

```
In [1]: import pandas as p
df=p.read_csv("creditcard.csv")
info=df.info()
print("\n\nis_na:\n\n",df.isna().head(7))
is_null_sum=df.isna().sum()
print("\n\nCount of all Missing values:\n\n",df.isna().sum())

<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

is_na:

      Time      V1      V2      V3      V4      V5      V6      V7      V8      V9      ...      \
0  False  False  False  False  False  False  False  False  False  False  ...
1  False  False  False  False  False  False  False  False  False  False  ...
2  False  False  False  False  False  False  False  False  False  False  ...
3  False  False  False  False  False  False  False  False  False  False  ...
4  False  False  False  False  False  False  False  False  False  False  ...
5  False  False  False  False  False  False  False  False  False  False  ...
6  False  False  False  False  False  False  False  False  False  False  ...

      V21      V22      V23      V24      V25      V26      V27      V28  Amount  Class
0  False  False  False  False  False  False  False  False  False  False
1  False  False  False  False  False  False  False  False  False  False
2  False  False  False  False  False  False  False  False  False  False
3  False  False  False  False  False  False  False  False  False  False
4  False  False  False  False  False  False  False  False  False  False
5  False  False  False  False  False  False  False  False  False  False
6  False  False  False  False  False  False  False  False  False  False

[7 rows x 31 columns]
```

Count of all Missing values:

```
Time      0
V1         0
V2         0
V3         0
V4         0
V5         0
V6         0
V7         0
V8         0
V9         0
V10        0
V11        0
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
```

```
In [3]: import pandas as pd
import numpy as np

# Create a sample credit card dataset
data = {
    'credit_limit': np.random.randint(1000, 10000, size=150),
    'balance': np.random.randint(0, 10000, size=150),
    'transactions': np.random.randint(1, 100, size=150),
    'age': np.random.randint(18, 70, size=150),
    'default': np.random.choice([0, 1], size=150) # 0: No default, 1: Default
}

# Convert to DataFrame
credit_card_df = pd.DataFrame(data)

# Calculate covariance and correlation
cov = credit_card_df.cov()
cor = credit_card_df.corr()

print("Covariance:\n", cov)
print("\nCorrelation:\n", cor)

Covariance:
      credit_limit      balance  transactions      age  default
credit_limit  6.195820e+06 -1.087753e+05  14749.685638 -1539.066309  29.448322
balance      -1.087753e+05  9.790485e+06 -8854.377181  1302.802685  48.026846
transactions  1.474969e+04 -8.854377e+03   848.102416   34.683356   0.293960
age           -1.539066e+03  1.302803e+03    34.683356   243.644787   0.162864
default       2.944832e+01  4.802685e+01     0.293960    0.162864   0.250559

Correlation:
      credit_limit  balance  transactions      age  default
credit_limit      1.000000 -0.013966   0.203474 -0.039612  0.023635
balance          -0.013966  1.000000   -0.097170  0.026675  0.030664
transactions      0.203474 -0.097170   1.000000  0.076299  0.020165
age               -0.039612  0.026675   0.076299  1.000000  0.020844
default           0.023635  0.030664   0.020165  0.020844  1.000000
```

```
In [25]: import pandas as pd
import numpy as np

# Create a sample credit card dataset
data = {
    'credit_limit': np.random.randint(1000, 10000, size=150),
    'balance': np.random.randint(0, 10000, size=150),
    'transactions': np.random.randint(1, 100, size=150),
    'age': np.random.randint(18, 70, size=150),
    'default': np.random.choice([0, 1], size=150) # 0: No default, 1: Default
}

# Convert to DataFrame
credit_card_df = pd.DataFrame(data)

# Calculate standard deviation
std_dev = credit_card_df.std()

print("Standard Deviation:\n", std_dev)

Standard Deviation:
      credit_limit      balance  transactions      age  default
credit_limit      2698.156885
balance          2996.652386
transactions      128.882215
age               14.632831
default           0.500559
dtype: float64
```

```
In [27]: import pandas as pd
import numpy as np

# Create a sample credit card dataset
data = {
    'credit_limit': np.random.randint(1000, 10000, size=150),
    'balance': np.random.randint(0, 10000, size=150),
    'transactions': np.random.randint(1, 100, size=150),
    'age': np.random.randint(18, 70, size=150),
    'default': np.random.choice([0, 1], size=150) # 0: No default, 1: Default
}

# Convert to DataFrame
credit_card_df = pd.DataFrame(data)

# Calculate median
median_values = credit_card_df.median()

print("Median Values:\n", median_values)

Median Values:
      credit_limit      balance  transactions      age  default
credit_limit      5424.0
balance          5315.5
transactions       55.0
age               43.5
default           0.0
dtype: float64
```

```
In [29]: import pandas as pd
import numpy as np

# Create a sample credit card dataset
data = {
    'credit_limit': np.random.randint(1000, 10000, size=150),
    'balance': np.random.randint(0, 10000, size=150),
    'transactions': np.random.randint(1, 100, size=150),
    'age': np.random.randint(18, 70, size=150),
    'default': np.random.choice([0, 1], size=150) # 0: No default, 1: Default
}

# Convert to DataFrame
credit_card_df = pd.DataFrame(data)

# Calculate mode
mode_values = credit_card_df.mode().iloc[0]

print("Mode Values:\n", mode_values)

Mode Values:
      credit_limit      balance  transactions      age  default
credit_limit      1023.0
balance           27.0
transactions       17.0
age               61.0
default           0.0
Name: 0, dtype: float64
```

```
In [31]: import pandas as pd
import numpy as np

# Create a sample credit card dataset
data = {
    'credit_limit': np.random.randint(1000, 10000, size=150),
    'balance': np.random.randint(0, 10000, size=150),
    'transactions': np.random.randint(1, 100, size=150),
    'age': np.random.randint(18, 70, size=150),
    'default': np.random.choice([0, 1], size=150) # 0: No default, 1: Default
}

# Convert to DataFrame
credit_card_df = pd.DataFrame(data)

# Calculate mean
mean_values = credit_card_df.mean()

print("Mean Values:\n", mean_values)

Mean Values:
      credit_limit      balance  transactions      age
credit_limit      5696.066667
balance          5285.906667
transactions       48.780000
age               42.400000
```

```
default      0.540000
dtype: float64
```

```
In [ ]:
```