```
import pandas as pd  # Data manipulation aur analysis ke liye
import numpy as np  # Numerical computations ke liye
import matplotlib.pyplot as plt  # Data visualization ke liye
import seaborn as sns  # Graphs aur charts ke liye
from sklearn.preprocessing import StandardScaler  # Data normalization
ke liye
from sklearn.impute import SimpleImputer  # Missing values ko fill
karne ke liye

df = pd.read_csv('Customer_Transactions.csv')  # CSV file ko Pandas
dataframe me load kar rahe hain
df.head()  # First 5 rows ko display karne ke liye
```

```
              tranDate          custName           cardNum  zipCode  \
0  2023-09-15 20:32:41    Catherine Bell  2294637276392057     8642
1  2023-05-16 23:18:37     Parker Riddle   342160763812707    80349
2  2023-09-11 18:38:23      Brenda Baird  4137641055044779    34346
3  2023-08-04 21:42:37   Kimberly Carter  3546070762859922    47715
4  2023-09-22 08:27:40  Daniel Rodriguez   213170012973743    77790

   tranAmount
0         848
1         574
2         600
3         583
4        3636
```

```
print(df.isnull().sum())  # Har column me kitne missing values hain wo
check karna
print(df.isnull().mean() * 100)  # Har column me missing values ka
percentage check karna
```

```
tranDate      0
custName      0
cardNum       0
zipCode       0
tranAmount    0
dtype: int64
tranDate      0.0
custName      0.0
cardNum       0.0
zipCode       0.0
tranAmount    0.0
dtype: float64
```

```
df_cleaned = df.dropna()  # Saare missing values wali rows ko drop kar
diya
df_cleaned
```

```
                 tranDate          custName              cardNum
zipCode  \
0     2023-09-15 20:32:41    Catherine Bell     2294637276392057
8642
1     2023-05-16 23:18:37     Parker Riddle      342160763812707
80349
2     2023-09-11 18:38:23      Brenda Baird     4137641055044779
34346
3     2023-08-04 21:42:37   Kimberly Carter     3546070762859922
47715
4     2023-09-22 08:27:40  Daniel Rodriguez      213170012973743
77790
...                   ...               ...                  ...
...
5495  2024-01-18 01:08:01       Luke Garner      180059140353879
71470
5496  2023-12-07 02:26:00    Darrell Vargas     3578703731370362
86141
5497  2023-08-30 23:21:48       Ricky Smith      376357770060994
31935
5498  2023-12-18 02:35:29    Raymond Garcia       30071114876795
69219
5499  2023-12-24 21:47:48      Brett Tucker  4970134866497942923
27947

      tranAmount
0            848
1            574
2            600
3            583
4           3636
...          ...
5495        4200
5496        3063
5497        2394
5498        2193
5499        1623

[5500 rows x 5 columns]

df_cleaned = df.dropna(subset=['cardNum'])  # Sirf 'cardNum' column me
missing values wali rows ko drop kiya
df_cleaned

# dropna(subset=['cardNum']) ka matlab:

# cardNum column ko check karega aur jisme NaN (missing value) hogi,
us row ko remove kar dega.
# Baaki columns me agar missing value ho, to koi effect nahi padega.
#df_cleaned me sirf woh rows rahengi jisme cardNum ki value missing
nahi hai.
```

```
                    tranDate          custName              cardNum
zipCode  \
0     2023-09-15 20:32:41    Catherine Bell     2294637276392057
8642
1     2023-05-16 23:18:37     Parker Riddle      342160763812707
80349
2     2023-09-11 18:38:23      Brenda Baird     4137641055044779
34346
3     2023-08-04 21:42:37   Kimberly Carter     3546070762859922
47715
4     2023-09-22 08:27:40  Daniel Rodriguez      213170012973743
77790
...                   ...               ...                  ...
...
5495  2024-01-18 01:08:01       Luke Garner      180059140353879
71470
5496  2023-12-07 02:26:00    Darrell Vargas     3578703731370362
86141
5497  2023-08-30 23:21:48       Ricky Smith      376357770060994
31935
5498  2023-12-18 02:35:29    Raymond Garcia       30071114876795
69219
5499  2023-12-24 21:47:48      Brett Tucker  4970134866497942923
27947

      tranAmount
0            848
1            574
2            600
3            583
4           3636
...          ...
5495        4200
5496        3063
5497        2394
5498        2193
5499        1623

[5500 rows x 5 columns]
```

```
df_cleaned = df.dropna(axis=1)  # Saare missing values wale columns ko
drop kiya
df_cleaned
```

```
                    tranDate          custName              cardNum
zipCode  \
0     2023-09-15 20:32:41    Catherine Bell     2294637276392057
8642
1     2023-05-16 23:18:37     Parker Riddle      342160763812707
80349
```

```
2     2023-09-11 18:38:23      Brenda Baird     4137641055044779
34346
3     2023-08-04 21:42:37   Kimberly Carter     3546070762859922
47715
4     2023-09-22 08:27:40  Daniel Rodriguez      213170012973743
77790
...                   ...               ...                  ...
...
5495  2024-01-18 01:08:01       Luke Garner      180059140353879
71470
5496  2023-12-07 02:26:00    Darrell Vargas     3578703731370362
86141
5497  2023-08-30 23:21:48       Ricky Smith      376357770060994
31935
5498  2023-12-18 02:35:29    Raymond Garcia       30071114876795
69219
5499  2023-12-24 21:47:48      Brett Tucker  4970134866497942923
27947

      tranAmount
0            848
1            574
2            600
3            583
4           3636
...          ...
5495        4200
5496        3063
5497        2394
5498        2193
5499        1623

[5500 rows x 5 columns]

imputer = SimpleImputer(strategy='mean')  # Missing values ko mean se
fill karne ke liye SimpleImputer use kar rahe hain
#SimpleImputer(strategy='mean') ka matlab:

#SimpleImputer ek sklearn ka built-in function hai jo missing values
ko replace karne ke liye use hota hai.
#strategy='mean' ka matlab hai ki missing values ko column ke mean
(average) se fill kiya jayega.
#Agar hum strategy='median' likhte to missing values median (middle
value) se fill hoti.

df['tranAmount'] = imputer.fit_transform(df[['tranAmount']])  #
'transactionAmount' column me missing values ko fill kar diya

#Missing Values Ko Fill Karna:
```

```
#fit_transform(df[['tranAmount']]) ka matlab hai:
#tranAmount column ka mean calculate karega.
#Jahan bhi missing values hain, wahan mean ka value fill kar dega.
#df['tranAmount'] ko update kar diya hai with the new values.
df['tranAmount']
```

```
0        848.0
1        574.0
2        600.0
3        583.0
4       3636.0
         ...  
5495    4200.0
5496    3063.0
5497    2394.0
5498    2193.0
5499    1623.0
Name: tranAmount, Length: 5500, dtype: float64
```

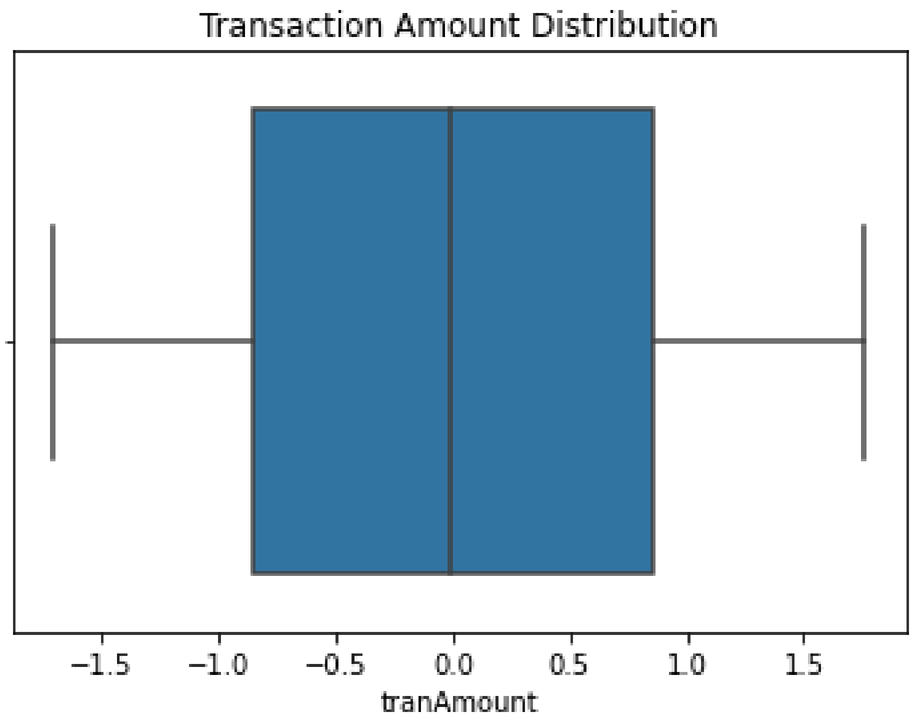
```
scaler = StandardScaler()  # Data ko normalize karne ke liye
StandardScaler use kar rahe hain
#StandardScaler() ek sklearn ka built-in function hai jo numerical
data ko standardize karta hai.
#Standardization ka matlab hota hai:

#μ (mean) ko subtract karta hai

#σ (standard deviation) se divide karta hai
#Resulting values ka mean 0 aur standard deviation 1 ho jata hai.
df[['tranAmount']] = scaler.fit_transform(df[['tranAmount']])  #
'transactionAmount' column ko scale kar diya
#fit_transform(df[['tranAmount']]) ka matlab hai:

#Pehle tranAmount column ka mean aur standard deviation calculate
karega.
#Har value ko standardization formula ke through scale karega.
#Naye standardized values ko tranAmount column me replace karega.

df['tranAmount']
```

```
0      -1.181430
1      -1.375590
2      -1.357166
3      -1.369213
4       0.794188
          ...   
5495    1.193847
5496    0.388152
5497   -0.085911
5498   -0.228343
```

```
5499   -0.632253
Name: tranAmount, Length: 5500, dtype: float64

# **Part 3: Filling Missing Values (Imputation) **


# Fill all missing values with a constant value, such as 0 or
"Unknown"
df_filled = df.fillna(0)  # For numeric columns
df_filled = df.fillna("Unknown")  # For categorical columns
df_filled

                 tranDate          custName              cardNum
zipCode  \
0     2023-09-15 20:32:41    Catherine Bell     2294637276392057
8642
1     2023-05-16 23:18:37     Parker Riddle      342160763812707
80349
2     2023-09-11 18:38:23      Brenda Baird     4137641055044779
34346
3     2023-08-04 21:42:37   Kimberly Carter     3546070762859922
47715
4     2023-09-22 08:27:40  Daniel Rodriguez      213170012973743
77790
...                   ...               ...                  ...
...
5495  2024-01-18 01:08:01       Luke Garner      180059140353879
71470
5496  2023-12-07 02:26:00    Darrell Vargas     3578703731370362
86141
5497  2023-08-30 23:21:48       Ricky Smith      376357770060994
31935
5498  2023-12-18 02:35:29    Raymond Garcia       30071114876795
69219
5499  2023-12-24 21:47:48      Brett Tucker  4970134866497942923
27947

      tranAmount
0      -1.181430
1      -1.375590
2      -1.357166
3      -1.369213
4       0.794188
...          ...
5495    1.193847
5496    0.388152
5497   -0.085911
5498   -0.228343
5499   -0.632253
```

```
[5500 rows x 5 columns]


df = df.drop_duplicates()
df

                 tranDate          custName              cardNum
zipCode  \
0     2023-09-15 20:32:41    Catherine Bell     2294637276392057
8642
1     2023-05-16 23:18:37     Parker Riddle      342160763812707
80349
2     2023-09-11 18:38:23       Brenda Baird     4137641055044779
34346
3     2023-08-04 21:42:37   Kimberly Carter     3546070762859922
47715
4     2023-09-22 08:27:40  Daniel Rodriguez      213170012973743
77790
...                   ...               ...                  ...
...
5495  2024-01-18 01:08:01       Luke Garner      180059140353879
71470
5496  2023-12-07 02:26:00    Darrell Vargas     3578703731370362
86141
5497  2023-08-30 23:21:48       Ricky Smith      376357770060994
31935
5498  2023-12-18 02:35:29    Raymond Garcia       30071114876795
69219
5499  2023-12-24 21:47:48      Brett Tucker  4970134866497942923
27947

      tranAmount
0      -1.181430
1      -1.375590
2      -1.357166
3      -1.369213
4       0.794188
...          ...
5495    1.193847
5496    0.388152
5497   -0.085911
5498   -0.228343
5499   -0.632253

[5500 rows x 5 columns]

# **Part 4: Handling  and Visualizing Outliers **
# A box plot is a good way to identify potential outliers in the
Amount column.
```

```
sns.boxplot(x=df['tranAmount'])
plt.title('Transaction Amount Distribution')
plt.show()
```



```
#Part 5: Removing Outliers Using IQR (Interquartile Range)
# Outliers ka matlab hai bahut chhoti ya bahut badi values, jo baaki
data se alag hoti hain.
# Yeh IQR method ka use karke outliers remove karne ka tarika hai.
Q1 = df['tranAmount'].quantile(0.25)
Q1
# Q1 (First Quartile / 25th Percentile):
# Yeh code 25th percentile (Q1) ka value nikalta hai tranAmount column
ke liye.
# Iska matlab hai ki 25% transactions is value se chhoti hain.
```

```
-0.8598959585936592
```

```
Q3 = df['tranAmount'].quantile(0.75)
Q3
#Q3 (Third Quartile / 75th Percentile):
# Yeh code 75th percentile (Q3) ka value nikalta hai tranAmount column
ke liye.
# Iska matlab hai ki 75% transactions is value se chhoti hain.
```

```
0.8597350386116023

IQR = Q3 - Q1
IQR
#IQR (Interquartile Range) Calculate Karna:
# IQR ka formula hota hai:
# IQR=Q3−Q1
# IQR middle 50% data ka range batata hai, jo outliers ko ignore karta
hai.

1.7196309972052615

df = df[(df['tranAmount'] >= (Q1 - 1.5 * IQR)) & (df['tranAmount'] <=
(Q3 + 1.5 * IQR))]
df
#Outliers ko Remove Karna:
# Lower Bound: Q1−1.5×IQR
# Upper Bound: Q3+1.5×IQR
# Yeh lower aur upper bound ke andar wale transactions ko dataframe me
rakhta hai.
# Jo values is range ke bahar hoti hain, unhe remove kar diya jata
hai.

                 tranDate          custName              cardNum
zipCode  \
0     2023-09-15 20:32:41    Catherine Bell     2294637276392057
8642
1     2023-05-16 23:18:37     Parker Riddle      342160763812707
80349
2     2023-09-11 18:38:23      Brenda Baird     4137641055044779
34346
3     2023-08-04 21:42:37   Kimberly Carter     3546070762859922
47715
4     2023-09-22 08:27:40  Daniel Rodriguez      213170012973743
77790
...                   ...               ...                  ...
...
5495  2024-01-18 01:08:01       Luke Garner      180059140353879
71470
5496  2023-12-07 02:26:00    Darrell Vargas     3578703731370362
86141
5497  2023-08-30 23:21:48       Ricky Smith      376357770060994
31935
5498  2023-12-18 02:35:29    Raymond Garcia       30071114876795
69219
5499  2023-12-24 21:47:48      Brett Tucker  4970134866497942923
27947

      tranAmount
0      -1.181430
```

```
1       -1.375590
2       -1.357166
3       -1.369213
4        0.794188
...           ...
5495     1.193847
5496     0.388152
5497    -0.085911
5498    -0.228343
5499    -0.632253

[5500 rows x 5 columns]

IQR

1.7196309972052615
```