Name:-Tejaswini Sunil Mahale

Practical:-10

Roll no:-26 Sub:-DV

Case Study of Loan Prediction

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Loading and Summarizing Data

```
train_data = pd.read_csv("train.csv")
train_data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Ter
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.

```
train_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

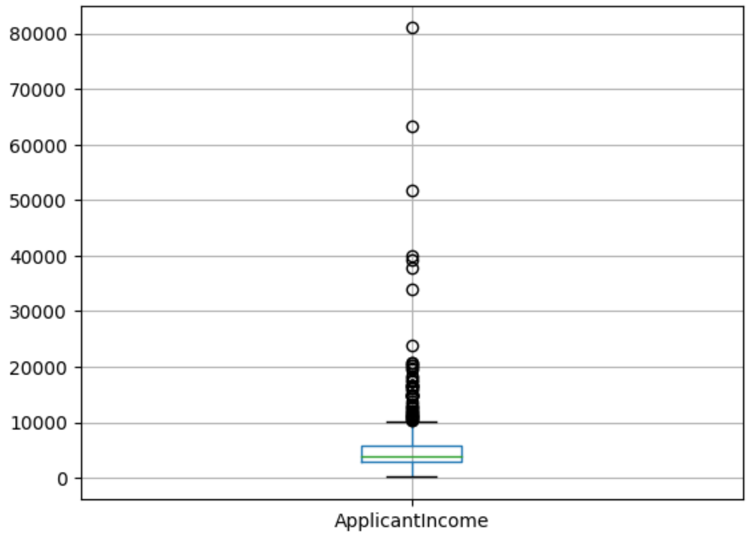
Distribution Analysis

```
train_data['ApplicantIncome'].hist(bins=70,grid=False)
```

<Axes: >

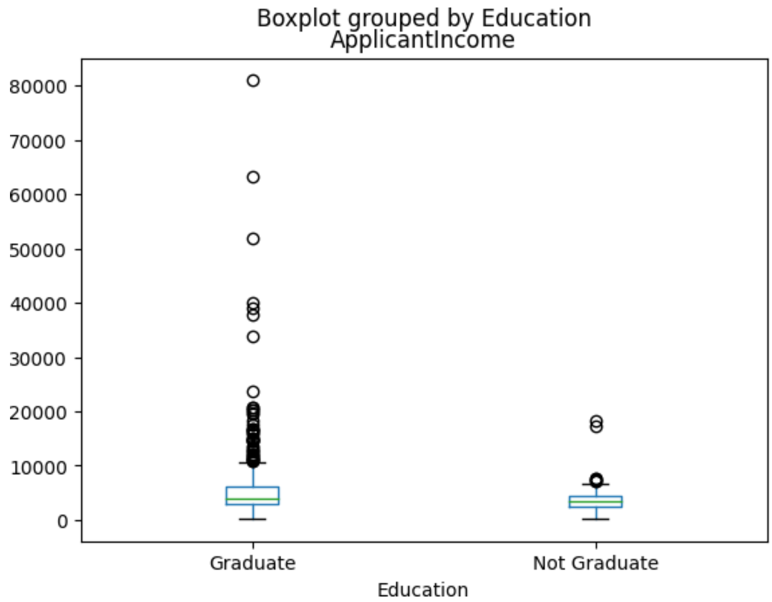
train_data.boxplot(column = 'ApplicantIncome')

<Axes: >



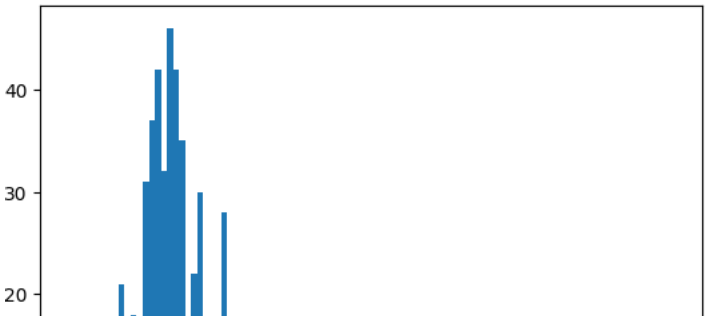
train_data.boxplot(column = 'ApplicantIncome', grid =False, by = 'Education')

<Axes: title={'center': 'ApplicantIncome'}, xlabel='Education'>



train_data['LoanAmount'].hist(bins=100,grid = False)

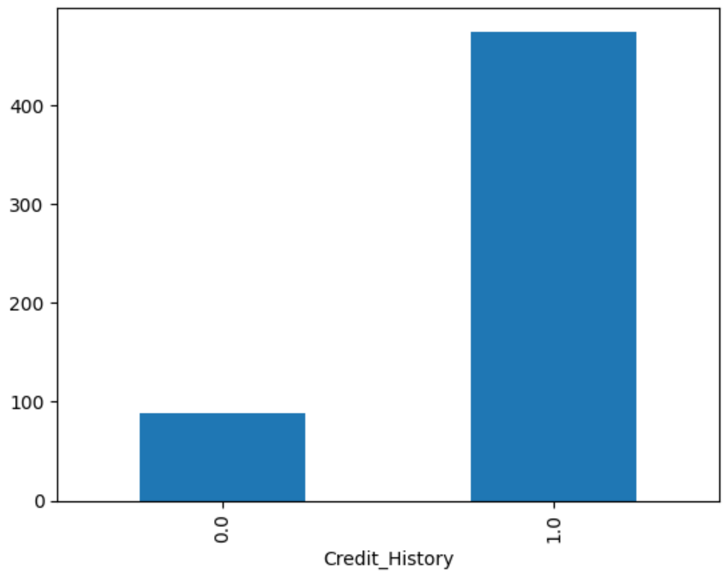
<Axes: >



Categorical Value Analysis

```
temp = train_data['Credit_History'].value_counts(ascending = True)
temp.plot(kind = 'bar')
```

<Axes: xlabel='Credit_History'>



Data Munging

```
train_data.apply(lambda x: sum(x.isnull()),axis=0)
```

```
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
```

```
train_data['LoanAmount'].fillna(train_data['LoanAmount'].mean(),inplace=True)
```

```
train_data['Self_Employed'].fillna('No',inplace=True)
```

```
train_data['Gender'].fillna(train_data['Gender'].mode()[0], inplace=True)
train_data['Married'].fillna(train_data['Married'].mode()[0], inplace=True)
train_data['Dependents'].fillna(train_data['Dependents'].mode()[0],  inplace=True)
```

```
train_data['Loan_Amount_Term'].fillna(train_data['Loan_Amount_Term'].mode()[0], inplace=True)
train_data['Credit_History'].fillna(train_data['Credit_History'].mode()[0], inplace=True)
```

```
train_data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Ter
0	LP001002	Male	No	0	Graduate	No	5849	0.0	146.412162	360.
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.000000	360.
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.000000	360.
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.000000	360.

```
train_data.apply(lambda x: sum(x.isnull()),axis=0)
```

```
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
```

```
from sklearn.preprocessing import LabelEncoder
var_mod = ['Gender','Married','Dependents','Education','Self_Employed','Property_Area','Loan_Status']
le = LabelEncoder()
for i in var_mod:
    train_data[i] = le.fit_transform(train_data[i])
train_data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Ter
0	LP001002	1	0	0	0	0	5849	0.0	146.412162	360.
1	LP001003	1	1	1	0	0	4583	1508.0	128.000000	360.
2	LP001005	1	1	0	0	1	3000	0.0	66.000000	360.
3	LP001006	1	1	0	1	0	2583	2358.0	120.000000	360.
4	LP001008	1	0	0	0	0	6000	0.0	141.000000	360.

Training Model

```
X = train_data[['Credit_History','Gender','Married','Education']]
y = train_data['Loan_Status']
```

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X,y)
predictions = model.predict(X)
```

```
from sklearn.metrics import accuracy_score
print(accuracy_score(predictions,y))
```

```
0.8094462540716613
```