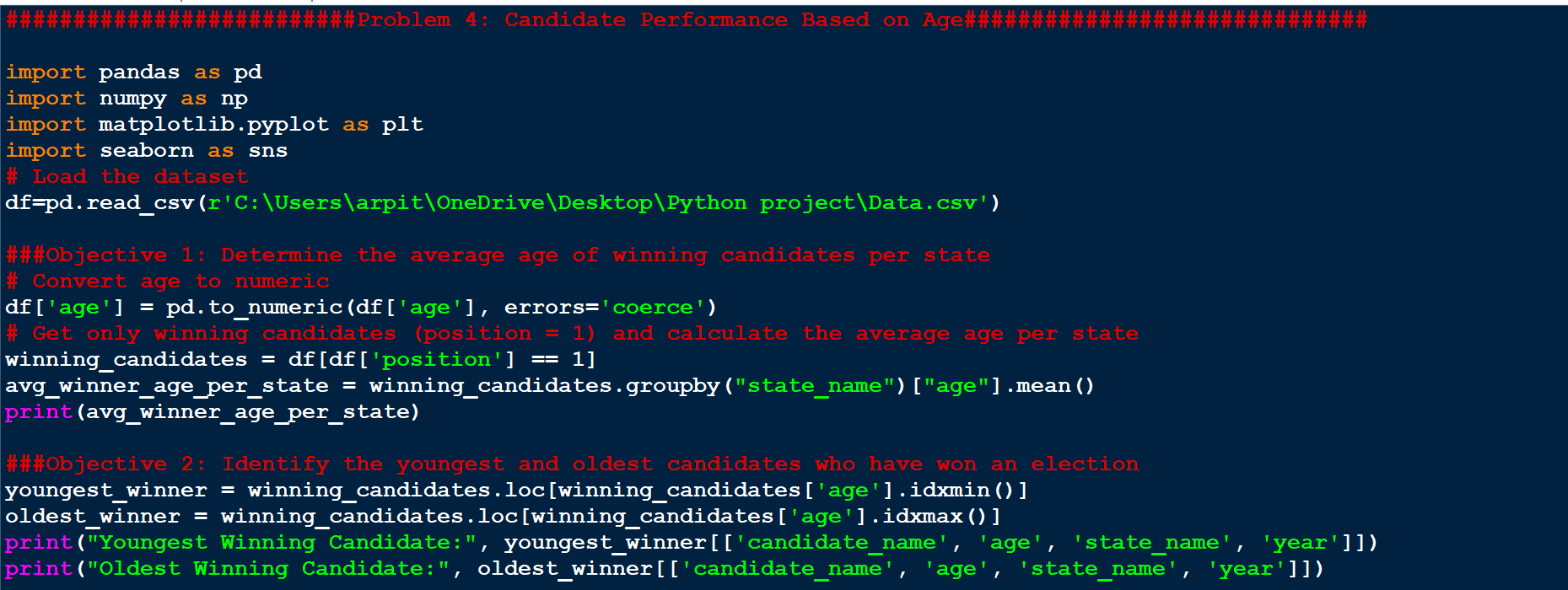
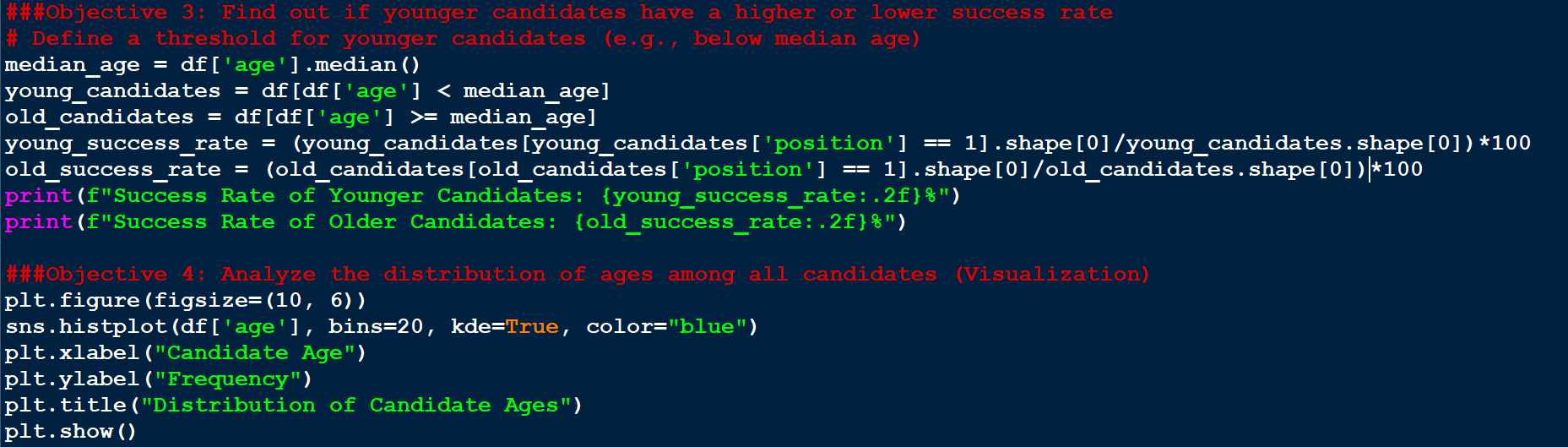
**Code:**





##########################Problem 4: Candidate Performance Based on Age##############################

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: Determine the average age of winning candidates per state

# Convert age to numeric

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

# Get only winning candidates (position = 1) and calculate the average age per state

winning\_candidates = df[df['position'] == 1]

avg\_winner\_age\_per\_state = winning\_candidates.groupby("state\_name")["age"].mean()

print(avg\_winner\_age\_per\_state)

###Objective 2: Identify the youngest and oldest candidates who have won an election

youngest\_winner = winning\_candidates.loc[winning\_candidates['age'].idxmin()]

oldest\_winner = winning\_candidates.loc[winning\_candidates['age'].idxmax()]

print("Youngest Winning Candidate:", youngest\_winner[['candidate\_name', 'age', 'state\_name', 'year']])

print("Oldest Winning Candidate:", oldest\_winner[['candidate\_name', 'age', 'state\_name', 'year']])

###Objective 3: Find out if younger candidates have a higher or lower success rate

# Define a threshold for younger candidates (e.g., below median age)

median\_age = df['age'].median()

young\_candidates = df[df['age'] < median\_age]

old\_candidates = df[df['age'] >= median\_age]

young\_success\_rate = (young\_candidates[young\_candidates['position'] == 1].shape[0]/young\_candidates.shape[0])\*100

old\_success\_rate = (old\_candidates[old\_candidates['position'] == 1].shape[0]/old\_candidates.shape[0])\*100

print(f"Success Rate of Younger Candidates: {young\_success\_rate:.2f}%")

print(f"Success Rate of Older Candidates: {old\_success\_rate:.2f}%")

###Objective 4: Analyze the distribution of ages among all candidates (Visualization)

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

sns.histplot(df['age'], bins=20, kde=True, color="blue")

plt.xlabel("Candidate Age")

plt.ylabel("Frequency")

plt.title("Distribution of Candidate Ages")

plt.show()

**Output:**

