```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
from scipy import stats
import statistics as st

df=pd.read_csv("loan_data_set.csv")

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]

df.head()

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

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

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

df.tail()
```

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

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
609             2900                0.0        71.0             360.0

610             4106                0.0        40.0             180.0

611             8072              240.0       253.0             360.0

612             7583                0.0       187.0             360.0

613             4583                0.0       133.0             360.0


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

```
df.shape
```

```
(614, 13)
```

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

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

```
df.describe()
```

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

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

```
df.shape
```

```
(614, 13)
```

```
df.columns
```

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

```
minimum = df.select_dtypes(include=['number']).min()

print(minimum)
```

```
ApplicantIncome      150.0
CoapplicantIncome      0.0
LoanAmount             9.0
Loan_Amount_Term      12.0
```

```
Credit_History          0.0
dtype: float64
```

```
print(df.select_dtypes(include=['object',
'category']).astype(str).min())
```

```
Loan_ID          LP001002
Gender             Female
Married                No
Dependents              0
Education        Graduate
Self_Employed          No
Property_Area       Rural
Loan_Status             N
dtype: object
```

```
df['ApplicantIncome'].mean()
```

```
np.float64(5403.459283387622)
```

```
df['ApplicantIncome'].median()
```

```
np.float64(3812.5)
```

```
df['ApplicantIncome'].count()
```

```
np.int64(614)
```

```
df['ApplicantIncome'].mode()
```

```
0    2500
Name: ApplicantIncome, dtype: int64
```

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

```
np.float64(6109.041673387178)
```

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

```
np.float64(37320390.16718121)
```

```
male_female=df.groupby("Gender")["Gender"].count()
print(male_female)
```

```
Gender
Female    112
Male      489
Name: Gender, dtype: int64
```

```
df.Education.unique()
```

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

```
mean_math=df.groupby("Gender")["ApplicantIncome"].mean()

print(mean_math)
```

```
Gender
Female    4643.473214
Male      5446.460123
Name: ApplicantIncome, dtype: float64
```

```
df.groupby("Gender")["ApplicantIncome"].sum()
```

```
Gender
Female     520069
Male      2663319
Name: ApplicantIncome, dtype: int64
```

```
df.groupby("Gender")["ApplicantIncome"].count()
```

```
Gender
Female    112
Male      489
Name: ApplicantIncome, dtype: int64
```

```
mean_math_education=df.groupby(['Gender','Education']).ApplicantIncome
.mean()
print(mean_math_education)
```

```
Gender  Education
Female  Graduate        4646.467391
        Not Graduate    4629.700000
Male    Graduate        5992.345745
        Not Graduate    3630.061947
Name: ApplicantIncome, dtype: float64
```

```
df.ApplicantIncome.unique()
```

```
array([ 5849,  4583,  3000,  2583,  6000,  5417,  2333,  3036,  4006,
       12841,  3200,  2500,  3073,  1853,  1299,  4950,  3596,  3510,
        4887,  2600,  7660,  5955,  3365,  3717,  9560,  2799,  4226,
        1442,  3750,  4166,  3167,  4692,  3500, 12500,  2275,  1828,
        3667,  3748,  3600,  1800,  2400,  3941,  4695,  3410,  5649,
        5821,  2645,  4000,  1928,  3086,  4230,  4616, 11500,  2708,
        2132,  3366,  8080,  3357,  3029,  2609,  4945,  5726, 10750,
        7100,  4300,  3208,  1875,  4755,  5266,  1000,  3333,  3846,
        2395,  1378,  3988,  2366,  8566,  5695,  2958,  6250,  3273,
        4133,  3620,  6782,  2484,  1977,  4188,  1759,  4288,  4843,
       13650,  4652,  3816,  3052, 11417,  7333,  3800,  2071,  5316,
        2929,  3572,  7451,  5050, 14583,  2214,  5568, 10408,  5667,
        2137,  2957,  3692, 23803,  3865, 10513,  6080, 20166,  2014,
        2718,  3459,  4895,  3316, 14999,  4200,  5042,  6950,  2698,
       11757,  2330, 14866,  1538, 10000,  4860,  6277,  2577,  9166,
```

```
       2281,  3254, 39999,  9538,  2980,  1863,  7933,  3089,  4167,
       9323,  3707,  2439,  2237,  8000,  1820, 51763,  3522,  5708,
       4344,  3497,  2045,  5516,  6400,  1916,  4600, 33846,  3625,
      39147,  2178,  2383,   674,  9328,  4885, 12000,  6033,  3858,
       4191,  3125,  8333,  1907,  3416, 11000,  4923,  3992,  3917,
       4408,  3244,  3975,  2479,  3418,  3430,  7787,  5703,  3173,
       3850,   150,  3727,  5000,  4283,  2221,  4009,  2971,  7578,
       3250,  4735,  4758,  2491,  3716,  3189,  3155,  5500,  5746,
       3463,  3812,  3315,  5819,  2510,  2965,  3406,  6050,  9703,
       6608,  2882,  1809,  1668,  3427,  2661, 16250,  3083,  6045,
       5250, 14683,  4931,  6083,  2060,  3481,  7200,  5166,  4095,
       4708,  4333,  2876,  3237, 11146,  2833,  2620,  3900,  2750,
       3993,  3103,  4100,  4053,  3927,  2301,  1811, 20667,  3158,
       3704,  4124,  9508,  3075,  4400,  3153,  4416,  6875,  4666,
       2875,  1625,  2000,  3762, 20233,  7667,  2917,  2927,  2507,
       2473,  3399,  2058,  3541,  4342,  3601,  3166, 15000,  8666,
       4917,  5818,  4384,  2935, 63337,  9833,  5503,  1830,  4160,
       2647,  2378,  4554,  2499,  3523,  6333,  2625,  9083,  8750,
       2666,  2423,  3813,  3875,  5167,  4723,  4750,  3013,  6822,
       6216,  5124,  6325, 19730, 15759,  5185,  3062,  2764,  4817,
       4310,  3069,  5391,  5941,  7167,  4566,  2346,  3010,  5488,
       9167,  9504,  1993,  3100,  3276,  3180,  3033,  3902,  1500,
       2889,  2755,  1963,  7441,  4547,  2167,  2213,  8300, 81000,
       3867,  6256,  6096,  2253,  2149,  2995,  1600,  1025,  3246,
       5829,  2720,  7250, 14880,  4606,  5935,  2920,  2717,  8624,
       6500, 12876,  2425, 10047,  1926, 10416,  7142,  3660,  7901,
       4707, 37719,  3466,  3539,  3340,  2769,  2309,  1958,  3948,
       2483,  7085,  3859,  4301,  3708,  4354,  8334,  2083,  7740,
       3015,  5191,  2947, 16692,   210,  3450,  2653,  4691,  5532,
      16525,  6700,  2873, 16667,  4350,  3095, 10833,  3547, 18333,
       2435,  2699,  5333,  3691, 17263,  3597,  3326,  4625,  2895,
       6283,   645,  3159,  4865,  4050,  3814, 20833,  3583, 13262,
       3598,  6065,  3283,  2130,  5815,  2031,  3074,  4683,  3400,
       2192,  5677,  7948,  4680, 17500,  3775,  5285,  2679,  6783,
       4281,  3588, 11250, 18165,  2550,  6133,  3617,  6417,  4608,
       2138,  3652,  2239,  3017,  2768,  3358,  2526,  2785,  6633,
       2492,  2454,  3593,  5468,  2667, 10139,  3887,  4180,  3675,
      19484,  5923,  5800,  8799,  4467,  3417,  5116, 16666,  6125,
       6406,  3087,  3229,  1782,  3182,  6540,  1836,  1880,  2787,
       2297,  2165,  2726,  9357, 16120,  3833,  6383,  2987,  9963,
       5780,   416,  2894,  3676,  3987,  3232,  2900,  4106,  8072,
       7583])
```

```
print(df.groupby('Gender').ApplicantIncome.mean())
```

```
Gender
Female    4643.473214
Male      5446.460123
Name: ApplicantIncome, dtype: float64
```

```
groups=pd.cut(df['ApplicantIncome'],bins=4)
groups
```

```
0      (69.15, 20362.5]
1      (69.15, 20362.5]
2      (69.15, 20362.5]
3      (69.15, 20362.5]
4      (69.15, 20362.5]
              ...
609    (69.15, 20362.5]
610    (69.15, 20362.5]
611    (69.15, 20362.5]
612    (69.15, 20362.5]
613    (69.15, 20362.5]
Name: ApplicantIncome, Length: 614, dtype: category
Categories (4, interval[float64, right]): [(69.15, 20362.5] <
(20362.5, 40575.0] < (40575.0, 60787.5] < (60787.5, 81000.0]]
```

```
df.groupby(groups)['ApplicantIncome'].count()
```

```
C:\Users\userm\AppData\Local\Temp\ipykernel_18740\3887112018.py:1:
FutureWarning: The default of observed=False is deprecated and will be
changed to True in a future version of pandas. Pass observed=False to
retain current behavior or observed=True to adopt the future default
and silence this warning.
  df.groupby(groups)['ApplicantIncome'].count()
```

```
ApplicantIncome
(69.15, 20362.5]      604
(20362.5, 40575.0]      7
(40575.0, 60787.5]      1
(60787.5, 81000.0]      2
Name: ApplicantIncome, dtype: int64
```

```
pd.crosstab(groups,df['Gender'])
```

```
Gender              Female  Male
ApplicantIncome
(69.15, 20362.5]       112   481
(20362.5, 40575.0]       0     6
(60787.5, 81000.0]       0     2
```