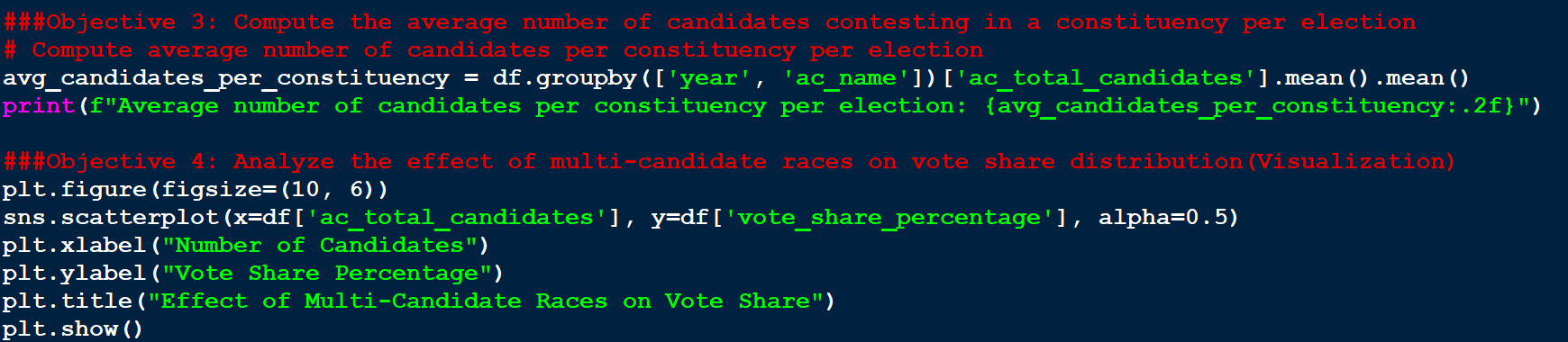
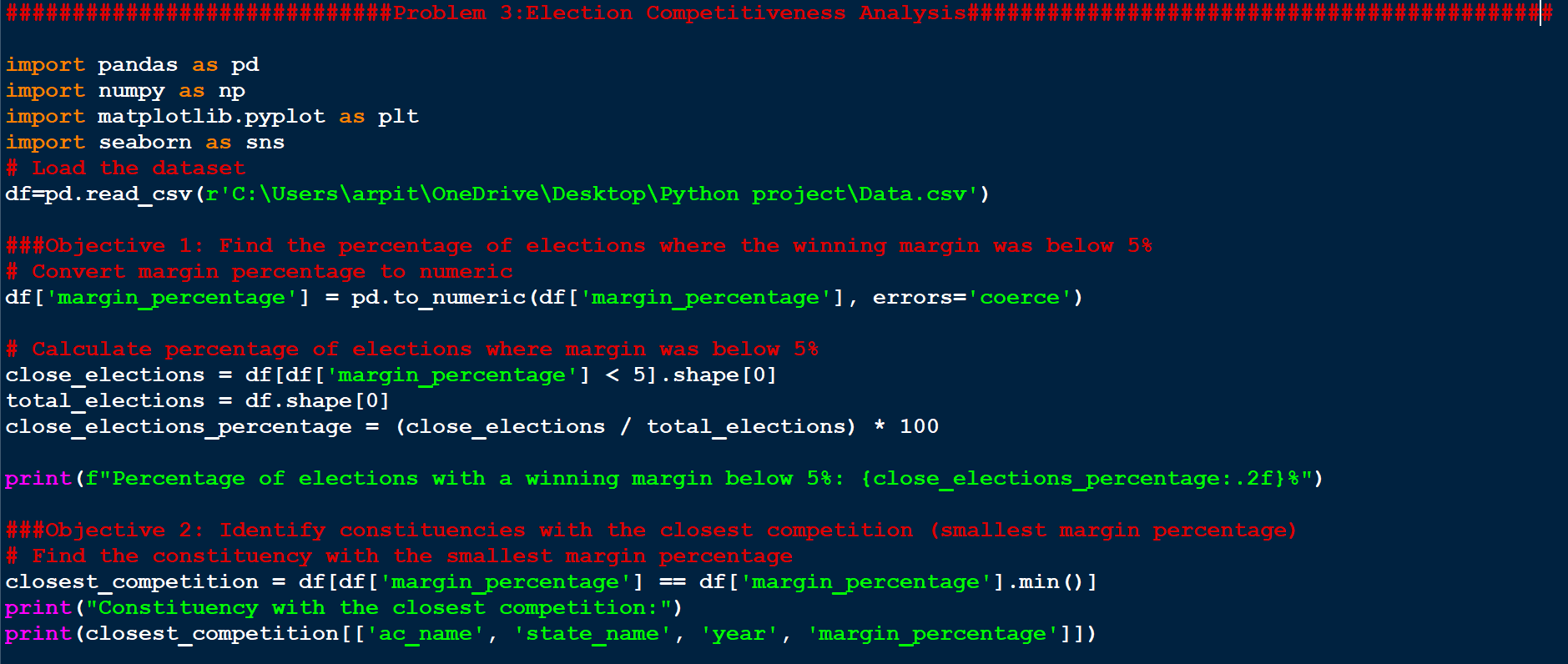
**Code:**



#############################Problem 3:Election Competitiveness Analysis#############################################

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: Find the percentage of elections where the winning margin was below 5%

# Convert margin percentage to numeric

df['margin\_percentage'] = pd.to\_numeric(df['margin\_percentage'], errors='coerce')

# Calculate percentage of elections where margin was below 5%

close\_elections = df[df['margin\_percentage'] < 5].shape[0]

total\_elections = df.shape[0]

close\_elections\_percentage = (close\_elections / total\_elections) \* 100

print(f"Percentage of elections with a winning margin below 5%: {close\_elections\_percentage:.2f}%")

###Objective 2: Identify constituencies with the closest competition (smallest margin percentage)

# Find the constituency with the smallest margin percentage

closest\_competition = df[df['margin\_percentage'] == df['margin\_percentage'].min()]

print("Constituency with the closest competition:")

print(closest\_competition[['ac\_name', 'state\_name', 'year', 'margin\_percentage']])

###Objective 3: Compute the average number of candidates contesting in a constituency per election

# Compute average number of candidates per constituency per election

avg\_candidates\_per\_constituency = df.groupby(['year', 'ac\_name'])['ac\_total\_candidates'].mean().mean()

print(f"Average number of candidates per constituency per election: {avg\_candidates\_per\_constituency:.2f}")

###Objective 4: Analyze the effect of multi-candidate races on vote share distribution(Visualization)

plt.figure(figsize=(10, 6))

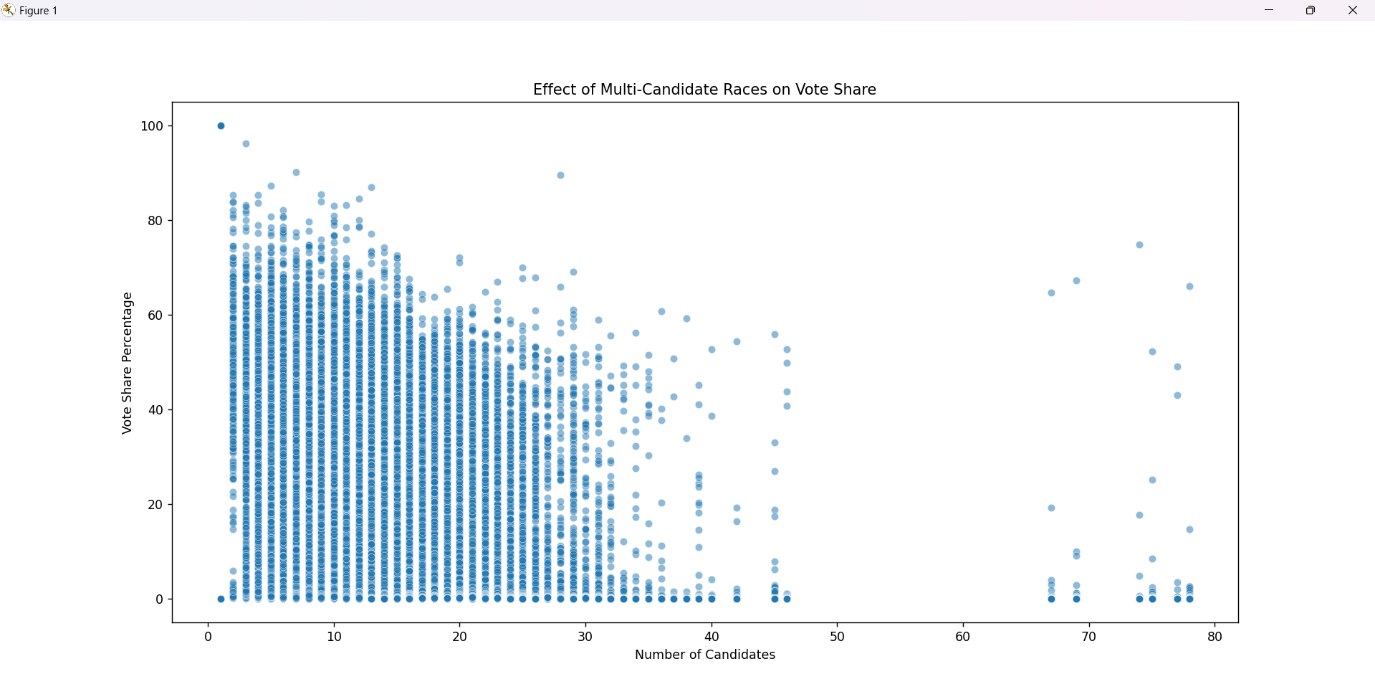
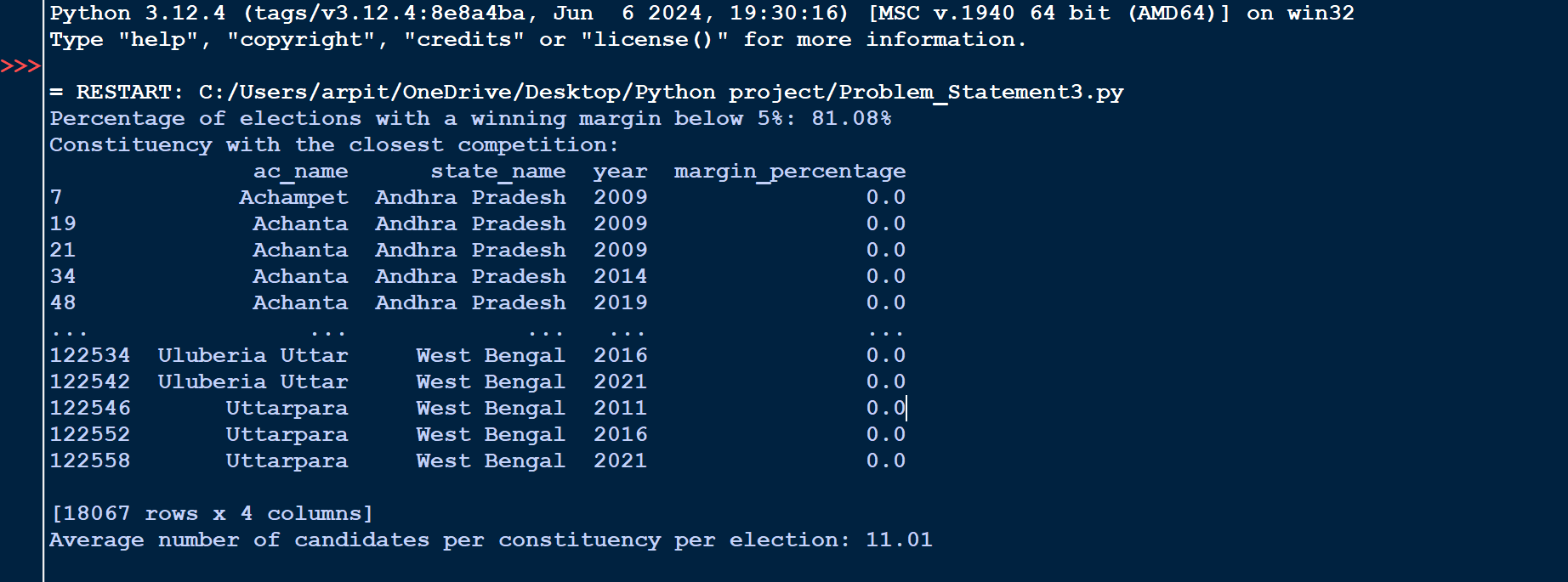
sns.scatterplot(x=df['ac\_total\_candidates'], y=df['vote\_share\_percentage'], alpha=0.5)

plt.xlabel("Number of Candidates")

plt.ylabel("Vote Share Percentage")

plt.title("Effect of Multi-Candidate Races on Vote Share")

plt.show()

**Output: **