[1]:
```
# Lib for extraction, manipulation, analysis
import numpy as np
import pandas as pd

# for visualization
import matplotlib.pyplot as plt
import seaborn as sns

# for stats
import scipy.stats
from scipy.stats import shapiro, chi2, normaltest, kstest, zscore

# train test split
from sklearn.model_selection import train_test_split
```

[2]:
```
df = pd.read_csv(r"C:\Users\riyac\Downloads\archive_1\Loan_Data.csv")
df
```

[2]:

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | LP001002 | Male | No | 0 | Graduate | No | 5849 | 0.0 | NaN | 360.0 | 1.0 | |
| 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 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 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 | Semi |

614 rows × 13 columns

## EDA (Exploratory Data Analysis)

[4]:
```
# check for no. of rows and no. of columns

df.shape
```

[4]:
```
(614, 13)
```

[5]:
```
# check for size

df.size
```

[5]:
```
7982
```

[6]:
```
# check for information

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
```

[7]:
```
# check for stats

df.describe()
```

[7]:

| | 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 |

[185]:
```
# check for columns datatype

df.dtypes
```

[185]:
```
Gender                int64
Married               int64
Dependents           object
Education             int64
Self_Employed         int64
ApplicantIncome     float64
CoapplicantIncome   float64
LoanAmount          float64
Loan_Amount_Term    float64
Credit_History      float64
Property_Area         int64
Loan_Status           int64
dtype: object
```

[9]:
```
df.isna().sum()
```

[9]:
```
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
```

[10]:
```
df['Gender'].fillna(df['Gender'].mode()[0],inplace=True)
df['Married'].fillna(df['Married'].mode()[0],inplace=True)
df['Dependents'].fillna(df['Dependents'].mode()[0],inplace=True)
df['Self_Employed'].fillna(df['Self_Employed'].mode()[0],inplace=True)
df['LoanAmount'].fillna(df['LoanAmount'].mean(),inplace=True)
```

```
df['Credit_History'].fillna(df['Credit_History'].mode()[0],inplace=True)
df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mode()[0],inplace=True)
```

```
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1311568322.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df['Gender'].fillna(df['Gender'].mode()[0],inplace=True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1311568322.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df['Married'].fillna(df['Married'].mode()[0],inplace=True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1311568322.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df['Dependents'].fillna(df['Dependents'].mode()[0],inplace=True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1311568322.py:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df['Self_Employed'].fillna(df['Self_Employed'].mode()[0],inplace=True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1311568322.py:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df['LoanAmount'].fillna(df['LoanAmount'].mean(),inplace=True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1311568322.py:6: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df['Credit_History'].fillna(df['Credit_History'].mode()[0],inplace=True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1311568322.py:7: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mode()[0],inplace=True)
```

[11]:
```
df.isnull().sum()
```

[11]:
```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

## Analysis for ApplicantIncome

[13]:
```
# check for ApplicantIncome

mean_ApplicantIncome= df["ApplicantIncome"].mean()
mean_ApplicantIncome
```

[13]:
```
5403.459283387622
```

[14]:
```
# check for median

median_ApplicantIncome= df["ApplicantIncome"].median()
median_ApplicantIncome
```

[14]:
```
3812.5
```

[15]:
```
# check for most repitative ApplicantINcome value
Mode_ApplicantIncome= df["ApplicantIncome"].mode()[0]
Mode_ApplicantIncome
```

[15]:
```
2500
```

[16]:
```
# check for spread of data

df["ApplicantIncome"].var()
```

[16]:
```
37320390.167181164
```

[17]:
```
# check for spread of data  around its mean

df["ApplicantIncome"].std()
```

[17]:
```
6109.041673387174
```

[18]:
```
# check for skewness

df["ApplicantIncome"].skew()
```

[18]:
```
6.539513113994625
```

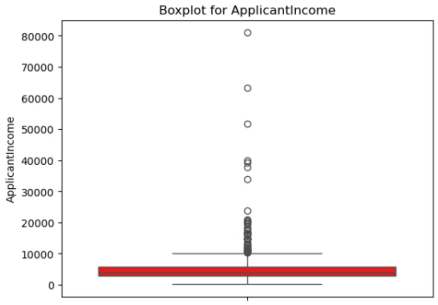
[19]:
```
df["ApplicantIncome"].max() - df["ApplicantIncome"].min()
```

[19]:
```
80850
```

[20]:
```
def Checking_and_Handling_Of_Outliers(df, col):
    sns.boxplot(df[col], color = "Red")
    plt.title(f"Boxplot for {col}")
    plt.show()

    q1 = df[col].quantile(0.25)
    q3 = df[col].quantile(0.75)

    iqr = q3 - q1

    LowerTail = q1 - 1.5*iqr
    UpperTail = q3 + 1.5*iqr

    print(f"25% Quantile q1 = {q1}\n75% Quantile q3 = {q3}\nIQR = {iqr}\n")
    print("-"*80)
    print(f"Lower Tail = {LowerTail}\nUpper Tail = {UpperTail}")
    print("-"*80)
```

```
    # Checking for Outliers
    Outliers = df[(df[col] < LowerTail) | (df[col] > UpperTail)]
    print("\nOutliers :\n",Outliers)
    print("-"*80)
    #Handling of Outliers :
    df.loc[df[col] < LowerTail, col] = LowerTail # all outliers less than lowertail, assigned by lowertail value
    df.loc[df[col] > UpperTail, col] = UpperTail # all outliers greater than uppertail, assigned by uppertail value
    print("After handling of Outliers data:\n")
    print(df.head())
```

[21]:
```
Checking_and_Handling_Of_Outliers(df,'ApplicantIncome')
```

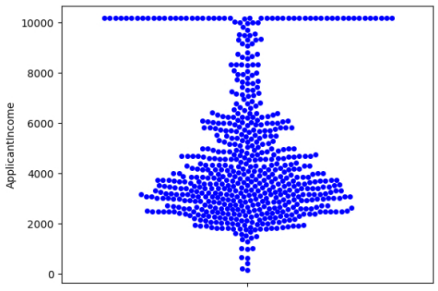



```
25% Quantile q1 = 2877.5
75% Quantile q3 = 5795.0
IQR = 2917.5

--------------------------------------------------------------------------------
Lower Tail = -1498.75
Upper Tail = 10171.25
--------------------------------------------------------------------------------

Outliers :
      Loan_ID  Gender Married Dependents     Education Self_Employed  \
9    LP001020    Male     Yes          1      Graduate            No
34   LP001100    Male      No         3+      Graduate            No
54   LP001186  Female     Yes          1      Graduate           Yes
67   LP001233    Male     Yes          1      Graduate            No
102  LP001350    Male     Yes          0      Graduate            No
106  LP001369    Male     Yes          2      Graduate            No
115  LP001401    Male     Yes          1      Graduate            No
119  LP001422  Female      No          0      Graduate            No
126  LP001448    Male     Yes         3+      Graduate            No
128  LP001451    Male     Yes          1      Graduate           Yes
130  LP001469    Male      No          0      Graduate           Yes
138  LP001492    Male      No          0      Graduate            No
144  LP001508    Male     Yes          2      Graduate            No
146  LP001516  Female     Yes          2      Graduate            No
155  LP001536    Male     Yes         3+      Graduate            No
171  LP001585    Male     Yes         3+      Graduate            No
183  LP001637    Male     Yes          1      Graduate            No
185  LP001640    Male     Yes          0      Graduate           Yes
191  LP001656    Male      No          0      Graduate            No
199  LP001673    Male      No          0      Graduate           Yes
254  LP001844    Male      No          0      Graduate           Yes
258  LP001859    Male     Yes          0      Graduate            No
271  LP001891    Male     Yes          0      Graduate            No
278  LP001907    Male     Yes          0      Graduate            No
284  LP001922    Male     Yes          0      Graduate            No
308  LP001996    Male      No          0      Graduate            No
324  LP002065    Male     Yes         3+      Graduate            No
333  LP002101    Male     Yes          0      Graduate            No
369  LP002191    Male     Yes          0      Graduate            No
370  LP002194  Female      No          0      Graduate           Yes
409  LP002317    Male     Yes         3+      Graduate            No
424  LP002364    Male     Yes          0      Graduate            No
432  LP002386    Male      No          0      Graduate            No
438  LP002403    Male      No          0      Graduate           Yes
443  LP002422    Male      No          1      Graduate            No
467  LP002501    Male     Yes          0      Graduate            No
475  LP002527    Male     Yes          2      Graduate           Yes
478  LP002531    Male     Yes          1      Graduate           Yes
483  LP002541    Male     Yes          0      Graduate            No
487  LP002547    Male     Yes          1      Graduate            No
493  LP002582  Female      No          0  Not Graduate           Yes
506  LP002624    Male     Yes          0      Graduate            No
509  LP002634  Female      No          1      Graduate            No
525  LP002699    Male     Yes          2      Graduate           Yes
533  LP002729    Male      No          1      Graduate            No
534  LP002731  Female      No          0  Not Graduate           Yes
561  LP002813  Female     Yes          1      Graduate           Yes
572  LP002855    Male     Yes          2      Graduate            No
594  LP002938    Male     Yes          0      Graduate           Yes
604  LP002959  Female     Yes          1      Graduate            No

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
9              12841            10968.0  349.000000             360.0
34             12500             3000.0  320.000000             360.0
54             11500                0.0  286.000000             360.0
67             10750                0.0  312.000000             360.0
102            13650                0.0  146.412162             360.0
106            11417             1126.0  225.000000             360.0
115            14583                0.0  185.000000             180.0
119            10408                0.0  259.000000             360.0
126            23803                0.0  370.000000             360.0
128            10513             3850.0  160.000000             180.0
130            20166                0.0  650.000000             480.0
138            14999                0.0  242.000000             360.0
144            11757                0.0  187.000000             180.0
146            14866                0.0   70.000000             360.0
155            39999                0.0  600.000000             180.0
171            51763                0.0  700.000000             300.0
183            33846                0.0  260.000000             360.0
185            39147             4750.0  120.000000             360.0
191            12000                0.0  164.000000             360.0
199            11000                0.0   83.000000             360.0
254            16250                0.0  192.000000             360.0
258            14683             2100.0  304.000000             360.0
271            11146                0.0  136.000000             360.0
278            14583                0.0  436.000000             360.0
284            20667                0.0  146.412162             360.0
308            20233                0.0  480.000000             360.0
324            15000                0.0  300.000000             360.0
333            63337                0.0  490.000000             180.0
369            19730             5266.0  570.000000             360.0
370            15759                0.0   55.000000             360.0
409            81000                0.0  360.000000             360.0
424            14880                0.0   96.000000             360.0
432            12876                0.0  405.000000             360.0
438            10416                0.0  187.000000             360.0
443            37719                0.0  152.000000             360.0
467            16692                0.0  110.000000             360.0
475            16525             1014.0  150.000000             360.0
478            16667             2250.0   86.000000             360.0
483            10833                0.0  234.000000             360.0
487            18333                0.0  500.000000             360.0
493            17263                0.0  225.000000             360.0
506            20833             6667.0  480.000000             360.0
509            13262                0.0   40.000000             360.0
525            17500                0.0  400.000000             360.0
533            11250                0.0  196.000000             360.0
534            18165                0.0  125.000000             360.0
561            19484                0.0  600.000000             360.0
572            16666                0.0  275.000000             360.0
594            16120                0.0  260.000000             360.0
604            12000                0.0  496.000000             360.0

     Credit_History Property_Area Loan_Status
9               1.0     Semiurban           N
34              1.0         Rural           N
54              0.0         Urban           N
67              1.0         Urban           Y
```

```
102            1.0        Urban          Y
106            1.0        Urban          Y
115            1.0        Rural          Y
119            1.0        Urban          Y
126            1.0        Rural          Y
128            0.0        Urban          N
130            1.0        Urban          Y
138            0.0    Semiurban          N
144            1.0        Urban          Y
146            1.0        Urban          Y
155            0.0    Semiurban          Y
171            1.0        Urban          Y
183            1.0    Semiurban          N
185            1.0    Semiurban          Y
191            1.0    Semiurban          N
199            1.0        Urban          N
254            0.0        Urban          N
258            1.0        Rural          N
271            1.0        Urban          Y
278            1.0    Semiurban          Y
284            1.0        Rural          N
308            1.0        Rural          N
324            1.0        Rural          Y
333            1.0        Urban          Y
369            1.0        Rural          N
370            1.0    Semiurban          Y
409            0.0        Rural          N
424            1.0    Semiurban          Y
432            1.0    Semiurban          Y
438            0.0        Urban          N
443            1.0    Semiurban          Y
467            1.0    Semiurban          Y
475            1.0        Rural          Y
478            1.0    Semiurban          Y
483            1.0    Semiurban          Y
487            1.0        Urban          N
493            1.0    Semiurban          Y
506            1.0        Urban          Y
509            1.0        Urban          Y
525            1.0        Rural          Y
533            1.0    Semiurban          N
534            1.0        Urban          Y
561            1.0    Semiurban          Y
572            1.0        Urban          Y
594            1.0        Urban          Y
604            1.0    Semiurban          Y
--------------------------------------------------------------------------------
After handling of Outliers 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

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0           5849.0                0.0  146.412162             360.0
1           4583.0             1508.0  128.000000             360.0
2           3000.0                0.0   66.000000             360.0
3           2583.0             2358.0  120.000000             360.0
4           6000.0                0.0  141.000000             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
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1527558741.py:24: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an er
ror in a future version of pandas. Value '-1498.75' has dtype incompatible with int64, please explicitly cast to a compatible dtype first.
  df.loc[df[col] < LowerTail, col] = LowerTail # all outliers less than lowertail, assigned by lowertail value
```

[31]:
```
sns.swarmplot(df["ApplicantIncome"], color = "blue")
```

[31]:
```
<Axes: ylabel='ApplicantIncome'>
```




## Analysis CoapplicantIncome

[43]:
```
# check for CoapplicantIncome

mean_ApplicantIncome= df["CoapplicantIncome"].mean()
mean_ApplicantIncome
```

[43]:
```
1621.2457980271008
```

[44]:
```
# check for median

median_ApplicantIncome= df["CoapplicantIncome"].median()
median_ApplicantIncome
```

[44]:
```
1188.5
```

[45]:
```
# check for most repitative Coapplicant value
Mode_ApplicantIncome= df["CoapplicantIncome"].mode()[0]
Mode_ApplicantIncome
```

[45]:
```
0.0
```

[46]:
```
# check for spread of data

df["CoapplicantIncome"].var()
```

[46]:
```
8562929.518387241
```

[47]:
```
# check for spread of data  around its mean

df["CoapplicantIncome"].std()
```

[47]:
```
2926.2483692241917
```

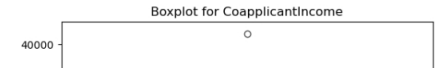
[48]:
```
# check for skewness

df["CoapplicantIncome"].skew()
```

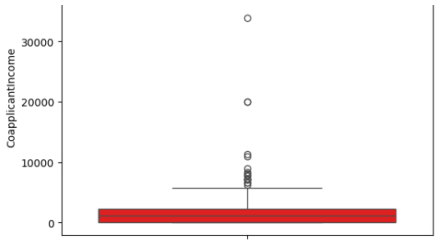
[48]:
```
7.491531216657306
```

[49]:
```
df["CoapplicantIncome"].max() - df["CoapplicantIncome"].min()
```

[49]:
```
41667.0
```

[50]:
```
Checking_and_Handling_Of_Outliers(df,'CoapplicantIncome')
```

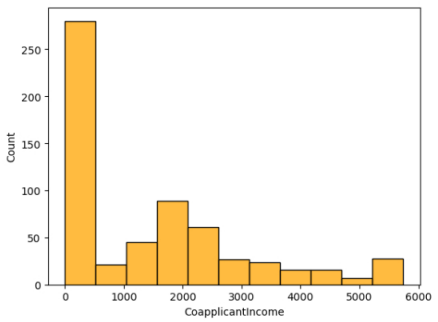





```
25% Quantile q1 = 0.0
75% Quantile q3 = 2297.25
IQR = 2297.25

--------------------------------------------------------------------------------
Lower Tail = -3445.875
Upper Tail = 5743.125
--------------------------------------------------------------------------------

Outliers :
      Loan_ID  Gender Married Dependents     Education Self_Employed  \
9    LP001020    Male     Yes          1      Graduate            No
12   LP001028    Male     Yes          2      Graduate            No
38   LP001114    Male      No          0      Graduate            No
122  LP001431  Female      No          0      Graduate            No
135  LP001488    Male     Yes         3+      Graduate            No
177  LP001610    Male     Yes         3+      Graduate            No
180  LP001633    Male     Yes          1      Graduate            No
253  LP001843    Male     Yes          1  Not Graduate            No
349  LP002138    Male     Yes          0      Graduate            No
372  LP002201    Male     Yes          2      Graduate           Yes
402  LP002297    Male      No          0      Graduate            No
417  LP002342    Male     Yes          2      Graduate           Yes
444  LP002424    Male     Yes          0      Graduate            No
506  LP002624    Male     Yes          0      Graduate            No
513  LP002648    Male     Yes          0      Graduate            No
523  LP002693    Male     Yes          2      Graduate           Yes
581  LP002893    Male      No          0      Graduate            No
600  LP002949  Female      No         3+      Graduate            No

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
9           10171.25            10968.0       349.0             360.0
12           3073.00             8106.0       200.0             360.0
38           4166.00             7210.0       184.0             360.0
122          2137.00             8980.0       137.0             360.0
135          4000.00             7750.0       290.0             360.0
177          5516.00            11300.0       495.0             360.0
180          6400.00             7250.0       180.0             360.0
253          2661.00             7101.0       279.0             180.0
349          2625.00             6250.0       187.0             360.0
372          9323.00             7873.0       308.0             300.0
402          2500.00            20000.0       103.0             360.0
417          1600.00            20000.0       239.0             360.0
444          7333.00             8333.0       175.0             300.0
506         10171.25             6667.0       480.0             360.0
513          2130.00             6666.0        70.0             180.0
523          7948.00             7166.0       480.0             360.0
581          1836.00            33837.0        90.0             360.0
600           416.00            41667.0       350.0             180.0

     Credit_History Property_Area Loan_Status
9               1.0     Semiurban           N
12              1.0         Urban           Y
38              1.0         Urban           Y
122             0.0     Semiurban           Y
135             1.0     Semiurban           N
177             0.0     Semiurban           N
180             0.0         Urban           N
253             1.0     Semiurban           Y
349             1.0         Rural           Y
372             1.0         Rural           Y
402             1.0     Semiurban           Y
417             1.0         Urban           N
444             1.0         Rural           Y
506             1.0         Urban           Y
513             1.0     Semiurban           N
523             1.0         Rural           Y
581             1.0         Urban           N
600             1.0         Urban           N
--------------------------------------------------------------------------------
After handling of Outliers 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

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0           5849.0                0.0  146.412162             360.0
1           4583.0             1508.0  128.000000             360.0
2           3000.0                0.0   66.000000             360.0
3           2583.0             2358.0  120.000000             360.0
4           6000.0                0.0  141.000000             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
```

[51]:
```
sns.histplot(df["CoapplicantIncome"], color = "orange")
```

[51]:
```
<Axes: xlabel='CoapplicantIncome', ylabel='Count'>
```




## Analysis for Loan Amount

[61]:
```
# check for LoanAmount

mean_LoanAmount= df["LoanAmount"].mean()
mean_LoanAmount
```

[61]:
```
146.41216216216216
```

[63]:
```
# check for median

median_LoanAmount= df["LoanAmount"].median()
median_LoanAmount
```

[63]: 129.0

[65]:
```
# check for most repitative Coapplicant value
Mode_loanamount= df["LoanAmount"].mode()[0]
Mode_loanamount
```

[65]: 146.41216216216216

[69]:
```
# check for spread of data

df["LoanAmount"].var()
```

[69]: 7062.295974604296

[71]:
```
# check for spread of data  around its mean

df["LoanAmount"].std()
```

[71]: 84.0374676831965

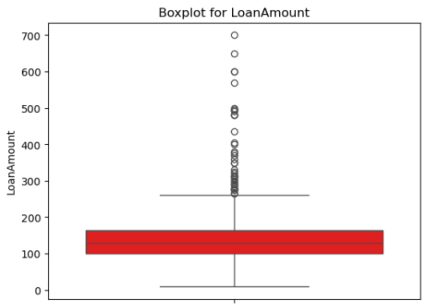
[73]:
```
# check for skewness

df["LoanAmount"].skew()
```

[73]: 2.726601144105299

[75]:
```
df["LoanAmount"].max() - df["LoanAmount"].min() ## range
```

[75]: 691.0

[77]:
```
Checking_and_Handling_Of_Outliers(df,'LoanAmount')
```

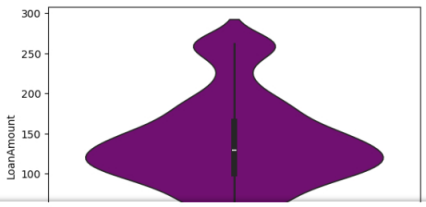



```
25% Quantile q1 = 100.25
75% Quantile q3 = 164.75
IQR = 64.5

--------------------------------------------------------------------------------
Lower Tail = 3.5
Upper Tail = 261.5
--------------------------------------------------------------------------------

Outliers :
      Loan_ID  Gender Married Dependents     Education Self_Employed  \
5    LP001011    Male     Yes          2      Graduate           Yes
9    LP001020    Male     Yes          1      Graduate            No
21   LP001046    Male     Yes          1      Graduate            No
34   LP001100    Male      No         3+      Graduate            No
54   LP001186  Female     Yes          1      Graduate           Yes
67   LP001233    Male     Yes          1      Graduate            No
83   LP001273    Male     Yes          0      Graduate            No
126  LP001448    Male     Yes         3+      Graduate            No
130  LP001469    Male      No          0      Graduate           Yes
135  LP001488    Male     Yes         3+      Graduate            No
155  LP001536    Male     Yes         3+      Graduate            No
161  LP001562    Male     Yes          0      Graduate            No
171  LP001585    Male     Yes         3+      Graduate            No
177  LP001610    Male     Yes         3+      Graduate            No
233  LP001776  Female      No          0      Graduate            No
253  LP001843    Male     Yes          1  Not Graduate            No
258  LP001859    Male     Yes          0      Graduate            No
260  LP001865    Male     Yes          1      Graduate            No
278  LP001907    Male     Yes          0      Graduate            No
308  LP001996    Male      No          0      Graduate            No
324  LP002065    Male     Yes         3+      Graduate            No
325  LP002067    Male     Yes          1      Graduate           Yes
333  LP002101    Male     Yes          0      Graduate            No
351  LP002140    Male      No          0      Graduate            No
369  LP002191    Male     Yes          0      Graduate            No
372  LP002201    Male     Yes          2      Graduate           Yes
381  LP002229    Male      No          0      Graduate            No
391  LP002262    Male     Yes         3+      Graduate            No
409  LP002317    Male     Yes         3+      Graduate            No
432  LP002386    Male      No          0      Graduate            No
487  LP002547    Male     Yes          1      Graduate            No
506  LP002624    Male     Yes          0      Graduate            No
514  LP002652    Male      No          0      Graduate            No
523  LP002693    Male     Yes          2      Graduate           Yes
525  LP002699    Male     Yes          2      Graduate           Yes
536  LP002734    Male     Yes          0      Graduate            No
561  LP002813  Female     Yes          1      Graduate           Yes
572  LP002855    Male     Yes          2      Graduate            No
592  LP002933    Male      No         3+      Graduate           Yes
600  LP002949  Female      No         3+      Graduate            No
604  LP002959  Female     Yes          1      Graduate            No

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
5             5417.00            4196.000       267.0             360.0
9            10171.25            5743.125       349.0             360.0
21            5955.00            5625.000       315.0             360.0
34           10171.25            3000.000       320.0             360.0
54           10171.25               0.000       286.0             360.0
67           10171.25               0.000       312.0             360.0
83            6000.00            2250.000       265.0             360.0
126          10171.25               0.000       370.0             360.0
130          10171.25               0.000       650.0             480.0
135           4000.00            5743.125       290.0             360.0
155          10171.25               0.000       600.0             180.0
161           7933.00               0.000       275.0             360.0
171          10171.25               0.000       700.0             300.0
177           5516.00            5743.125       495.0             360.0
233           8333.00               0.000       280.0             360.0
253           2661.00            5743.125       279.0             180.0
258          10171.25            2100.000       304.0             360.0
260           6083.00            4250.000       330.0             360.0
278          10171.25               0.000       436.0             360.0
308          10171.25               0.000       480.0             360.0
324          10171.25               0.000       300.0             360.0
325           8666.00            4983.000       376.0             360.0
333          10171.25               0.000       490.0             180.0
351           8750.00            4167.000       308.0             360.0
369          10171.25            5266.000       570.0             360.0
372           9323.00            5743.125       380.0             300.0
381           5941.00            4232.000       296.0             360.0
391           9504.00               0.000       275.0             360.0
409          10171.25               0.000       360.0             360.0
432          10171.25               0.000       405.0             360.0
487          10171.25               0.000       500.0             360.0
506          10171.25            5743.125       480.0             360.0
514           5815.00            3666.000       311.0             360.0
523           7948.00            5743.125       480.0             360.0
525          10171.25               0.000       400.0             360.0
536           6133.00            3906.000       324.0             360.0
561          10171.25               0.000       600.0             360.0
572          10171.25               0.000       275.0             360.0
592           9357.00               0.000       292.0             360.0
600            416.00            5743.125       350.0             180.0
604          10171.25               0.000       496.0             360.0

     Credit_History Property_Area Loan_Status
5               1.0         Urban           Y
9               1.0     Semiurban           N
```

```
21             1.0        Urban           Y
34             1.0        Rural           N
54             0.0        Urban           N
67             1.0        Urban           Y
83             1.0    Semiurban           N
126            1.0        Rural           Y
130            1.0        Urban           Y
135            1.0    Semiurban           N
155            0.0    Semiurban           Y
161            1.0        Urban           N
171            1.0        Urban           Y
177            0.0    Semiurban           N
233            1.0    Semiurban           Y
253            1.0    Semiurban           Y
258            1.0        Rural           N
260            1.0        Urban           Y
278            1.0    Semiurban           Y
308            1.0        Rural           N
324            1.0        Rural           Y
325            0.0        Rural           N
333            1.0        Urban           Y
351            1.0        Rural           N
369            1.0        Rural           N
372            1.0        Rural           Y
381            1.0    Semiurban           Y
391            1.0        Rural           Y
409            0.0        Rural           N
432            1.0    Semiurban           Y
487            1.0        Urban           N
506            1.0        Urban           Y
514            1.0        Rural           N
523            1.0        Rural           Y
525            1.0        Rural           Y
536            1.0        Urban           Y
561            1.0    Semiurban           Y
572            1.0        Urban           Y
592            1.0    Semiurban           Y
600            1.0        Urban           N
604            1.0    Semiurban           Y
--------------------------------------------------------------------------------
After handling of Outliers 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

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0           5849.0                0.0  146.412162             360.0
1           4583.0             1508.0  128.000000             360.0
2           3000.0                0.0   66.000000             360.0
3           2583.0             2358.0  120.000000             360.0
4           6000.0                0.0  141.000000             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
```

[79]:
```
sns.violinplot(df["LoanAmount"],  color = "purple")
```

[79]:
```
<Axes: ylabel='LoanAmount'>
```




## Analysis for Loan_Amount_Term

[82]:
```
# check for Loan Amount term

mean_loanamountterm= df["Loan_Amount_Term"].mean()
mean_loanamountterm
```

[82]:
```
342.4104234527687
```

[84]:
```
# check for median

median_LoanAmountterm= df["Loan_Amount_Term"].median()
median_LoanAmountterm
```

[84]:
```
360.0
```

[86]:
```
# check for most repitative Coapplicant value

Mode_loanamountterm= df["Loan_Amount_Term"].mode()[0]
Mode_loanamountterm
```

[86]:
```
360.0
```

[88]:
```
# check for spread of data

df["Loan_Amount_Term"].var()
```

[88]:
```
4151.048243539801
```

[90]:
```
# check for spread of data  around its mean

df["Loan_Amount_Term"].std()
```

[90]:
```
64.42862906767301
```

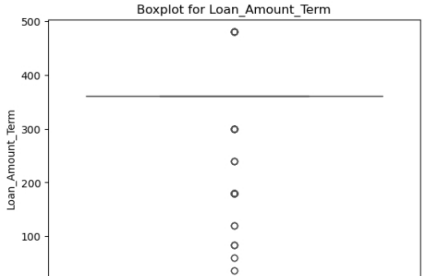
[92]:
```
# check for skewness

df["Loan_Amount_Term"].skew()
```

[92]:
```
-2.4021122563890396
```

[94]:
```
df["Loan_Amount_Term"].max() - df["Loan_Amount_Term"].min() ## range
```

[94]:
```
468.0
```

[96]:
```
Checking_and_Handling_Of_Outliers(df,'Loan_Amount_Term')
```



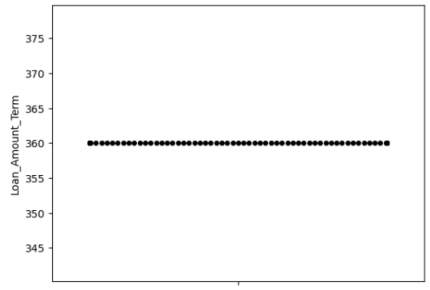
```
25% Quantile q1 = 360.0
75% Quantile q3 = 360.0
IQR = 0.0

--------------------------------------------------------------------------------
Lower Tail = 360.0
Upper Tail = 360.0
--------------------------------------------------------------------------------

Outliers :
      Loan_ID  Gender Married Dependents     Education Self_Employed  \
14   LP001030    Male     Yes          2      Graduate            No
16   LP001034    Male      No          1  Not Graduate            No
62   LP001207    Male     Yes          0  Not Graduate           Yes
66   LP001228    Male      No          0  Not Graduate            No
68   LP001238    Male     Yes         3+  Not Graduate           Yes
..        ...     ...     ...        ...           ...           ...
591  LP002931    Male     Yes          2      Graduate           Yes
593  LP002936    Male     Yes          0      Graduate            No
600  LP002949  Female      No         3+      Graduate            No
605  LP002960    Male     Yes          0  Not Graduate            No
610  LP002979    Male     Yes         3+      Graduate            No

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
14            1299.0           1086.000   17.000000             120.0
16            3596.0              0.000  100.000000             240.0
62            2609.0           3449.000  165.000000             180.0
66            3200.0           2254.000  126.000000             180.0
68            7100.0              0.000  125.000000              60.0
..               ...                ...         ...               ...
591           6000.0              0.000  205.000000             240.0
593           3859.0           3300.000  142.000000             180.0
600            416.0           5743.125  261.500000             180.0
605           2400.0           3800.000  146.412162             180.0
610           4106.0              0.000   40.000000             180.0

     Credit_History Property_Area Loan_Status
14              1.0         Urban           Y
16              1.0         Urban           Y
62              0.0         Rural           N
66              0.0         Urban           N
68              1.0         Urban           Y
..              ...           ...         ...
591             1.0     Semiurban           N
593             1.0         Rural           Y
600             1.0         Urban           N
605             1.0         Urban           N
610             1.0         Rural           Y

[88 rows x 13 columns]
--------------------------------------------------------------------------------
After handling of Outliers 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

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0           5849.0                0.0  146.412162             360.0
1           4583.0             1508.0  128.000000             360.0
2           3000.0                0.0   66.000000             360.0
3           2583.0             2358.0  120.000000             360.0
4           6000.0                0.0  141.000000             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
```

[249]:
```
sns.swarmplot(df["Loan_Amount_Term"], color = "Black")
```

```
C:\Users\riyac\anaconda3\Lib\site-packages\seaborn\categorical.py:3399: UserWarning: 19.4% of the points cannot be placed; you may want to decrease the siz
e of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

[249]:
```
<Axes: ylabel='Loan_Amount_Term'>
```

```
C:\Users\riyac\anaconda3\Lib\site-packages\seaborn\categorical.py:3399: UserWarning: 91.0% of the points cannot be placed; you may want to decrease the siz
e of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```




## Analysis for Property_Area column

[101]:
```
df["Property_Area"].nunique()
```

[101]:
```
3
```

[103]:
```
df["Property_Area"].unique()
```

[103]:
```
array(['Urban', 'Rural', 'Semiurban'], dtype=object)
```

[105]:
```
## Encoding

df["Property_Area"].replace({'Urban' : 1, 'Rural' :2 , 'Semiurban':3}, inplace = True)
df
```

```
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\4190641057.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df["Property_Area"].replace({'Urban' : 1, 'Rural' :2 , 'Semiurban':3}, inplace = True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\4190641057.py:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a
future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('futur
e.no_silent_downcasting', True)`
  df["Property_Area"].replace({'Urban' : 1, 'Rural' :2 , 'Semiurban':3}, inplace = True)
```

[105]:

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | LP001002 | Male | No | 0 | Graduate | No | 5849.0 | 0.0 | 146.412162 | 360.0 | 1.0 | |
| **1** | LP001003 | Male | Yes | 1 | Graduate | No | 4583.0 | 1508.0 | 128.000000 | 360.0 | 1.0 | |
| **2** | LP001005 | Male | Yes | 0 | Graduate | Yes | 3000.0 | 0.0 | 66.000000 | 360.0 | 1.0 | |
| **3** | LP001006 | Male | Yes | 0 | Not Graduate | No | 2583.0 | 2358.0 | 120.000000 | 360.0 | 1.0 | |
| **4** | LP001008 | Male | No | 0 | Graduate | No | 6000.0 | 0.0 | 141.000000 | 360.0 | 1.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **609** | LP002978 | Female | No | 0 | Graduate | No | 2900.0 | 0.0 | 71.000000 | 360.0 | 1.0 | |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **610** | LP002979 | Male | Yes | 3+ | Graduate | No | 4106.0 | 0.0 | 40.000000 | 360.0 | 1.0 |
| **611** | LP002983 | Male | Yes | 1 | Graduate | No | 8072.0 | 240.0 | 253.000000 | 360.0 | 1.0 |
| **612** | LP002984 | Male | Yes | 2 | Graduate | No | 7583.0 | 0.0 | 187.000000 | 360.0 | 1.0 |
| **613** | LP002990 | Female | No | 0 | Graduate | Yes | 4583.0 | 0.0 | 133.000000 | 360.0 | 0.0 |

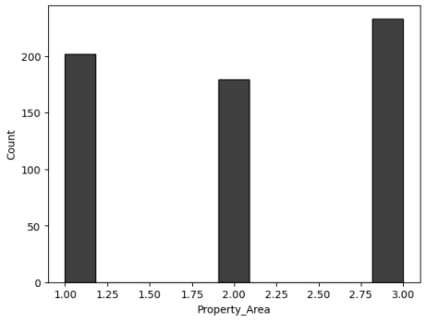
614 rows × 13 columns

[107]:
```
df["Property_Area"].value_counts()
```

[107]:
```
Property_Area
3    233
1    202
2    179
Name: count, dtype: int64
```

[241]:
```
sns.histplot(x = df["Property_Area"], color = "Black")
```

[241]:
```
<Axes: xlabel='Property_Area', ylabel='Count'>
```




## Analysis for Loan_Status column

[112]:
```
df["Loan_Status"].nunique()
```

[112]:
```
2
```

[114]:
```
df["Loan_Status"].unique()
```

[114]:
```
array(['Y', 'N'], dtype=object)
```

[116]:
```
df['Loan_Status'] .replace({'Y':1,'N':2},inplace = True)
df                                                    ### Encoding
```

```
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\975600492.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through cha
ined assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df['Loan_Status'] .replace({'Y':1,'N':2},inplace = True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\975600492.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a f
uture version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('futur
e.no_silent_downcasting', True)`
  df['Loan_Status'] .replace({'Y':1,'N':2},inplace = True)
```

[116]:

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | LP001002 | Male | No | 0 | Graduate | No | 5849.0 | 0.0 | 146.412162 | 360.0 | 1.0 | |
| **1** | LP001003 | Male | Yes | 1 | Graduate | No | 4583.0 | 1508.0 | 128.000000 | 360.0 | 1.0 | |
| **2** | LP001005 | Male | Yes | 0 | Graduate | Yes | 3000.0 | 0.0 | 66.000000 | 360.0 | 1.0 | |
| **3** | LP001006 | Male | Yes | 0 | Not Graduate | No | 2583.0 | 2358.0 | 120.000000 | 360.0 | 1.0 | |
| **4** | LP001008 | Male | No | 0 | Graduate | No | 6000.0 | 0.0 | 141.000000 | 360.0 | 1.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **609** | LP002978 | Female | No | 0 | Graduate | No | 2900.0 | 0.0 | 71.000000 | 360.0 | 1.0 | |
| **610** | LP002979 | Male | Yes | 3+ | Graduate | No | 4106.0 | 0.0 | 40.000000 | 360.0 | 1.0 | |
| **611** | LP002983 | Male | Yes | 1 | Graduate | No | 8072.0 | 240.0 | 253.000000 | 360.0 | 1.0 | |
| **612** | LP002984 | Male | Yes | 2 | Graduate | No | 7583.0 | 0.0 | 187.000000 | 360.0 | 1.0 | |
| **613** | LP002990 | Female | No | 0 | Graduate | Yes | 4583.0 | 0.0 | 133.000000 | 360.0 | 0.0 | |

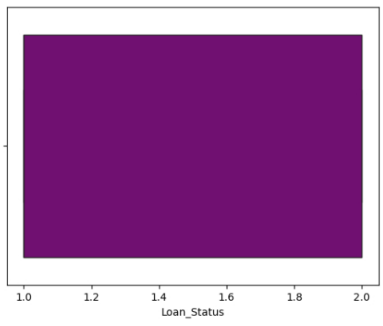
614 rows × 13 columns

[118]:
```
df["Loan_Status"].value_counts()
```

[118]:
```
Loan_Status
1    422
2    192
Name: count, dtype: int64
```

[239]:
```
sns.boxplot(x = df["Loan_Status"], color = "Purple")
```

[239]:
```
<Axes: xlabel='Loan_Status'>
```




## Analysis for Education column

[123]:
```
df["Education"].nunique()
```

[123]:
```
2
```

[125]:
```
df["Education"].unique()
```

[125]:
```
array(['Graduate', 'Not Graduate'], dtype=object)
```

[127]:
```
## Encoding
```

```
df["Education"].replace({'Graduate' : 0, 'Not Graduate' :1 }, inplace = True)
df
```

```
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1565263739.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df["Education"].replace({'Graduate' : 0, 'Not Graduate' :1 }, inplace = True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\1565263739.py:3: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a
future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('futur
e.no_silent_downcasting', True)`
  df["Education"].replace({'Graduate' : 0, 'Not Graduate' :1 }, inplace = True)
```

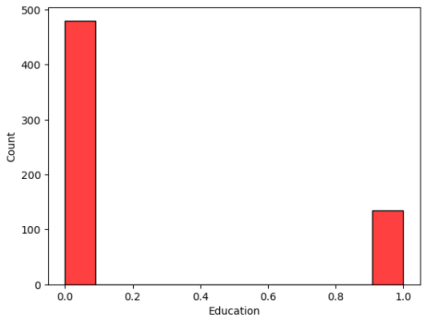
[127]:

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | LP001002 | Male | No | 0 | 0 | No | 5849.0 | 0.0 | 146.412162 | 360.0 | 1.0 | |
| **1** | LP001003 | Male | Yes | 1 | 0 | No | 4583.0 | 1508.0 | 128.000000 | 360.0 | 1.0 | |
| **2** | LP001005 | Male | Yes | 0 | 0 | Yes | 3000.0 | 0.0 | 66.000000 | 360.0 | 1.0 | |
| **3** | LP001006 | Male | Yes | 0 | 1 | No | 2583.0 | 2358.0 | 120.000000 | 360.0 | 1.0 | |
| **4** | LP001008 | Male | No | 0 | 0 | No | 6000.0 | 0.0 | 141.000000 | 360.0 | 1.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **609** | LP002978 | Female | No | 0 | 0 | No | 2900.0 | 0.0 | 71.000000 | 360.0 | 1.0 | |
| **610** | LP002979 | Male | Yes | 3+ | 0 | No | 4106.0 | 0.0 | 40.000000 | 360.0 | 1.0 | |
| **611** | LP002983 | Male | Yes | 1 | 0 | No | 8072.0 | 240.0 | 253.000000 | 360.0 | 1.0 | |
| **612** | LP002984 | Male | Yes | 2 | 0 | No | 7583.0 | 0.0 | 187.000000 | 360.0 | 1.0 | |
| **613** | LP002990 | Female | No | 0 | 0 | Yes | 4583.0 | 0.0 | 133.000000 | 360.0 | 0.0 | |

614 rows × 13 columns

[237]:
```
sns.histplot(x = df["Education"],color = "Red")
```

[237]:
```
<Axes: xlabel='Education', ylabel='Count'>
```




## Analysis for Self_Employed column

[131]:
```
df["Self_Employed"].nunique()
```

[131]:
```
2
```

[132]:
```
df["Self_Employed"].unique()
```

[132]:
```
array(['No', 'Yes'], dtype=object)
```

[134]:
```
df["Self_Employed"].replace({'No' : 0, 'Yes' :1 ,'nan':2}, inplace = True)
df
```

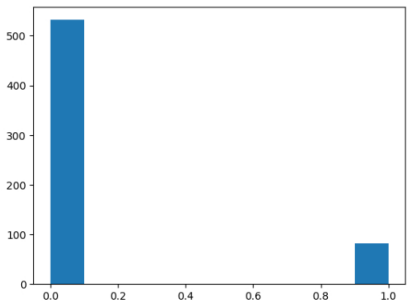
```
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\274255789.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through cha
ined assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df["Self_Employed"].replace({'No' : 0, 'Yes' :1 ,'nan':2}, inplace = True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\274255789.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a f
uture version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('futur
e.no_silent_downcasting', True)`
  df["Self_Employed"].replace({'No' : 0, 'Yes' :1 ,'nan':2}, inplace = True)
```

[134]:

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | LP001002 | Male | No | 0 | 0 | 0 | 5849.0 | 0.0 | 146.412162 | 360.0 | 1.0 | |
| **1** | LP001003 | Male | Yes | 1 | 0 | 0 | 4583.0 | 1508.0 | 128.000000 | 360.0 | 1.0 | |
| **2** | LP001005 | Male | Yes | 0 | 0 | 1 | 3000.0 | 0.0 | 66.000000 | 360.0 | 1.0 | |
| **3** | LP001006 | Male | Yes | 0 | 1 | 0 | 2583.0 | 2358.0 | 120.000000 | 360.0 | 1.0 | |
| **4** | LP001008 | Male | No | 0 | 0 | 0 | 6000.0 | 0.0 | 141.000000 | 360.0 | 1.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **609** | LP002978 | Female | No | 0 | 0 | 0 | 2900.0 | 0.0 | 71.000000 | 360.0 | 1.0 | |
| **610** | LP002979 | Male | Yes | 3+ | 0 | 0 | 4106.0 | 0.0 | 40.000000 | 360.0 | 1.0 | |
| **611** | LP002983 | Male | Yes | 1 | 0 | 0 | 8072.0 | 240.0 | 253.000000 | 360.0 | 1.0 | |
| **612** | LP002984 | Male | Yes | 2 | 0 | 0 | 7583.0 | 0.0 | 187.000000 | 360.0 | 1.0 | |
| **613** | LP002990 | Female | No | 0 | 0 | 1 | 4583.0 | 0.0 | 133.000000 | 360.0 | 0.0 | |

614 rows × 13 columns

[138]:
```
plt.hist(x = df["Self_Employed"])
plt.show()
```



## Analysis for Gender

[141]:
```
df["Gender"].nunique()
```

[141]:
```
2
```

[143]:
```
df["Gender"].unique()
```

[143]:
```
array(['Male', 'Female'], dtype=object)
```

[145]:
```
df["Gender"].replace({'Male' : 0, 'Female' :1 }, inplace = True)
df
```

```
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\2496245181.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df["Gender"].replace({'Male' : 0, 'Female' :1 }, inplace = True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\2496245181.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a
future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('futur
e.no_silent_downcasting', True)`
  df["Gender"].replace({'Male' : 0, 'Female' :1 }, inplace = True)
```

[145]:

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | LP001002 | 0 | No | 0 | 0 | 0 | 5849.0 | 0.0 | 146.412162 | 360.0 | 1.0 | |
| **1** | LP001003 | 0 | Yes | 1 | 0 | 0 | 4583.0 | 1508.0 | 128.000000 | 360.0 | 1.0 | |
| **2** | LP001005 | 0 | Yes | 0 | 0 | 1 | 3000.0 | 0.0 | 66.000000 | 360.0 | 1.0 | |
| **3** | LP001006 | 0 | Yes | 0 | 1 | 0 | 2583.0 | 2358.0 | 120.000000 | 360.0 | 1.0 | |
| **4** | LP001008 | 0 | No | 0 | 0 | 0 | 6000.0 | 0.0 | 141.000000 | 360.0 | 1.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **609** | LP002978 | 1 | No | 0 | 0 | 0 | 2900.0 | 0.0 | 71.000000 | 360.0 | 1.0 | |
| **610** | LP002979 | 0 | Yes | 3+ | 0 | 0 | 4106.0 | 0.0 | 40.000000 | 360.0 | 1.0 | |
| **611** | LP002983 | 0 | Yes | 1 | 0 | 0 | 8072.0 | 240.0 | 253.000000 | 360.0 | 1.0 | |
| **612** | LP002984 | 0 | Yes | 2 | 0 | 0 | 7583.0 | 0.0 | 187.000000 | 360.0 | 1.0 | |
| **613** | LP002990 | 1 | No | 0 | 0 | 1 | 4583.0 | 0.0 | 133.000000 | 360.0 | 0.0 | |

614 rows × 13 columns

[147]:
```
plt.pie(x = df["Gender"])
plt.show()
```

## Analysis for Married

[149]:
```
df["Married"].nunique()
```

[149]:
```
2
```

[150]:
```
df["Married"].unique()
```

[150]:
```
array(['No', 'Yes'], dtype=object)
```

[154]:
```
df["Married"].replace({'No' : 0, 'Yes' :1 }, inplace = True)
df
```

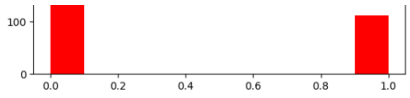
```
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\2187282038.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
ained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves a
s a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  df["Married"].replace({'No' : 0, 'Yes' :1 }, inplace = True)
C:\Users\riyac\AppData\Local\Temp\ipykernel_20072\2187282038.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a
future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('futur
e.no_silent_downcasting', True)`
  df["Married"].replace({'No' : 0, 'Yes' :1 }, inplace = True)
```

[154]:

| Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property_Area | Loan_Sta |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 5849.0 | 0.0 | 146.412162 | 360.0 | 1.0 | 1 | |
| 0 | 1 | 1 | 0 | 0 | 4583.0 | 1508.0 | 128.000000 | 360.0 | 1.0 | 2 | |
| 0 | 1 | 0 | 0 | 1 | 3000.0 | 0.0 | 66.000000 | 360.0 | 1.0 | 1 | |
| 0 | 1 | 0 | 1 | 0 | 2583.0 | 2358.0 | 120.000000 | 360.0 | 1.0 | 1 | |
| 0 | 0 | 0 | 0 | 0 | 6000.0 | 0.0 | 141.000000 | 360.0 | 1.0 | 1 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1 | 0 | 0 | 0 | 0 | 2900.0 | 0.0 | 71.000000 | 360.0 | 1.0 | 2 | |
| 0 | 1 | 3+ | 0 | 0 | 4106.0 | 0.0 | 40.000000 | 360.0 | 1.0 | 2 | |
| 0 | 1 | 1 | 0 | 0 | 8072.0 | 240.0 | 253.000000 | 360.0 | 1.0 | 1 | |
| 0 | 1 | 2 | 0 | 0 | 7583.0 | 0.0 | 187.000000 | 360.0 | 1.0 | 1 | |
| 1 | 0 | 0 | 0 | 1 | 4583.0 | 0.0 | 133.000000 | 360.0 | 0.0 | 3 | |

olumns

[156]:
```
plt.hist(x = df["Gender"],color = "Red")
plt.show()
```




[158]:
```
# Bivariate Analysis - corr between Age and charges column

corr = df.iloc[:, [5,6]].corr()
corr
```

[158]:

| | Self_Employed | ApplicantIncome |
|---|---|---|
| **Self_Employed** | 1.000000 | 0.237122 |
| **ApplicantIncome** | 0.237122 | 1.000000 |

[160]:
```
sns.heatmap(corr, annot=True)
```

[160]: <Axes: >


[162]:
```
q1 = df["Self_Employed"].quantile(0.25)
q2 = df["Self_Employed"].quantile(0.50)
q3 = df["Self_Employed"].quantile(0.75)
```

[164]:
```
iqr = q3 - q1
iqr
```

[164]: 0.0

[166]:
```
LowerTail = q1 - 1.5*iqr
LowerTail
```

[166]: 0.0

[168]:
```
UpperTail = q3 + 1.5*iqr
UpperTail
```

[168]: 0.0

[170]:
```
outliers = df[(df["Self_Employed"] < LowerTail) & (df["Self_Employed"] > UpperTail)]
outliers
```

[170]:

| Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property_Area |
|---|---|---|---|---|---|---|---|---|---|---|---|

[172]:
```
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             614 non-null    int64
 2   Married            614 non-null    int64
 3   Dependents         614 non-null    object
 4   Education          614 non-null    int64
 5   Self_Employed      614 non-null    int64
 6   ApplicantIncome    614 non-null    float64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         614 non-null    float64
 9   Loan_Amount_Term   614 non-null    float64
 10  Credit_History     614 non-null    float64
 11  Property_Area      614 non-null    int64
 12  Loan_Status        614 non-null    int64
dtypes: float64(5), int64(6), object(2)
memory usage: 62.5+ KB
```

[174]:
```
df.drop(columns=['Loan_ID'],inplace=True)
```

[176]:
```
df
```

[176]:

| | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property_Area | Loa |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 5849.0 | 0.0 | 146.412162 | 360.0 | 1.0 | 1 | |
| **1** | 0 | 1 | 1 | 0 | 0 | 4583.0 | 1508.0 | 128.000000 | 360.0 | 1.0 | 2 | |
| **2** | 0 | 1 | 0 | 0 | 1 | 3000.0 | 0.0 | 66.000000 | 360.0 | 1.0 | 1 | |
| **3** | 0 | 1 | 0 | 1 | 0 | 2583.0 | 2358.0 | 120.000000 | 360.0 | 1.0 | 1 | |
| **4** | 0 | 0 | 0 | 0 | 0 | 6000.0 | 0.0 | 141.000000 | 360.0 | 1.0 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **609** | 1 | 0 | 0 | 0 | 0 | 2900.0 | 0.0 | 71.000000 | 360.0 | 1.0 | 2 | |
| **610** | 0 | 1 | 3+ | 0 | 0 | 4106.0 | 0.0 | 40.000000 | 360.0 | 1.0 | 2 | |
| **611** | 0 | 1 | 1 | 0 | 0 | 8072.0 | 240.0 | 253.000000 | 360.0 | 1.0 | 1 | |
| **612** | 0 | 1 | 2 | 0 | 0 | 7583.0 | 0.0 | 187.000000 | 360.0 | 1.0 | 1 | |
| **613** | 1 | 0 | 0 | 0 | 1 | 4583.0 | 0.0 | 133.000000 | 360.0 | 0.0 | 3 | |

614 rows × 12 columns

[ ]:
```
sns.pairplot(df.corr())
```

[187]:
```
df.dtypes
```

[187]:
```
Gender                int64
Married               int64
Dependents           object
Education             int64
Self_Employed         int64
ApplicantIncome     float64
CoapplicantIncome   float64
LoanAmount          float64
Loan_Amount_Term    float64
Credit_History      float64
Property_Area         int64
Loan_Status           int64
dtype: object
```

[197]:
```
df['Dependents'].unique()
```

[197]:
```
array(['0', '1', '2', '3+'], dtype=object)
```

[203]:
```
df['Dependents']=df['Dependents'].replace({'0':0, '1':1, '2':2, '3+':3})
df
```

[203]:

| Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property_Area | Loa |
|---|---|---|---|---|---|---|---|---|---|---|---|

| | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property_Area | Loa |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 5849.0 | 0.0 | 146.412162 | 360.0 | 1.0 | 1 | |
| **1** | 0 | 1 | 1 | 0 | 0 | 4583.0 | 1508.0 | 128.000000 | 360.0 | 1.0 | 2 | |
| **2** | 0 | 1 | 0 | 0 | 1 | 3000.0 | 0.0 | 66.000000 | 360.0 | 1.0 | 1 | |
| **3** | 0 | 1 | 0 | 1 | 0 | 2583.0 | 2358.0 | 120.000000 | 360.0 | 1.0 | 1 | |
| **4** | 0 | 0 | 0 | 0 | 0 | 6000.0 | 0.0 | 141.000000 | 360.0 | 1.0 | 1 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **609** | 1 | 0 | 0 | 0 | 0 | 2900.0 | 0.0 | 71.000000 | 360.0 | 1.0 | 2 | |
| **610** | 0 | 1 | 3 | 0 | 0 | 4106.0 | 0.0 | 40.000000 | 360.0 | 1.0 | 2 | |
| **611** | 0 | 1 | 1 | 0 | 0 | 8072.0 | 240.0 | 253.000000 | 360.0 | 1.0 | 1 | |
| **612** | 0 | 1 | 2 | 0 | 0 | 7583.0 | 0.0 | 187.000000 | 360.0 | 1.0 | 1 | |
| **613** | 1 | 0 | 0 | 0 | 1 | 4583.0 | 0.0 | 133.000000 | 360.0 | 0.0 | 3 | |

614 rows × 12 columns

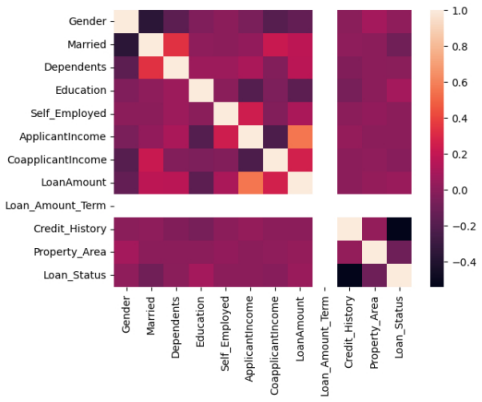
*[205]:
```
# feature selection and check for association between all columns

df.corr()
```

[205]:

| | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Gender** | 1.000000 | -0.364569 | -0.172914 | -0.045364 | 0.000525 | -0.065040 | -0.192438 | -0.144853 | NaN | -0.009170 | |
| **Married** | -0.364569 | 1.000000 | 0.334216 | 0.012304 | 0.004489 | 0.022480 | 0.221217 | 0.178150 | NaN | 0.010938 | |
| **Dependents** | -0.172914 | 0.334216 | 1.000000 | 0.055752 | 0.056798 | 0.111565 | -0.031510 | 0.164333 | NaN | -0.040160 | |
| **Education** | -0.045364 | 0.012304 | 0.055752 | 1.000000 | -0.010383 | -0.203087 | -0.049895 | -0.173348 | NaN | -0.073658 | |
| **Self_Employed** | 0.000525 | 0.004489 | 0.056798 | -0.010383 | 1.000000 | 0.237122 | -0.032784 | 0.114407 | NaN | -0.001550 | |
| **ApplicantIncome** | -0.065040 | 0.022480 | 0.111565 | -0.203087 | 0.237122 | 1.000000 | -0.228956 | 0.558624 | NaN | 0.035345 | |
| **CoapplicantIncome** | -0.192438 | 0.221217 | -0.031510 | -0.049895 | -0.032784 | -0.228956 | 1.000000 | 0.262322 | NaN | -0.001710 | |
| **LoanAmount** | -0.144853 | 0.178150 | 0.164333 | -0.173348 | 0.114407 | 0.558624 | 0.262322 | 1.000000 | NaN | 0.003080 | |
| **Loan_Amount_Term** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| **Credit_History** | -0.009170 | 0.010938 | -0.040160 | -0.073658 | -0.001550 | 0.035345 | -0.001710 | 0.003080 | NaN | 1.000000 | |
| **Property_Area** | 0.082045 | 0.003071 | 0.001781 | -0.003592 | 0.021996 | -0.001184 | 0.018449 | 0.037123 | NaN | 0.030248 | |
| **Loan_Status** | 0.017987 | -0.091478 | -0.010118 | 0.085884 | 0.003700 | 0.000442 | -0.011983 | 0.052050 | NaN | -0.540556 | |

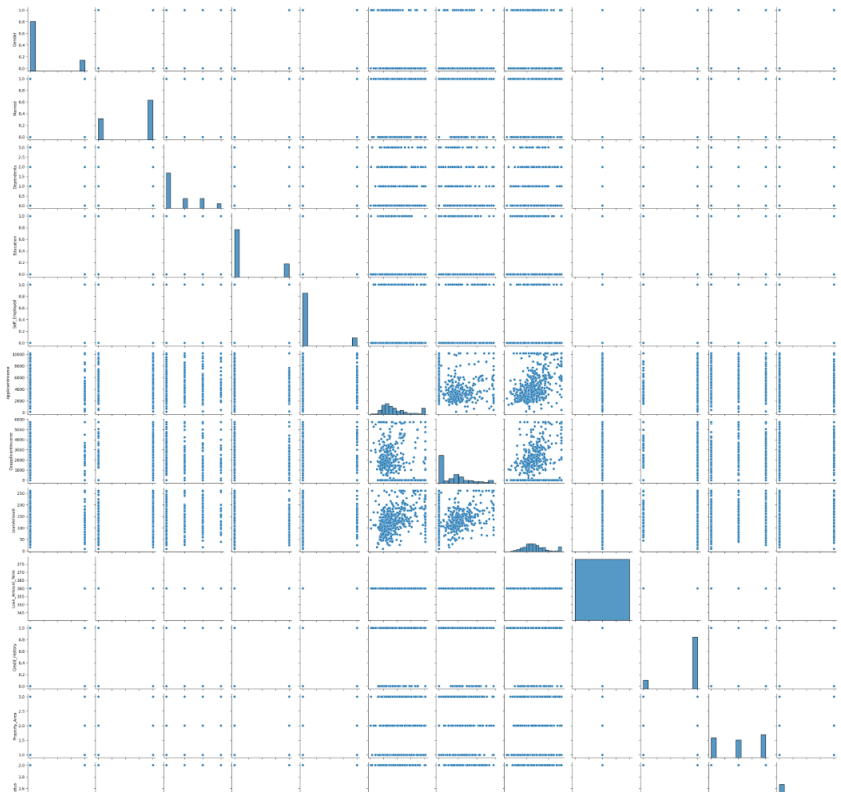
[211]:
```
sns.heatmap(df.corr())
```

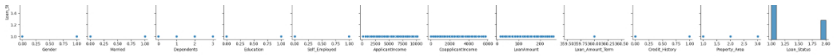
[211]:
```
<Axes: >
```




[213]:
```
# multivariate analysis
sns.pairplot(df)
```

[213]:
```
<seaborn.axisgrid.PairGrid at 0x1522192b980>
```





[214]:
```
x = df.iloc[:, :5]
y = df["Dependents"]
```

[215]:
```
x.head()
```

[215]:

| | Gender | Married | Dependents | Education | Self_Employed |
|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 1 | 1 | 0 | 0 |
| **2** | 0 | 1 | 0 | 0 | 1 |
| **3** | 0 | 1 | 0 | 1 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 |

[219]:
```
y.head()
```

[219]:
```
0    0
1    1
2    0
3    0
4    0
Name: Dependents, dtype: int64
```

## Train test split

[221]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x,y, test_size=0.3)
```

[223]:
```
xtrain.shape
```

[223]:
```
(429, 5)
```

[225]:
```
xtest.shape
```

[225]:
```
(185, 5)
```

[227]:
```
ytrain.shape
```

[227]:
```
(429,)
```

[229]:
```
ytest.shape
```

[229]:
```
(185,)
```

[ ]: