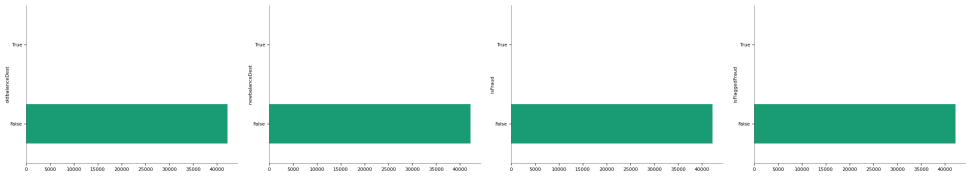
```
import pandas as pd
a=pd.read_csv("onlinefraud.csv")
a.isna()
```

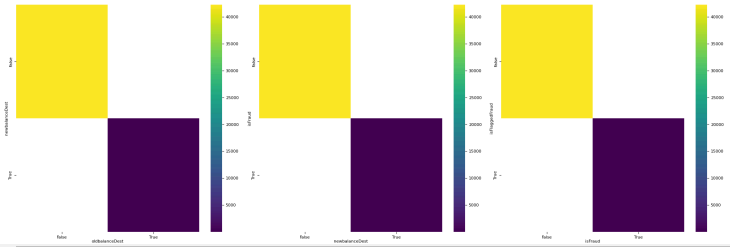
	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFrau
0	False	False	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	
42266	False	False	False	False	False	False	False	False	False	False	
42267	False	False	False	False	False	False	False	False	False	False	
42268	False	False	False	False	False	False	False	False	False	False	Fals
42269	False	False	False	False	False	False	False	False	False	False	Fals
42270	False	False	False	False	False	False	False	True	True	True	Tru

42271 rows × 11 columns

Categorical distributions

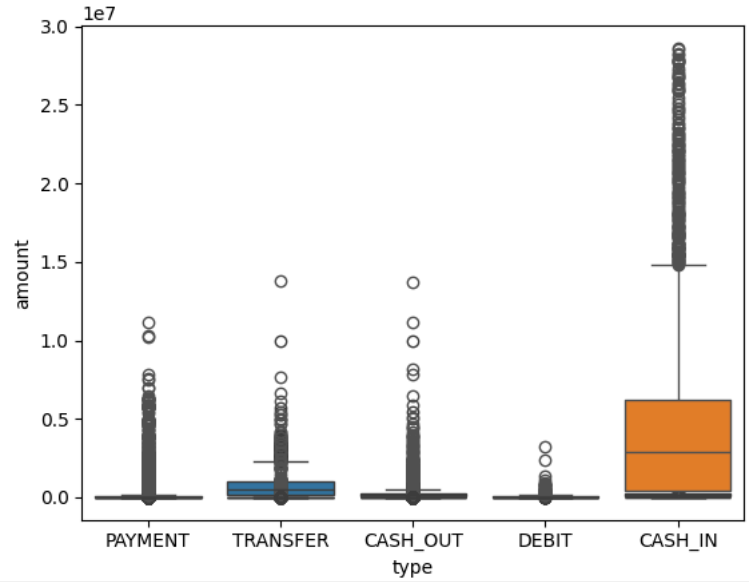


2-d categorical distributions



```
print("******GRAPHICAL MODEL OF THE GIVEN DATA ******")
import seaborn as sd
sd.boxplot(x="type",y="amount",data=a)
sd.boxplot(x="type",y="newbalanceOrig",data=a)
sd.boxplot(x="type",y="isFraud",data=a)
```

```
******GRAPHICAL MODEL OF THE GIVEN DATA ******
<Axes: xlabel='type', ylabel='amount'>
```



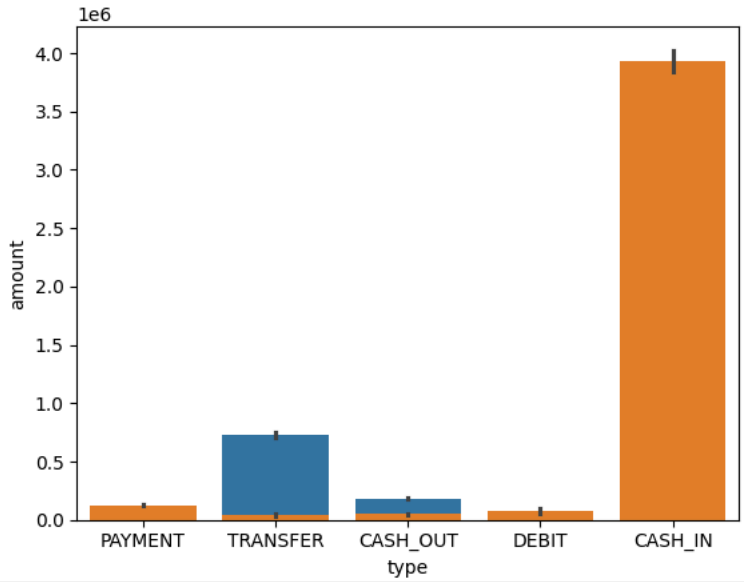
```
print("******DISTRIBUSION PLOT******")
sd.distplot(a.isFraud)
sd.distplot(a.oldbalanceDest)
sd.distplot(a.newbalanceOrig)
sd.distplot(a.oldbalanceOrg)
sd.distplot(a.amount)
```

```
******DISTRIBUSION PLOT******
<ipython-input-3-01b7865c155d>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sd.distplot(a.isFraud)
<ipython-input-3-01b7865c155d>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sd.distplot(a.oldbalanceDest)
<ipython-input-3-01b7865c155d>:4: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sd.distplot(a.newbalanceOrig)
<ipython-input-3-01b7865c155d>:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sd.distplot(a.oldbalanceOrg)
<ipython-input-3-01b7865c155d>:6: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sd.distplot(a.amount)
<Axes: xlabel='amount', ylabel='Density'>
```



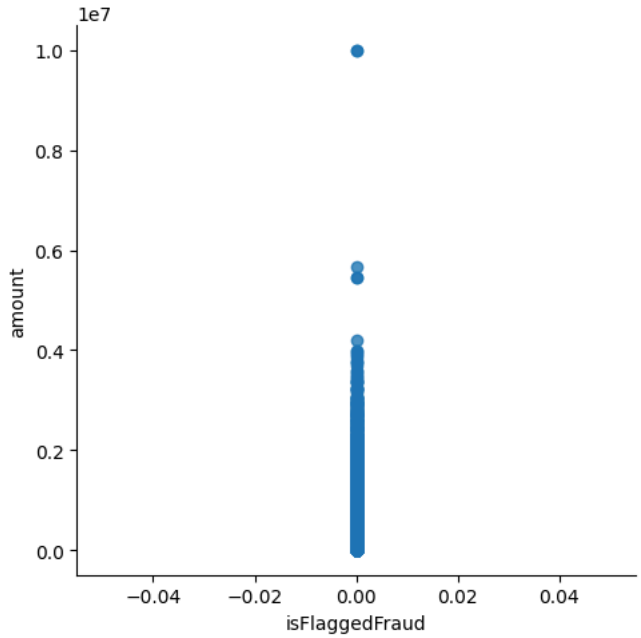
```
print("******BARPLOT******")
sd.barplot(x="type",y="amount",data=a)
sd.barplot(x="type",y="newbalanceOrig",data=a)
sd.barplot(x="type",y="isFraud",data=a)
```

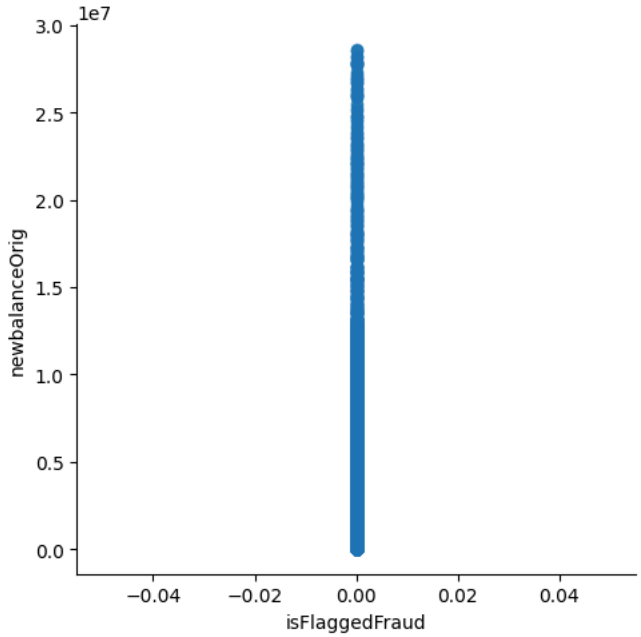
```
******BARPLOT******
<Axes: xlabel='type', ylabel='amount'>
```




```
print("******LMPLOT******")
sd.lmplot(x="isFlaggedFraud",y="amount",data=a)
sd.lmplot(x="isFlaggedFraud",y="newbalanceOrig",data=a)
sd.lmplot(x="isFlaggedFraud",y="isFraud",data=a)
```

```
******LMPLOT******
<seaborn.axisgrid.FacetGrid at 0x7baf83515780>
```






```
print("******* PLOT GRAPH *******")
from matplotlib import pyplot as plt
plt.plot(a["oldbalanceOrg"],a["newbalanceOrig"])
```

```
******* PLOT GRAPH *******
```

```
print("******SPLIT TEST AND TRAIN DATA******")
x=a.iloc[:,:-1].values
print(x)
y=a["Volume"]
print(y)
```

```
******SPLIT TEST AND TRAIN DATA******
[[1 'PAYMENT' 9839.64 ... 0.0 0.0 0.0]
 [1 'PAYMENT' 1864.28 ... 0.0 0.0 0.0]
 [1 'TRANSFER' 181.0 ... 0.0 0.0 1.0]
 ...
 [9 'CASH_OUT' 111003.87 ... 2533159.94 2644163.81 0.0]
 [9 'CASH_OUT' 101025.44 ... 156646.32 491301.04 0.0]
 [9 'CASH_OUT' 271441.28 ... nan nan nan]]
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3652         try:
-> 3653             return self._engine.get_loc(casted_key)
   3654         except KeyError as err:

                                 4 frames

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Volume'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3653             return self._engine.get_loc(casted_key)
   3654         except KeyError as err:
-> 3655             raise KeyError(key) from err
   3656         except TypeError:
   3657             # If we have a listlike key, _check_indexing_error will raise

KeyError: 'Volume'
```

```
print("******TRAIN_TEST_SPLIT ALGORITHM******")
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x,y,test_size=0.20,shuffle=True)
print(x_train)
print(x_test)
```

```
******TRAIN_TEST_SPLIT ALGORITHM******
[['18-11-2019' 296.0 304.98999 293.279999 302.570007 302.570007]
 ['28-08-2020' 532.0 539.0 522.0 523.890015 523.890015]
 ['27-11-2019' 313.929993 316.820007 312.75 315.929993 315.929993]
 ...
 ['25-03-2019' 359.0 367.040009 357.440002 366.230011 366.230011]
 ['28-09-2020' 489.109985 492.0 477.880005 490.649994 490.649994]
 ['25-08-2021' 550.159973 552.840027 545.450012 547.580017 547.580017]]
[['23-02-2021' 525.0 548.539978 518.280029 546.150024 546.150024]
 ['10-08-2021' 520.0 520.789978 512.969971 515.840027 515.840027]
 ['01-08-2019' 324.25 328.579987 318.73999 319.5 319.5]
 ...
 ['10-08-2020' 493.350006 497.459991 478.630005 483.380005 483.380005]
 ['06-06-2018' 367.779999 369.679993 363.329987 367.450012 367.450012]
 ['26-10-2020' 487.029999 496.820007 478.899994 488.23999 488.23999]]
```

```
print("******PRINTING THE TRAIN DATAS******")
print(x_train)
print(x_test)
```

```
******PRINTING THE TRAIN DATAS******
[['18-11-2019' 296.0 304.98999 293.279999 302.570007 302.570007]
 ['28-08-2020' 532.0 539.0 522.0 523.890015 523.890015]
 ['27-11-2019' 313.929993 316.820007 312.75 315.929993 315.929993]
 ...
 ['25-03-2019' 359.0 367.040009 357.440002 366.230011 366.230011]
 ['28-09-2020' 489.109985 492.0 477.880005 490.649994 490.649994]
 ['25-08-2021' 550.159973 552.840027 545.450012 547.580017 547.580017]]
[['23-02-2021' 525.0 548.539978 518.280029 546.150024 546.150024]
```