In [91]:
```
import pandas as pd
import numpy as np
data=pd.read_csv("Credit_Card.csv")
print(data.head())
```

```
   step      type    amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0     1   PAYMENT   9839.64  C1231006815       170136.0       160296.36
1     1   PAYMENT   1864.28  C1666544295        21249.0        19384.72
2     1  TRANSFER    181.00  C1305486145          181.0            0.00
3     1  CASH_OUT    181.00   C840083671          181.0            0.00
4     1   PAYMENT  11668.14  C2048537720        41554.0        29885.86

      nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud
0  M1979787155               0             0.0        0               0
1  M2044282225               0             0.0        0               0
2   C553264065               0             0.0        1               0
3    C38997010           21182             0.0        1               0
4  M1230701703               0             0.0        0               0
```

In [5]:
```
print(data.isnull().sum())
```

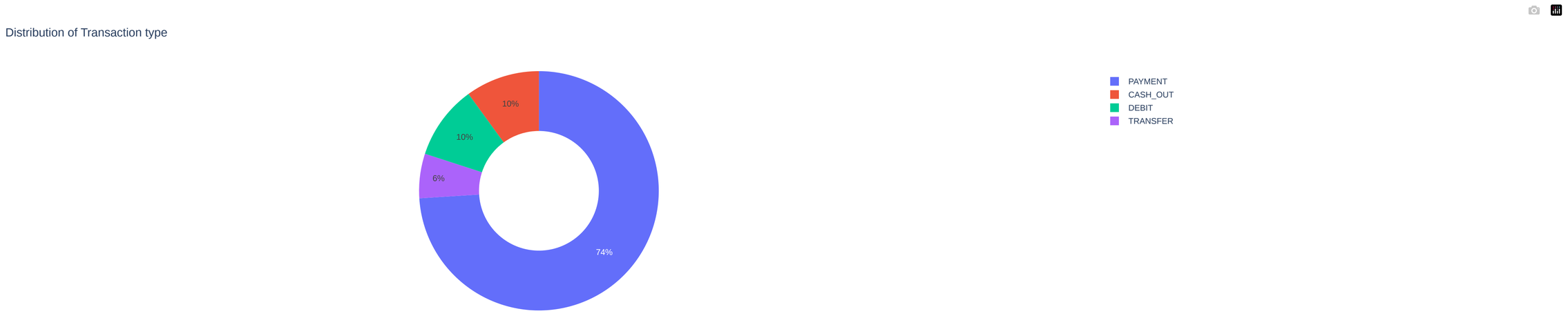
```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

In [7]:
```
print(data.type.value_counts())
```

```
type
PAYMENT     37
CASH_OUT     5
DEBIT        5
TRANSFER     3
Name: count, dtype: int64
```

In [9]:
```
import plotly
```

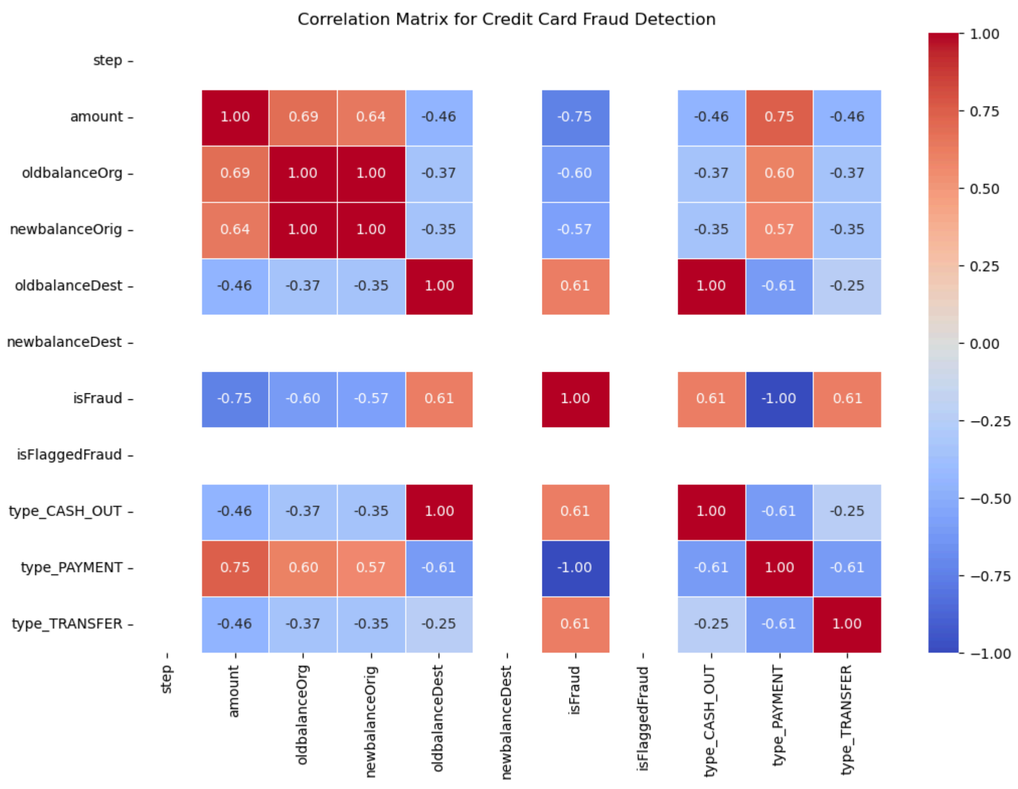
In [11]:
```
type = data["type"].value_counts()
transactions = type. index
quantity = type.values
import plotly.express as px
figure = px. pie (data,
values = quantity,
names = transactions, hole = 0.5,
title = "Distribution of Transaction type")
figure.show()
```



In [25]:
```
data = {
    'type': ['PAYMENT', 'REFUND', 'PAYMENT', 'PAYMENT'],
    'amount': [100, 200, 300, 400],
    'isFraud': [0, 1, 0, 1]
}
data = pd.DataFrame(data)
type_mapping = {"CASH_OUT": 1, "PAYMENT": 2, "CASH_IN": 3, "TRANSFER": 4, "DEBIT": 5, "REFUND": 6}
data["type"] = data["type"].map(type_mapping)
data["isFraud"] = pd.to_numeric(data["isFraud"], errors='coerce')
correlation = data.corr()
print(correlation["isFraud"].sort_values(ascending=False))
data["isFraud"] = data["isFraud"].map({0: "Nofraud", 1: "fraud"})
print(data.head())
```

```
isFraud    1.000000
type       0.577350
amount     0.447214
Name: isFraud, dtype: float64
   type  amount  isFraud
0     2     100  Nofraud
1     6     200    fraud
2     2     300  Nofraud
3     2     400    fraud
```

In [47]:
```
df = pd.DataFrame(data)
df = pd.get_dummies(df, columns=['type'])
df = df.drop(columns=['nameOrig', 'nameDest'])
correlation_matrix = df.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix for Credit Card Fraud Detection')
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: