

```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES,  
# THEN FEEL FREE TO DELETE THIS CELL.  
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON  
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR  
# NOTEBOOK.  
import kagglehub  
mlg_ulb_creditcardfraud_path = kagglehub.dataset_download('mlg-ulb/creditcardfraud')  
  
print('Data source import complete.')
```

Using Colab cache for faster access to the 'creditcardfraud' dataset.
Data source import complete.

```
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
import plotly.express as px  
from sklearn.linear_model import LogisticRegression  
from xgboost import XGBClassifier  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import confusion_matrix, accuracy_score , precision_score , recall_score  
  
import warnings  
warnings.filterwarnings('ignore')
```

```
df=pd.read_csv('/kaggle/input/creditcardfraud/creditcard.csv')
```

```
df.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941

5 rows × 31 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 284807 entries, 0 to 284806  
Data columns (total 31 columns):  
 #   Column  Non-Null Count  Dtype
```

```
---  -----  -----  -----  
0  Time    284807 non-null  float64  
1  V1      284807 non-null  float64  
2  V2      284807 non-null  float64  
3  V3      284807 non-null  float64  
4  V4      284807 non-null  float64  
5  V5      284807 non-null  float64  
6  V6      284807 non-null  float64  
7  V7      284807 non-null  float64  
8  V8      284807 non-null  float64  
9  V9      284807 non-null  float64  
10 V10     284807 non-null  float64  
11 V11     284807 non-null  float64  
12 V12     284807 non-null  float64  
13 V13     284807 non-null  float64  
14 V14     284807 non-null  float64  
15 V15     284807 non-null  float64  
16 V16     284807 non-null  float64  
17 V17     284807 non-null  float64  
18 V18     284807 non-null  float64  
19 V19     284807 non-null  float64  
20 V20     284807 non-null  float64  
21 V21     284807 non-null  float64  
22 V22     284807 non-null  float64  
23 V23     284807 non-null  float64  
24 V24     284807 non-null  float64  
25 V25     284807 non-null  float64  
26 V26     284807 non-null  float64  
27 V27     284807 non-null  float64  
28 V28     284807 non-null  float64  
29 Amount   284807 non-null  float64  
30 Class    284807 non-null  int64  
dtypes: float64(30), int64(1)  
memory usage: 67.4 MB
```

```
df.isna().sum()
```


0

Time 0

df.tail()

V2	0	V3	0	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				

5 rows × 31 columns

V9 0

V10 0

df.describe().T

V12 0

V13 0

V14 0

V15 0

V16 0

V17 0

V18 0

V19 0

V20 0

V21 0

V22 0

V23 0

V24 0

V25 0

V26 0

V27 0

V28 0

Amount 0

Class 0

dtype: int64

	count	mean	std	min	25%	50%	75%
Time	284807.0	9.481386e+04	47488.145955	0.000000	54201.500000	84692.000000	109300.000000
V1	284807.0	1.168375e-15	1.958696	-56.407510	-0.920373	0.000000	0.000000
V2	284807.0	3.416908e-16	1.651309	-72.715728	-0.598550	0.000000	0.000000
V3	284807.0	-1.379537e-15	1.516255	-48.325589	-0.890365	0.100000	0.100000
V4	284807.0	2.074005e-15	1.415869	-5.683171	-0.848640	-0.000000	-0.000000

df.nunique()

V6	284807.0	1.487313e-15	1.332271	-26.160506	-0.768296	-0.200000	0.100000
V7	284807.0	-5.556467e-16	1.237094	-43.557242	-0.554076	0.000000	0.000000
V8	284807.0	1.213481e-16	1.194353	-73.216718	-0.208630	0.000000	0.000000
V9	284807.0	-2.406331e-15	1.098632	-13.434066	-0.643098	-0.000000	0.000000
V10	284807.0	2.239053e-15	1.088850	-24.588262	-0.535426	-0.000000	0.000000
V11	284807.0	1.673327e-15	1.020713	-4.797473	-0.762494	-0.000000	0.000000
V12	284807.0	-1.247012e-15	0.999201	-18.683715	-0.405571	0.100000	0.100000
V13	284807.0	8.190001e-16	0.995274	-5.791881	-0.648539	-0.000000	0.000000
V14	284807.0	1.207294e-15	0.958596	-19.214325	-0.425574	0.000000	0.000000
V15	284807.0	4.887456e-15	0.915316	-4.498945	-0.582884	0.000000	0.000000
V16	284807.0	1.437716e-15	0.876253	-14.129855	-0.468037	0.000000	0.000000
V17	284807.0	-3.772171e-16	0.849337	-25.162799	-0.483748	-0.000000	0.000000
V18	284807.0	9.564149e-16	0.838176	-9.498746	-0.498850	-0.000000	0.000000
V19	284807.0	1.039917e-15	0.814041	-7.213527	-0.456299	0.000000	0.000000
V20	284807.0	6.406204e-16	0.770925	-54.497720	-0.211721	-0.000000	0.000000
V21	284807.0	1.654067e-16	0.734524	-34.830382	-0.228395	-0.000000	0.000000
V22	284807.0	-3.568593e-16	0.725702	-10.933144	-0.542350	0.000000	0.000000
V23	284807.0	2.578648e-16	0.624460	-44.807735	-0.161846	-0.000000	0.000000
V24	284807.0	4.473266e-15	0.605647	-2.836627	-0.354586	0.000000	0.000000
V25	284807.0	5.340915e-16	0.521278	-10.295397	-0.317145	0.000000	0.000000
V26	284807.0	1.683437e-15	0.482227	-2.604551	-0.326984	-0.000000	0.000000

V27	284807.0	-3.660091e-16	0.403632	-22.565679	-0.070840	0.01
V28	284807.0	-1.227390e-16	0.330083	-15.430084	-0.052960	0.01
Amount	284807.0	8.834962e+01	250.120109	0.000000	5.600000	22.01
Class	284807.0	1.727486e-03	0.041527	0.000000	0.000000	0.01

	0
Time	124592
V1	275663
▼ EDA	v2 275663
	V3 275663
	<pre>df.corr()['Class'].sort_values(ascending=False)</pre>
	V5 275663
	V6 275663
	V7 275663
	V8 275663
	V9 275663
	V10 275663
	V11 275663
	V12 275663
	V13 275663
	V14 275663
	V15 275663
	V16 275663
	V17 275663
	V18 275663
	V19 275663
	V20 275663
	V21 275663
	V22 275663
	V23 275663
	V24 275663
	V25 275663
	V26 275663
	V27 275663
	V28 275663
Amount	32767

Class

2

dtype: int64

Class

Class 1.000000

```
df.duplicated().sum()
```

V4 0.133447
np.int64(1081)
V2 0.091289

```
df=df.drop_duplicates()
```

V10 0.004782

```
df['Class'].value_counts()
```

V8 count9875

Class 0.017580

0 283253 536

Amount 04005632

V26 0.004455
dtype: int64

V25 0.003308

```
fraud = df[df['Class'] == 1]  
non_fraud = df[df['Class'] == 0]
```

V11 0.002000

```
fraud.Amount.describe()
```

V13 -0.004570
Amount

V14 0.007001

count 473.000000

mean -0.012520

std 123.871860

V6 0.012612

min 260.211041

V5 -0.094974

25% 0.000000

V10 0.007722

50% 1.000000

V1 -0.101547

75% 9.820000

V10 0.144185

max 105.890000

V1 -0.101547

max 2125.870000

V3 -0.192961

dtype: float64

V16 -0.196539

```
non_fraud.Amount.describe()
```

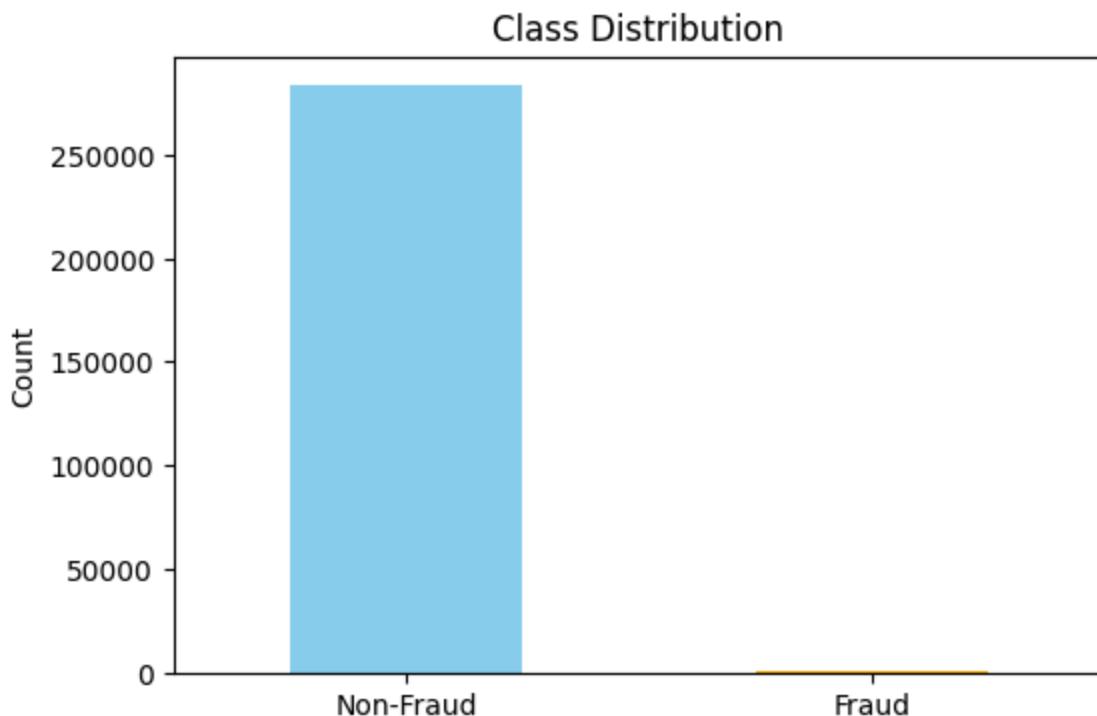
V12 -0.200000

V14 -0.302544

```
V17      -0.326481  
        Amount  
  
count    283253.000000  
mean      88.413575  
std       250.379023  
min       0.000000  
25%      5.670000  
50%      22.000000  
75%      77.460000  
max     25691.160000  
  
dtype: float64
```

▼ Class Distribution (Fraud vs Non-Fraud)

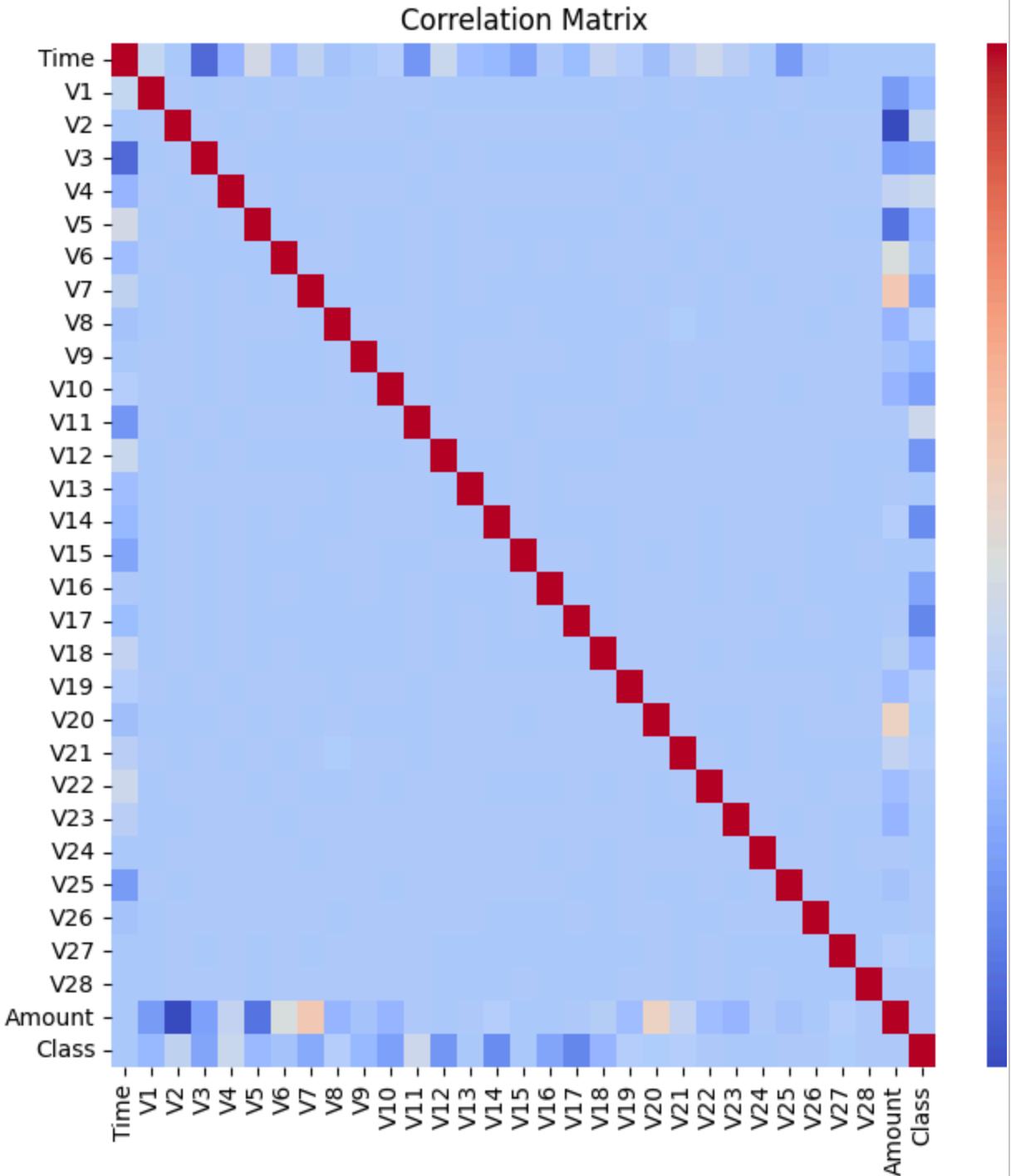
```
class_counts = df['Class'].value_counts()  
class_labels = ['Non-Fraud', 'Fraud']  
plt.figure(figsize=(6, 4))  
class_counts.plot(kind='bar', color=['skyblue', 'orange'])  
plt.title('Class Distribution')  
plt.xticks(ticks=[0, 1], labels=class_labels, rotation=0)  
plt.ylabel('Count')  
plt.show()
```



```

plt.figure(figsize=(8, 8))
sns.heatmap(df.corr(), cmap='coolwarm', annot=False, fmt=".2f")
plt.title('Correlation Matrix')
plt.show()

```

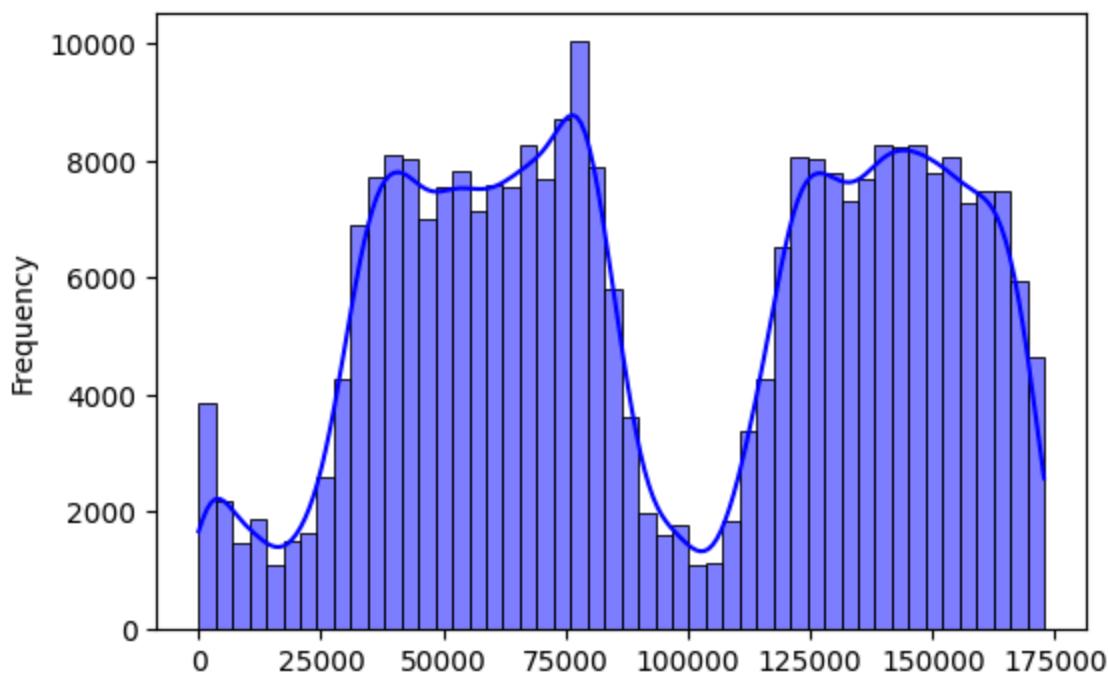


```

plt.figure(figsize=(6, 4))
sns.histplot(df['Time'], bins=50, kde=True, color='blue')
plt.title('Distribution of Time')
plt.xlabel('Time')
plt.ylabel('Frequency')
plt.show()

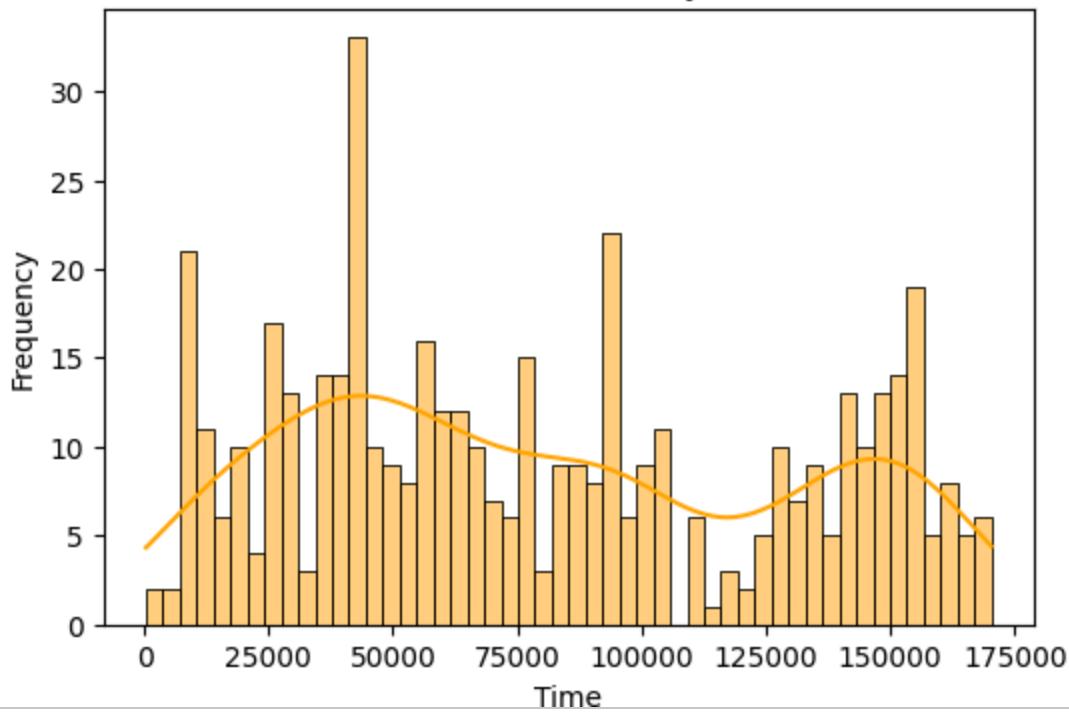
```

Distribution of Time



```
plt.figure(figsize=(6, 4))
sns.histplot(fraud['Time'], bins=50, kde=True, color='orange')
plt.title('Fraud Transactions by Time')
plt.xlabel('Time')
plt.ylabel('Frequency')
plt.show()
```

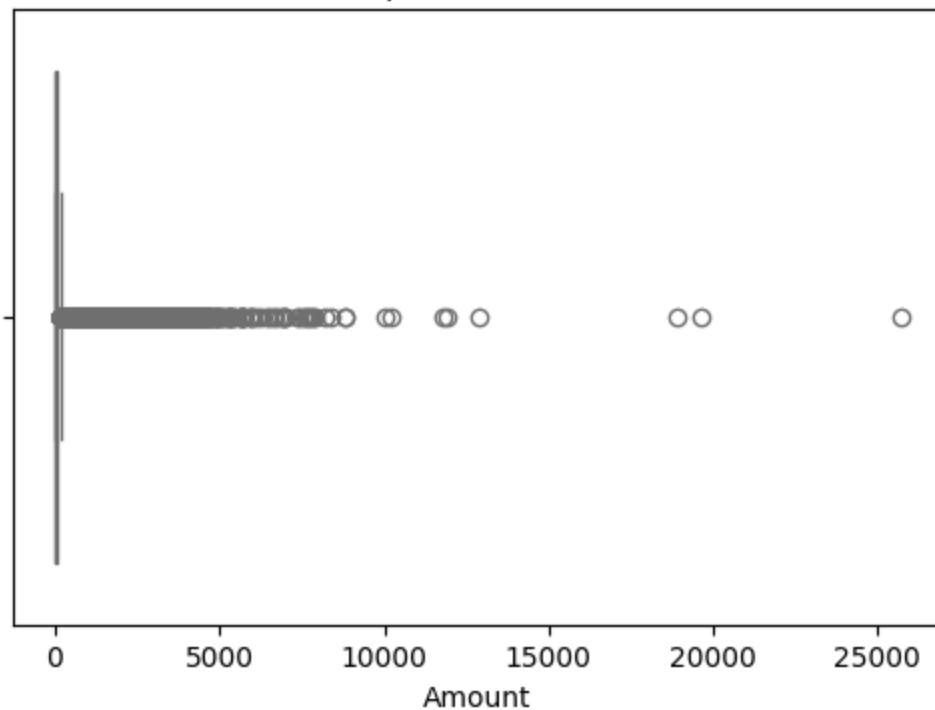
Fraud Transactions by Time



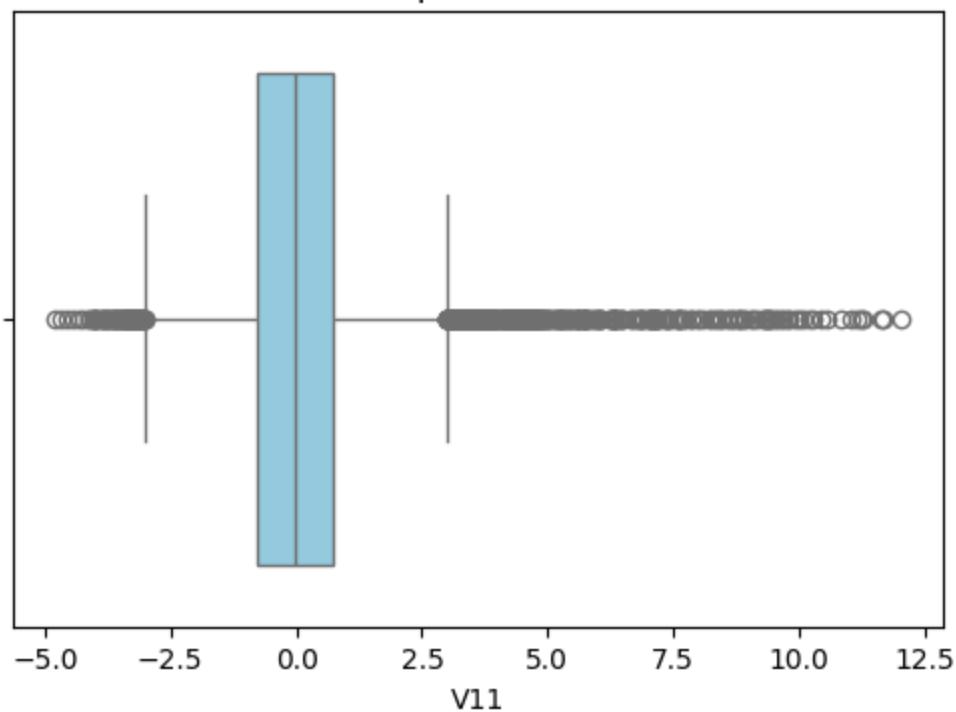
Outliers

```
features = ['Amount', 'V11', 'V2', 'V17', 'V4']
for feature in features:
    plt.figure(figsize=(6, 4))
    sns.boxplot(x=df[feature], color='skyblue')
    plt.title(f'Boxplot of {feature}')
    plt.xlabel(feature)
    plt.show()
```

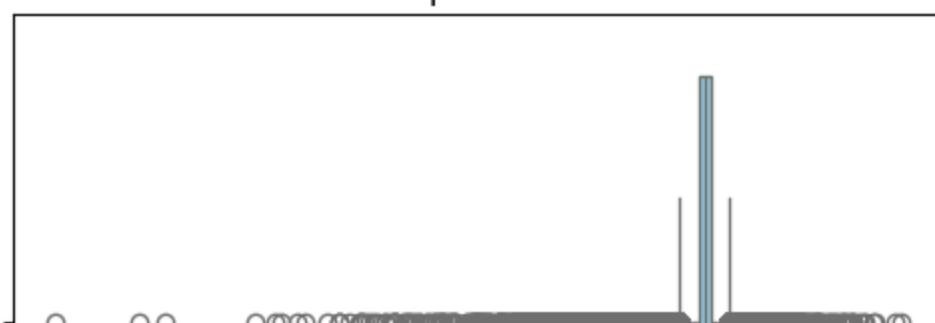

Boxplot of Amount

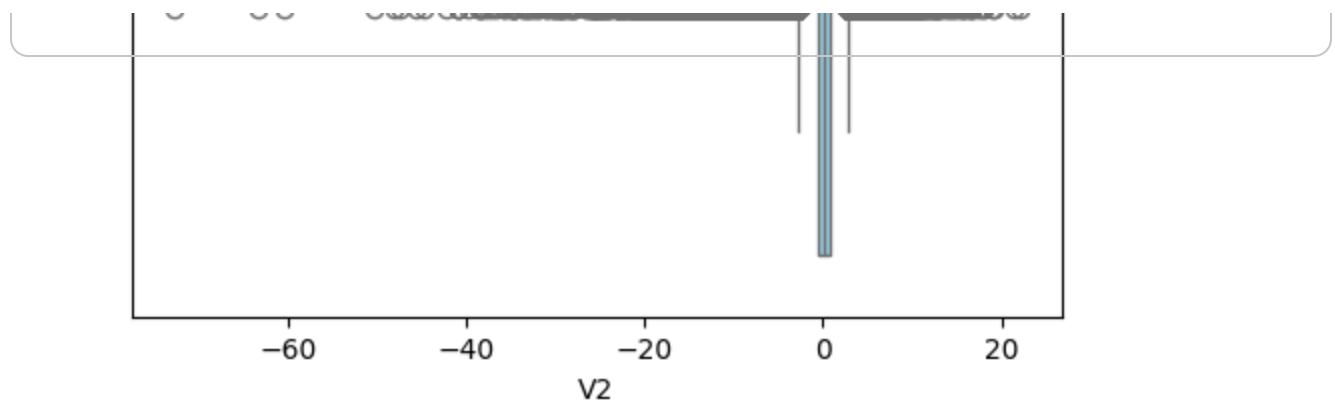


Boxplot of V11

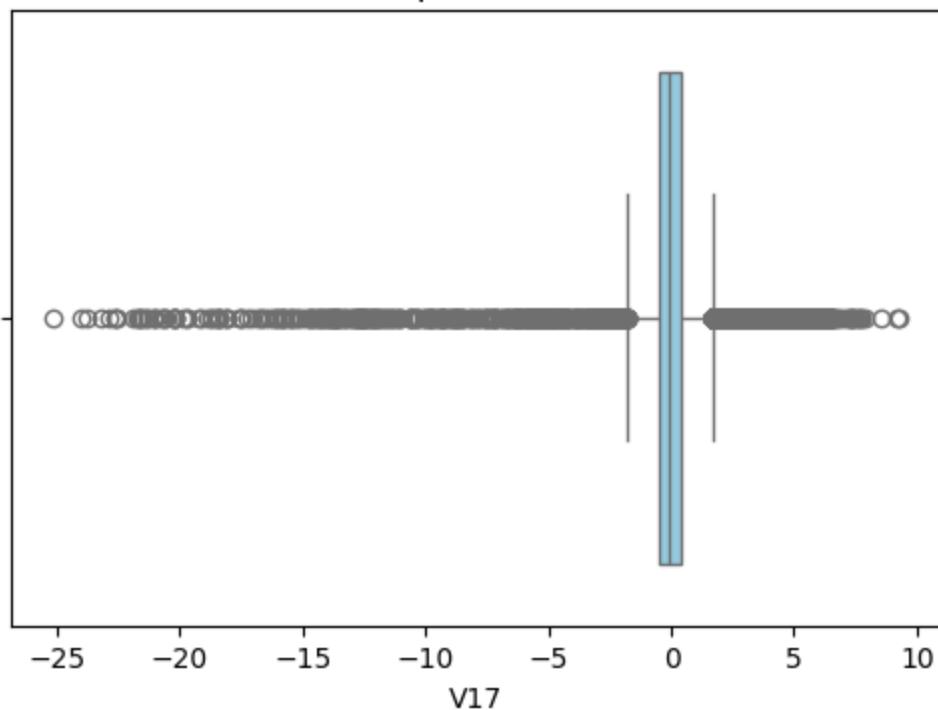


Boxplot of V2





Boxplot of V17



Boxplot of V4

