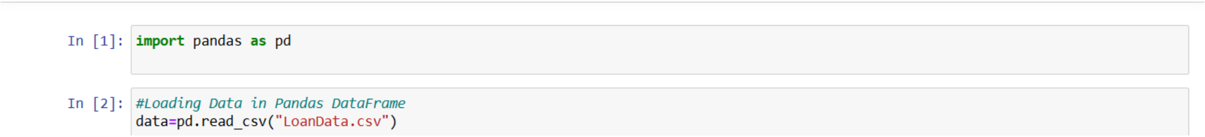
Case Study Python

Name: Divya Sree Murali

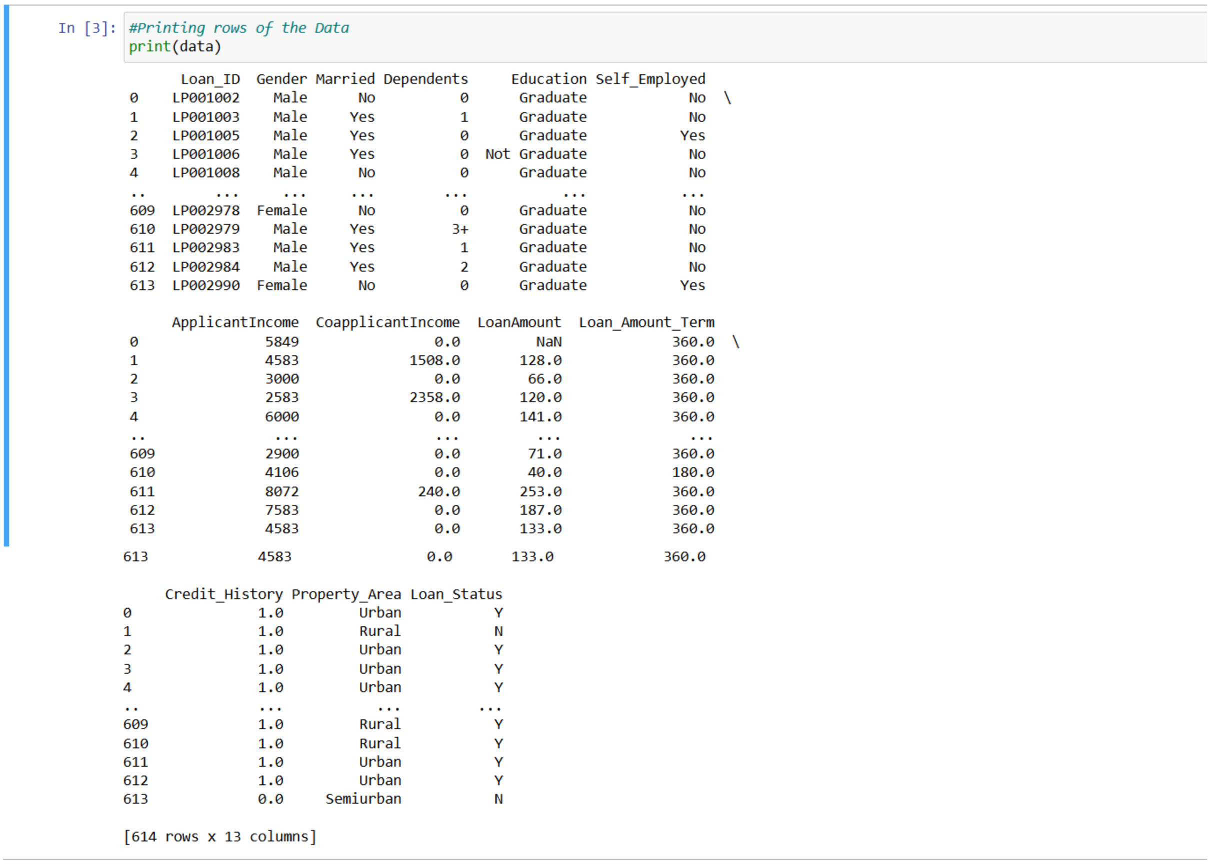
Date: 17/11/2024

1.Loading Data using pandas



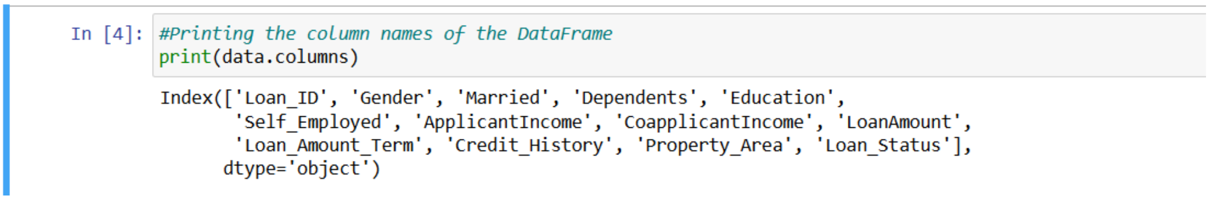
```
In [1]: import pandas as pd

In [2]: #Loading Data in Pandas DataFrame
        data=pd.read_csv("LoanData.csv")
```

2.Printing rows of data



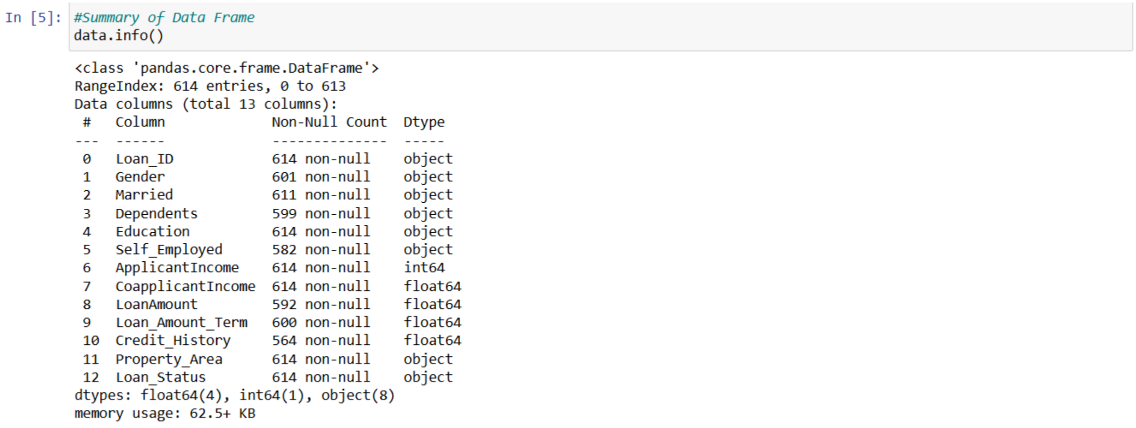
```
In [3]: #Printing rows of the Data
        print(data)

         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
   ..        ...     ...     ...        ...           ...           ...
   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  
   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
   ..               ...                ...         ...               ...
   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

  613              4583               0.0        133.0             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
  ..              ...           ...         ...
  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

  [614 rows x 13 columns]
```

3.Printing column names of data



```
In [4]: #Printing the column names of the DataFrame
        print(data.columns)

        Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
               'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
               'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
              dtype='object')
```

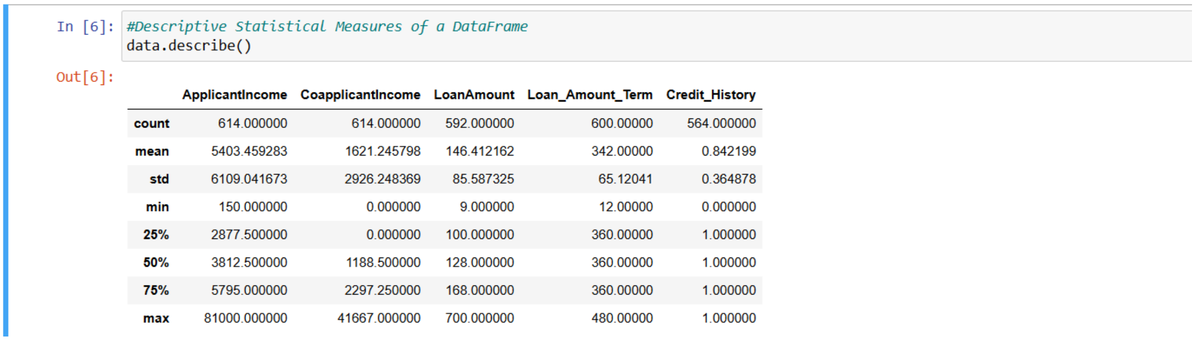
4.Summary of dataframe



```
In [5]: #Summary of Data Frame
        data.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
```

5.Descriptive Statistical measures of a dataframe

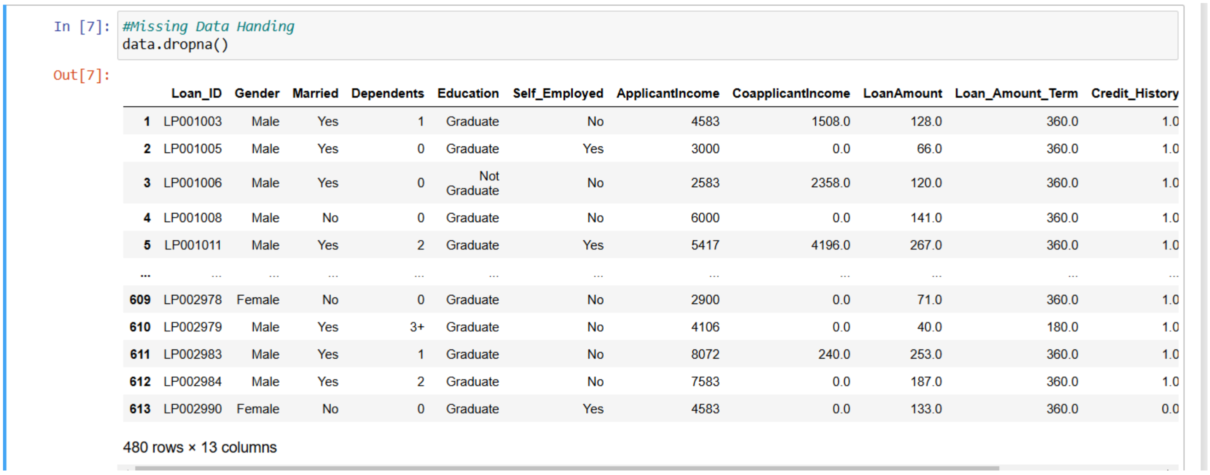


```
In [6]: #Descriptive Statistical Measures of a DataFrame
        data.describe()
```

Out[6]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

6.Missing data handling

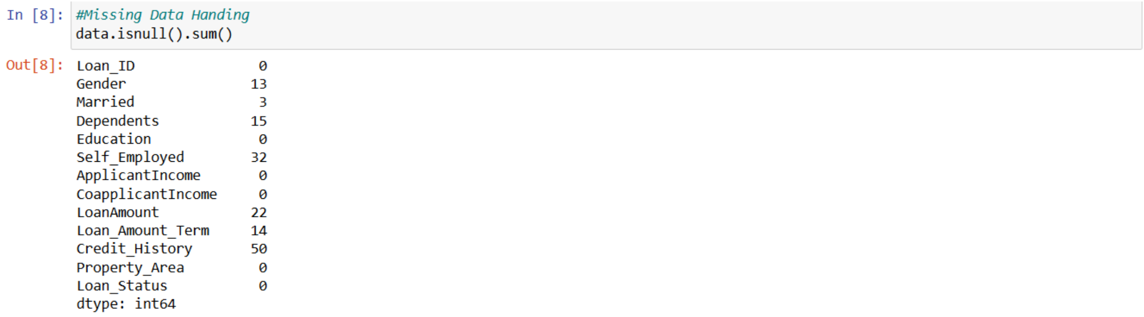


```
In [7]: #Missing Data Handing
        data.dropna()
```

Out[7]:

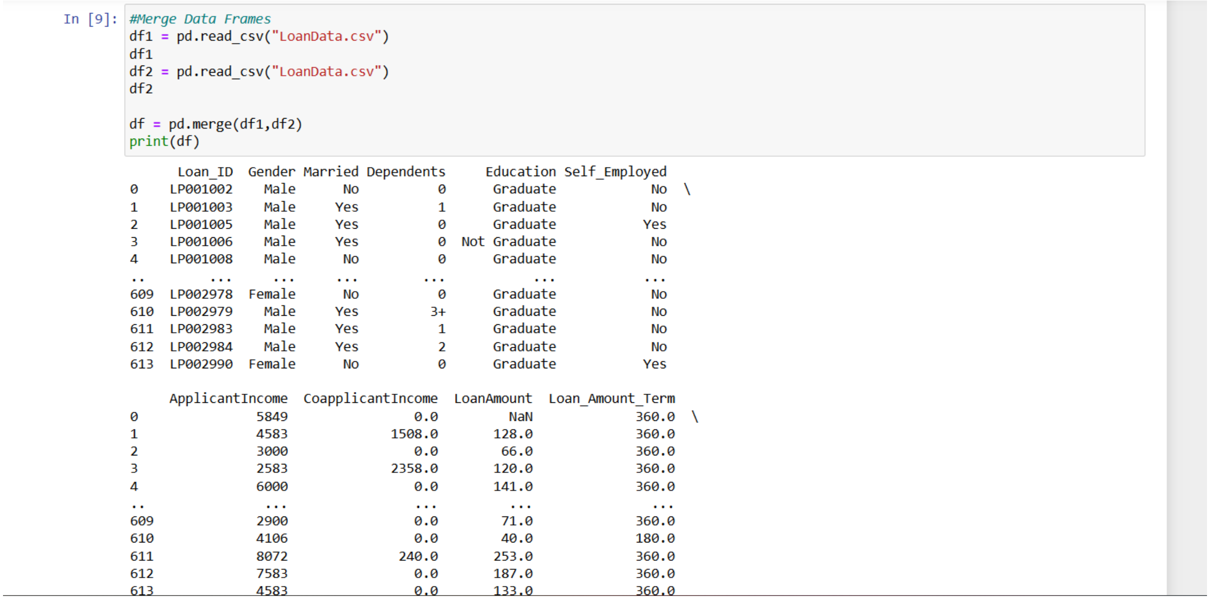
	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0
5	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0	267.0	360.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.0
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.0
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0

480 rows × 13 columns

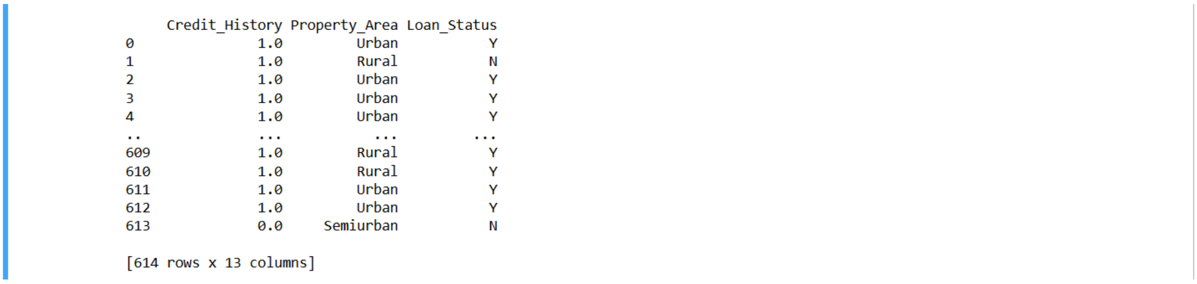


```
In [8]: #Missing Data Handing
        data.isnull().sum()

Out[8]: 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
```

7.Mergee data frames

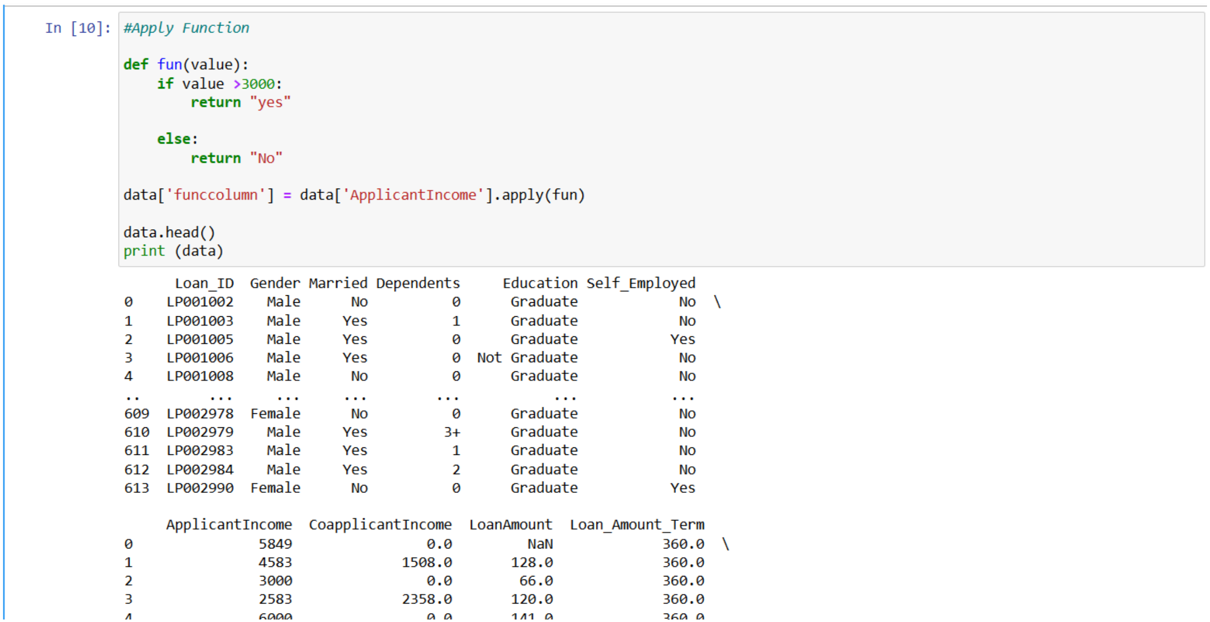


```
In [9]: #Merge Data Frames
        df1 = pd.read_csv("LoanData.csv")
        df1
        df2 = pd.read_csv("LoanData.csv")
        df2

        df = pd.merge(df1,df2)
        print(df)

      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   
..        ...     ...     ...        ...           ...           ...   
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  \
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   
..               ...                ...         ...               ...   
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   
```



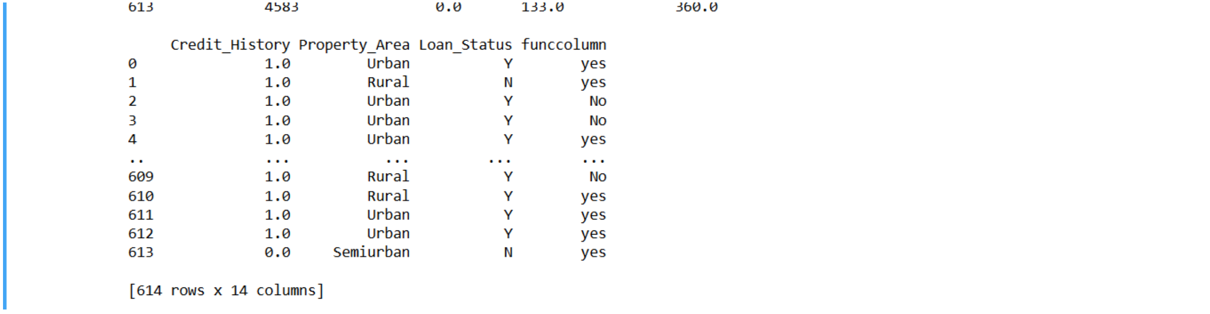
```
     Credit_History Property_Area Loan_Status  
0               1.0         Urban           Y  
1               1.0         Rural           N  
2               1.0         Urban           Y  
3               1.0         Urban           Y  
4               1.0         Urban           Y  
..      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

[614 rows x 13 columns]
```

8.Apply function



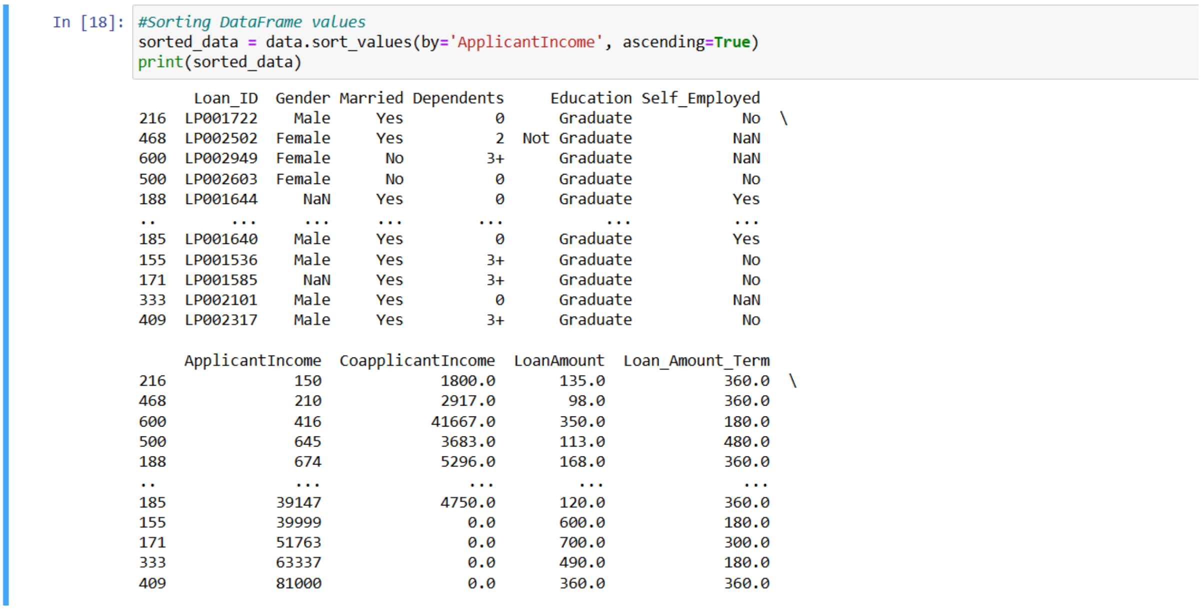
```
In [10]: #Apply Function

         def fun(value):
             if value >3000:
                 return "yes"

             else:
                 return "No"

         data['funccolumn'] = data['ApplicantIncome'].apply(fun)

         data.head()
         print (data)
```

```
      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   
..        ...     ...     ...        ...           ...           ...   
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  \
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   
```



```
613              4583                0.0       133.0             360.0   

     Credit_History Property_Area Loan_Status funccolumn  
0               1.0         Urban           Y        yes  
1               1.0         Rural           N        yes  
2               1.0         Urban           Y         No  
3               1.0         Urban           Y         No  
4               1.0         Urban           Y        yes  
..              ...           ...         ...        ...  
609             1.0         Rural           Y         No  
610             1.0         Rural           Y        yes  
611             1.0         Urban           Y        yes  
612             1.0         Urban           Y        yes  
613             0.0     Semiurban           N        yes  

[614 rows x 14 columns]
```

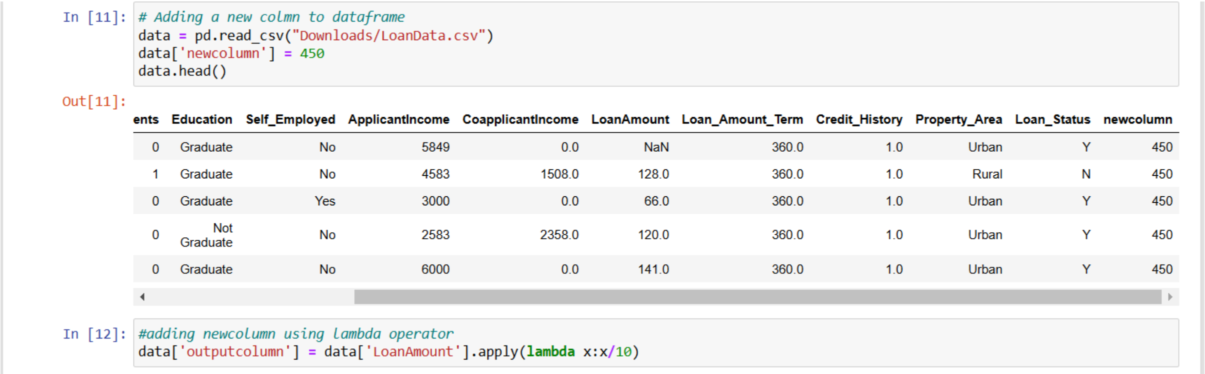
9.Sorting values



```
In [18]: #Sorting DataFrame values
         sorted_data = data.sort_values(by='ApplicantIncome', ascending=True)
         print(sorted_data)
```

```
      Loan_ID  Gender Married Dependents     Education Self_Employed  \
216  LP001722    Male     Yes          0      Graduate            No   
468  LP002502  Female     Yes          2  Not Graduate           NaN   
600  LP002949  Female      No         3+      Graduate           NaN   
500  LP002603  Female      No          0      Graduate            No   
188  LP001644     NaN     Yes          0      Graduate           Yes   
..        ...     ...     ...        ...           ...           ...   
185  LP001640    Male     Yes          0      Graduate           Yes   
155  LP001536    Male     Yes         3+      Graduate            No   
171  LP001585     NaN     Yes         3+      Graduate            No   
333  LP002101    Male     Yes          0      Graduate           NaN   
409  LP002317    Male     Yes         3+      Graduate            No   

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
216              150             1800.0       135.0             360.0   
468              210             2917.0        98.0             360.0   
600              416            41667.0       350.0             180.0   
500              645             3683.0       113.0             480.0   
188              674             5296.0       168.0             360.0   
..               ...                ...         ...               ...   
185            39147             4750.0       120.0             360.0   
155            39999                0.0       600.0             180.0   
171            51763                0.0       700.0             300.0   
333            63337                0.0       490.0             180.0   
409            81000                0.0       360.0             360.0   
```

10.Adding a new column and using lambda operator

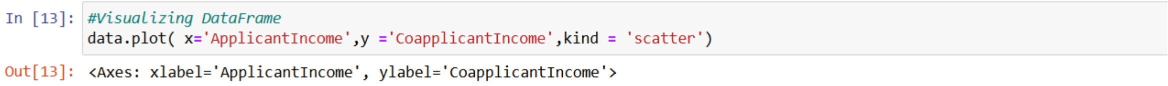


```
In [11]: # Adding a new colmn to dataframe
         data = pd.read_csv("Downloads/LoanData.csv")
         data['newcolumn'] = 450
         data.head()

Out[11]:
```

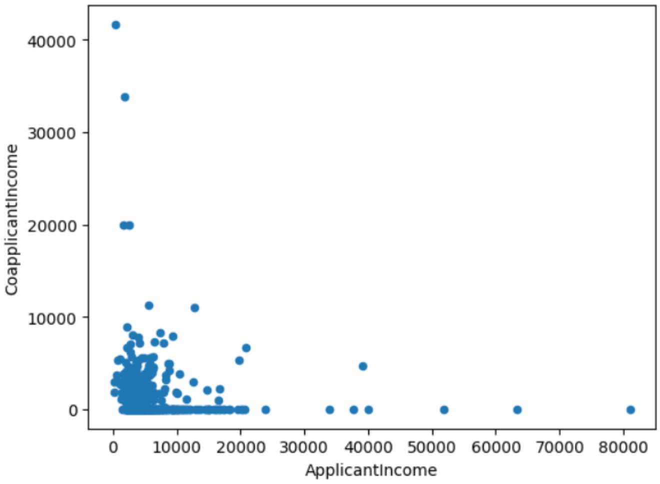
ents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	newcolumn
0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y	450
1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N	450
0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y	450
0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y	450
0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y	450

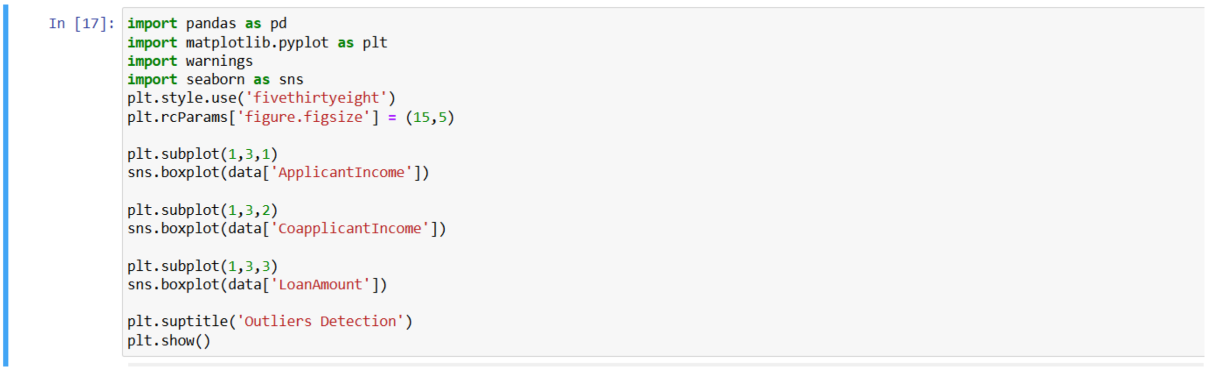
```
In [12]: #adding newcolumn using lambda operator
         data['outputcolumn'] = data['LoanAmount'].apply(lambda x:x/10)
```

11.Visualizing dataframe

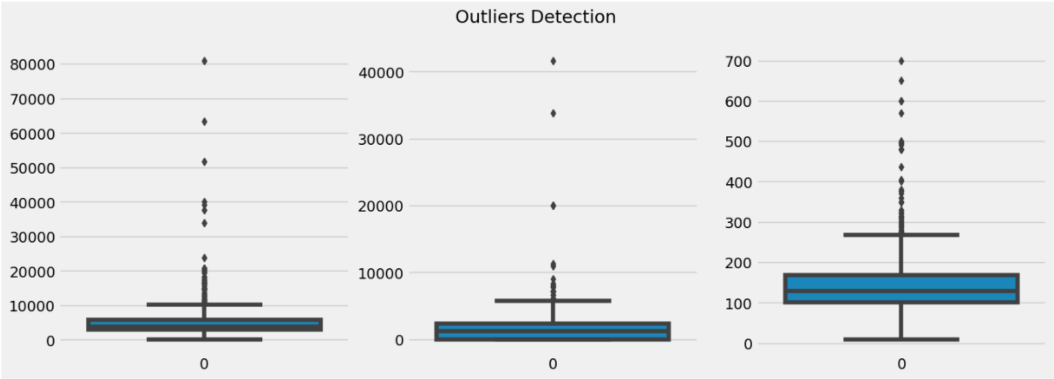


```
In [13]: #Visualizing DataFrame
         data.plot( x='ApplicantIncome',y ='CoapplicantIncome',kind = 'scatter')

Out[13]: <Axes: xlabel='ApplicantIncome', ylabel='CoapplicantIncome'>
```





```
In [17]: import pandas as pd
         import matplotlib.pyplot as plt
         import warnings
         import seaborn as sns
         plt.style.use('fivethirtyeight')
         plt.rcParams['figure.figsize'] = (15,5)

         plt.subplot(1,3,1)
         sns.boxplot(data['ApplicantIncome'])

         plt.subplot(1,3,2)
         sns.boxplot(data['CoapplicantIncome'])

         plt.subplot(1,3,3)
         sns.boxplot(data['LoanAmount'])

         plt.suptitle('Outliers Detection')
         plt.show()
```


Outliers Detection
80000
70000
60000
50000
40000
30000
20000
10000
0
0
40000
30000
20000
10000
0
0
700
600
500
400
300
200
100
0
0