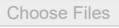


```
1 from google.colab import files
2 uploaded = files.upload()
```

 No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

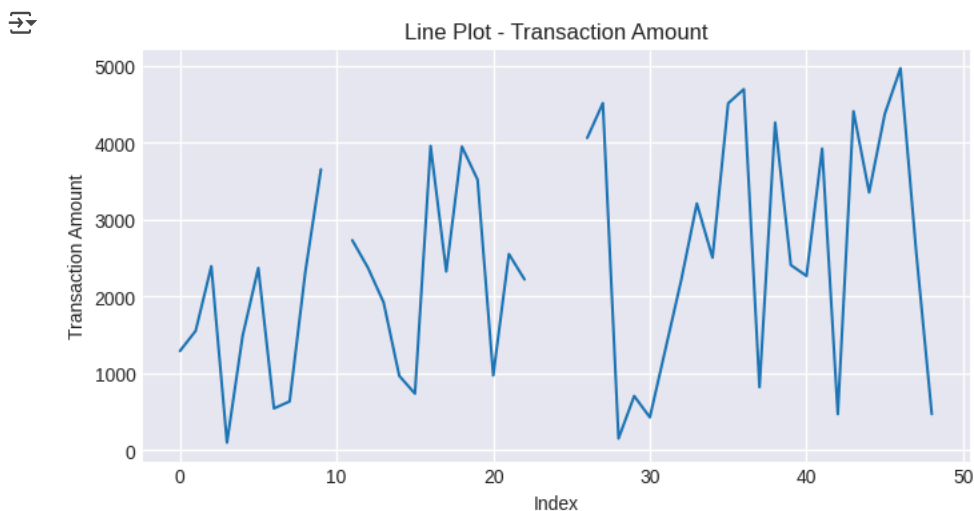
Saving Fraud Detection Dataset.csv to Fraud Detection Dataset.csv

```
1 import pandas as pd
2 df = pd.read_csv('Fraud Detection Dataset.csv')
```

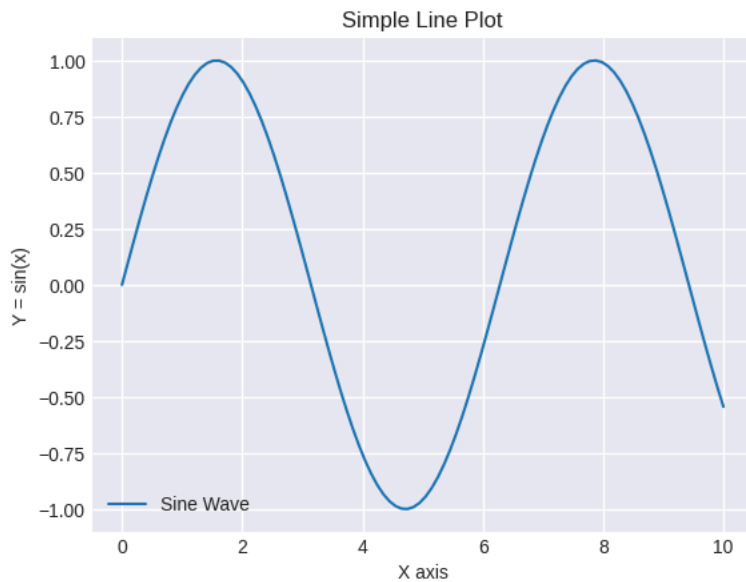
```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from mpl_toolkits.mplot3d import Axes3D
```

```
1 # Use a consistent style
2 plt.style.use('seaborn-v0_8-darkgrid')
```

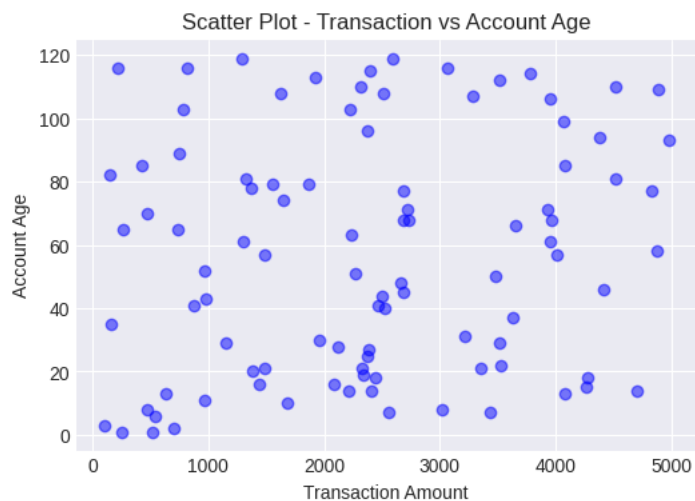
```
1 # ----- 1. Simple Line Plot -----
2 plt.figure(figsize=(8, 4))
3 df['Transaction_Amount'].head(50).plot(title='Line Plot - Transaction Amount')
4 plt.xlabel("Index")
5 plt.ylabel("Transaction Amount")
6 plt.show()
```



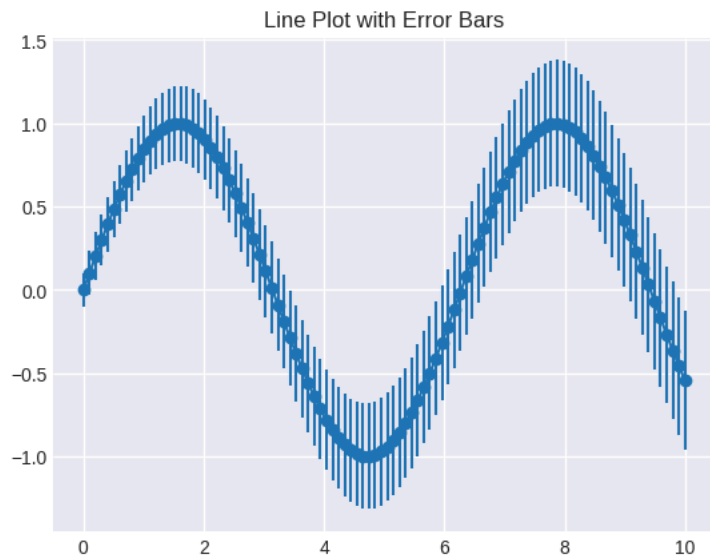
```
1 # ----- 1a). Simple Line Plot -----
2 x = np.linspace(0, 10, 100)
3 y = np.sin(x)
4 plt.plot(x, y, label='Sine Wave')
5 plt.title("Simple Line Plot")
6 plt.xlabel("X axis")
7 plt.ylabel("Y = sin(x)")
8 plt.legend()
9 plt.show()
10
```



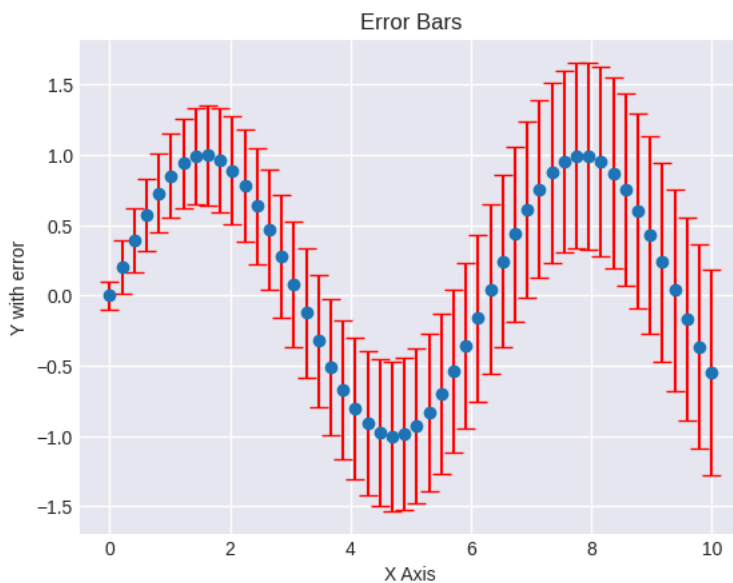
```
1 # ----- 2. Scatter Plot -----
2 plt.figure(figsize=(6, 4))
3 plt.scatter(df['Transaction_Amount'][:100], df['Account_Age'][:100], c='blue', alpha=0.5)
4 plt.title("Scatter Plot - Transaction vs Account Age")
5 plt.xlabel("Transaction Amount")
6 plt.ylabel("Account Age")
7 plt.show()
```



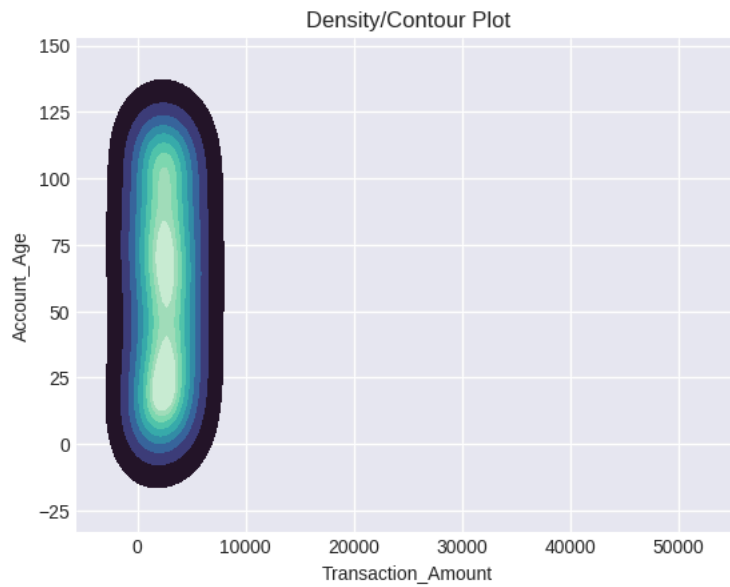
```
1 # ----- 3. Visualizing Errors -----
2 x = np.linspace(0, 10, 100)
3 y = np.sin(x)
4 error = 0.1 + 0.1 * np.sqrt(x)
5 plt.figure()
6 plt.errorbar(x, y, yerr=error, fmt='-o')
7 plt.title("Line Plot with Error Bars")
8 plt.show()
```



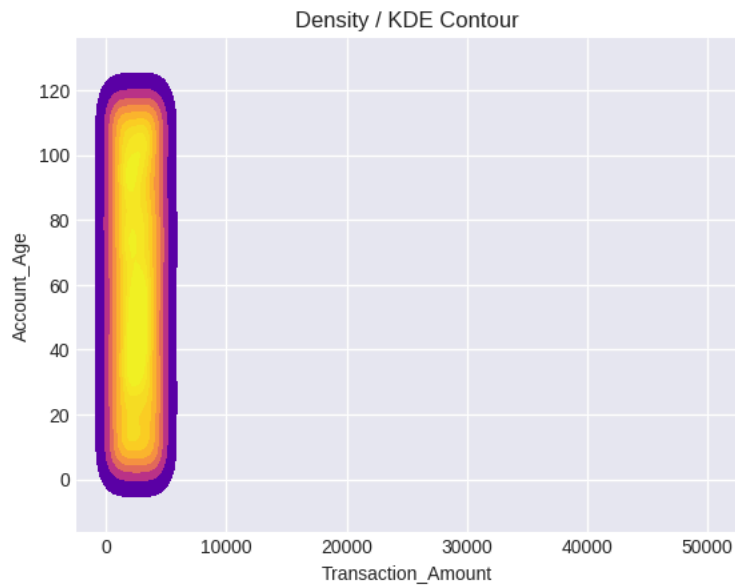
```
1 # ----- 3a). Visualizing Errors -----
2 x = np.linspace(0, 10, 50)
3 y = np.sin(x)
4 errors = 0.1 + 0.2 * np.sqrt(x)
5 plt.errorbar(x, y, yerr=errors, fmt='o', ecolor='red', capsize=5)
6 plt.title("Error Bars")
7 plt.xlabel("X Axis")
8 plt.ylabel("Y with error")
9 plt.show()
```



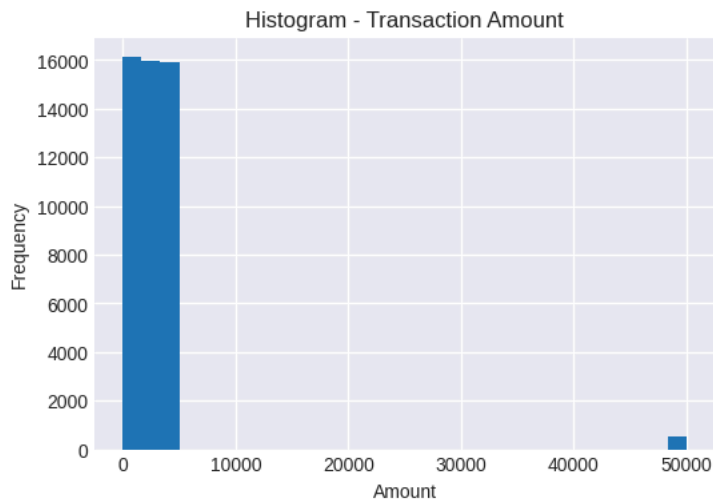
```
1 # ----- 4. Density and Contour Plot -----
2 x = df['Transaction_Amount'][:1000]
3 y = df['Account_Age'][:1000]
4 xy = np.vstack([x, y])
5 kde = sns.kdeplot(x=x, y=y, fill=True, cmap="mako")
6 plt.title("Density/Contour Plot")
7 plt.show()
```



```
1 # ----- 4. Density and Contour Plot -----
2 sns.kdeplot(x=df['Transaction_Amount'], y=df['Account_Age'], fill=True, cmap='plasma')
3 plt.title("Density / KDE Contour")
4 plt.show()
```



```
1 # ----- 5. Histogram -----
2 plt.figure(figsize=(6, 4))
3 df['Transaction_Amount'].hist(bins=30)
4 plt.title("Histogram - Transaction Amount")
5 plt.xlabel("Amount")
6 plt.ylabel("Frequency")
7 plt.show()
```



```
1 # ----- 6. Binning -----
2 df['Amount_Bin'] = pd.cut(df['Transaction_Amount'], bins=5)
3 print("\nBinned Transaction Amount:\n", df['Amount_Bin'].value_counts())
```

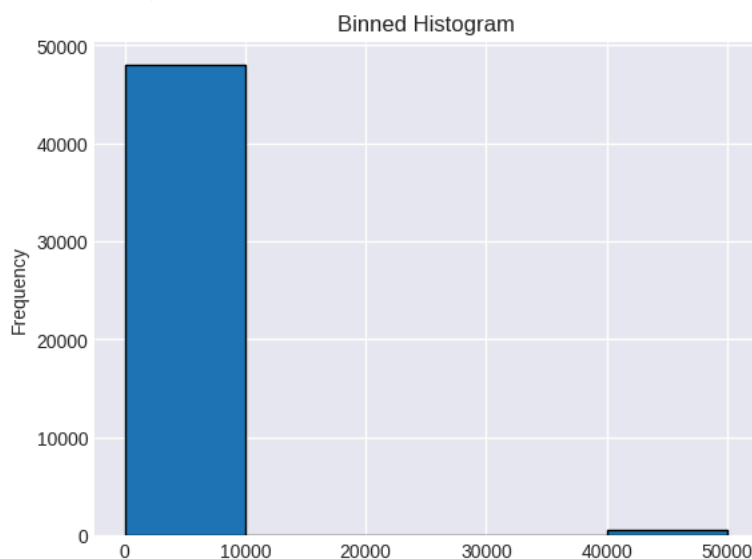


```
Binned Transaction Amount:
Amount_Bin
(-44.963, 10003.584]    47972
(39999.246, 49997.8]      508
(10003.584, 20002.138]      0
(20002.138, 30000.692]      0
(30000.692, 39999.246]      0
Name: count, dtype: int64
```

```
1 # ----- 6a). Binning -----
2 bins = pd.cut(df['Transaction_Amount'], bins=5)
3 print(bins.value_counts())
4 # Plot histogram by bins
5 df['Transaction_Amount'].plot.hist(bins=5, edgecolor='black')
6 plt.title("Binned Histogram")
7 plt.show()
```



```
Transaction_Amount
(-44.963, 10003.584]    47972
(39999.246, 49997.8]      508
(10003.584, 20002.138]      0
(20002.138, 30000.692]      0
(30000.692, 39999.246]      0
Name: count, dtype: int64
```

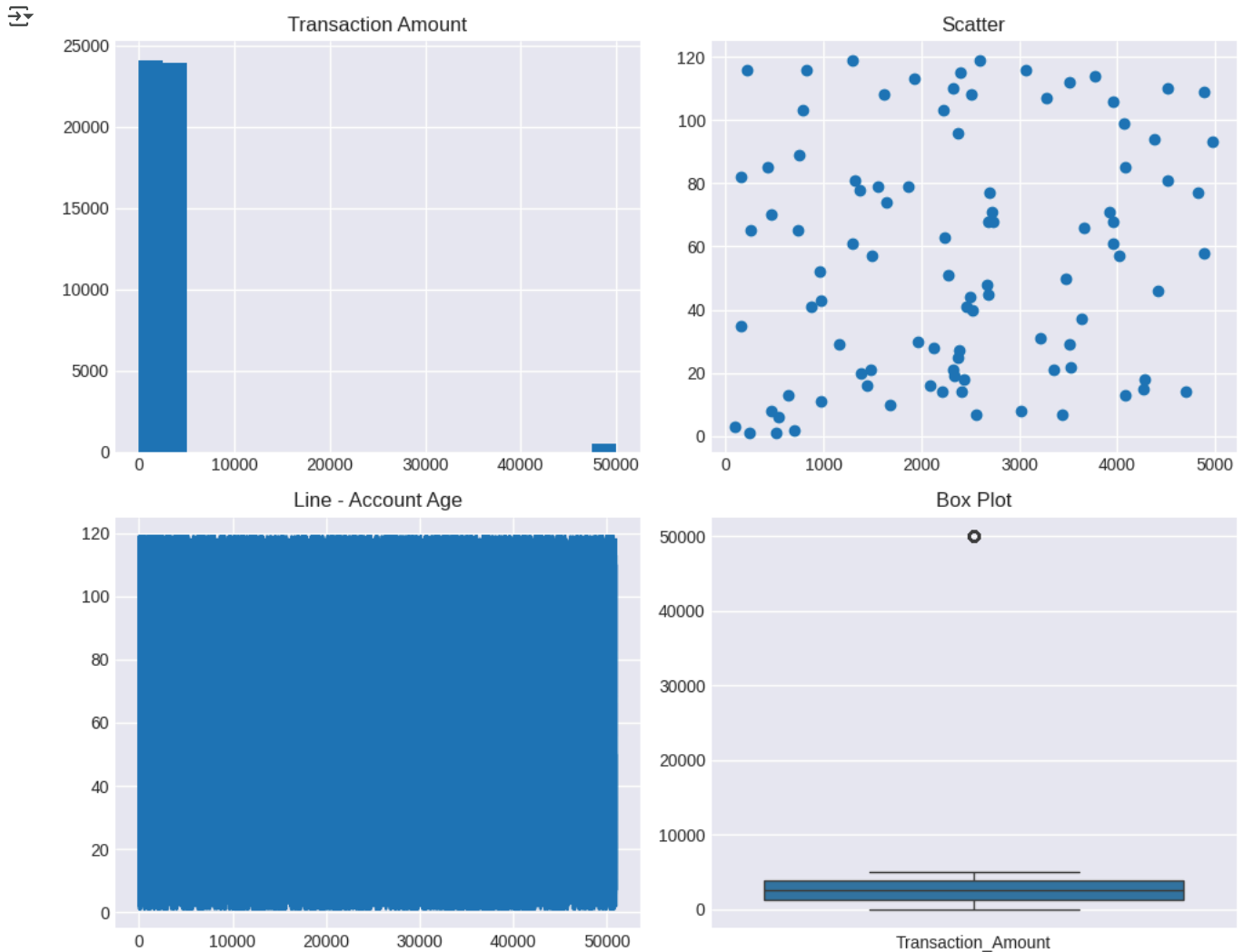


```
1 # ----- 7. Multiple Subplots -----
2 fig, axs = plt.subplots(2, 2, figsize=(10, 8))
3 df['Transaction_Amount'].hist(ax=axs[0, 0], bins=20)
4 axs[0, 0].set_title("Transaction Amount")
5 axs[0, 1].scatter(df['Transaction_Amount'][:100], df['Account_Age'][:100])
6 axs[0, 1].set_title("Scatter")
7 df['Account_Age'].plot(ax=axs[1, 0])
```

```

8 axs[1, 0].set_title("Line - Account Age")
9 sns.boxplot(data=df[['Transaction_Amount']], ax=axs[1, 1])
10 axs[1, 1].set_title("Box Plot")
11 plt.tight_layout()
12 plt.show()

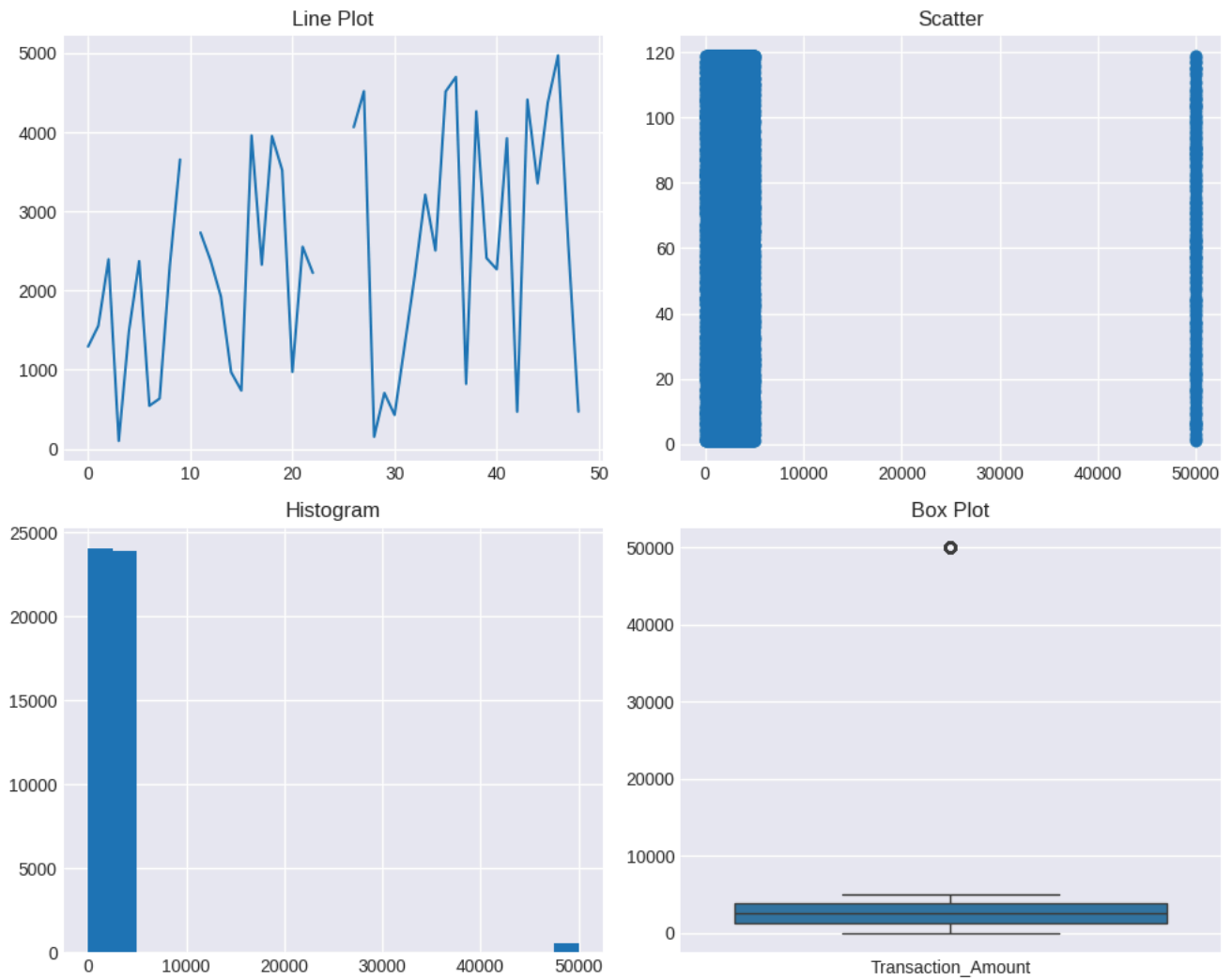
```



```

1 # ----- 7a). Multiple Subplots -----
2 fig, axs = plt.subplots(2, 2, figsize=(10, 8))
3
4 # Subplot 1 - Line
5 axs[0, 0].plot(df['Transaction_Amount'].head(50))
6 axs[0, 0].set_title('Line Plot')
7
8 # Subplot 2 - Scatter
9 axs[0, 1].scatter(df['Transaction_Amount'], df['Account_Age'], alpha=0.5)
10 axs[0, 1].set_title('Scatter')
11
12 # Subplot 3 - Histogram
13 axs[1, 0].hist(df['Transaction_Amount'], bins=20)
14 axs[1, 0].set_title('Histogram')
15
16 # Subplot 4 - Boxplot
17 sns.boxplot(data=df[['Transaction_Amount']], ax=axs[1, 1])
18 axs[1, 1].set_title('Box Plot')
19
20 plt.tight_layout()
21 plt.show()
22

```



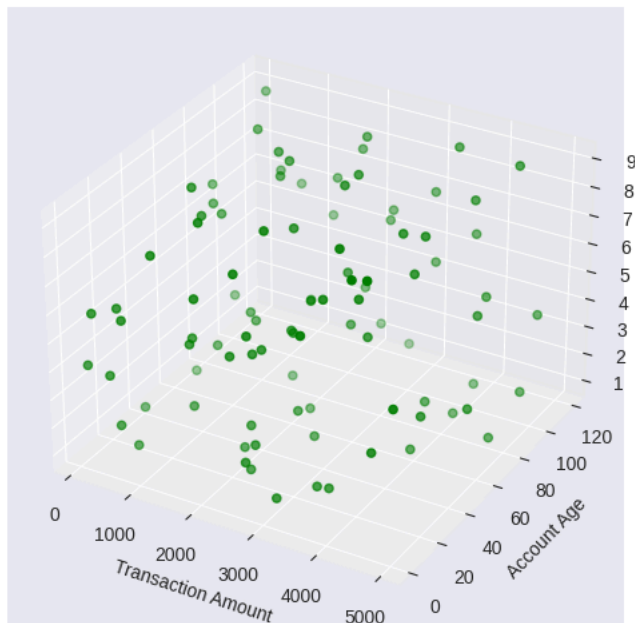
```

1 # ----- 8. 3D Plot -----
2 fig = plt.figure(figsize=(8, 6))
3 ax = fig.add_subplot(111, projection='3d')
4
5 if 'Num_Transactions' not in df.columns:
6     df['Num_Transactions'] = np.random.randint(1, 10, size=len(df))
7
8 ax.scatter(df['Transaction_Amount'][:100], df['Account_Age'][:100], df['Num_Transactions'][:100], c='green')
9
10 ax.set_xlabel("Transaction Amount")
11 ax.set_ylabel("Account Age")
12 ax.set_zlabel("Num Transactions")
13 plt.title("3D Scatter Plot")
14 plt.show()

```



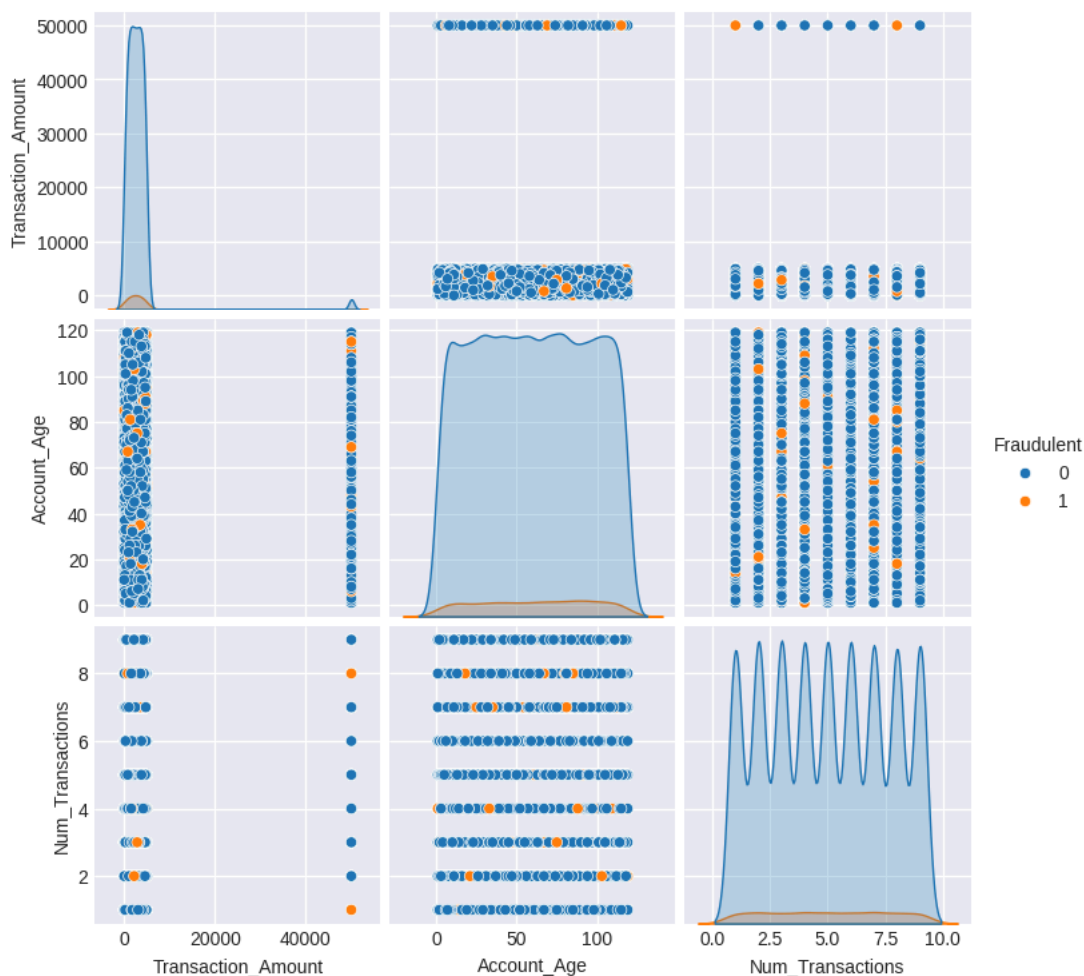
3D Scatter Plot



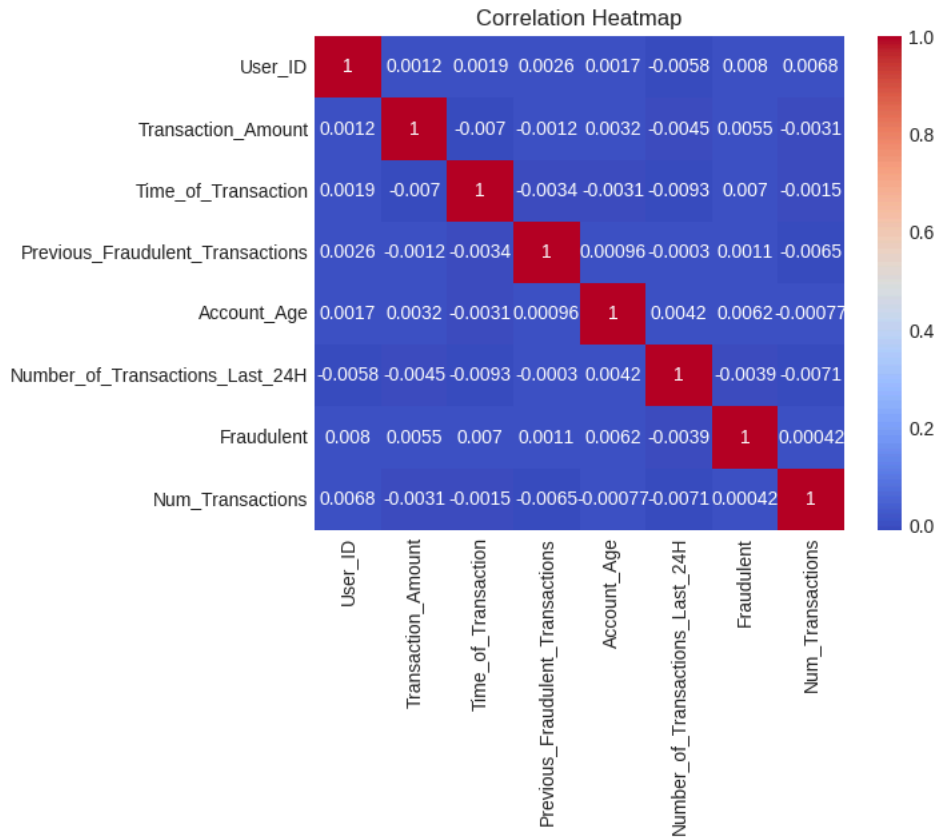
```
1 # ----- 9. Seaborn Visualization -----
2 sns.pairplot(df[['Transaction_Amount', 'Account_Age', 'Num_Transactions', 'Fraudulent']].dropna(), hue='Fraudulent')
3 plt.suptitle("Seaborn Pair Plot", y=1.02)
4 plt.show()
```



Seaborn Pair Plot



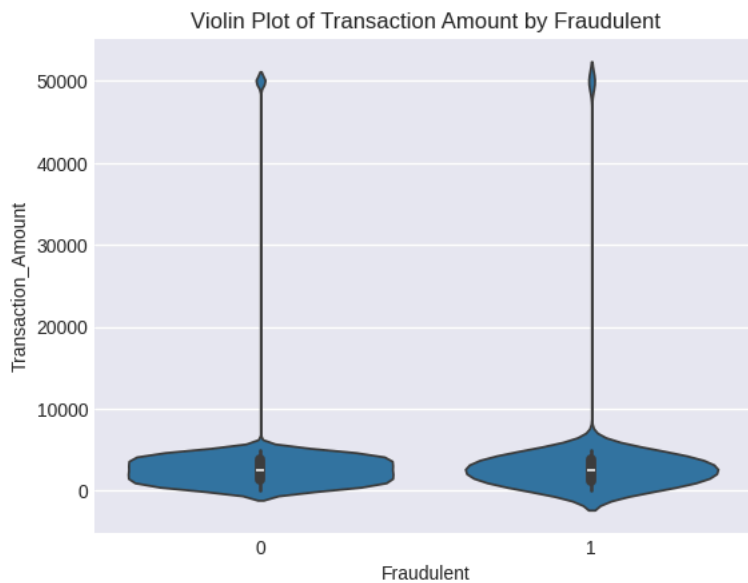
```
1 # ----- 10. Heatmap -----
2 correlation_matrix = df.corr(numeric_only=True)
3 sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
4 plt.title("Correlation Heatmap")
5 plt.show()
```

```

1 # ----- 11. Violin Plot -----
2 sns.violinplot(x='Fraudulent', y='Transaction_Amount', data=df)
3 plt.title("Violin Plot of Transaction Amount by Fraudulent")
4 plt.show()

```



```

1 # ----- 12. Boxen Plot -----

```