In [ ]:
```
#Lut Lat Aung,  6511163,  542
```

In [108]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import uniform
from statsmodels.formula.api import ols

from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
```

In [117]:
```
# Q1 (1.1)
train_1 = pd.read_csv("train_1.csv")
train_1.head()

train_2 = pd.read_csv("train_2.csv")
train_2.head()



train = pd.concat([train_1, train_2], sort = True)
train.shape
```

Out[117]:
```
(614, 13)
```

In [110]:
```
# Q1 (1.2)

print(train.dtypes)
print("\nThe number of missing values are -\n")

print(train.isna().sum())
```

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

The number of missing values are -

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

In [111]:
```
# Q1 (1.3)

threshold = len(train) * 0.05

train_drop = train.columns[train.isna().sum() <= threshold]
train.dropna(subset = train_drop, inplace = True)
train.isna().sum()
```

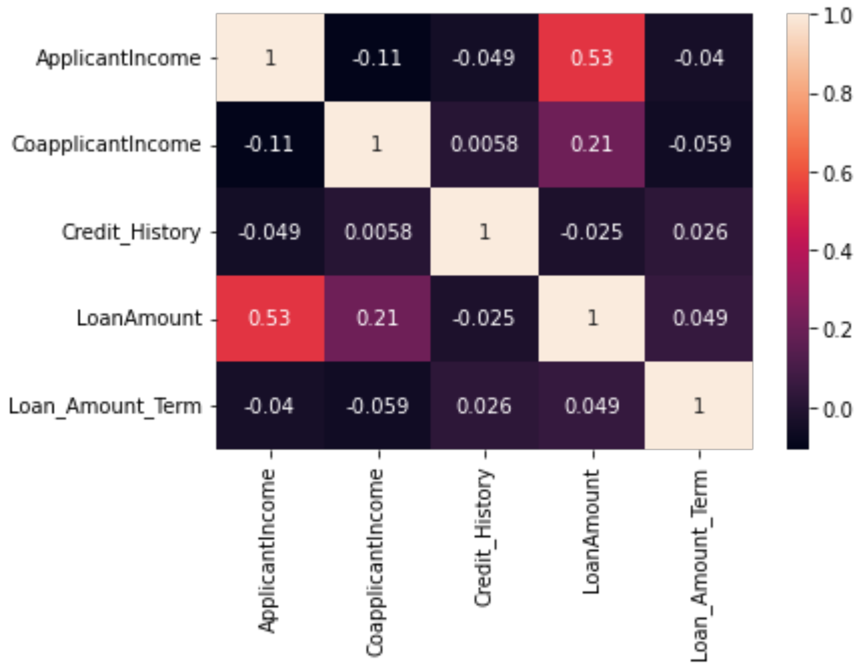
Out[111]:
```
ApplicantIncome       0
CoapplicantIncome     0
Credit_History       48
Dependents            0
Education             0
Gender                0
LoanAmount            0
Loan_Amount_Term      0
Loan_ID               0
Loan_Status           0
Married               0
Property_Area         0
Self_Employed        30
dtype: int64
```

In [112]:
```
# Q1 (1.4)


train_cols_with_missing_values = train.columns[train.isna().sum() > 0]
print(train_cols_with_missing_values)

for col in train_cols_with_missing_values[:-1]:

    train[col] = train[col].fillna(train[col].max())


train.isna().sum()
```

```
Index(['Credit_History', 'Self_Employed'], dtype='object')
```

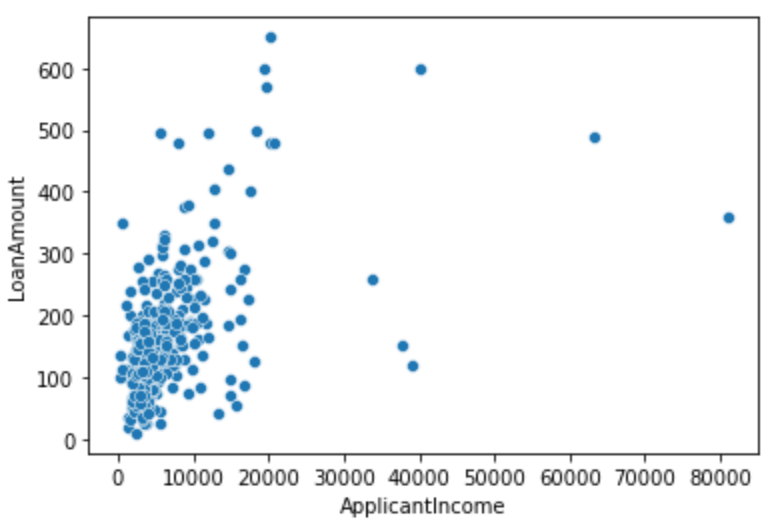
Out[112]:
```
ApplicantIncome       0
CoapplicantIncome     0
Credit_History        0
Dependents            0
Education             0
Gender                0
LoanAmount            0
Loan_Amount_Term      0
Loan_ID               0
Loan_Status           0
Married               0
Property_Area         0
Self_Employed        30
dtype: int64
```

In [113]:
```
# Q1 (1.5)

sns.heatmap(train.corr(), annot=True)
plt.show()

#The correlation between ApplicantIncome and Loan Amount is 0.53.
# There is a weak correlation.
```

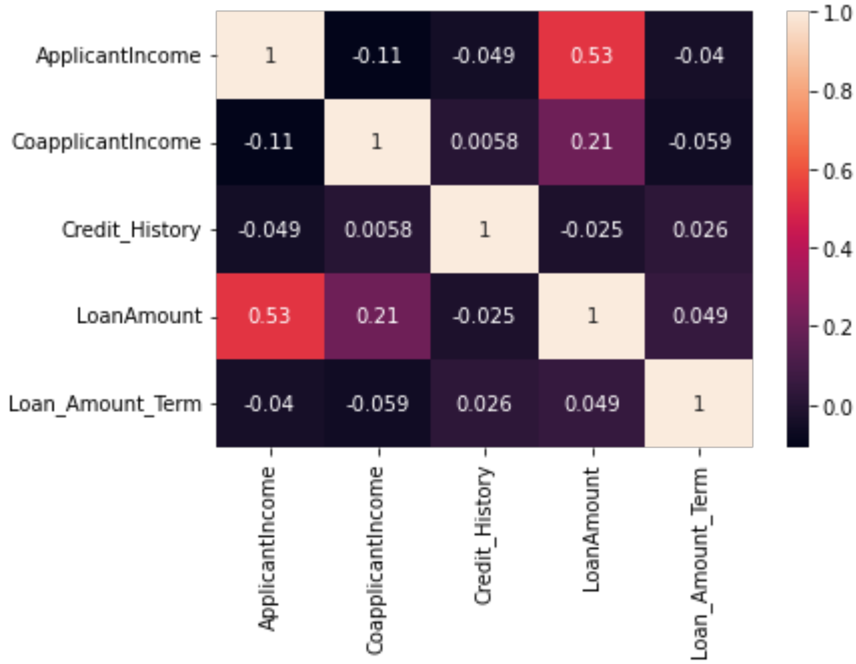


In [114]:
```
# Q1 (1.6)

# This is group by Education and Property_Area
sns.scatterplot(data=train, x="ApplicantIncome",y= "LoanAmount")
```

Out[114]:
```
<AxesSubplot:xlabel='ApplicantIncome', ylabel='LoanAmount'>
```



In [115]:
```
# Q1 (1.7)

sns.heatmap(train.corr(), annot =True)
plt.show()
```



Q1 (1.7)

The correlation between CoapplicantIncome and LoadAmount is 0.21
It is very normal correlation considering the whole heatmap values are not that high.

The correlation between Credit_History and ApplicationIncome is -0.049
It is very weak correlation. It indicate the small relation between them.

In [ ]:
```
#Lut Lat Aung,  6511163,  542
```