Experiment-1

IMPLEMENT PROGRAM FOR TIME SERIES DATA CLEANING, LOADING AND HANDLING TIME SERIES DATA AND PREPROCESSING TECHNIQUES

AIM:
TO WRITE A TO IMPLEMENT PROGRAM FOR TIME SERIES DATA CLEANING, LOADING AND HANDLING TIME SERIES DATA AND PREPROCESSING TECHNIQUES

PROCEDURE:
1) Import necessary libraries.
2)Load the necessary libraries
3)Generate visualizations using different plots like, scatter, line, pie
4)Label the x-axis and y-axis , then give the title

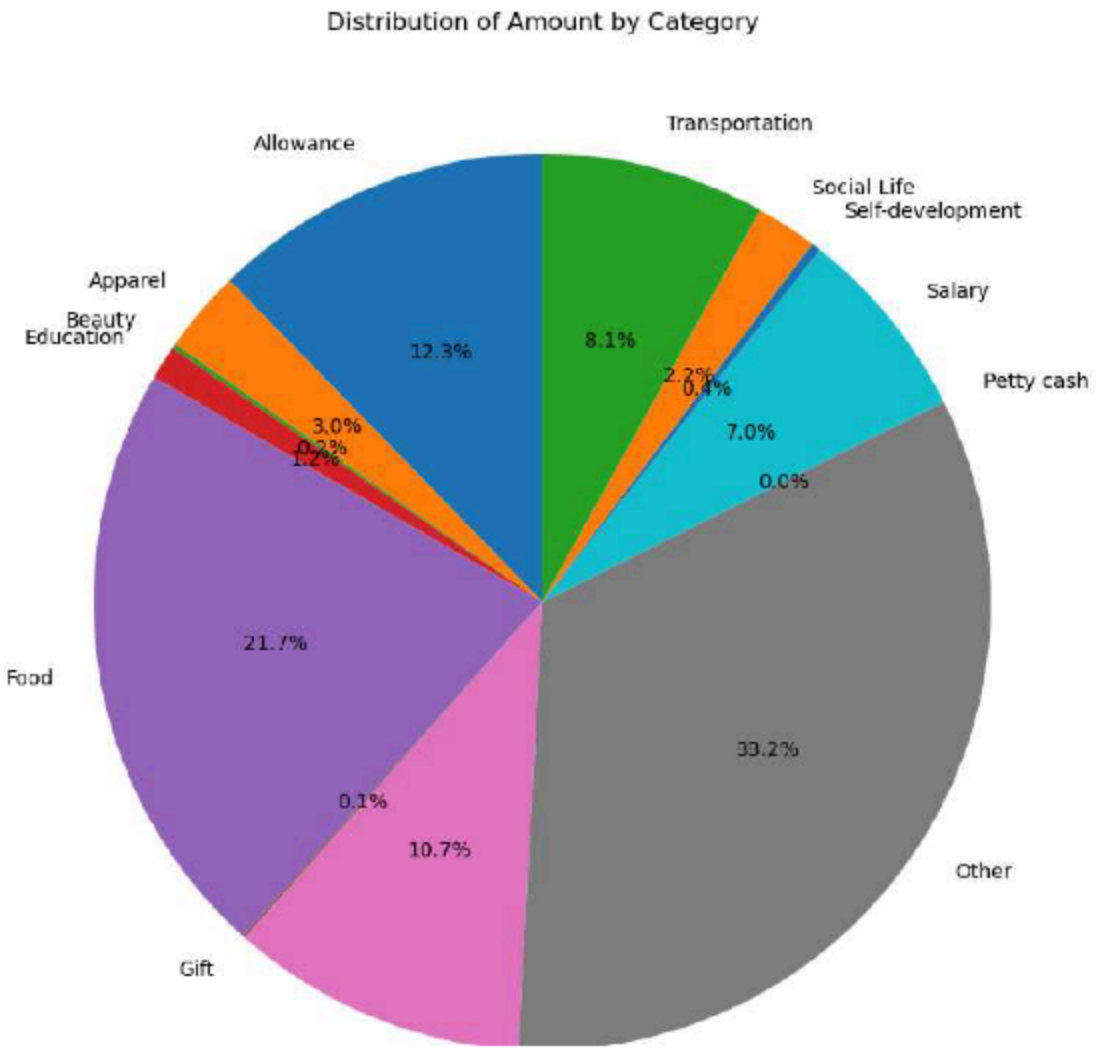
CODE:

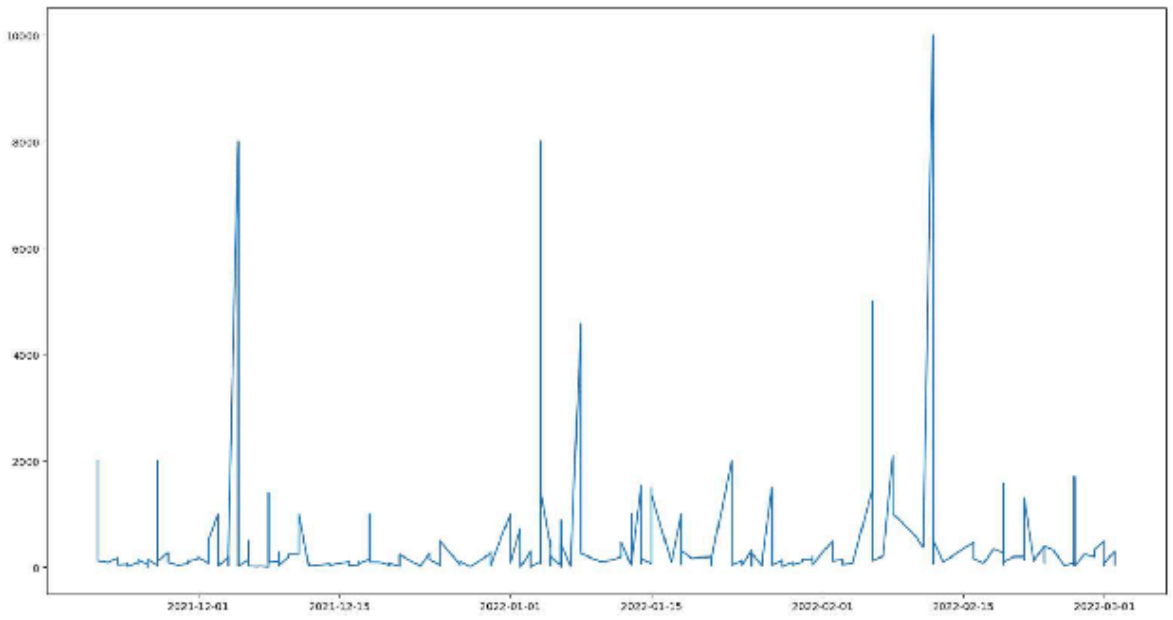
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("expense_data_1.csv")

# Display the first few rows of the dataset
df.head()

# Drop duplicate rows
df.drop_duplicates()

# Check for missing values
df.isna().sum()

# Drop unnecessary columns
df.drop(['Subcategory', "Note.1"], axis=1, inplace=True)
```

```
# Display the first few rows after dropping columns
df.head()

# Generate descriptive statistics
df.describe()

# Display information about the DataFrame
df.info()

# Split the 'Date' column into 'Date' and 'time'
df['time'] = df['Date'].str.split(" ").str[1]
df['Date'] = df['Date'].str.split(" ").str[0]

# Display the first few rows after splitting the 'Date' column
df.head()

# Check unique values in the 'Currency' column
df['Currency'].unique()

# Convert 'Amount' to INR if the currency is USD
df['Amount'] = df.apply(lambda row: row['Amount'] * 93 if row['Currency'] == 'USD' else
row['Amount'], axis=1)

# Drop the 'Currency' and 'Account.1' columns
df.drop(['Currency', 'Account.1'], axis=1, inplace=True)

# Display the first few rows after dropping columns
df.head()

# Convert 'Date' to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Display information about the DataFrame after converting 'Date'
df.info()

# Check unique values in the 'Category' column
df['Category'].unique()

# Check unique values in the 'Account' column
df['Account'].unique()

# Replace 'CUB - online payment' with 'online' in the 'Account' column
df['Account'] = df.apply(lambda row: 'online' if row['Account'] == 'CUB - online payment' else
row['Account'], axis=1)
```

```
# Check unique values in the 'Account' column after replacement
df['Account'].unique()

# Plot a bar plot of 'Category' vs 'Amount'
plt.figure(figsize=(16,18))
sns.barplot(data=df, x='Category', y='Amount')

# Set 'Date' as the index
df.set_index('Date', inplace=True)

# Plot a line plot of 'Date' vs 'Amount'
plt.figure(figsize=(19,10))
plt.plot(df.index, df['Amount'])

# Group by 'Category' and sum the 'Amount'
category_amount = df.groupby('Category')['Amount'].sum().reset_index()

# Plot a pie chart of the distribution of 'Amount' by 'Category'
plt.figure(figsize=(10, 10))
plt.pie(category_amount['Amount'], labels=category_amount['Category'], autopct='%1.1f%%',
startangle=90)
plt.title('Distribution of Amount by Category')
plt.show()

# Plot a heatmap of the correlation between 'INR' and 'Amount'
plt.figure(figsize=(8, 6))
sns.heatmap(df[['INR', 'Amount']].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# Plot a scatter plot of 'INR' vs 'Amount'
plt.figure(figsize=(8, 6))
plt.scatter(df['INR'], df['Amount'], alpha=0.6, color='purple')
plt.title('INR vs Amount')
plt.xlabel('INR')
plt.ylabel('Amount')
plt.grid(True)
plt.show()
```

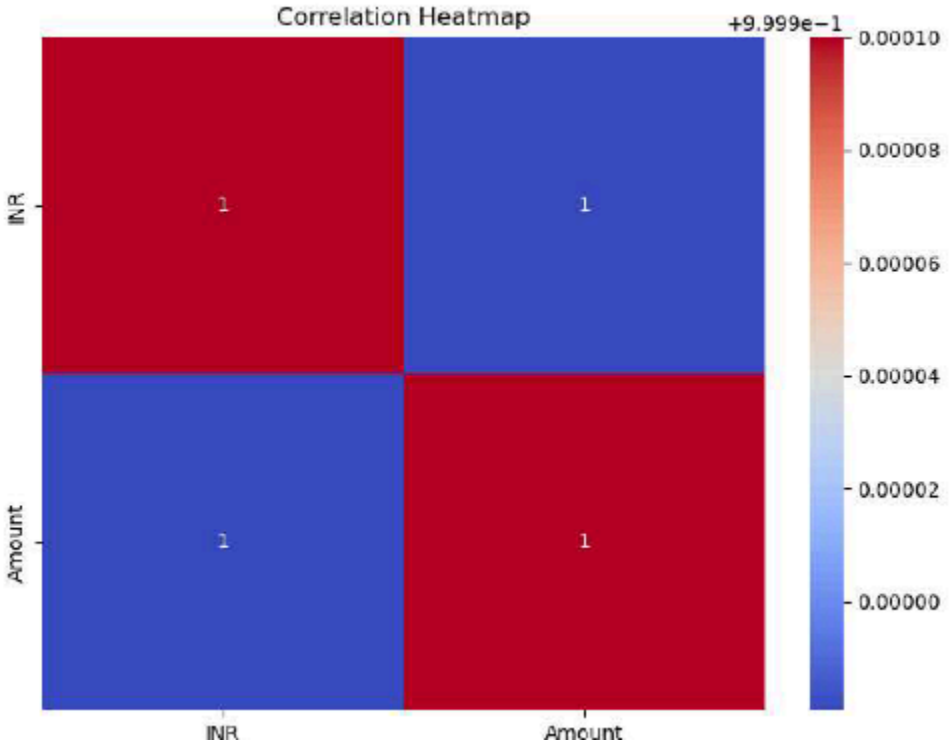
OUTPUT:

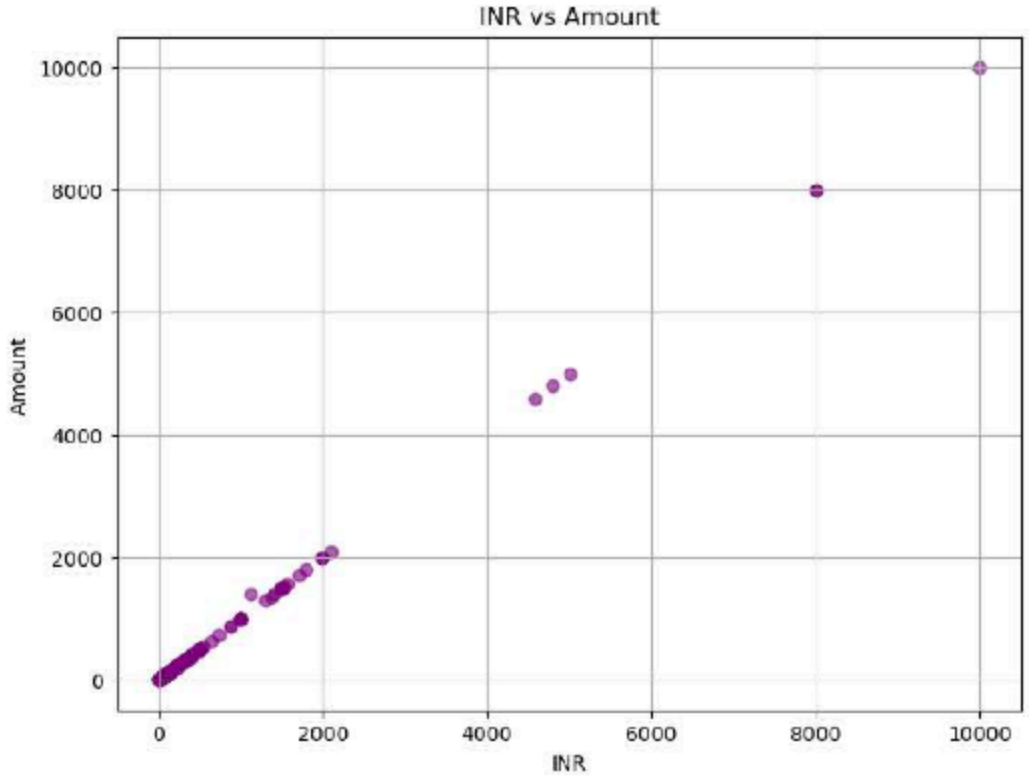

Distribution of Amount by Category
Transportation
Allowance
Social Life
Self-development
Salary
Apparel
Beauty
Education
Petty cash
12.3%
8.1%
2.2%
0.4%
7.0%
3.0%
0.2%
1.2%
0.0%
21.7%
Food
33.2%
0.1%
10.7%
Other
Gift

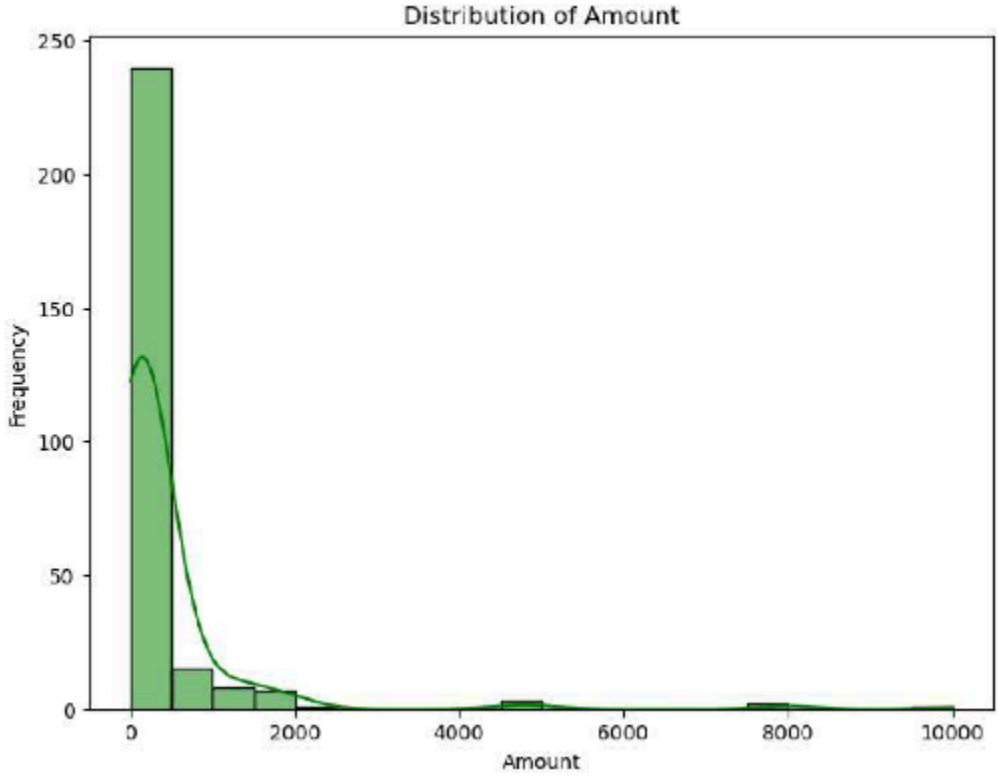

10000
8000
6000
4000
2000
0
2021-12-01
2021-12-15
2022-01-01
2022-01-15
2022-02-01
2022-02-15
2022-03-01

AM

Untitled


Correlation Heatmap
+9.999e−1
0.00010
0.00008
0.00006
0.00004
0.00002
0.00000
INR
Amount
1
1
1
1
INR
Amount


INR vs Amount
Amount
INR
0
2000
4000
6000
8000
10000


Distribution of Amount
Frequency
Amount
0
50
100
150
200
250
0
2000
4000
6000
8000
10000

RESULT:
The program to implementing different preprocessing techniques is implemented successfully.