# **Import Libraries**

In [1]:

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

# **Read the dataset**

In [2]:

data **=** pd**.**read\_csv('Downloads/Fraud.csv') data**.**head()

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Out[2]: | **step** | **type** | **amount** | **nameOrig** | **oldbalanceOrg** | **newbalanceOrig** | **nameDest** | **oldbalanceDest** | **newb** |
|  | **0** 1 | PAYMENT | 9839.64 | C1231006815 | 170136.0 | 160296.36 | M1979787155 | 0.0 |  |
|  | **1** 1 | PAYMENT | 1864.28 | C1666544295 | 21249.0 | 19384.72 | M2044282225 | 0.0 |  |
|  | **2** 1 | TRANSFER | 181.00 | C1305486145 | 181.0 | 0.00 | C553264065 | 0.0 |  |
|  | **3** 1 | CASH\_OUT | 181.00 | C840083671 | 181.0 | 0.00 | C38997010 | 21182.0 |  |
|  | **4** 1 | PAYMENT | 11668.14 | C2048537720 | 41554.0 | 29885.86 | M1230701703 | 0.0 |  |

# **Data types**

In [3]:

<class 'pandas.core.frame.DataFrame'> RangeIndex: 6362620 entries, 0 to 6362619 Data columns (total 11 columns):

data**.**info()

# Column Dtype

1. step int64
2. type object
3. amount float64
4. nameOrig object
5. oldbalanceOrg float64
6. newbalanceOrig float64
7. nameDest object
8. oldbalanceDest float64
9. newbalanceDest float64
10. isFraud int64
11. isFlaggedFraud int64

dtypes: float64(5), int64(3), object(3) memory usage: 534.0+ MB

# **Checking for null values**

In [4]:

Out[4]:

step 0

data**.**isnull()**.**sum()

type 0

amount 0

nameOrig 0

oldbalanceOrg 0

newbalanceOrig 0

nameDest 0

oldbalanceDest 0

newbalanceDest 0

isFraud 0

isFlaggedFraud 0

dtype: int64

# **Describe some statistic**

In [5]:

data**.**describe()

Out[5]:

**step amount oldbalanceOrg newbalanceOrig oldbalanceDest newbalanceDest isFrau count** 6.362620e+06 6.362620e+06 6.362620e+06 6.362620e+06 6.362620e+06 6.362620e+06 6.362620e+

**std** 1.423320e+02 6.038582e+05 2.888243e+06 2.924049e+06 3.399180e+06 3.674129e+06 3.590480e-

**mean** 2.433972e+02 1.798619e+05 8.338831e+05

8.551137e+05 1.100702e+06

1.224996e+06 1.290820e-

**min** 1.000000e+00 0.000000e+00 0.000000e+00

0.000000e+00 0.000000e+00

0.000000e+00 0.000000e+

**25%** 1.560000e+02 1.338957e+04 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+

**50%** 2.390000e+02 7.487194e+04 1.420800e+04

0.000000e+00 1.327057e+05

2.146614e+05 0.000000e+

**75%** 3.350000e+02 2.087215e+05 1.073152e+05 1.442584e+05 9.430367e+05 1.111909e+06 0.000000e+

**max** 7.430000e+02 9.244552e+07 5.958504e+07

4.958504e+07 3.560159e+08

3.561793e+08 1.000000e+

# **Correlation of the variables**

In [ ]:

data**.**corr()

In [7]:

Out[7]:

<AxesSubplot:>

sns**.**heatmap(data**.**corr(),cmap**=**'coolwarm',annot **=True**)



Only two pages were converted.

Please **Sign Up** to convert the full document.

[**www.freepdfconvert.com/membership**](https://www.freepdfconvert.com/membership)