EDS PROJECT

GROUP MEMBERS:

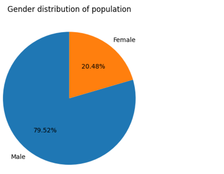
401. ALVIN A

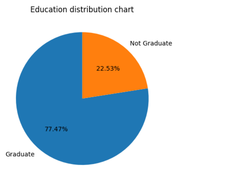
404. ARHANT N,

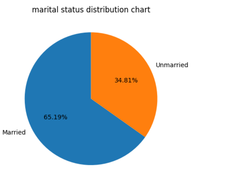
407. ARPAN B,

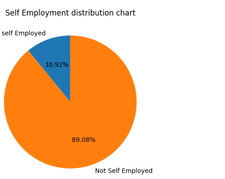
414. ANKIT K

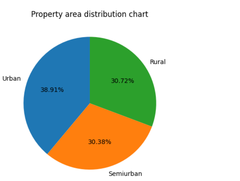
In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt

#Reading csv file
df = pd.read_csv('Loan_dataset.csv')

#----- cleaning Data -----#

# Dropping such rows where the value for the given column is either na or blank
df.dropna(subset=['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'Credit_History', 'Property_Area', 'Loan_Amount_Term'], inplace=True)

#calculating mean to replace NaN or missing values
applicant_income_mean = df['ApplicantIncome'].mean()
co_applicant_income_mean = df['CoapplicantIncome'].mean()
loan_amt_mean = df['LoanAmount'].mean()

# replacing Nan or missing values with their mean of theie respective column
df.fillna({'ApplicantIncome': applicant_income_mean, 'CoapplicantIncome': co_applicant_income_mean, 'LoanAmount': loan_amt_mean}, inplace=True)

#----- population data charts -----#

#number of male v/s number of female pi-chart
labels = df['Gender'].unique()
values = [df[df['Gender'] == 'Male'].shape[0], df[df['Gender'] == 'Female'].shape[0]]
plt.pie(values, labels=labels, startangle= 90, autopct='%1.2f%%')
plt.title('Gender distribution of population')
plt.show()

#number of grduated v/s number of non graduates pi chart
labels = df['Education'].unique()
values = [df[df['Education'] == 'Graduate'].shape[0], df[df['Education'] == 'Not Graduate'].shape[0]]
plt.pie(values, labels=labels, startangle=90, autopct='%1.2f%%')
plt.title('Education distribution chart')
plt.show()

#married v/s unmarried pi chart
labels = ['Married', 'Unmarried']
values = [df[df['Married'] == 'Yes'].shape[0], df[df['Married'] == 'No'].shape[0]]
plt.pie(values, labels=labels, startangle=90, autopct='%1.2f%%')
plt.title('marital status distribution chart')
plt.show()

#self-employe v/s not self-employed pi chart
labels = ['self Employed', 'Not Self Employed']
values = [df[df['Self_Employed'] == 'Yes'].shape[0], df[df['Self_Employed'] == 'No'].shape[0]]
plt.pie(values, labels=labels, startangle=90, autopct='%1.2f%%')
plt.title('Self Employment distribution chart')
plt.show()

#property area pi chart
labels = df['Property_Area'].unique()
values = [df[df['Property_Area'] == 'Urban'].shape[0], df[df['Property_Area'] == 'Semiurban'].shape[0], df[df['Property_Area'] == 'Rural'].shape[0]]
plt.pie(values, labels=labels, startangle=90, autopct='%1.2f%%')
plt.title('Property area distribution chart')
plt.show()


#----- plotting graphs acceptance rate-----#

#Gender v/s acceptance plot
labels = df['Gender'].unique()
values = [df[df['Gender'] == 'Male') & (df['Credit_History'] == 1)].shape[0] / df[(df['Gender'] == 'Male')].shape[0] * 100,
          df[df['Gender'] == 'Female') & (df['Credit_History'] == 1)].shape[0] / df[(df['Gender'] == 'Female')].shape[0] * 100]
print(values)
plt.bar(labels, values)
plt.show()

#Marital status v/s acceptance rate

labels = ['Married', 'Not married']
values = [df[df['Married'] == 'Yes') & (df['Credit_History'] == 1)].shape[0] / df[df['Married'] == 'Yes'].shape[0] * 100,
          df[df['Married'] == 'No') & (df['Credit_History'] == 1)].shape[0] / df[df['Married'] == 'No'].shape[0] * 100 ]
print(values)
plt.bar(labels, values)
plt.show()

#Education v/s acceptance rate
labels = ['Graduated', 'Not Graduated']
values = [df[df['Education'] == 'Graduate') & (df['Credit_History'] == 1)].shape[0] / df[df['Education'] == 'Graduate'].shape[0] * 100,
          df[df['Education'] == 'Not Graduate') & (df['Credit_History'] == 1)].shape[0] / df[df['Education'] == 'Not Graduate'].shape[0] * 100 ]
print(values)
plt.bar(labels, values)
plt.show()

#property area v/s acceptance rate

labels = (df['Property_Area'].unique())
values = [df[df['Property_Area'] == 'Urban') & (df['Credit_History'] == 1)].shape[0] / df[df['Property_Area'] == 'Urban'].shape[0] * 100,
          df[df['Property_Area'] == 'Semiurban') & (df['Credit_History'] == 1)].shape[0] / df[df['Property_Area'] == 'Semiurban'].shape[0] * 100,
          df[df['Property_Area'] == 'Rural') & (df['Credit_History'] == 1)].shape[0] / df[df['Property_Area'] == 'Rural'].shape[0] * 100 ]
print(values)
plt.bar(labels, values)
plt.show()
```



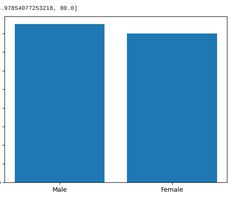




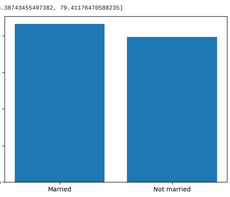




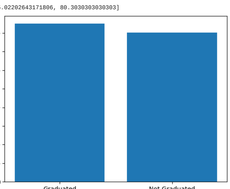

```
[84.97854077253218, 80.0]
```



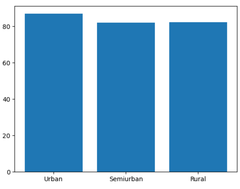

```
[86.38743554597302, 79.41176470588235]
```




```
[85.02202643171806, 80.30303030303033]
```




```
[86.8421052631579, 82.02247191011236, 82.22222222222221]
```


80
60
40
20
0
Urban
Semiurban
Rural