```
Start coding or generate with AI.
```

Basic Information: The dataset contains various columns such as id, sex, age, federal_district, type_of_city, knows_election_date, will_vote, candidate, television_usage, internet_usage, education, income, employment, job_type, company_type, and weight1.

```
#import lib
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load the dataset
df = pd.read_csv('election2024 (1).csv')
```

```
df.sample(5)
```

	id	sex	age	federal_district	type_of_city	knows_election_date	will_vote	candidate	television_usag	
1062	0716c7c3a84ceb94	male	76.0	central	city with population of 50-100k	named correct date	likely yes	Putin	over 4 hours a day	le
236	07169a48c4cce476	male	54.0	north caucasian	village	named correct date	definitely yes	Putin	less than 4 hours a day	le
433	07169758e64ce981	female	70.0	northwestern	city with population of 100-500k	named correct date	likely yes	Slutskiy	over 4 hours a day	le
624	0716cc6de98cea31	female	83.0	central	settlement	named correct date	definitely yes	Putin	less than 4 hours a day	
1161	07168addf04ce9c1	female	42.0	central	city with population of 100-500k	named correct date	likely yes	struggle to answer	less than 4 hours a day	

```
df.dtypes
```

	0
id	object
sex	object
age	float64
federal_district	object
type_of_city	object
knows_election_date	object
will_vote	object
candidate	object
television_usage	object
internet_usage	object
education	object
income	float64
employment	object
job_type	object
company_type	object
weight1	float64
age_group	category
income_group	category

dtype: object

```
# Basic Information
print("Basic Information:")
print(df.info())
```

```
Basic Information:
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1600 entries, 0 to 1599
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   id                   1600 non-null   object 
 1   sex                  1600 non-null   object 
 2   age                  1600 non-null   float64
 3   federal_district     1600 non-null   object 
 4   type_of_city         1600 non-null   object 
 5   knows_election_date  1600 non-null   object 
 6   will_vote            1600 non-null   object 
 7   candidate            1600 non-null   object 
 8   television_usage     1600 non-null   object 
 9   internet_usage       1600 non-null   object 
 10  education            1600 non-null   object 
 11  income               1600 non-null   object 
 12  employment           1600 non-null   object 
 13  job_type             692 non-null    object 
 14  company_type         879 non-null    object 
 15  weight1              1600 non-null   float64
dtypes: float64(2), object(14)
memory usage: 200.1+ KB
None
```

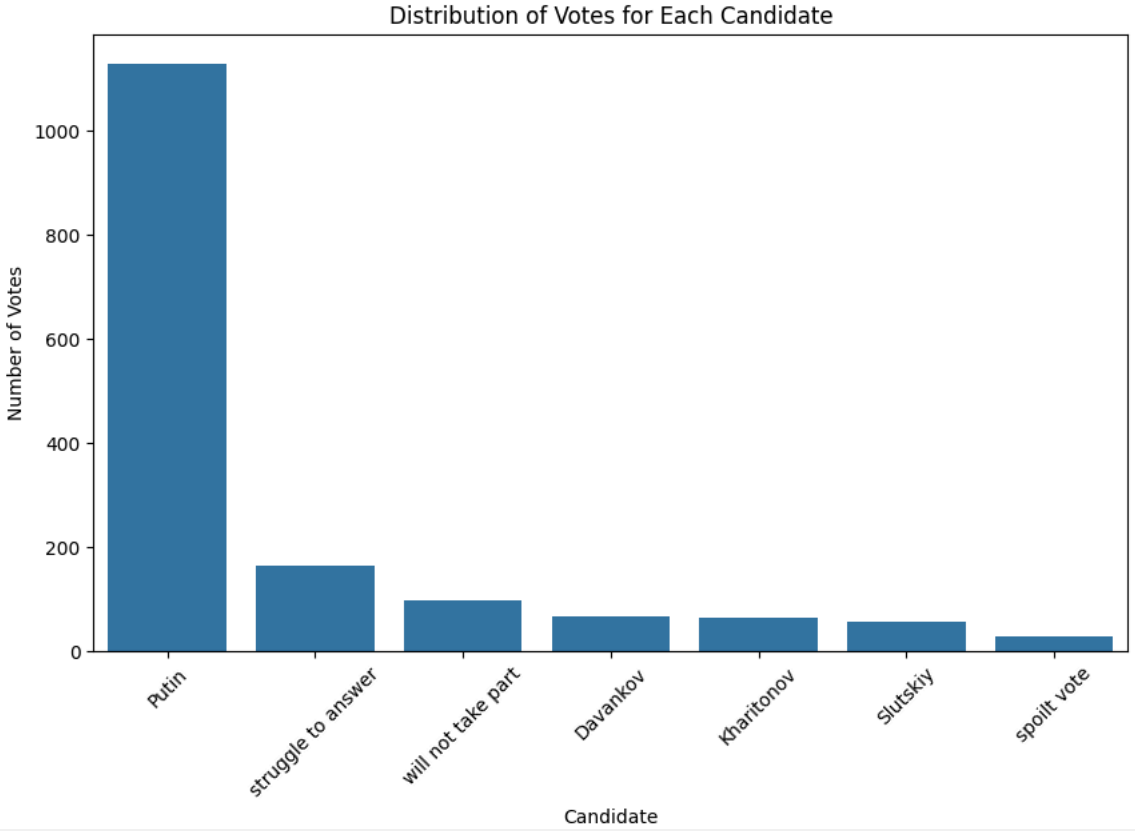
```
# Summary Statistics
print("\nSummary Statistics:")
print(df.describe())
```

```
Summary Statistics:
               age      weight1
count  1600.000000  1600.000000
mean     49.936250     1.000000
std      16.901797     0.327084
min      18.000000     0.468226
25%      37.000000     0.772224
50%      49.000000     0.921724
75%      64.000000     1.158913
max      90.000000     2.515072
```

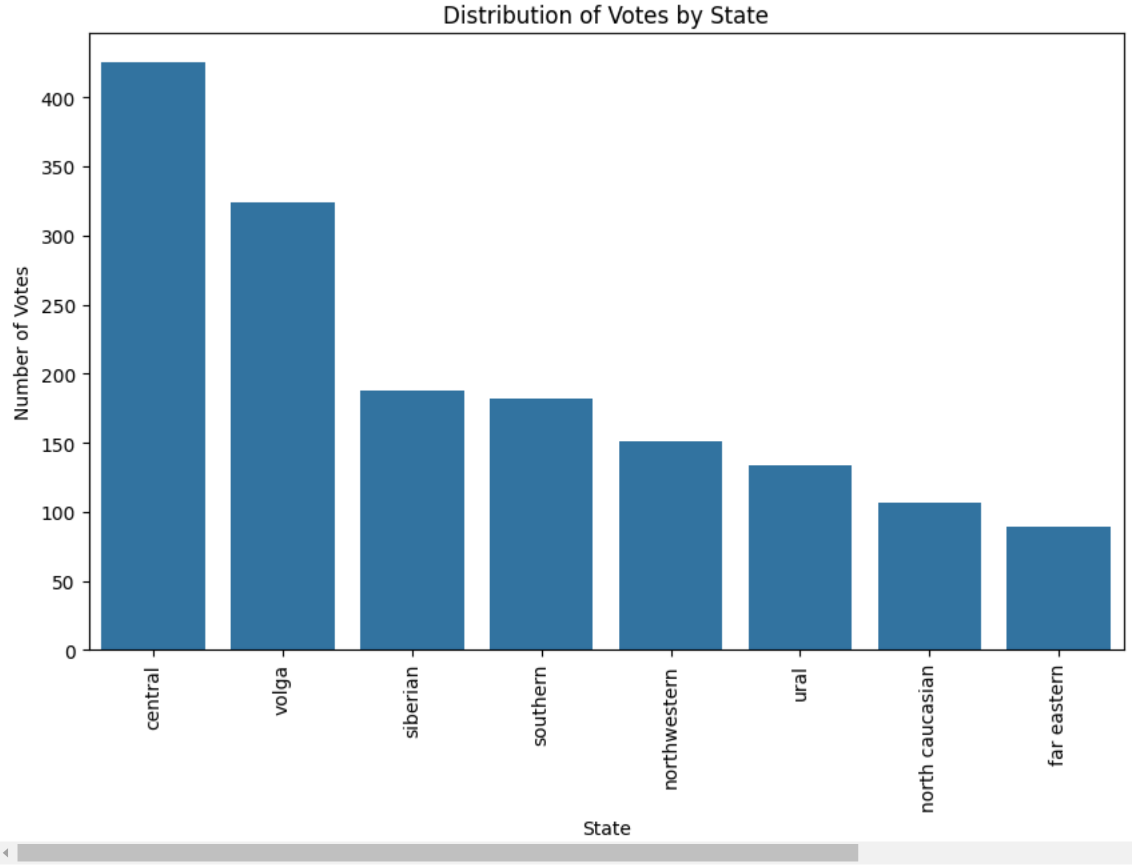
```
# Check for missing values
print("\nMissing Values:")
print(df.isnull().sum())
```

```
Missing Values:
id                       0
sex                      0
age                      0
federal_district         0
type_of_city             0
knows_election_date      0
will_vote                0
candidate                0
television_usage         0
internet_usage           0
education                0
income                   0
employment               0
job_type               908
company_type           721
weight1                  0
dtype: int64
```

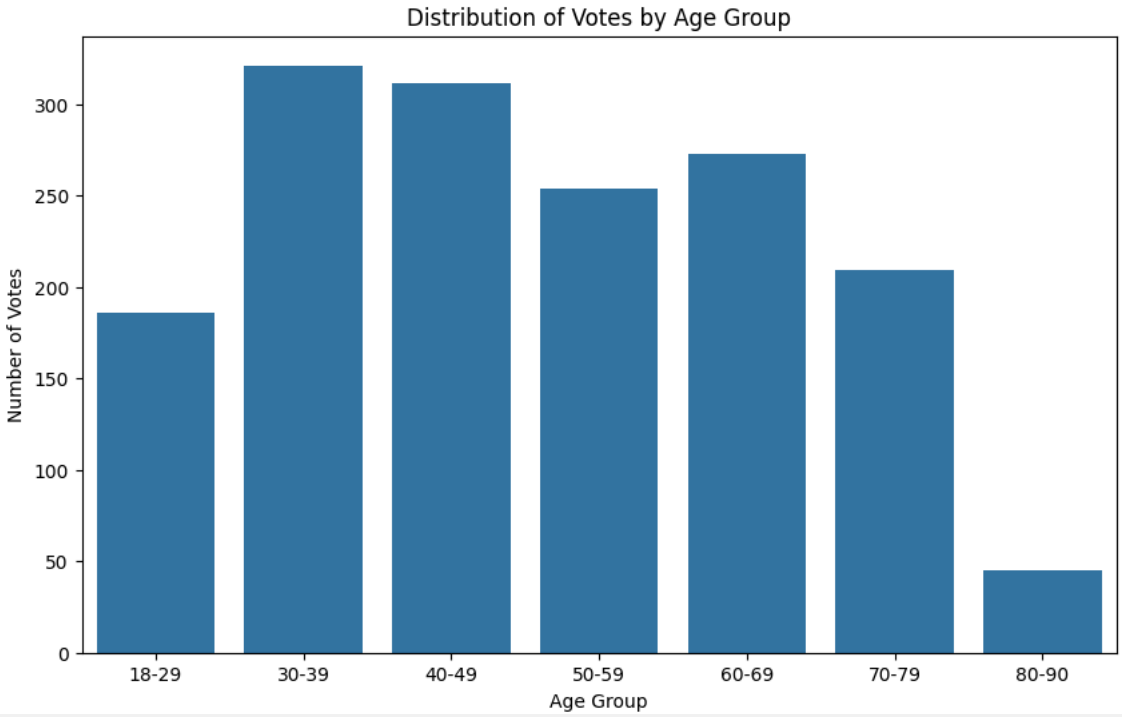
```
# Distribution of Votes for Each Candidate
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='candidate', order=df['candidate'].value_counts().index)
plt.title('Distribution of Votes for Each Candidate')
plt.xlabel('Candidate')
plt.ylabel('Number of Votes')
plt.xticks(rotation=45)
plt.show()
```



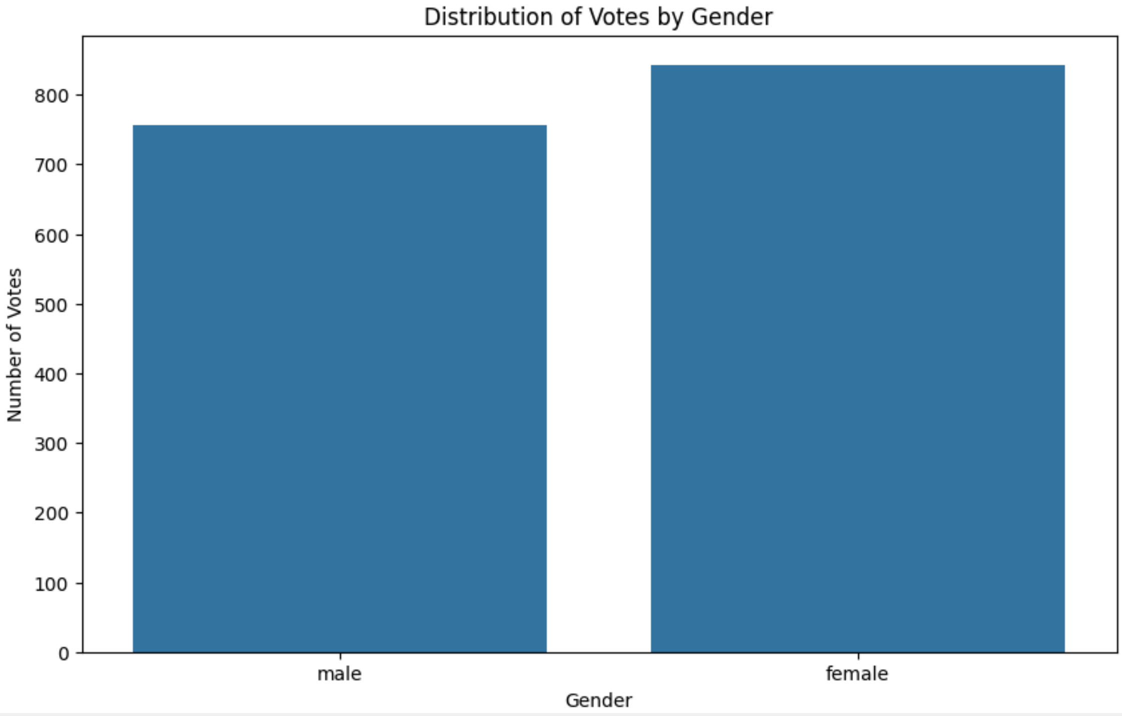
```
# Distribution of Votes by State
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='federal_district', order=df['federal_district'].value_counts().index)
plt.title('Distribution of Votes by State')
plt.xlabel('State')
plt.ylabel('Number of Votes')
plt.xticks(rotation=90)
plt.show()
```



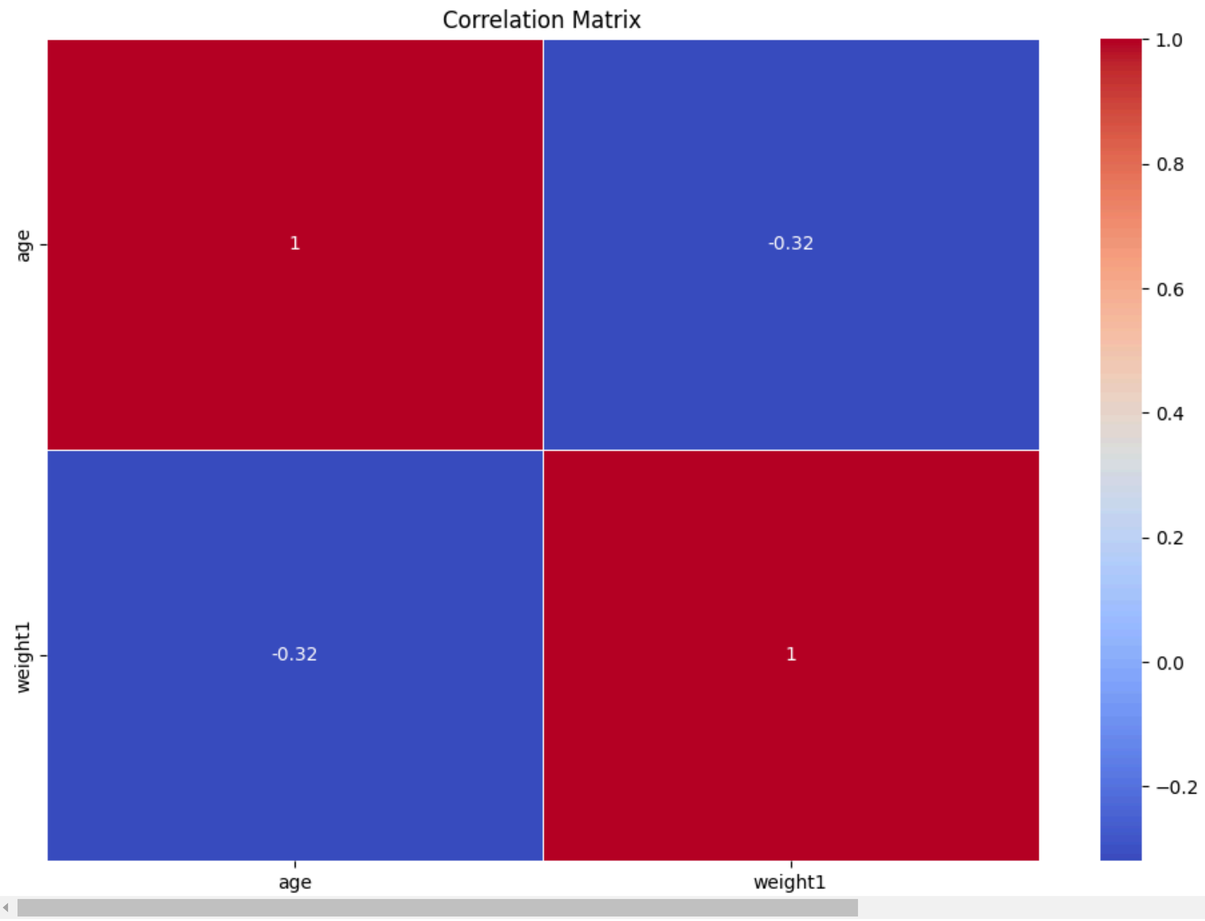
```
# Distribution of Votes by Age Group
age_bins = [18, 30, 40, 50, 60, 70, 80, 90]
age_labels = ['18-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80-90']
age_labels = ['18-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80-90']
df['age_group'] = pd.cut(df['age'], bins=age_bins, labels=age_labels, right=False)
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='age_group', order=age_labels)
plt.title('Distribution of Votes by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Number of Votes')
plt.show()
```



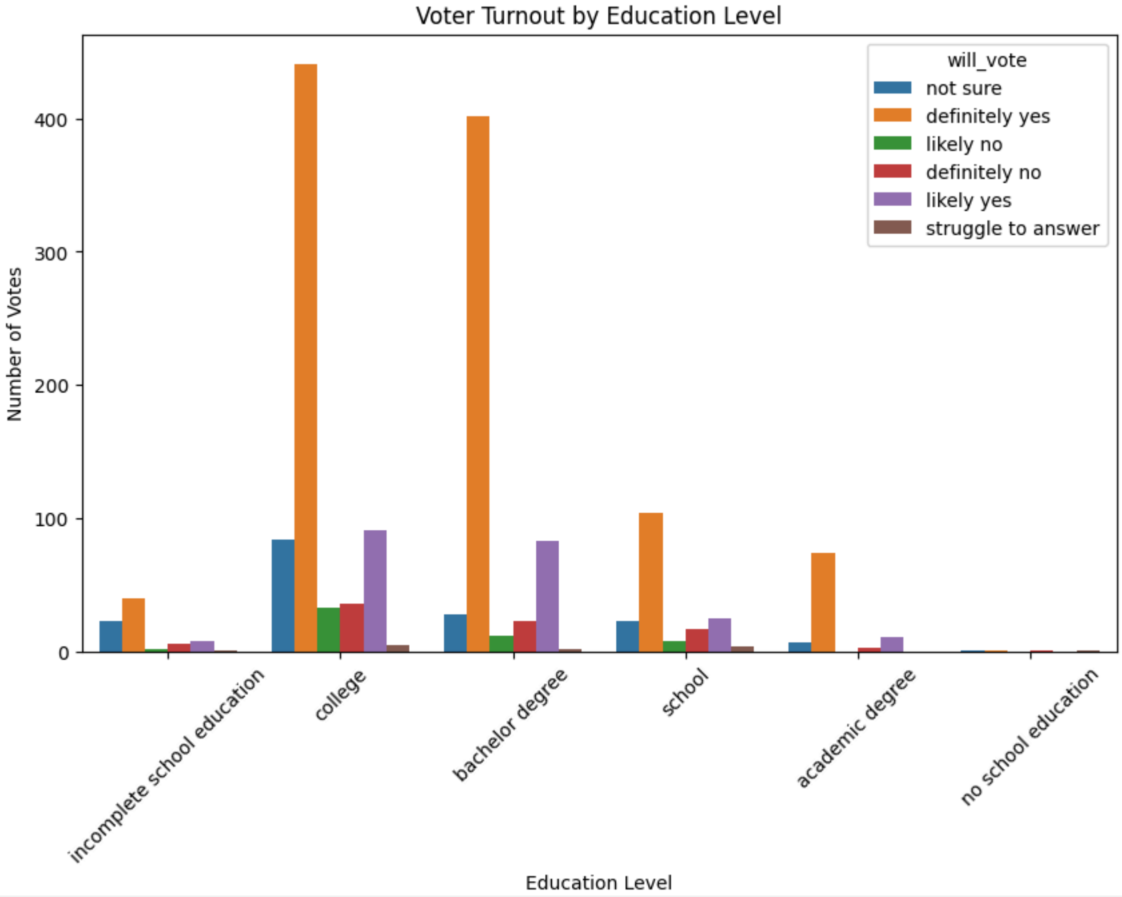
```
# Distribution of Votes by Gender
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='sex')
plt.title('Distribution of Votes by Gender')
plt.xlabel('Gender')
plt.ylabel('Number of Votes')
plt.show()
```



```
# Correlation Matrix
plt.figure(figsize=(12, 8))
correlation_matrix = df.corr(numeric_only=True)
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```



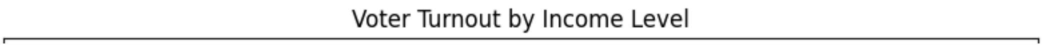
```
# Advanced Analysis: Voter Turnout by Education Level
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='education', hue='will_vote')
plt.title('Voter Turnout by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Number of Votes')
plt.xticks(rotation=45)
plt.show()
```



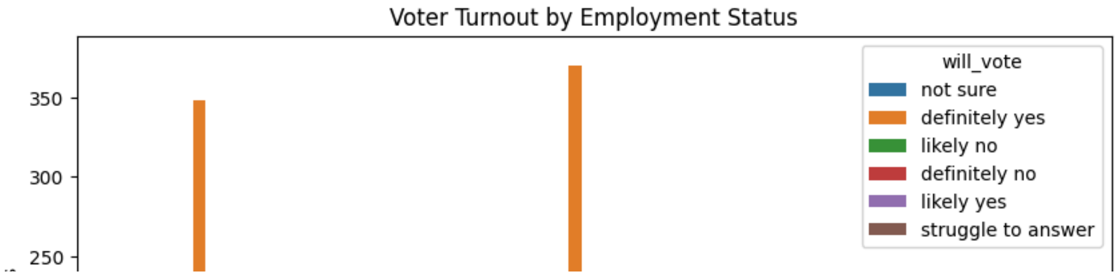
```
# Advanced Analysis: Voter Turnout by Income Level
# Convert 'income' column to numeric, handling non-numeric values
df['income'] = pd.to_numeric(df['income'], errors='coerce')

# Proceed with binning
income_bins = [0, 20000, 40000, 60000, 80000, 100000, 120000]
income_labels = ['0-20k', '20k-40k', '40k-60k', '60k-80k', '80k-100k', '100k-120k']
df['income_group'] = pd.cut(df['income'], bins=income_bins, labels=income_labels)

# Plotting Voter Turnout by Income Level
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='income_group', hue='will_vote')
plt.title('Voter Turnout by Income Level')
plt.xlabel('Income Level')
plt.ylabel('Number of Votes')
plt.show()
```



```
# Advanced Analysis: Voter Turnout by Employment Status
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='employment', hue='will_vote')
plt.title('Voter Turnout by Employment Status')
plt.xlabel('Employment Status')
plt.ylabel('Number of Votes')
plt.xticks(rotation=45)
plt.show()
```


Voter Turnout by Employment Status
will_vote
not sure
definitely yes
likely no
definitely no
likely yes
struggle to answer