```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
     Mounted at /content/drive
```

```
# Code to read csv file into Colaboratory:
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
# Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

```
link = 'https://drive.google.com/file/d/1g1KW6UgHCbDFJ_9X0a2O2qnQeDoIUUlH/view?usp=sharing'
```

```
fluff, id = link.split('=')
print (id) # Verify that you have everything after '='
```

```
     sharing
```

```
downloaded = drive.CreateFile({'id':id})
#downloaded.GetContentFile('/content/drive/My Drive/train_u6lujuX_CVtuZ9i.csv')
#data=pd.read_csv('/content/drive/My Drive/train_u6lujuX_CVtuZ9i.csv')
```

```
#downloaded.GetContentFile('/content/drive/My Drive/train.csv')
data=pd.read_csv('/content/drive/My Drive/train.csv')
```

```
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
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

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
```

```
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
data.shape
```

```
(614, 13)
```

```
data.describe()
```

	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

```
data.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
```

```
data['LoanAmount']=data['LoanAmount'].fillna(data['LoanAmount'].mean())
data['Credit_History']=data['Credit_History'].fillna(data['Credit_History'].median())
```

```
data.dropna(inplace=True)
```

```
data.shape
```

```
(542, 13)
```

```
data['Gender'].value_counts()
```

```
Male      444
Female     98
Name: Gender, dtype: int64
```

```
data['Married'].value_counts()
```

```
Yes    355
No     187
Name: Married, dtype: int64
```

```
data['Education'].value_counts()
```

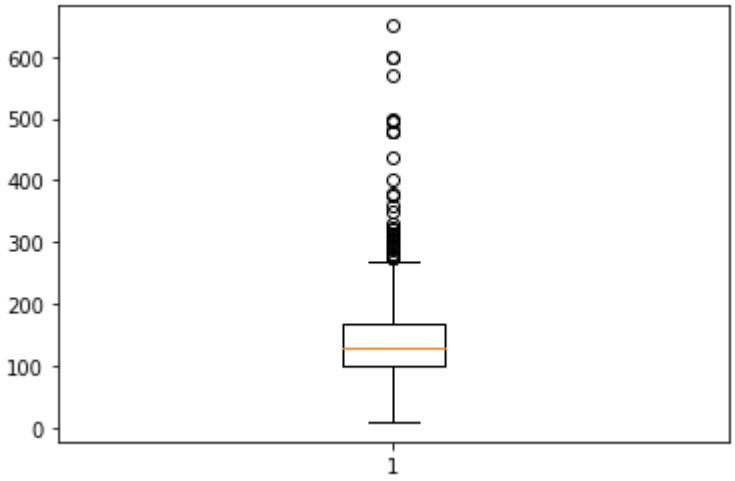
```
Graduate        425
Not Graduate    117
Name: Education, dtype: int64
```

```
data['Property_Area'].value_counts()
```

```
Semiurban    209
Urban        174
Rural        159
Name: Property_Area, dtype: int64
```

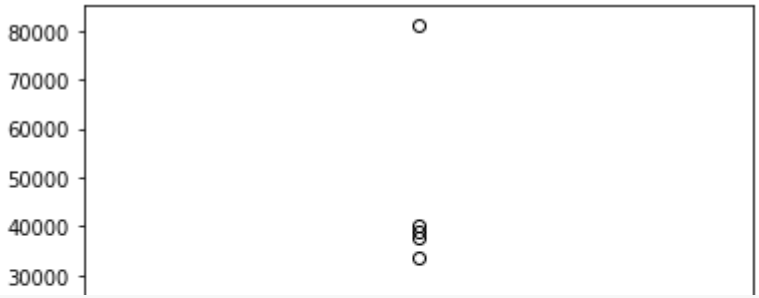
```
plt.boxplot(data['LoanAmount'])
```

```
{'boxes': [<matplotlib.lines.Line2D at 0x7fceee177d10>],
 'caps': [<matplotlib.lines.Line2D at 0x7fceee104dd0>,
  <matplotlib.lines.Line2D at 0x7fceee10e350>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fceee10ee10>],
 'means': [],
 'medians': [<matplotlib.lines.Line2D at 0x7fceee10e8d0>],
 'whiskers': [<matplotlib.lines.Line2D at 0x7fceee104350>,
  <matplotlib.lines.Line2D at 0x7fceee104890>]}
```




```
plt.boxplot(data['ApplicantIncome'])
```

```
{'boxes': [<matplotlib.lines.Line2D at 0x7fceedcab0d0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fceedc31190>,
  <matplotlib.lines.Line2D at 0x7fceedc316d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fceeed4add0>],
 'means': [],
 'medians': [<matplotlib.lines.Line2D at 0x7fceedc31c50>],
 'whiskers': [<matplotlib.lines.Line2D at 0x7fceedcab6d0>,
  <matplotlib.lines.Line2D at 0x7fceedcabc10>]}
```

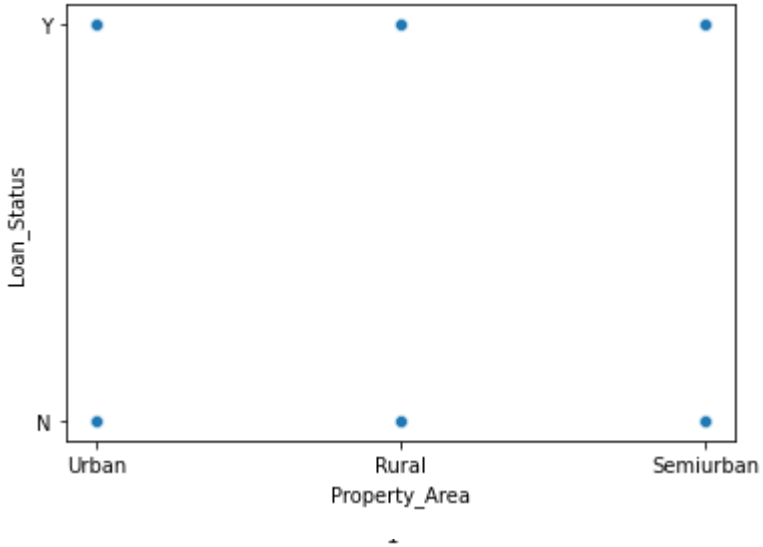



```
plt.boxplot(data['Loan_Amount_Term'])
```

```
{'boxes': [<matplotlib.lines.Line2D at 0x7fceedc29450>],
```

```
sns.scatterplot(x='Property_Area',y='Loan_Status',data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fceedc1df50>
```

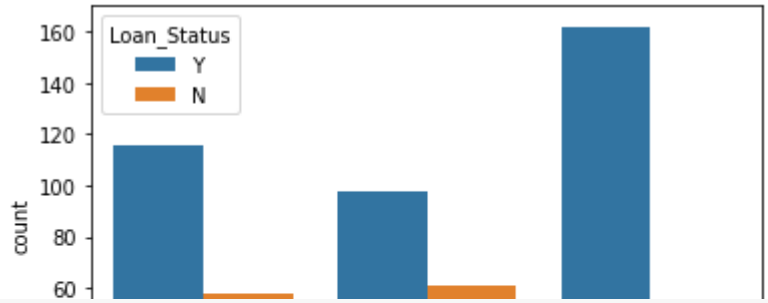



```
print(pd.crosstab(data['Property_Area'],data['Loan_Status']))
```

```
Loan_Status     N    Y
Property_Area
Rural          61   98
Semiurban      47  162
Urban          58  116
```

```
sns.countplot(data['Property_Area'],hue=data['Loan_Status'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following var
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fceedb41390>
```




```
print(pd.crosstab(data['Gender'],data['Loan_Status']))
```

```
Loan_Status    N    Y
Gender
Female        33   65
Male         133  311
```

```
sns.countplot(data['Gender'],hue=data['Loan_Status'])
```

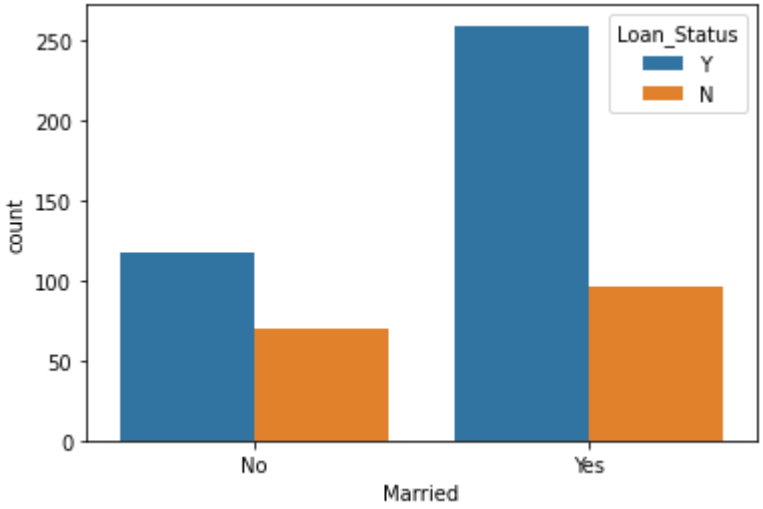
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following var
```

```
print(pd.crosstab(data['Married'],data['Loan_Status']))
```

```
Loan_Status    N    Y
Married
No            70  117
Yes           96  259
```

```
sns.countplot(data['Married'],hue=data['Loan_Status'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following var
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fceeda4cb50>
```

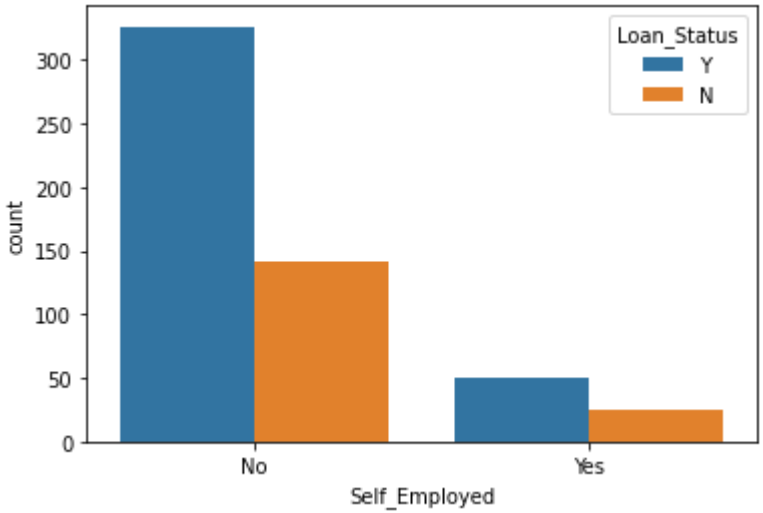



```
print(pd.crosstab(data['Self_Employed'],data['Loan_Status']))
```

```
Loan_Status      N    Y
Self_Employed
No             141  326
Yes             25   50
```

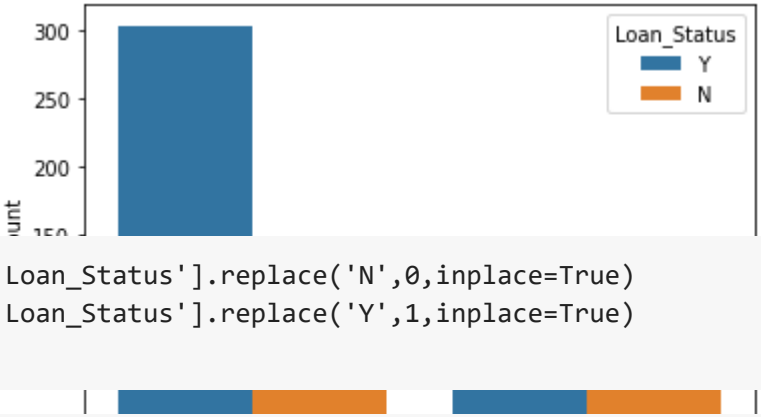
```
sns.countplot(data['Self_Employed'],hue=data['Loan_Status'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following var
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fceed9c7ed0>
```




```
print(pd.crosstab(data['Education'],data['Loan_Status']))


# In[34]:


sns.countplot(data['Education'],hue=data['Loan_Status'])
```

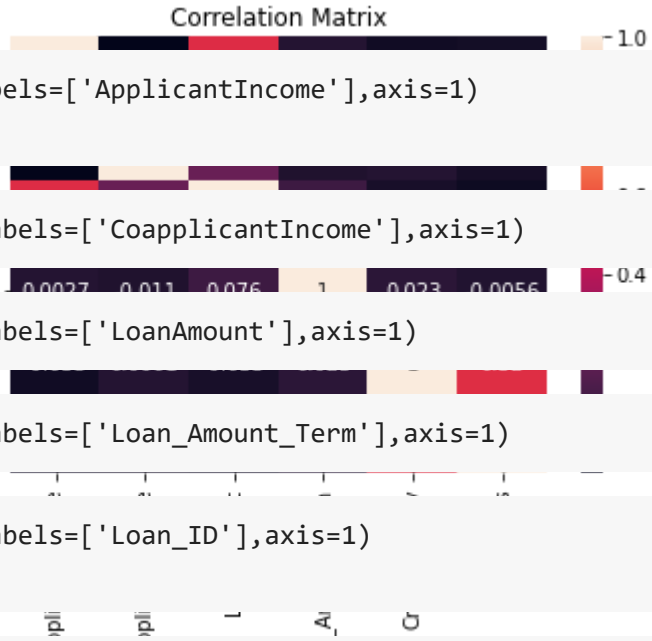
```
Loan_Status     N    Y
Education
Graduate      122  303
Not Graduate   44   73
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keywo
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fceed930d10>
```




```
data['Loan_Status'].replace('N',0,inplace=True)
data['Loan_Status'].replace('Y',1,inplace=True)
```

```
plt.title('Correlation Matrix')
sns.heatmap(data.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fceed933a90>
```




```
data2=data.drop(labels=['ApplicantIncome'],axis=1)
```

```
data2=data2.drop(labels=['CoapplicantIncome'],axis=1)
```

```
data2=data2.drop(labels=['LoanAmount'],axis=1)
```

```
data2=data2.drop(labels=['Loan_Amount_Term'],axis=1)
```

```
data2=data2.drop(labels=['Loan_ID'],axis=1)
```

```
data2.head()
```

	Gender	Married	Dependents	Education	Self_Employed	Credit_History	Property_Area	Loan_Status
0	Male	No	0	Graduate	No	1.0	Urban	1
1	Male	Yes	1	Graduate	No	1.0	Rural	0
2	Male	Yes	0	Graduate	Yes	1.0	Urban	1
3	Male	Yes	0	Not Graduate	No	1.0	Urban	1
4	Male	No	0	Graduate	No	1.0	Urban	1

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
le=LabelEncoder()
ohe=OneHotEncoder()
```

```
data2['Property_Area']=le.fit_transform(data2['Property_Area'])
```

```
data2['Dependents']=le.fit_transform(data2['Dependents'])
```

```
data2=pd.get_dummies(data2)
```

```
data2.dtypes
```

```
Dependents                 int64
Credit_History           float64
Property_Area              int64
Loan_Status                int64
Gender_Female              uint8
Gender_Male                uint8
Married_No                 uint8
Married_Yes                uint8
Education_Graduate         uint8
Education_Not Graduate     uint8
Self_Employed_No           uint8
Self_Employed_Yes          uint8
dtype: object
```

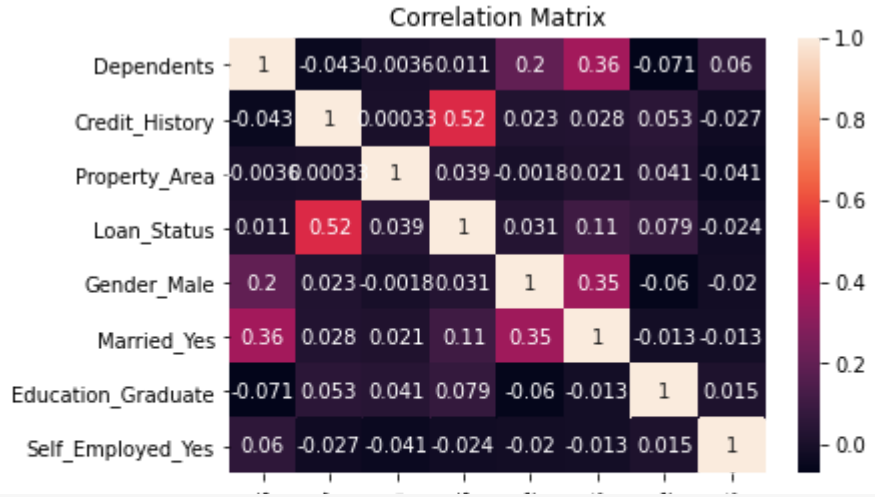
```
data2=data2.drop(labels=['Gender_Female'],axis=1)


# In[55]:


data2=data2.drop(labels=['Married_No'],axis=1)


# In[56]:
```

```
data2=data2.drop(labels=['Education_Not Graduate'],axis=1)


# In[58]:


data2=data2.drop(labels=['Self_Employed_No'],axis=1)


# In[60]:


data2.head()
```

	Dependents	Credit_History	Property_Area	Loan_Status	Gender_Male	Married_Yes	Education_Graduate
0	0	1.0	2	1	1	0	1
1	1	1.0	0	0	1	1	1
2	0	1.0	2	1	1	1	1
3	0	1.0	2	1	1	1	0
4	0	1.0	2	1	1	0	1

```
plt.title('Correlation Matrix')
sns.heatmap(data2.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcee3ca8bd0>
```




```
data2=data2.drop('Self_Employed_Yes',1)


# In[65]:


data2=data2.drop('Dependents',1)


# In[67]:


data2=data2.drop('Education_Graduate',1)


# In[68]:


X=data2.drop('Loan_Status',1)
```

```
Y=data2['Loan_Status']
```

```
from sklearn.model_selection import train_test_split


# In[72]:


x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=6)


# In[74]:


print('X_train is: ',x_train.shape)
print('X_test is: ',x_test.shape)
print('Y_train is: ',y_train.shape)
print('y_test is: ',y_test.shape)
```

```
X_train is:  (433, 4)
X_test is:  (109, 4)
Y_train is:  (433,)
y_test is:  (109,)
```

```
from sklearn.linear_model import LogisticRegression
log=LogisticRegression()


# In[77]:


log.fit(x_train,y_train)


# In[78]:


log.score(x_train,y_train)


# In[79]
```

```
# In[79]:


pred=log.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,pred)
```

```
     0.8440366972477065
```

```
from sklearn import metrics


# In[84]:


# CONFUSION Matrix
metrics.confusion_matrix(y_test,pred)
```

```
     array([[23, 16],
            [ 1, 69]])
```

```
metrics.f1_score(y_test,pred)
```

```
     0.8903225806451613
```

```
metrics.precision_score(y_test,pred)
```

```
     0.8117647058823529
```

```
metrics.recall_score(y_test,pred)
```

```
0.9857142857142858
```

```
data={'y_test':y_test,'pred':pred}
pd.DataFrame(data=data)
```

	y_test	pred
572	1	1
133	1	1
371	1	1
487	0	1
277	1	1
...	...	...
457	0	1
310	1	1
553	0	0
186	0	0
340	0	1

109 rows × 2 columns

```
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier()


# In[91]:


clf.fit(x_train,y_train)
```

```
# In[92]:


pred1=clf.predict(x_test)


# In[93]:



```

```
accuracy_score(y_test,pred1)
```

```
     0.8440366972477065
```

```
metrics.confusion_matrix(y_test,pred1)
```

```
     array([[23, 16],
            [ 1, 69]])
```

```
metrics.f1_score(y_test,pred1)
```

```
     0.8903225806451613
```

```
metrics.recall_score(y_test,pred1)
```

```
     0.9857142857142858
```

```
metrics.precision_score(y_test,pred1)
```

```
     0.8117647058823529
```