In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
```

In [2]:
```
data_set=pd.read_csv("loan_prediction.csv")
```

In [3]:
```
data_set.head()
```

Out[3]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128	360	1	Rural	N
1	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66	360	1	Urban	Y
2	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120	360	1	Urban	Y
3	LP001008	Male	No	0	Graduate	No	6000	0.0	141	360	1	Urban	Y
4	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0	267	360	1	Urban	Y

In [4]:
```
data_set.shape
```

Out[4]:
```
(480, 13)
```

In [5]:
```
data_set.describe()
```

Out[5]:

	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	480.000000	480.000000	480.000000	480.000000	480.000000	480.000000
mean	0.862500	5364.231250	1581.093583	144.735417	342.050000	0.854167
std	1.220107	5668.251251	2617.692267	80.508164	65.212401	0.353307
min	0.000000	150.000000	0.000000	9.000000	36.000000	0.000000
25%	0.000000	2898.750000	0.000000	100.000000	360.000000	1.000000
50%	0.000000	3859.000000	1164.500000	129.000000	360.000000	1.000000
75%	2.000000	5852.500000	2253.250000	170.000000	360.000000	1.000000
max	4.000000	63337.000000	33837.000000	600.000000	480.000000	1.000000

In [6]:
```
data_set.isnull().sum()
```

Out[6]:
```
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
```

In [7]:
```
data_set.replace({'Loan_Status':{'N':0,'Y':1}},inplace=True)
```

In [8]:
```
data_set.head()
```

Out[8]:

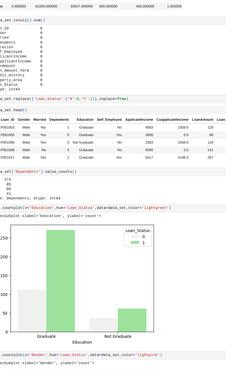
	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128	360	1	Rural	0
1	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66	360	1	Urban	1
2	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120	360	1	Urban	1
3	LP001008	Male	No	0	Graduate	No	6000	0.0	141	360	1	Urban	1
4	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0	267	360	1	Urban	1

In [9]:
```
data_set['Dependents'].value_counts()
```

Out[9]:
```
0   274
2   86
1   80
3+  41
Name: Dependents, dtype: int64
```

In [10]:
```
sns.countplot(x='Education',hue='Loan_Status',data=data_set,color='lightgreen')
```

Out[10]:
```
<AxesSubplot:xlabel="Education", ylabel="count">
```

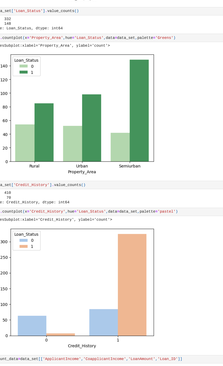


In [11]:
```
sns.countplot(x='Gender',hue='Loan_Status',data=data_set,color='lightpink')
```

Out[11]:
```
<AxesSubplot:xlabel="Gender", ylabel="count">
```

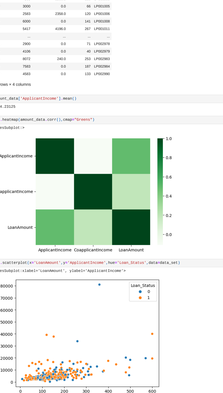
In [12]:
```
sns.countplot(x='Married',hue='Loan_Status',data=data_set,color='lightblue')
```

Out[12]:
```
<AxesSubplot:xlabel="Married", ylabel="count">
```



In [13]:
```
sns.countplot(x='Self_Employed',hue='Loan_Status',data=data_set,palette='BuPu')
```

Out[13]:
```
<AxesSubplot:xlabel="Self_Employed", ylabel="count">
```



In [14]:
```
data_set['Loan_Status'].value_counts()
```

Out[14]:
```
1   332
0   148
Name: Loan_Status, dtype: int64
```

In [15]:
```
sns.countplot(x='Property_Area',hue='Loan_Status',data=data_set,palette='Greens')
```

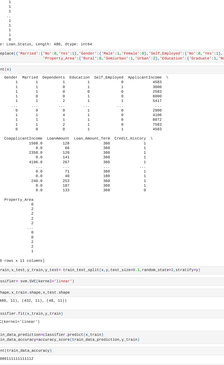
Out[15]:
```
<AxesSubplot:xlabel="Property_Area", ylabel="count">
```

In [16]:
```
data_set['Credit_History'].value_counts()
```

Out[16]:
```
1   410
0   70
Name: Credit_History, dtype: int64
```

In [17]:
```
sns.countplot(x='Credit_History',hue='Loan_Status',data=data_set,palette='pastel')
```

Out[17]:
```
<AxesSubplot:xlabel="Credit_History", ylabel="count">
```



In [18]:
```
amount_data=data_set[['ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_ID']]
```

In [19]:
```
amount_data
```

Out[19]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_ID
0	4583	1508.0	128	LP001003
1	3000	0.0	66	LP001005
2	2583	2358.0	120	LP001006
3	6000	0.0	141	LP001008
4	5417	4196.0	267	LP001011
...	...	...	...	...
475	2900	0.0	71	LP002978
476	4106	0.0	40	LP002979
477	8072	240.0	253	LP002984
478	7583	0.0	187	LP002990
479	4583	0.0	133	LP002989

480 rows × 4 columns

In [20]:
```
amount_data['ApplicantIncome'].mean()
```

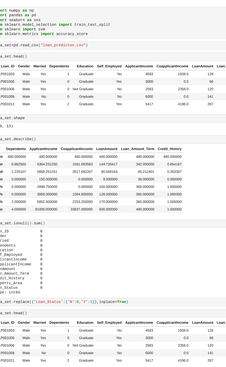
Out[20]:
```
5364.231250
```

In [21]:
```
sns.heatmap(amount_data.corr(),cmap="Greens")
```

Out[21]:
```
<AxesSubplot:>
```


In [22]:
```
sns.scatterplot(x='LoanAmount',y='ApplicantIncome',hue='Loan_Status',data=data_set)
```

Out[22]:
```
<AxesSubplot:xlabel='LoanAmount', ylabel='ApplicantIncome'>
```



In [23]:
```
sns.scatterplot(x='LoanAmount',y='CoapplicantIncome',hue='Loan_Status',data=data_set)
```

Out[23]:
```
<AxesSubplot:xlabel='LoanAmount', ylabel='CoapplicantIncome'>
```


In [24]:
```
x=data_set.drop(columns=['Loan_ID','Loan_Status'],axis=1)
y=data_set['Loan_Status']
```

In [25]:
```
print(x)
print(y)
```

```
     Gender Married  Dependents  Education Self_Employed  ApplicantIncome  \
0      Male     Yes           1   Graduate            No             4583
1      Male     Yes           0   Graduate           Yes             3000
2      Male     Yes           0  Not Graduate         No             2583
3      Male      No           0   Graduate            No             6000
4      Male     Yes           2   Graduate           Yes             5417
..      ...     ...         ...        ...           ...              ...
475  Female      No           0   Graduate            No             2900
476    Male     Yes           4   Graduate            No             4106
477    Male     Yes           1   Graduate            No             8072
478    Male     Yes           0   Graduate            No             7583
479  Female      No           0   Graduate           Yes             4583

     CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  \
0               1508.0         128               360               1
1                  0.0          66               360               1
2               2358.0         120               360               1
3                  0.0         141               360               1
4               4196.0         267               360               1
..                 ...         ...               ...             ...
475                0.0          71               360               1
476                0.0          40               180               1
477              240.0         253               360               1
478                0.0         187               360               1
479                0.0         133               360               0

    Property_Area
0           Rural
1           Urban
2           Urban
3           Urban
4           Urban
..            ...
475         Rural
476         Rural
477         Rural
478         Urban
479     Semiurban

[480 rows x 11 columns]
0      0
1      1
2      1
3      1
4      1
      ..
475    1
476    1
477    1
478    1
479    1
Name: Loan_Status, Length: 480, dtype: int64
```

In [26]:
```
x.replace({'Married':{'No':0,'Yes':1},'Gender':{'Male':1,'Female':0},'Self_Employed':{'No':0,'Yes':1},
           'Property_Area':{'Rural':0,'Semiurban':1,'Urban':2},'Education':{'Graduate':1,'Not Graduate':0}},inplace=True)
```

In [27]:
```
print(x)
```

```
     Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  \
0         1        1           1          1              0             4583
1         1        1           0          1              1             3000
2         1        1           0          0              0             2583
3         1        0           0          1              0             6000
4         1        1           2          1              1             5417
..      ...      ...         ...        ...            ...              ...
475       0        0           0          1              0             2900
476       1        1           4          1              0             4106
477       1        1           1          1              0             8072
478       1        1           0          1              0             7583
479       0        0           0          0              1             4583

     CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  \
0               1508.0         128               360               1
1                  0.0          66               360               1
2               2358.0         120               360               1
3                  0.0         141               360               1
4               4196.0         267               360               1
..                 ...         ...               ...             ...
475                0.0          71               360               1
476                0.0          40               180               1
477              240.0         253               360               1
478                0.0         187               360               1
479                0.0         133               360               0

     Property_Area
0                2
1                2
2                2
3                2
4                2
..             ...
475              0
476              0
477              0
478              2
479              1

[480 rows x 11 columns]
```

In [28]:
```
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.1,random_state=2,stratify=y)
```

In [29]:
```
classifier= svm.SVC(kernel='linear')
```

In [30]:
```
x.shape,x_train.shape,x_test.shape
```

Out[30]:
```
((480, 11), (432, 11), (48, 11))
```

In [31]:
```
classifier.fit(x_train,y_train)
```

Out[31]:
```
SVC(kernel='linear')
```

In [32]:
```
train_data_prediction=classifier.predict(x_train)
train_data_accuracy=accuracy_score(train_data_prediction,y_train)
```

In [33]:
```
print(train_data_accuracy)
```

```
0.7986111111111112
```

In [34]:
```
test_data_prediction=classifier.predict(x_test)
test_data_accuracy=accuracy_score(test_data_prediction,y_test)
```

In [35]:
```
print(test_data_accuracy)
```

```
0.8333333333333334
```

In [ ]:
```

```