EXPERIMENT – 2

PANDAS LIBRARY – BASIC CONCEPT

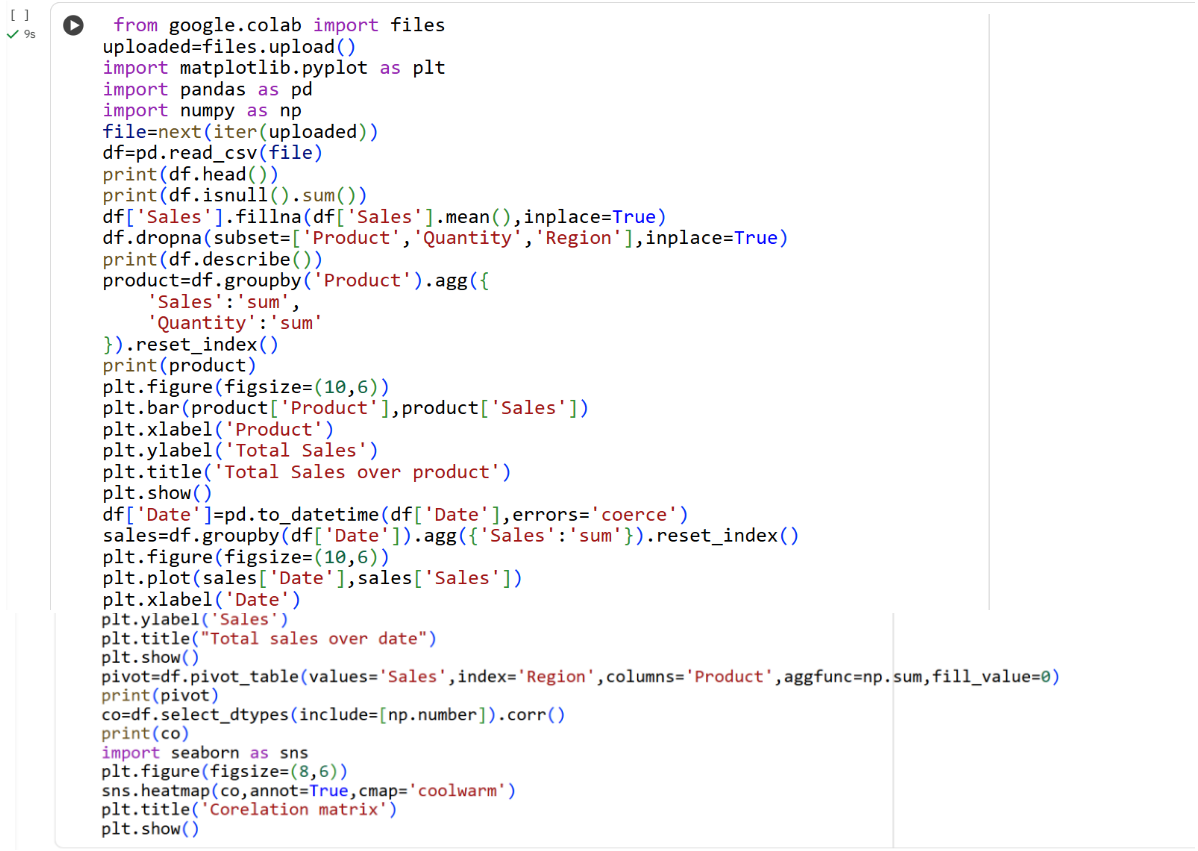
Aim:

To upload and analyze dataset given in csv format and perform data preprocessing and visualization

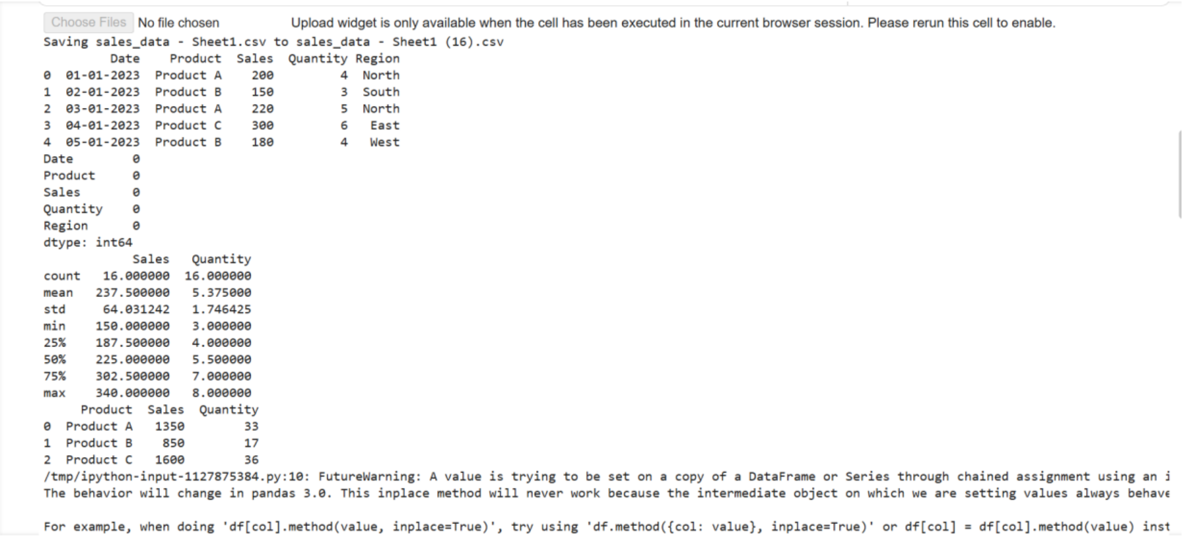
Procedure:

- Upload the csv file and read it
- Import the necessities like pandas ,numpy and seaborn
- Now visualize bar plot for sales over products , line plot for sales over time and correlation matrix

Program:



```
from google.colab import files
uploaded=files.upload()
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
file=next(iter(uploaded))
df=pd.read_csv(file)
print(df.head())
print(df.isnull().sum())
df['Sales'].fillna(df['Sales'].mean(),inplace=True)
df.dropna(subset=['Product','Quantity','Region'],inplace=True)
print(df.describe())
product=df.groupby('Product').agg({
    'Sales':'sum',
    'Quantity':'sum'
}).reset_index()
print(product)
plt.figure(figsize=(10,6))
plt.bar(product['Product'],product['Sales'])
plt.xlabel('Product')
plt.ylabel('Total Sales')
plt.title('Total Sales over product')
plt.show()
df['Date']=pd.to_datetime(df['Date'],errors='coerce')
sales=df.groupby(df['Date']).agg({'Sales':'sum'}).reset_index()
plt.figure(figsize=(10,6))
plt.plot(sales['Date'],sales['Sales'])
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title("Total sales over date")
plt.show()
pivot=df.pivot_table(values='Sales',index='Region',columns='Product',aggfunc=np.sum,fill_value=0)
print(pivot)
co=df.select_dtypes(include=[np.number]).corr()
print(co)
import seaborn as sns
plt.figure(figsize=(8,6))
sns.heatmap(co,annot=True,cmap='coolwarm')
plt.title('Corelation matrix')
plt.show()
```

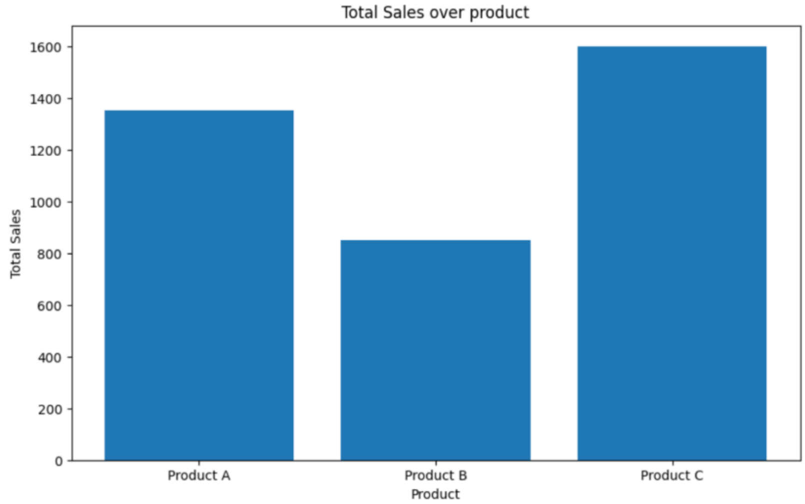


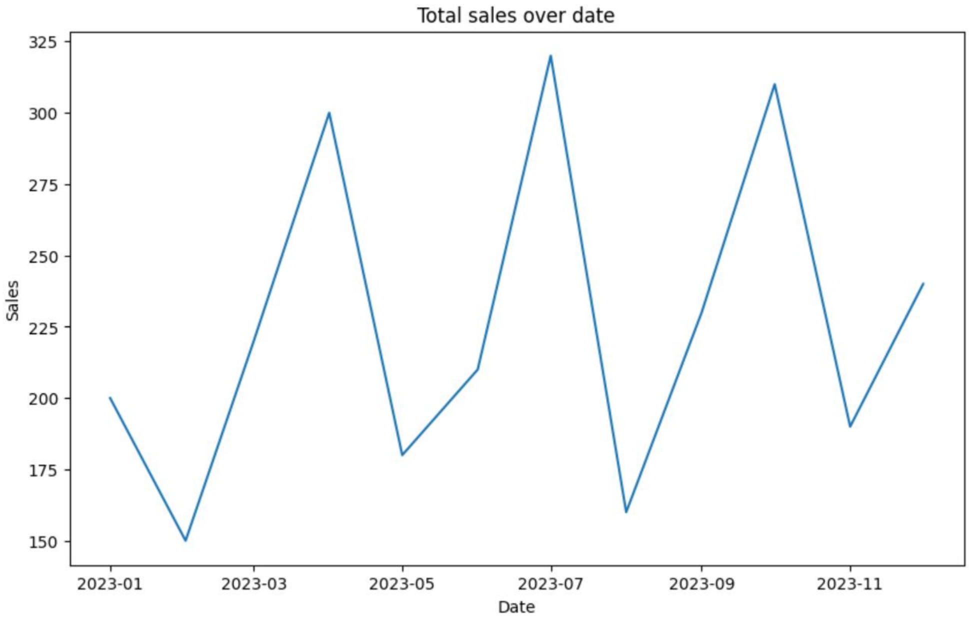
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

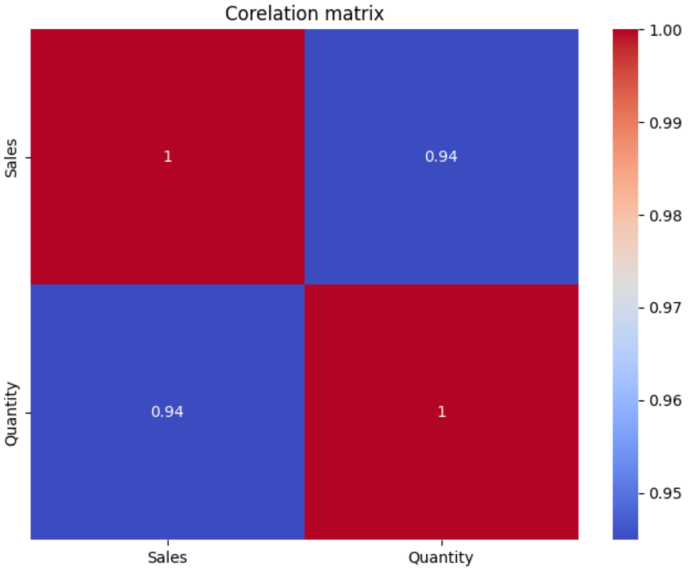
```
Saving sales_data - Sheet1.csv to sales_data - Sheet1 (16).csv
         Date    Product  Sales  Quantity Region
0  01-01-2023  Product A    200         4  North
1  02-01-2023  Product B    150         3  South
2  03-01-2023  Product A    220         5  North
3  04-01-2023  Product C    300         6   East
4  05-01-2023  Product B    180         4   West
Date        0
Product     0
Sales       0
Quantity    0
Region      0
dtype: int64
            Sales   Quantity
count   16.000000  16.000000
mean   237.500000   5.375000
std     64.031242   1.746425
min    150.000000   3.000000
25%    187.500000   4.000000
50%    225.000000   5.500000
75%    302.500000   7.000000
max    340.000000   8.000000
     Product  Sales  Quantity
0  Product A   1350        33
1  Product B    850        17
2  Product C   1600        36
/tmp/ipython-input-1127875384.py:10: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an i
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behave

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inst
```

|< < 1 / 2 > >|







Result:

Thus the python program to analyze the given dataset and perform data processing is executed and verified