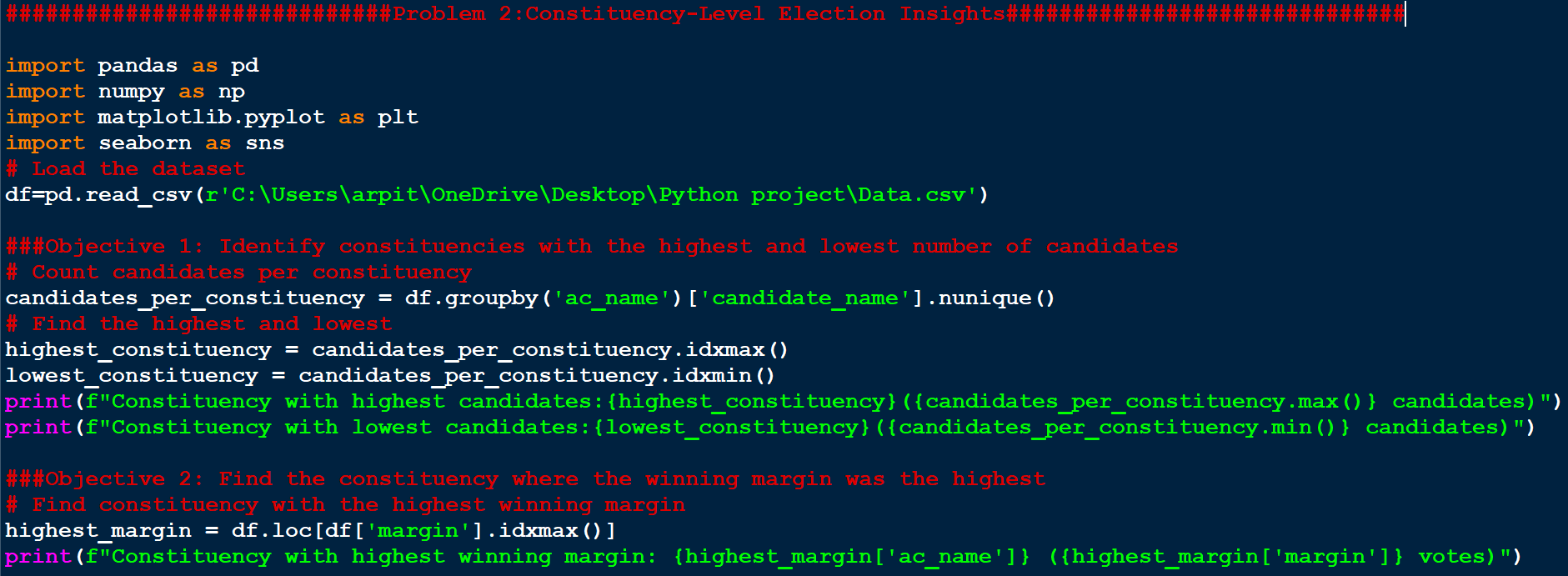
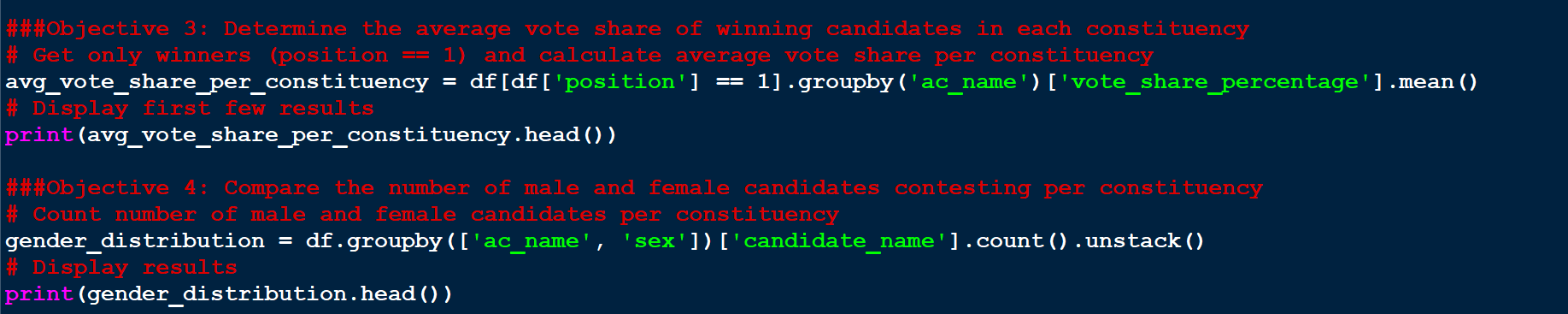
**Code:**





#############################Problem 2:Constituency-Level Election Insights##############################

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

df=pd.read\_csv(r'C:\Users\arpit\OneDrive\Desktop\Python project\Data.csv')

###Objective 1: Identify constituencies with the highest and lowest number of candidates

# Count candidates per constituency

candidates\_per\_constituency = df.groupby('ac\_name')['candidate\_name'].nunique()

# Find the highest and lowest

highest\_constituency = candidates\_per\_constituency.idxmax()

lowest\_constituency = candidates\_per\_constituency.idxmin()

print(f"Constituency with highest candidates:{highest\_constituency}({candidates\_per\_constituency.max()} candidates)")

print(f"Constituency with lowest candidates:{lowest\_constituency}({candidates\_per\_constituency.min()} candidates)")

###Objective 2: Find the constituency where the winning margin was the highest

# Find constituency with the highest winning margin

highest\_margin = df.loc[df['margin'].idxmax()]

print(f"Constituency with highest winning margin: {highest\_margin['ac\_name']} ({highest\_margin['margin']} votes)")

###Objective 3: Determine the average vote share of winning candidates in each constituency

# Get only winners (position == 1) and calculate average vote share per constituency

avg\_vote\_share\_per\_constituency = df[df['position'] == 1].groupby('ac\_name')['vote\_share\_percentage'].mean()

# Display first few results

print(avg\_vote\_share\_per\_constituency.head())

###Objective 4: Compare the number of male and female candidates contesting per constituency

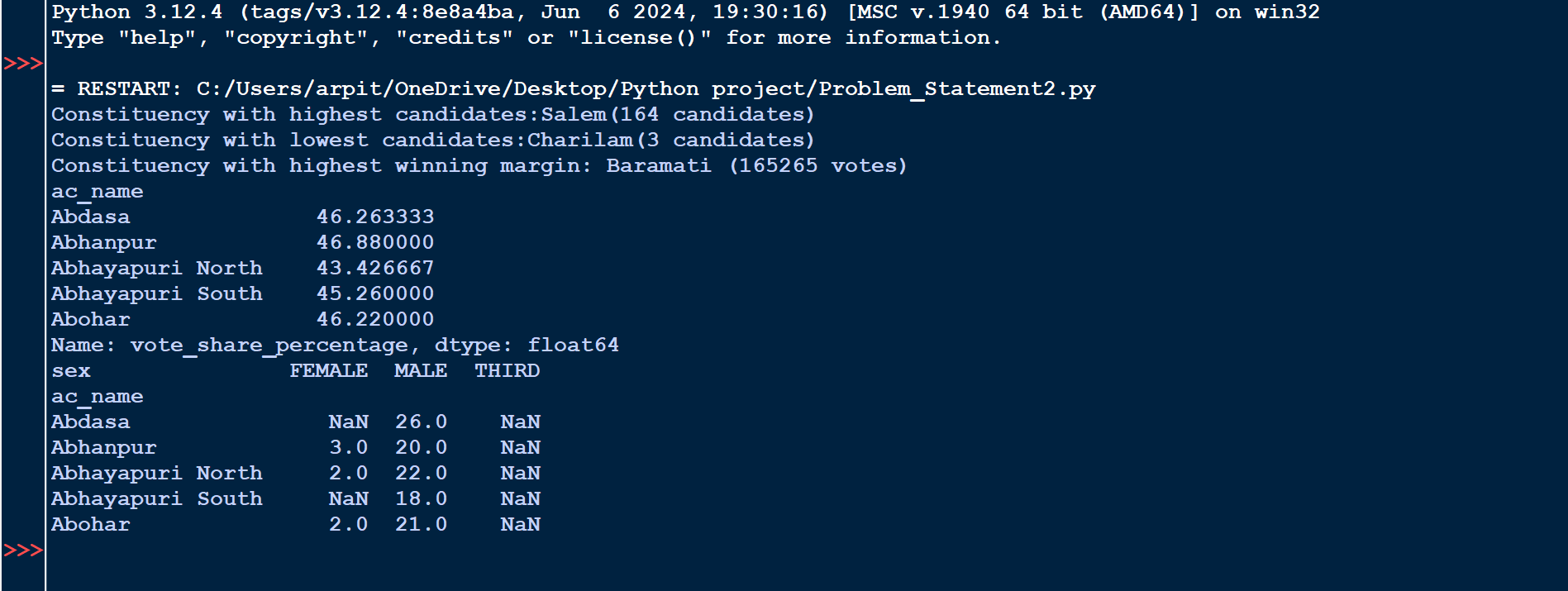
# Count number of male and female candidates per constituency

gender\_distribution = df.groupby(['ac\_name', 'sex'])['candidate\_name'].count().unstack()

# Display results

print(gender\_distribution.head())

**Output:**

****