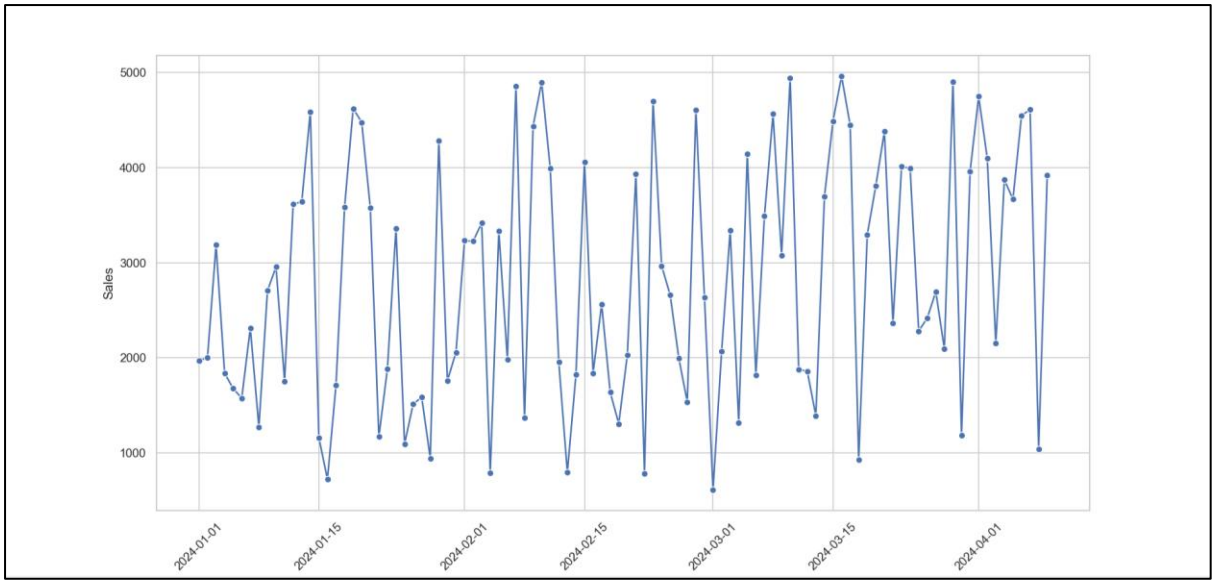
A. Perform data visualization using Python on any sales data

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
def visualize_sales(file_path):
    df = pd.read_csv(file_path)
    print(df.info(), "\n", df.head(), "\n")
    sns.set(style="whitegrid")
    if 'Date' in df.columns:
        df['Date'] = pd.to_datetime(df['Date'])
        df.sort_values(by='Date', inplace=True)
        sns.lineplot(x='Date', y='Sales', data=df, marker='o')
        plt.xticks(rotation=45)
        plt.show()
    sns.histplot(df['Sales'], bins=20, kde=True, color='blue')
    plt.show()
    if 'Category' in df.columns:
        sns.barplot(x='Category', y='Sales', estimator=sum, ci=None, palette='viridis', data=df)
        plt.xticks(rotation=45)
        plt.show()
if __name__ == "__main__":
    visualize_sales("sales_data.csv")
```

Output:




Sales
5000
4000
3000
2000
1000
2024-01-01
2024-01-15
2024-02-01
2024-02-15
2024-03-01
2024-03-15
2024-04-01

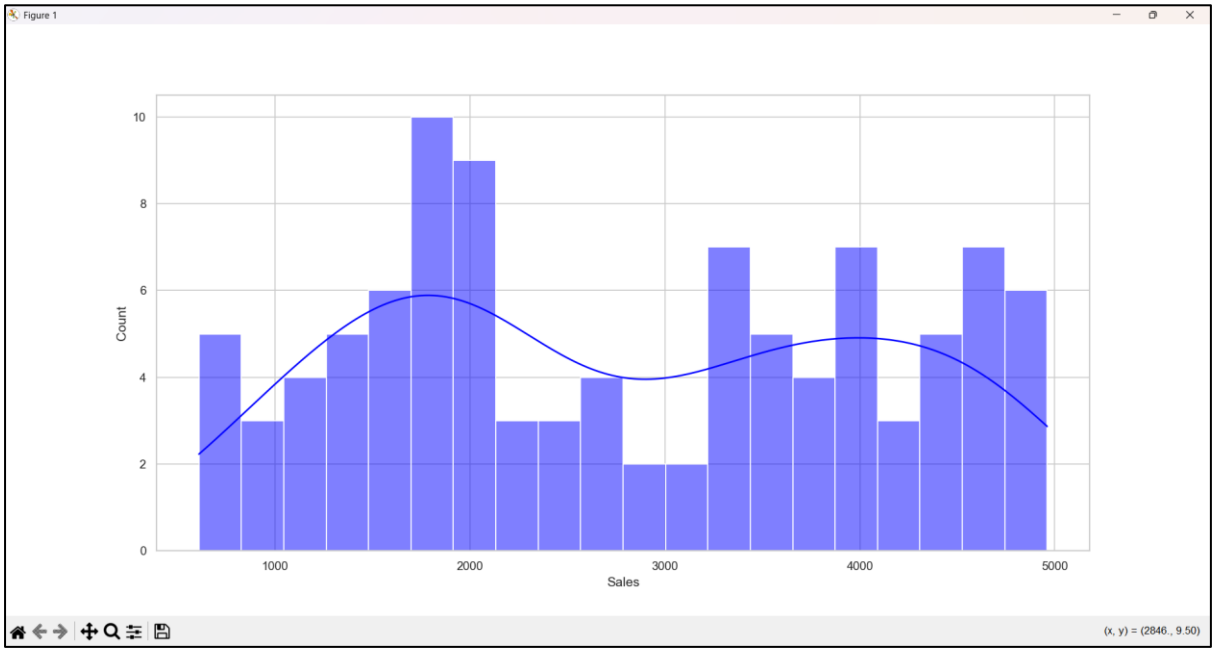


Figure 1
Count
Sales
10
8
6
4
2
0
1000
2000
3000
4000
5000
(x, y) = (2846., 9.50)

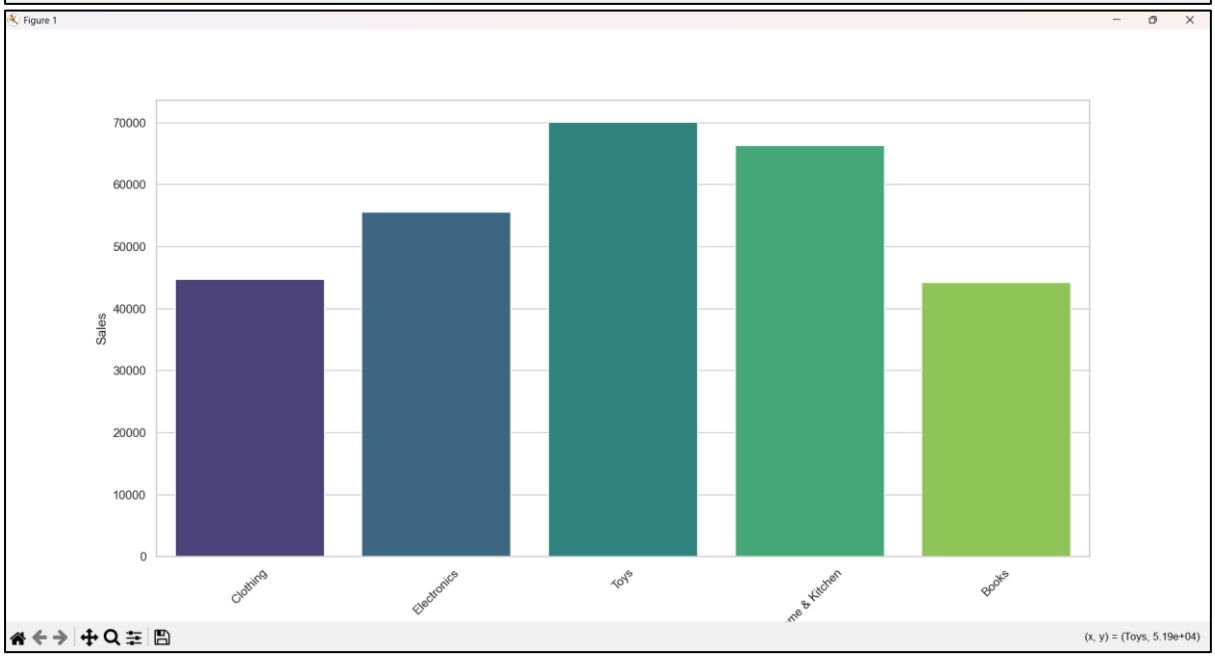


Figure 1
Sales
70000
60000
50000
40000
30000
20000
10000
0
Clothing
Electronics
Toys
Books
(x, y) = (Toys, 5.19e+04)

B. Perform data visualization using PowerBI on any sales data.

Solution:

Step1: Open Power BI

Step2: Take blank report template.

Step3: download "sales_data.xlsx" file from google chrome.

Step4: go to home -> excel workbook -> select the excel file from download.

Step5: Open -> select excel file >select transform data

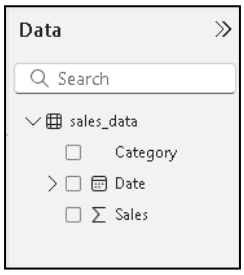
Step6: your data will be enable in “Power Query Editor window

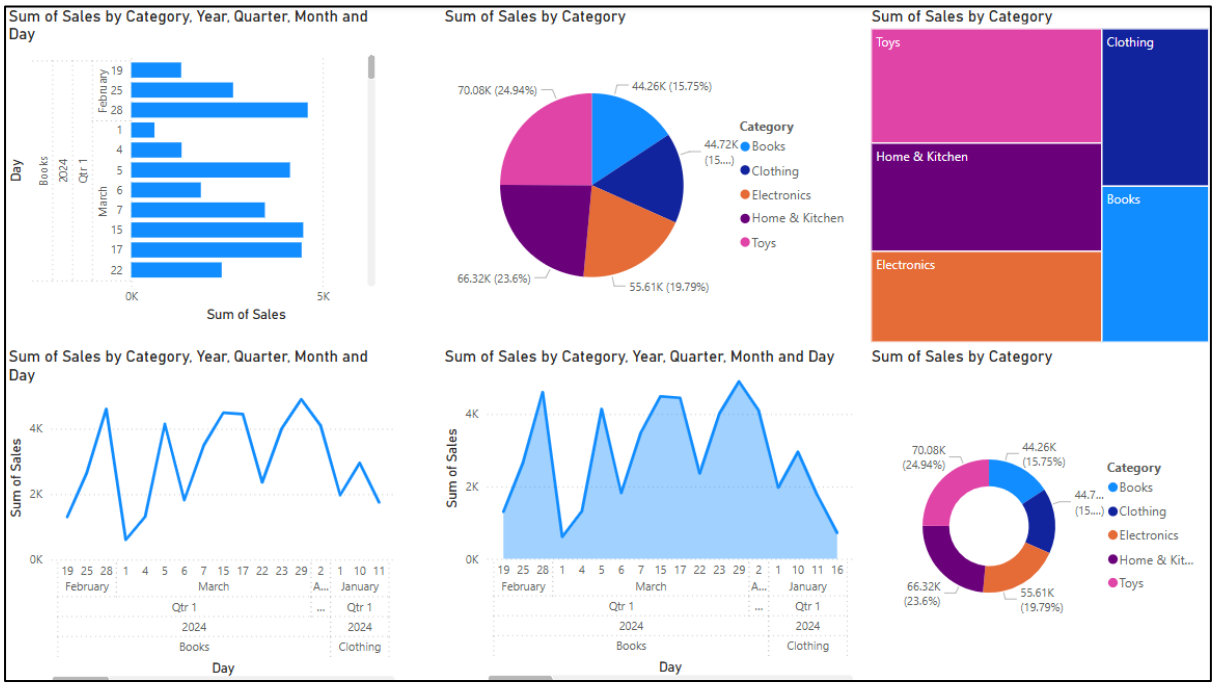
Step 7: go to back again on Power BI desktop window ->click on apply changes.

Step 8: below the file menu -> click on report view -> click right side of report view you will see the visualization select it.

Step 9: take any graph you want to see the visualized data -> after the visualization step you will see the near data option click on it and select the column you want to add in your graph.

Output:





Practical No.9

Aim: Create the Data staging area for the selected database using SQL.

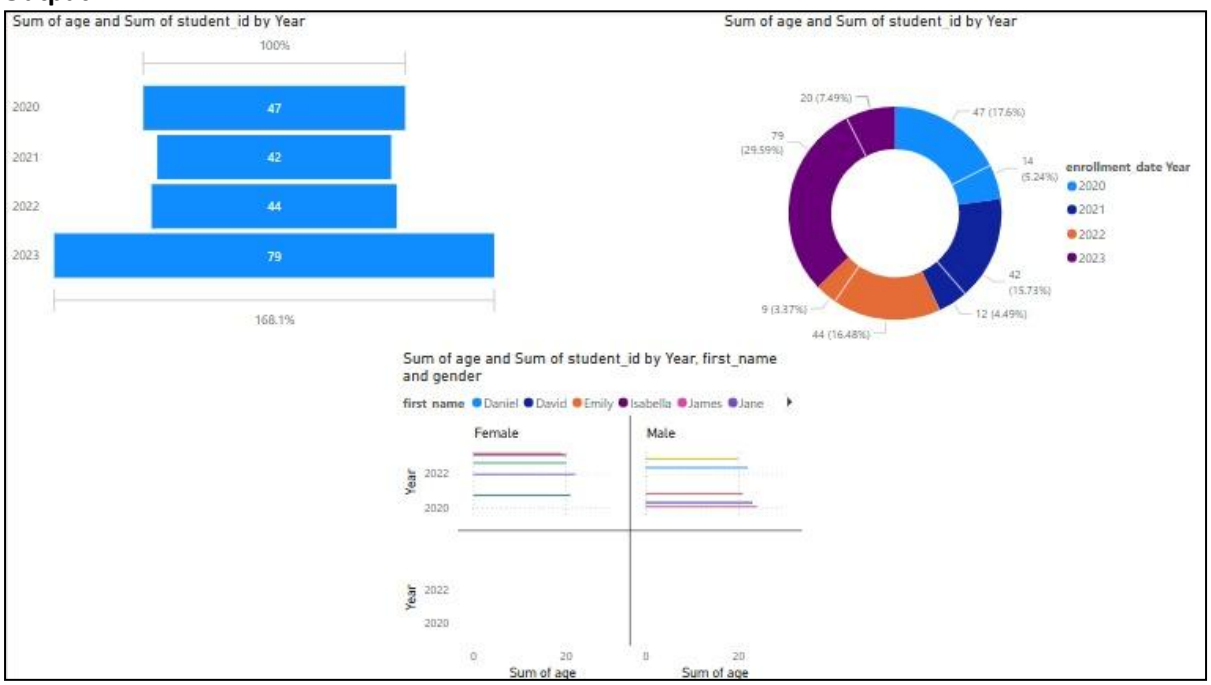
Query:

```
CREATE TABLE students (
   student_id INT PRIMARY KEY,
   first_name VARCHAR(50),
   last_name VARCHAR(50),
   age INT,
   gender VARCHAR(10),
   grade VARCHAR(5),
   enrollment_date DATE
);
```

Data:

```
INSERT INTO students (student_id, first_name, last_name, age, gender, grade, enrollment_date)
VALUES
(1, 'John', 'Doe', 20, 'Male', 'A', '2023-09-01'),
(2, 'Jane', 'Smith', 22, 'Female', 'B', '2022-05-15'),
(3, 'Emily', 'Johnson', 19, 'Female', 'A', '2023-01-10'),
(4, 'Michael', 'Williams', 21, 'Male', 'C', '2021-08-23'),
(5, 'David', 'Brown', 23, 'Male', 'B', '2020-12-05'),
(6, 'Sophia', 'Davis', 20, 'Female', 'A', '2023-02-10'),
(7, 'Daniel', 'Martinez', 22, 'Male', 'B', '2022-07-19'),
(8, 'Olivia', 'Garcia', 21, 'Female', 'A', '2021-11-30'),
(9, 'James', 'Rodriguez', 24, 'Male', 'C', '2020-03-15'),
(10, 'Isabella', 'Wilson', 20, 'Female', 'B', '2023-05-25');
```

Output:



Practical No.10

Aim: Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model.

Solution:

Step 1: Start BIDS Environment

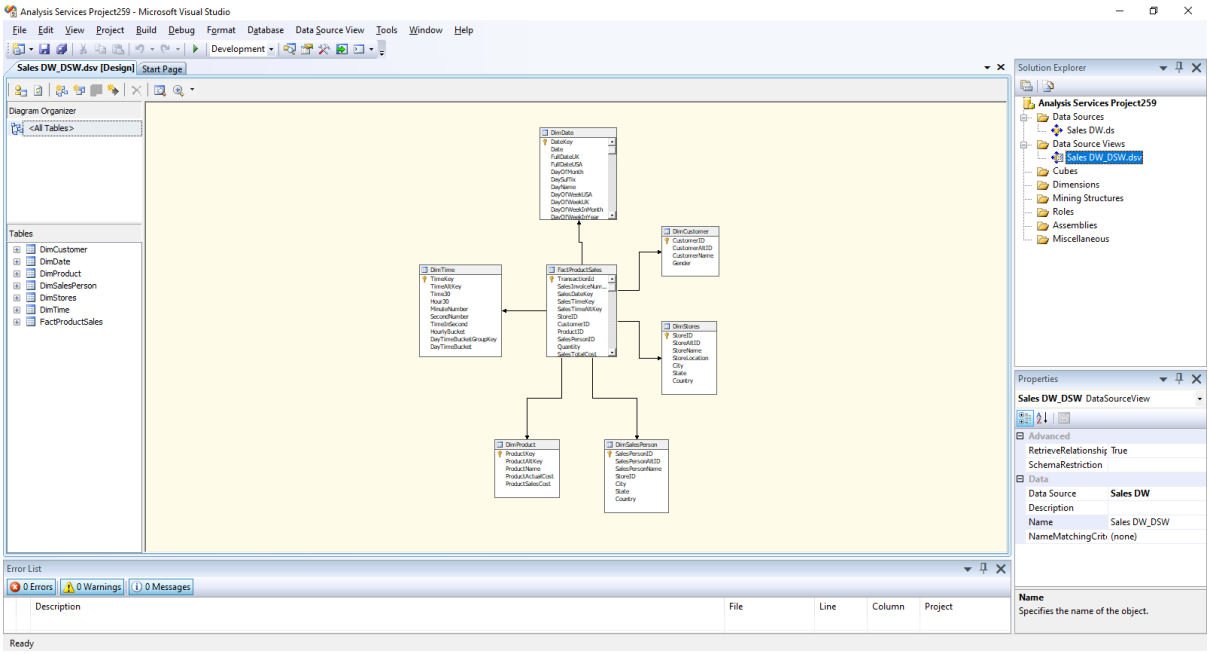
Click on Start Menu -> Microsoft SQL Server 2008 R2 -> Click SQL Server Business Intelligence Development Studio.

Step 2: Start Analysis Services Project

Click File -> New -> Project ->Business Intelligence Projects ->select Analysis Services Project-> Assign Project Name (cubepract4)->Click OK

Step 3: Create New Data Source

Step 4: Create New Data Source View



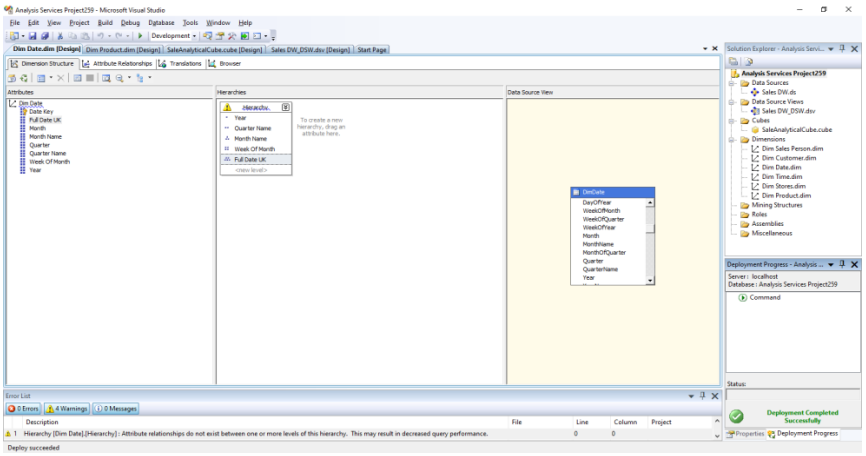
Steps 5: Create New Cube

Step 6: In Solution Explorer, Double Click on Dimension Dim Product-> Drag and Drop Product Name from Table in Data Source View and Add in Attribute Pane at left side.

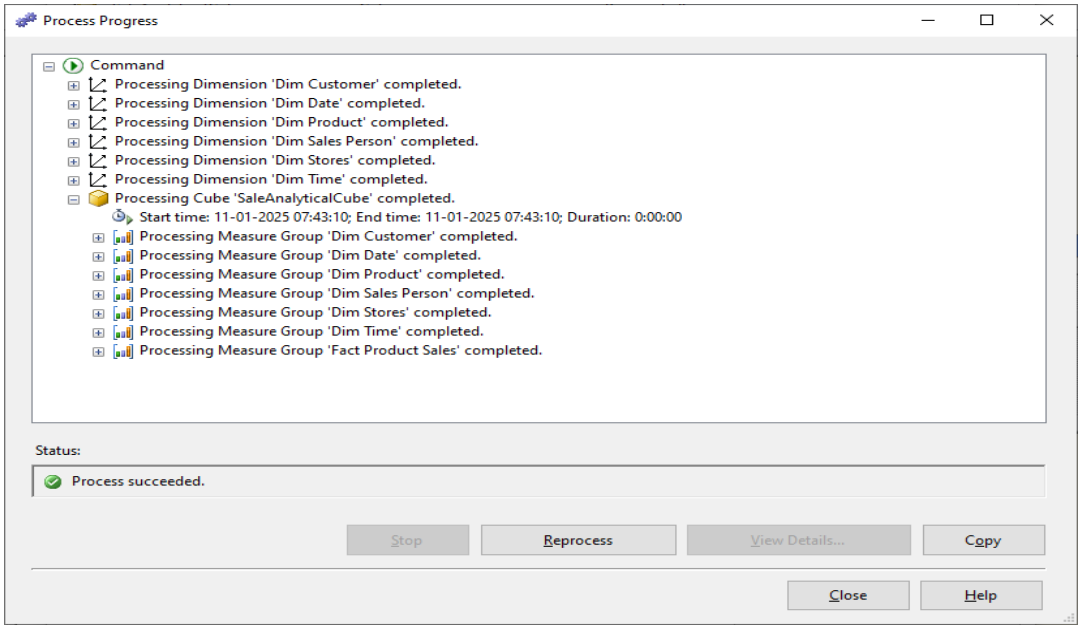
Step 7: Double click On Dim Date dimension -> Drag and Drop Fields from Table shown in Data Source View to Attributes-> Drag and Drop attributes from leftmost pane of attributes to middle pane of Hierarchy.

Drag fields in sequence from Attributes to Hierarchy window (Year, Quarter Name. Month Name, Week of the Month, Full Date UK),

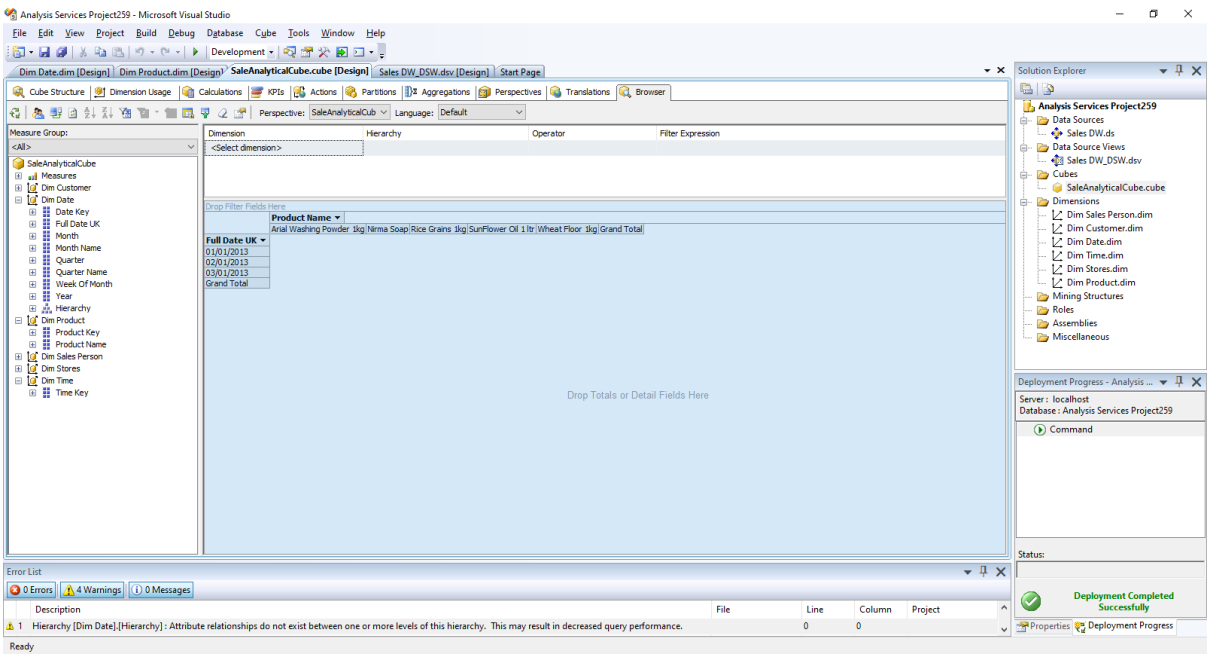
Step 8: Deploy the Cube



Step 9: Process the Cube In Solution Explorer, right click on Project Name (SalesDataAnalysis) -> Click Process Click on Run button to process the Cube



Step 10: Browse the Cube for Analysis In Solution Explorer, right click on Cube Name (SalesAnalyticalCube) -- > Click Browse



Drag and drop measures in to Detail fields, & Drag and Drop Dimension Attributes in Row Field or Column fields.

Now to Browse Our Cube

1.. Product Name Drag & Drop into Column

2. Full Date UK Drag & Drop into Row Field

3. FactProductSalesCount Drop this measure in Detail area