

Descriptive Analysis

The dataset is already download in .csv format

IMPORTING THE PACKAGE

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

import warnings
warnings.filterwarnings('ignore')
```

Load the dataset

```
In [2]: df=pd.read_csv("C:\loan_prediction.csv")
```

```
In [3]: df
```

O...	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Lo
0	LP001002	Male	No	0	Graduate	No	5849	0.0	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	
...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	

614 rows × 13 columns



Perform descriptive statistics on the dataset

```
In [4]: pwd #path where the dataset is stored
```

```
Out[4]: 'C:\\Users\\kothai\\Desktop'
```

```
In [5]: df.head()
```

O...	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Loan
0	LP001002	Male	No	0	Graduate	No	5849	0.0	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Loan
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	

In [6]:
df.tail()

O...	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Lo
609	LP002978	Female	No	0	Graduate	No	2900	0.0	
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	

In [7]:
df.Loan_Status

```
Out[7]:0      Y
      1      N
      2      Y
      3      Y
      4      Y
      ..
     609     Y
     610     Y
     611     Y
     612     Y
     613     N
      Name: Loan_Status, Length: 614, dtype: object
```

In [8]:
df.shape

Out[8]:(614, 13)

In [9]:
checking null values
df.isnull().sum()

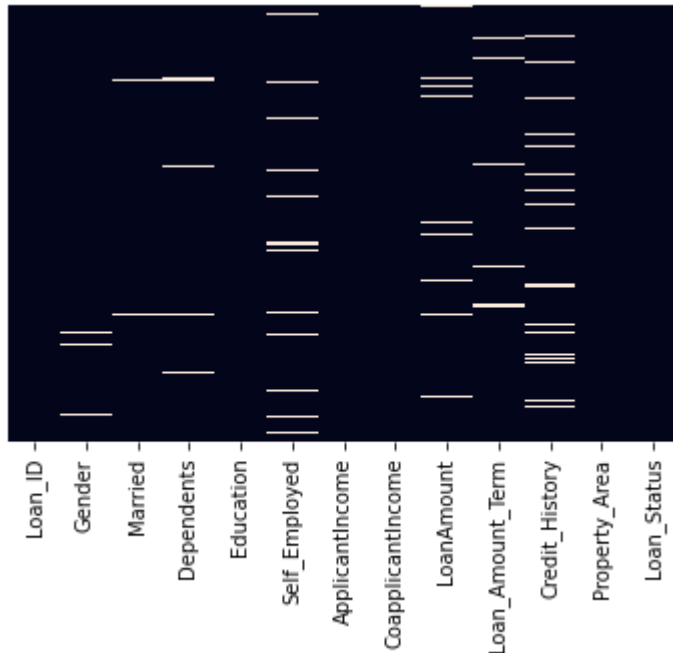
```
Out[9]:Loan_ID      0
      Gender      13
      Married      3
      Dependents    15
      Education      0
      Self_Employed  32
      ApplicantIncome  0
      CoapplicantIncome  0
      LoanAmount     22
      Loan_Amount_Term  14
      Credit_History  50
      Property_Area      0
```

```
Loan_Status      0
dtype: int64
```

```
In [10]: import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(df.isnull(),yticklabels=False,cbar=False)
```

Out[10]:



```
In [11]: df.isnull().any()
```

```
Out[11]: Loan_ID      False
Gender             True
Married            True
Dependents          True
Education           False
Self_Employed       True
ApplicantIncome     False
CoapplicantIncome   False
LoanAmount          True
Loan_Amount_Term    True
Credit_History      True
Property_Area       False
Loan_Status         False
dtype: bool
```

```
In [12]: numerical_features = df.select_dtypes(include = [np.number]).columns
categorical_features = df.select_dtypes(include = [np.object]).columns
```

```
In [13]: numerical_features
```

```
Out[13]: Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
               'Loan_Amount_Term', 'Credit_History'],
              dtype='object')
```

```
In [14]: categorical_features
```

```
Out[14]:Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
              'Self_Employed', 'Property_Area', 'Loan_Status'],
              dtype='object')
```

```
In [15]: df.info()
```

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

```
In [16]: df.describe()
```

```
Out[16]:
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.000000	564.000000
mean	5403.459283	1621.245798	146.412162	342.000000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.000000	0.000000
25%	2877.500000	0.000000	100.000000	360.000000	1.000000
50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

```
In [17]: df.describe().T
```

```
Out[17]:
```

	count	mean	std	min	25%	50%	75%	max
ApplicantIncome	614.0	5403.459283	6109.041673	150.0	2877.5	3812.5	5795.00	81000.0
CoapplicantIncome	614.0	1621.245798	2926.248369	0.0	0.0	1188.5	2297.25	41667.0
LoanAmount	592.0	146.412162	85.587325	9.0	100.0	128.0	168.00	700.0
Loan_Amount_Term	600.0	342.000000	65.120410	12.0	360.0	360.0	360.00	480.0
Credit_History	564.0	0.842199	0.364878	0.0	1.0	1.0	1.00	1.0

```
In [18]: df.columns
```

```
Out[18]:Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
              'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
              'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
              dtype='object')
```

```
In [19]: df.dtypes
```

```
Out[19]:Loan_ID      object
Gender      object
Married     object
Dependents  object
Education   object
Self_Employed object
ApplicantIncome    int64
CoapplicantIncome  float64
LoanAmount         float64
Loan_Amount_Term   float64
Credit_History     float64
Property_Area      object
Loan_Status        object
dtype: object
```

```
In [20]: df.var()
```

```
Out[20]:ApplicantIncome    3.732039e+07
CoapplicantIncome         8.562930e+06
LoanAmount                7.325190e+03
Loan_Amount_Term          4.240668e+03
Credit_History            1.331362e-01
dtype: float64
```

```
In [21]: df.skew()
```

```
Out[21]:ApplicantIncome    6.539513
CoapplicantIncome         7.491531
LoanAmount                2.677552
Loan_Amount_Term          -2.362414
Credit_History            -1.882361
dtype: float64
```

```
In [22]: df.corr()
```

```
Out[2...
      ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History
ApplicantIncome      1.000000      -0.116605      0.570909      -0.045306      -0.014715
CoapplicantIncome    -0.116605      1.000000      0.188619      -0.059878      -0.002056
LoanAmount           0.570909      0.188619      1.000000      0.039447      -0.008433
Loan_Amount_Term     -0.045306     -0.059878      0.039447      1.000000      0.001470
Credit_History       -0.014715     -0.002056     -0.008433      0.001470      1.000000
```

```
In [23]: df.std()
```

```
Out[23]:ApplicantIncome    6109.041673
CoapplicantIncome         2926.248369
```

```

LoanAmount      85.587325
Loan_Amount_Term 65.120410
Credit_History   0.364878
dtype: float64

```

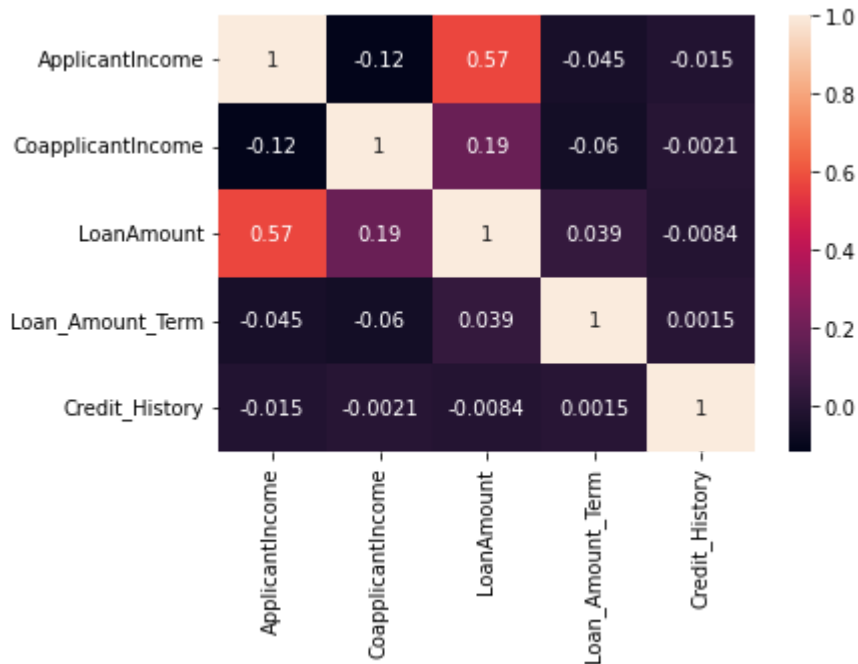
```

In [24]: import seaborn as sns
import matplotlib.pyplot as plt

df.corr()
#plotting the correlation
plt.figure(1)
sns.heatmap(df.corr(), annot = True)

```

Out[24]:



```

In [25]: data=df
data = np.random.randint(low=1,high=100,size=(10,10))
print(data)

```

```

[[56 76 33 16 14 82 68 44 41 18]
 [15 88 39 49 32 78 43 46 63 17]
 [75 92 52 21 13 95 96 27 99 91]
 [46 14 53 68 20  4  5 10 21 59]
 [15 68 78 74 19 30 68  9 50 95]
 [98 61 25 98 76 56 47 12 45 81]
 [14 94 61 69 57 67 21 35 83 50]
 [55 97 94 91 40 98 96  5 92 28]
 [66 36 31 78 63 22 12 61 36 30]
 [32 92 38  7 34  5 44 32 14 38]]

```

In []: