```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable

```
import pandas as pd
import numpy as np

# Load the CSV file (update filename accordingly)
df = pd.read_csv("Fraud Detection Dataset.csv")

# Display first few rows
print(df.head())
```

```
  Transaction_ID  User_ID  Transaction_Amount Transaction_Type  \
0             T1     4174             1292.76   ATM Withdrawal
1             T2     4507             1554.58   ATM Withdrawal
2             T3     1860             2395.02   ATM Withdrawal
3             T4     2294              100.10     Bill Payment
4             T5     2130             1490.50      POS Payment

   Time_of_Transaction Device_Used       Location  \
0                 16.0      Tablet  San Francisco
1                 13.0      Mobile       New York
2                  NaN      Mobile            NaN
3                 15.0     Desktop        Chicago
4                 19.0      Mobile  San Francisco

   Previous_Fraudulent_Transactions  Account_Age  \
0                                 0          119
1                                 4           79
2                                 3          115
3                                 4            3
4                                 2           57

   Number_of_Transactions_Last_24H Payment_Method  Fraudulent
0                               13     Debit Card           0
1                                3    Credit Card           0
2                                9            NaN           0
3                                4            UPI           0
4                                7    Credit Card           0
```

```
# Extract numerical columns as NumPy arrays
transaction_amounts = df["Transaction_Amount"].fillna(0).to_numpy()
fraud_labels = df["Fraudulent"].to_numpy()
account_ages = df["Account_Age"].to_numpy()

```

```
# Fixed type structured array
structured_array = np.array(
    list(zip(transaction_amounts, fraud_labels, account_ages)),
    dtype=[("Transaction_Amount", "f8"), ("Fraudulent", "i4"), ("Account_Age", "i4")]
)

```

```
# Indexing
indexed_values = transaction_amounts[:5]
```

```
# Slicing
sliced_values = fraud_labels[:10]
```

```
# Reshaping
reshaped_array = transaction_amounts[:15].reshape(5, 3)
```

```
# Concatenation
concatenated_array = np.concatenate((transaction_amounts[:5], account_ages[:5]))
```

```
# Splitting
split_arrays = np.split(transaction_amounts[:10], 2)

```

```
# Universal functions (UFUNCs)
mean_transaction = np.mean(transaction_amounts)
sum_transactions = np.sum(transaction_amounts)

```

```
# Broadcasting
broadcasted_array = account_ages + 10
```

```
# Boolean Masking
high_value_transactions = transaction_amounts[transaction_amounts > 2000]
```

```
# Fancy Indexing
fancy_indexing_example = transaction_amounts[[0, 5, 10]]
```

```
# Sorting
sorted_transactions = np.sort(transaction_amounts)
```

```
# Partial Sorting (Top 5 highest transactions)
top_5_transactions = np.partition(transaction_amounts, -5)[-5:]
```

```
# Additional NumPy Operations
# Mathematical Functions
sqrt_transactions = np.sqrt(transaction_amounts)
log_transactions = np.log(transaction_amounts + 1)

```

```
# Statistical Functions
median_transaction = np.median(transaction_amounts)
std_transaction = np.std(transaction_amounts)
```

```
# Linear Algebra
dot_product = np.dot(account_ages[:5], fraud_labels[:5])
```

```
# Random Numbers
random_values = np.random.normal(loc=50, scale=10, size=5)

```

```
# Advanced Indexing
high_value_indices = np.where(transaction_amounts > 2000)
taken_values = np.take(transaction_amounts, [0, 5, 10])

```

```
# Stacking Arrays
hstacked_array = np.hstack((account_ages[:5].reshape(-1, 1), fraud_labels[:5].reshape(-1, 1)))
```

```
# Unique and Counting
unique_fraud_labels, fraud_counts = np.unique(fraud_labels, return_counts=True)

```

```
# Clipping and Rounding
clipped_transactions = np.clip(transaction_amounts, 0, 5000)
rounded_transactions = np.round(transaction_amounts, 2)

```

```
# Finding Min/Max Locations
max_index = np.argmax(transaction_amounts)
min_index = np.argmin(transaction_amounts)

```

```
# Tile & Repeat Functions
repeated_array = np.repeat(transaction_amounts[:5], 3)
tiled_array = np.tile(transaction_amounts[:5], 3)

```

```
# Correlation and Covariance
correlation = np.corrcoef(df["Account_Age"], df["Fraudulent"])[0, 1]
covariance = np.cov(transaction_amounts, fraud_labels)[0, 1]

```

```
# Generating Custom NumPy Arrays
zeros_array = np.zeros(10)
ones_array = np.ones(10)
linspace_array = np.linspace(0, 100, 10)
logspace_array = np.logspace(1, 3, 10)

```

```
# Printing results
print("Indexed Values:", indexed_values)
print()
print("Sliced Values:", sliced_values)
print()
print("Reshaped Array:\n", reshaped_array)
print()
print("Mean Transaction Amount:", mean_transaction)
print()
print("Sum of Transactions:", sum_transactions)
```

```
Indexed Values: [1292.76 1554.58 2395.02  100.1  1490.5 ]

Sliced Values: [0 0 0 0 0 0 1 0 0 0]

Reshaped Array:
 [[1292.76 1554.58 2395.02]
 [ 100.1  1490.5  2372.04]
 [ 544.81  635.75 2318.87]
 [3656.17    0.   2733.84]
 [2376.37 1924.48  968.78]]

Mean Transaction Amount: 2848.1997950980394

Sum of Transactions: 145258189.55
```

```
print("High Value Transactions:", high_value_transactions[:5])
print()
print("Top 5 Transactions:", top_5_transactions)
print()
print("Square Root Transactions:", sqrt_transactions[:5])
print()
print("Log Transactions:", log_transactions[:5])
print()
print("Median Transaction Amount:", median_transaction)
print()
```

```
High Value Transactions: [2395.02 2372.04 2318.87 3656.17 2733.84]

Top 5 Transactions: [49997.8 49997.8 49997.8 49997.8 49997.8]

Square Root Transactions: [35.95497184 39.42816252 48.93894155 10.00499875 38.60699418]

Log Transactions: [7.16530799 7.34960375 7.78156431 4.61611013 7.3075376 ]

Median Transaction Amount: 2392.0600000000004
```

```
print("Standard Deviation:", std_transaction)
print()
print("Dot Product:", dot_product)
print()
print("Random Values:", random_values)
print()
print("High Value Indices:", high_value_indices)
print()
print("Taken Values:", taken_values)
print()

```

```
Standard Deviation: 4960.376536284364

Dot Product: 0

Random Values: [38.20658443 31.03091709 49.78132788 40.27913498 57.15971118]

High Value Indices: (array([    2,     5,     8, ..., 50996, 50997, 50998]),)

Taken Values: [1292.76 2372.04    0.  ]
```

```
print("Horizontally Stacked Array:\n", hstacked_array)
print()
print("Unique Fraud Labels and Counts:", unique_fraud_labels, fraud_counts)
print()
print("Clipped Transactions:", clipped_transactions[:5])
print()
print("Rounded Transactions:", rounded_transactions[:5])
print()
print("Max Transaction Index:", max_index)
print()
```

```
Horizontally Stacked Array:
 [[119   0]
 [ 79   0]
 [115   0]
 [  3   0]
 [ 57   0]]

Unique Fraud Labels and Counts: [0 1] [48490  2510]

Clipped Transactions: [1292.76 1554.58 2395.02  100.1  1490.5 ]

Rounded Transactions: [1292.76 1554.58 2395.02  100.1  1490.5 ]

Max Transaction Index: 166
```

```
print("Min Transaction Index:", min_index)
print()
print("Repeated Array:", repeated_array)
print()
print("Tiled Array:", tiled_array)
print()
print("Correlation:", correlation)
print()
print("Covariance:", covariance)
```

```
Min Transaction Index: 10

Repeated Array: [1292.76 1292.76 1292.76 1554.58 1554.58 1554.58 2395.02 2395.02 2395.02
  100.1   100.1   100.1  1490.5  1490.5  1490.5 ]

Tiled Array: [1292.76 1554.58 2395.02  100.1  1490.5  1292.76 1554.58 2395.02  100.1
 1490.5  1292.76 1554.58 2395.02  100.1  1490.5 ]

Correlation: 0.006202800889533029

Covariance: 5.6535303496916045
```

```
print()
print("Zeros Array:", zeros_array)
print()
print("Ones Array:", ones_array)
print()
print("Linspace Array:", linspace_array)
print()
print("Logspace Array:", logspace_array)
```

```

Zeros Array: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Ones Array: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

Linspace Array: [  0.          11.11111111  22.22222222  33.33333333  44.44444444
  55.55555556  66.66666667  77.77777778  88.88888889 100.        ]

Logspace Array: [  10.           16.68100537   27.82559402   46.41588834   77.42636827
  129.1549665   215.443469    359.38136638  599.48425032 1000.        ]
```