Python - Case Study

Data Processing with Pandas case study

- Loading Data in Pandas DataFrame

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('/content/LoanData (1).csv')
```

- Printing rows of the Data

```
print(df.head())
```

```
    Loan_ID Gender Married Dependents     Education Self_Employed  \
0  LP001002   Male      No          0      Graduate            No
1  LP001003   Male     Yes          1      Graduate            No
2  LP001005   Male     Yes          0      Graduate           Yes
3  LP001006   Male     Yes          0  Not Graduate            No
4  LP001008   Male      No          0      Graduate            No

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0             5849                0.0         NaN             360.0
1             4583             1508.0       128.0             360.0
2             3000                0.0        66.0             360.0
3             2583             2358.0       120.0             360.0
4             6000                0.0       141.0             360.0

   Credit_History Property_Area Loan_Status
0             1.0         Urban           Y
1             1.0         Rural           N
2             1.0         Urban           Y
3             1.0         Urban           Y
4             1.0         Urban           Y
```

```
print(df.tail())
```

```
     Loan_ID  Gender Married Dependents Education Self_Employed  \
609  LP002978  Female      No          0  Graduate            No
610  LP002979    Male     Yes         3+  Graduate            No
611  LP002983    Male     Yes          1  Graduate            No
612  LP002984    Male     Yes          2  Graduate            No
613  LP002990  Female      No          0  Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
609             2900                0.0        71.0             360.0
610             4106                0.0        40.0             180.0
611             8072              240.0       253.0             360.0
612             7583                0.0       187.0             360.0
613             4583                0.0       133.0             360.0

     Credit_History Property_Area Loan_Status
609             1.0         Rural           Y
610             1.0         Rural           Y
611             1.0         Urban           Y
612             1.0         Urban           Y
613             0.0     Semiurban           N
```

```
[4] print(df.sample(5))
```

```
      Loan_ID  Gender Married Dependents     Education Self_Employed  \
571  LP002847    Male     Yes        NaN      Graduate            No
516  LP002670  Female     Yes          2      Graduate            No
485  LP002544    Male     Yes          1  Not Graduate            No
64   LP001222  Female      No          0      Graduate            No
99   LP001343    Male     Yes          0      Graduate            No

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
571             5116             1451.0       165.0             360.0
516             2031             1632.0       113.0             480.0
485             1958             2436.0       131.0             360.0
64              4166                0.0       116.0             360.0
99              1759             3541.0       131.0             360.0

     Credit_History Property_Area Loan_Status
571             0.0         Urban           N
516             1.0     Semiurban           Y
485             1.0         Rural           Y
64              0.0     Semiurban           N
99              1.0     Semiurban           Y
```

- Printing the column names of the DataFrame

```
[5] print(df.columns.tolist())
```

```
['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term', 'Credit_History', 'Property_Area'
```

- Summary of Data Frame

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
None
```

- Descriptive Statistical Measures of a DataFrame

```
print(df.describe())
```

```
       ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
count       614.000000         614.000000  592.000000         600.00000
mean       5403.459283        1621.245798  146.412162         342.00000
std        6109.041673        2926.248369   85.587325          65.12041
min         150.000000           0.000000    9.000000          12.00000
25%        2877.500000           0.000000  100.000000         360.00000
50%        3812.500000        1188.500000  128.000000         360.00000
75%        5795.000000        2297.250000  168.000000         360.00000
max       81000.000000       41667.000000  700.000000         480.00000

       Credit_History
count      564.000000
mean         0.842199
std          0.364878
min          0.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          1.000000
```

```
print(df.describe(include='all'))
```

```
         Loan_ID Gender Married Dependents Education Self_Employed  \
count        614    601     611        599       614           582
unique       614      2       2          4         2             2
top     LP002990   Male     Yes          0  Graduate            No
freq           1    489     398        345       480           500
mean         NaN    NaN     NaN        NaN       NaN           NaN
std          NaN    NaN     NaN        NaN       NaN           NaN
min          NaN    NaN     NaN        NaN       NaN           NaN
25%          NaN    NaN     NaN        NaN       NaN           NaN
50%          NaN    NaN     NaN        NaN       NaN           NaN
75%          NaN    NaN     NaN        NaN       NaN           NaN
max          NaN    NaN     NaN        NaN       NaN           NaN

        ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
count        614.000000         614.000000  592.000000         600.00000
unique              NaN                NaN         NaN               NaN
top                 NaN                NaN         NaN               NaN
freq                NaN                NaN         NaN               NaN
mean        5403.459283        1621.245798  146.412162         342.00000
std         6109.041673        2926.248369   85.587325          65.12041
min          150.000000           0.000000    9.000000          12.00000
25%         2877.500000           0.000000  100.000000         360.00000
50%         3812.500000        1188.500000  128.000000         360.00000
75%         5795.000000        2297.250000  168.000000         360.00000
max        81000.000000       41667.000000  700.000000         480.00000

        Credit_History Property_Area Loan_Status
count       564.000000           614         614
unique             NaN             3           2
top                NaN     Semiurban           Y
freq               NaN           233         422
mean          0.842199           NaN         NaN
std           0.364878           NaN         NaN
min           0.000000           NaN         NaN
25%           1.000000           NaN         NaN
```

- Missing Data Handing

```
print(df.isnull().sum())
```

```
Loan_ID               0
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```

```
categorical_cols = ['Gender', 'Married', 'Dependents', 'Self_Employed', 'Loan_Amount_Term', 'Credit_History']
for col in categorical_cols:
    df[col] = df[col].fillna(df[col].mode()[0])
```

```
df = df.drop(columns=['Loan_ID'])
print(df.head())
```

```
  Gender Married Dependents     Education Self_Employed  ApplicantIncome  \
0   Male      No          0      Graduate            No             5849
1   Male     Yes          1      Graduate            No             4583
2   Male     Yes          0      Graduate           Yes             3000
3   Male     Yes          0  Not Graduate            No             2583
4   Male      No          0      Graduate            No             6000

   CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  \
0                0.0         NaN             360.0             1.0
1             1508.0       128.0             360.0             1.0
2                0.0        66.0             360.0             1.0
3             2358.0       120.0             360.0             1.0
4                0.0       141.0             360.0             1.0

  Property_Area Loan_Status
0         Urban           Y
1         Rural           N
2         Urban           Y
3         Urban           Y
4         Urban           Y
```

```
df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount'].median())
print(df.head())
```

```
  Gender Married Dependents     Education Self_Employed  ApplicantIncome  \
0   Male      No          0      Graduate            No             5849
1   Male     Yes          1      Graduate            No             4583
2   Male     Yes          0      Graduate           Yes             3000
3   Male     Yes          0  Not Graduate            No             2583
4   Male      No          0      Graduate            No             6000

   CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  \
0                0.0       128.0             360.0             1.0
1             1508.0       128.0             360.0             1.0
2                0.0        66.0             360.0             1.0
3             2358.0       120.0             360.0             1.0
4                0.0       141.0             360.0             1.0

  Property_Area Loan_Status
0         Urban           Y
1         Rural           N
2         Urban           Y
3         Urban           Y
4         Urban           Y
```

- Sorting DataFrame values

```
df = df.sort_values(by='ApplicantIncome', ascending=True)
print(df.head())
```

```
     Gender Married Dependents     Education Self_Employed  ApplicantIncome  \
216    Male     Yes          0      Graduate            No              150
468  Female     Yes          2  Not Graduate            No              210
600  Female      No         3+      Graduate            No              416
500  Female      No          0      Graduate            No              645
188    Male     Yes          0      Graduate           Yes              674

     CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  \
216             1800.0       135.0             360.0             1.0
468             2917.0        98.0             360.0             1.0
600            41667.0       350.0             180.0             1.0
500             3683.0       113.0             480.0             1.0
188             5296.0       168.0             360.0             1.0

    Property_Area Loan_Status
216         Rural           N
468     Semiurban           Y
600         Urban           N
500         Rural           Y
188         Rural           Y
```

```
df = df.sort_values(by=['Education', 'LoanAmount'], ascending=[True, True])
print(df.head())
```

```
     Gender Married Dependents Education Self_Employed  ApplicantIncome  \
568  Female      No          0  Graduate            No             2378
14     Male     Yes          2  Graduate            No             1299
133    Male     Yes          0  Graduate           Yes             3459
555    Male     Yes          1  Graduate            No             5468
147    Male     Yes          1  Graduate            No             1538

     CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  \
568                0.0         9.0             360.0             1.0
14              1086.0        17.0             120.0             1.0
133                0.0        25.0             120.0             1.0
555             1032.0        26.0             360.0             1.0
147             1425.0        30.0             360.0             1.0

    Property_Area Loan_Status
568         Urban           N
14          Urban           Y
133     Semiurban           Y
555     Semiurban           Y
147         Urban           Y
```

- Merge Data Frames

```
discount_info = pd.DataFrame({
    'Property_Area': ['Urban', 'Rural', 'Semiurban'],
    'Discount': [10, 5, 8]
})
merged_df = pd.merge(df, discount_info, on='Property_Area', how='left')
print(merged_df[['Education', 'Property_Area', 'LoanAmount', 'Discount']].head())
```

```
  Education Property_Area  LoanAmount  Discount
0  Graduate         Urban         9.0        10
1  Graduate         Urban        17.0        10
2  Graduate     Semiurban        25.0         8
3  Graduate     Semiurban        26.0         8
4  Graduate         Urban        30.0        10
```

- Apply Function

```
def loan_category(amount):
    if amount >= 250:
        return 'High'
    elif amount >= 150:
        return 'Medium'
    else:
        return 'Low'

df['LoanCategory'] = df['LoanAmount'].apply(loan_category)
print(df[['LoanAmount', 'LoanCategory']].head())
```

```
     LoanAmount LoanCategory
568         9.0          Low
14         17.0          Low
133        25.0          Low
555        26.0          Low
147        30.0          Low
```

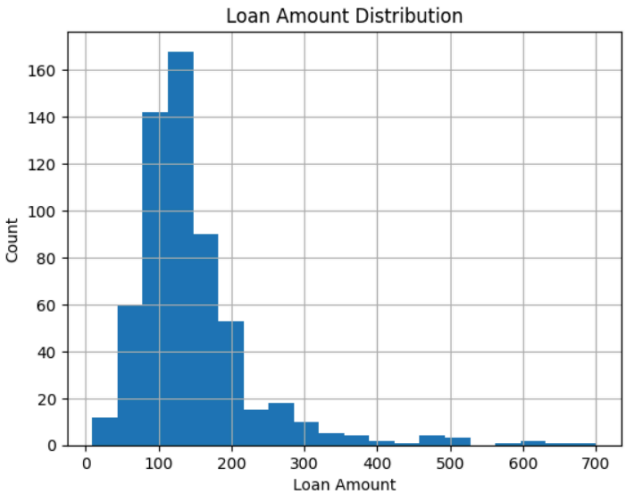
- By using the lambda operator

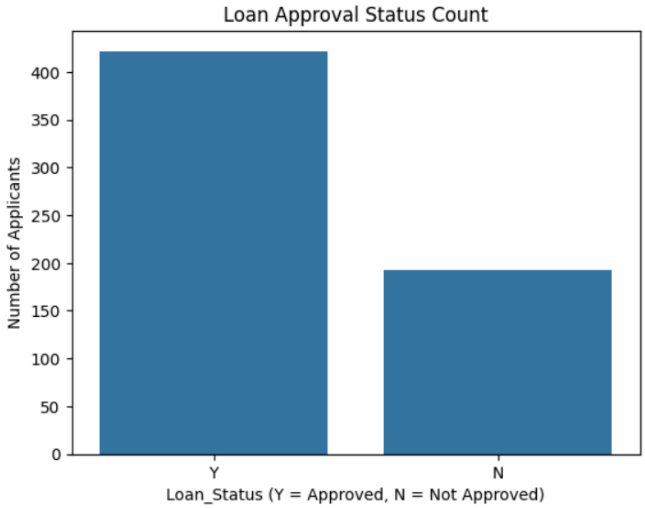
```
df['IncomeGroup'] = df['ApplicantIncome'].apply(lambda income: 'Low' if income < 2500 else 'Medium' if income < 6000 else 'High')
print(df[['ApplicantIncome', 'IncomeGroup']].head())
```

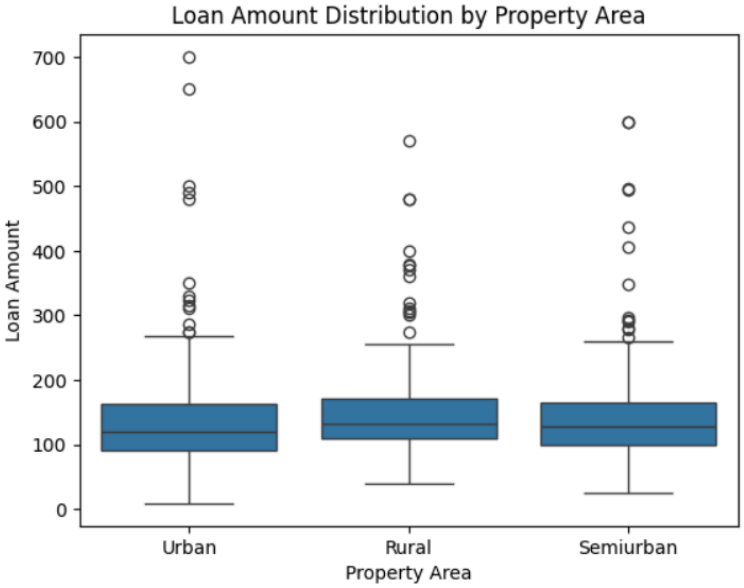
```
     ApplicantIncome IncomeGroup
568             2378         Low
14              1299         Low
133             3459      Medium
555             5468      Medium
147             1538         Low
```

- Visualizing DataFrame

```
df['LoanAmount'].hist(bins=20)
plt.title("Loan Amount Distribution")
plt.xlabel("Loan Amount")
plt.ylabel("Count")
plt.show()
sns.countplot(x='Loan_Status', data=df)
plt.title("Loan Approval Status Count")
plt.xlabel("Loan_Status (Y = Approved, N = Not Approved)")
plt.ylabel("Number of Applicants")
plt.show()
sns.boxplot(x='Property_Area', y='LoanAmount', data=df)
plt.title("Loan Amount Distribution by Property Area")
plt.xlabel("Property Area")
plt.ylabel("Loan Amount")
plt.show()
```


Loan Amount Distribution
Count
Loan Amount
0
20
40
60
80
100
120
140
160
0
100
200
300
400
500
600
700


Loan Approval Status Count
Number of Applicants
Loan_Status (Y = Approved, N = Not Approved)
Y
N
0
50
100
150
200
250
300
350
400


Loan Amount Distribution by Property Area
Loan Amount
Property Area
Urban
Rural
Semiurban
0
100
200
300
400
500
600
700