This *'ipynb'* file shows the Data Analysis and Visualizations of the 2019 Andhra Pradesh Elections

```
# connecting to Google Drive
from google.colab import drive
drive.mount('/content/drive')
%cd /content/drive/My Drive/AP elections
```

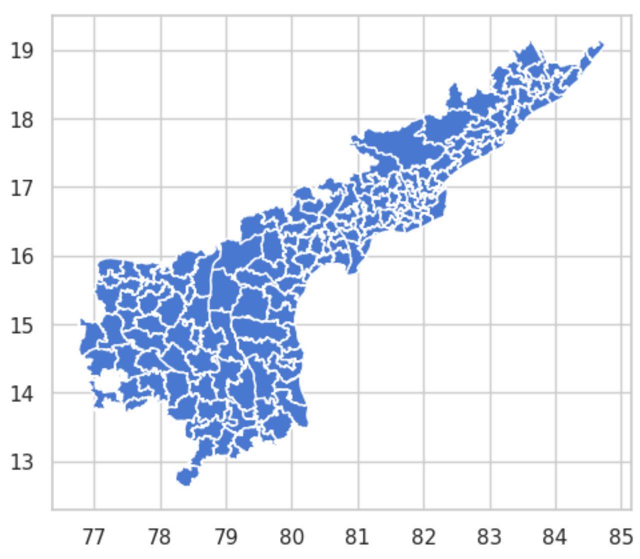
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
/content/drive/My Drive/AP elections
```

```
import pandas as pd
import numpy as np
import geopandas as gpd
import matplotlib.pyplot as plt
from matplotlib import pyplot as plt
import seaborn as sns
```

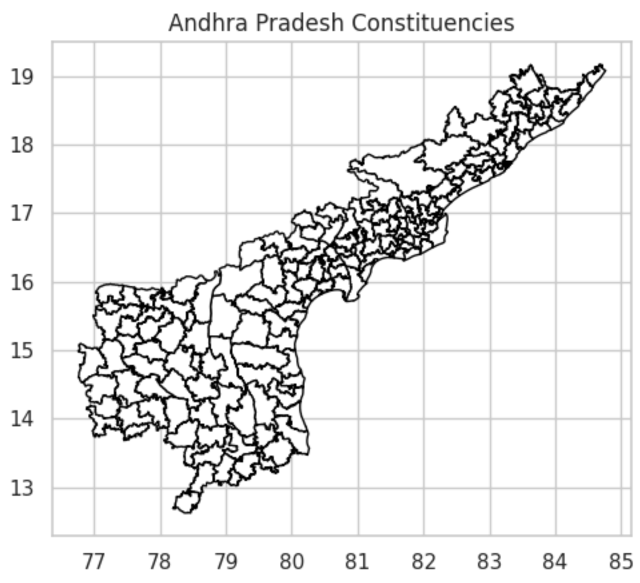
```
# Load the dataset
data = pd.read_csv("/content/drive/MyDrive/AP elections/data 2.csv", encoding="windows-1252")

print(data.head())

# Handling missing values
data['age'].fillna(data['age'].mean(), inplace=True)

data['sex'].fillna(data['sex'].mode()[0], inplace=True)

data['category'].fillna(data['category'].mode()[0], inplace=True)

data['postal'].fillna(data['postal'].median(), inplace=True)

print(data.isnull().sum())
```

```
                candidate_name     sex   age category party         symbol  \
0             GANAPA VANAJAKSHI  FEMALE  31.0  GENERAL   IND           Ring
1                          NOTA     NaN   NaN      NaN  NOTA           NOTA
2               ESWARA RAO KOLLI    MALE  50.0  GENERAL   INC           Hand
3                    DASARI RAJU    MALE  42.0  GENERAL   JnP  Glass Tumbler
4  JANNALA SURYAVARA PRASADA RAO    MALE  67.0  GENERAL   BJP          Lotus

   general  postal  total  percentage_votes_polled  total_electors
0      613     4.0    617                 0.357841          247941
1     3872     8.0   3880                 2.250280          247941
2     2100    38.0   2138                 1.239974          247941
3    10940   183.0  11123                 6.450996          247941
4     1656   170.0   1826                 1.059023          247941
year                       0
state                      0
ac_number                  0
ac_name                    0
candidate_name             0
sex                        0
age                        0
category                   0
party                      0
symbol                     0
general                    0
postal                     0
total                      0
percentage_votes_polled    0
total_electors             0
dtype: int64
<ipython-input-388-6ddddf4ea398>:7: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using a
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always be

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)


  data['age'].fillna(data['age'].mean(), inplace=True)
<ipython-input-388-6ddddf4ea398>:9: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using a
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always be

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)


  data['sex'].fillna(data['sex'].mode()[0], inplace=True)
<ipython-input-388-6ddddf4ea398>:11: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always be

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)


  data['category'].fillna(data['category'].mode()[0], inplace=True)
<ipython-input-388-6ddddf4ea398>:13: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always be

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)


  data['postal'].fillna(data['postal'].median(), inplace=True)
```

```
# Load the shapefile
shapefile_path = "/content/drive/MyDrive/AP elections/ANDHRA PRADESH_ASSEMBLY.geojson"

andhra_map = gpd.read_file(shapefile_path)
andhra_map.plot()
plt.show()
```



```
# Plot the map with added labels or additional styling
andhra_map.plot(edgecolor='black', color='white')
plt.title("Andhra Pradesh Constituencies")
plt.show()
```



```
# Load your dataset
file_path = "/content/drive/MyDrive/AP elections/data 2.csv"
data = pd.read_csv(file_path, encoding="windows-1252")

# Create a dictionary for spelling corrections
spelling_corrections = {
    "Vizianagaram": "VIZIANAGARM",
    "Srungavarapukota": "SRUNGAVARAPUKOTA",
    "Bhimli": "BHIMILI",
    "Visakhapatnam East": "VISHAKAPATNAM EAST",
    "Visakhapatnam  West": "VISHAKAPATNAM WEST",
    "Visakhapatnam  North": "VISHAKAPATNAM NORTH",
    "Visakhapatnam South": "VISHAKAPATNAM SOUTH",
    "Gajuwaka": "GAJUWAKA",
    "Chodavaram": "CHODAVARM",
    "V.Madugula": "MADUGULA",
    "Araku valley": "ARAKU VALLEY (ST)",
    "Paderu": "PADERU (ST)",
    "Anakapalli": "ANAKAPALLE",
    "Pendurthi": "PENDURTHI",
    "ELAMANCHILI": "YELAMANCHILI",
    "PAYAKARAOPETA": "PAYAKARAOPET (SC)",
    "Narsipatnam": "NARSIPATNAM",
    "Tuni": "TUNI",
    "Prathipadu": "PRATHIPADU",
    "Pithapuram": "PITHAPURAM",
    "Kakinada Rural": "KAKINADA RURAL",
    "Peddapuram": "PEDDAPURAM",
    "Anaparthy": "ANAPARTHY",
    "Kakinada City": "KAKINADA URBAN",
    "Ramachandrapuram": "RAMACHANDRAPURAM",
    "Mummidivaram": "MUMMIDIVARAM",
    "Amalapuram": "AMALAPURAM (SC)",
    "Razole": "RAZOLE (SC)",
    "Gannavaram": "GANNAVARAM (SC)",
    "Kothapeta": "KOTHAPETA",
    "Mandapeta": "MANDAPETA",
    "Rajanagaram": "RAJANAGARAM",
    "Rajahmundry City": "RAJAHMUNDRY URBAN",
    "Rajamundry Rural": "RAJAHMUNDRY RURAL",
    "Jaggampeta": "JAGGAMPETA",
    "Rampachodavaram": "RAMPACHODAVARAM (ST)",
    "Kovvur": "KOVVUR (SC)",
    "Nidadavole": "NIDADAVOLE",
    "Achanta": "ACHANTA",
    "Palacole": "PALACOLE",
    "Narasapuram": "NARSAPURAM",
    "Bhimavaram": "BHIMAVARAM",
    "Undi": "UNDI",
    "Tadepalligudem": "TADEPALLIGUDEM",
    "Unguturu": "UNGUTUR",
    "Denduluru": "DENDULURU",
    "Eluru": "ELURU",
    "Gopalapuram": "GOPALAPURAM (SC)",
    "Polavaram": "POLAVARAM (ST)",
    "Chintalapudi": "CHINTALAPUDI (SC)",
    "Tiruvuru": "TIRUVURU (SC)",
    "Nuzvid": "NUZVID",
    "Gannavaram": "GANGAVARAM",
    "Gudivada": "GUDIVADA",
    "Kaikalur": "KAIKALUR",
    "Pedana": "PEDANA",
    "Machilipatnam": "MACHILIPATNAM",
    "Avanigadda": "AVANIGADDA",
    "Pamarru": "PAMARRU (SC)",
    "Penamaluru": "PENAMALURU",
    "Vijaywada West": "VIJAYAWADA WEST",
    "Vijayawada central": "VIJAYAWADA CENTRAL",
    "Vijayawada East": "VIJAYAWADA EAST",
    "Mylavaram": "MYLAVARAM",
    "Nandigama": "NANDIGAMA (SC)",
    "Jaggayyapeta": "JAGGAYYAPETA",
    "Pedakurapadu": "PEDAKURAPADU",
    "Mangalagiri": "MANGALAGIRI",
    "Ponnur": "PONNUR",
    "Vemuru (SC)": "VEMURU (SC)",
    "Repalle": "REPALLE",
    "Tenali": "TENALI",
    "Bapatla": "BAPATLA",
    "Prathipadu (SC)": "PRATHIPADU (SC)",
    "Guntur West": "GUNTUR WEST",
    "Guntur East": "GUNTUR EAST",
    "Chilakaluripet": "CHILAKALURIPET",
    "Narasaraopet": "NARASARAOPET",
    "Sattenapalli": "SATTENAPALLE",
    "Vinukonda": "VINUKONDA",
    "Gurazala": "GURAZALA",
    "Macherla": "MACHERLA",
    "Yerragondapalem": "YERRAGONDAPALEM (SC)",
    "Darsi": "DARSI",
    "Parchur": "PARCHUR",
    "Addanki": "ADDANKI",
    "Chirala": "CHIRALA",
    "Santhanuthalapadu": "SANTANUTHALAPADU (SC)",
    "Ongole": "ONGOLE",
    "Kandukur": "KANDUKUR",
    "Kondapi": "KONDAPI (SC)",
    "Markapuram": "MARKAPURAM",
    "Giddalur": "GIDDALUR",
    "Kanigiri": "KANIGIRI",
    "Kavali": "KAVALI",
    "Atmakur": "ATMAKUR",
    "Kovur": "KOVURU",
    "Nellore City": "NELLORE URBAN",
    "Nellore Rural": "NELLORE RURAL",
    "Sarvepalli": "SARVEPALLI",
    "Gudur": "GUDUR (SC)",
    "Sullurpeta": "SULLURUPETA (SC)",
    "Venkatagiri": "VENKATAGIRI",
    "Udayagiri": "UDAYAGIRI",
    "Badvel": "BADVEL (SC)",
    "Rajampet": "RAJAMPET",
    "Kadapa": "YSR KADAPA",
    "Kodur": "KODUR (SC)",
    "Rayachoti": "RAYACHOTI",
    "Pulivendla": "PULIVENDLA",
    "Kamalapuram": "KAMALAPURAM",
    "Jammalamadugu": "JAMMALAMADUGU",
    "Proddatur": "PRODDATUR",
    "Mydukur": "S.MYDUKUR",
    "Allagadda": "ALLAGADDA",
    "Srisailam": "SRISAILAM",
    "Nandikotkur": "NANDIKOTKUR (SC)",
    "Kurnool": "KURNOOL",
    "Panyam": "PANYAM",
    "Nandyal": "NANDYAL",
    "Banaganapalle": "BANAGANAPALLE",
    "Dhone": "DHONE",
    "Pattikonda": "PATTIKONDA",
    "Kodumur": "KODUMURU (SC)",
    "Yemmiganur": "YEMMIGANUR",
    "Mantralayam": "MANTRALAYAM",
    "Adoni": "ADONI",
    "Alur": "ALUR",
    "Rayadurg": "RAYADURG",
    "Uravakonda": "URAVAKONDA",
    "Guntakal": "GUNTAKAL",
    "Tadipatri": "TADIPATRI",
    "Singanamala": "SINGANAMALA (SC)",
    "Anantapur urban": "ANANTAPUR URBAN",
    "Raptadu": "RAPTADU",
    "Madakasira": "MADAKASIRA (SC)",
    "Hindupur": "HINDUPUR",
    "Penukonda": "PENUKONDA",
    "Puttaparthi": "PUTTAPARTHI",
    "Dharmavaram": "DHARMAVARAM",
    "Kadiri": "KADIRI",
    "Thamballapalle": "THAMALLAPALLE",
    "Pileru": "PILERU",
    "Madanapalle": "MADANAPALLE",
    "Punganur": "PUNGANUR",
    "Chandragiri": "CHANDRAGIRI",
    "Tirupati": "TIRUPATI",
    "Srikalahasti": "SRIKALAHASTI",
    "Satyavedu": "SATYAVEDU (SC)",
    "Nagari": "NAGARI",
    "Gangadhara Nellore": "GANGADHARANELLORE (SC)",
    "Puthalapattu": "PUTHALAPATTU (SC)",
    "Palamaner": "PALAMANER",
    "Kuppam": "KUPPAM",
    "Chittoor": "CHITTOOR",
}

# Apply the corrections
data['ac_name'] = data['ac_name'].replace(spelling_corrections)

# Save the corrected dataset
corrected_file_path = "corrected_data.csv"
data.to_csv(corrected_file_path, index=False)

print(f"Spelling corrections applied. Corrected dataset saved to {corrected_file_path}.")
```

```
Spelling corrections applied. Corrected dataset saved to corrected_data.csv.
```

```
# Load Shapefile and Election Results Data
shapefile_path = "/content/drive/MyDrive/AP elections/ANDHRA_PRADESH_MERGED.geojson"  # Update with the shapefile path
election_results_path = "/content/drive/MyDrive/AP elections/corrected_data.csv"  # Update with the CSV file path

# Load the shapefile
andhra_map = gpd.read_file(shapefile_path)

# Load election results data
election_results = pd.read_csv(election_results_path)

# Find the Winning Party for Each Constituency

andhra_map = andhra_map.apply(lambda x: x.str.lower() if x.dtype == "object" else x)
election_results = election_results.applymap(lambda x: x.lower() if isinstance(x, str) else x)
```

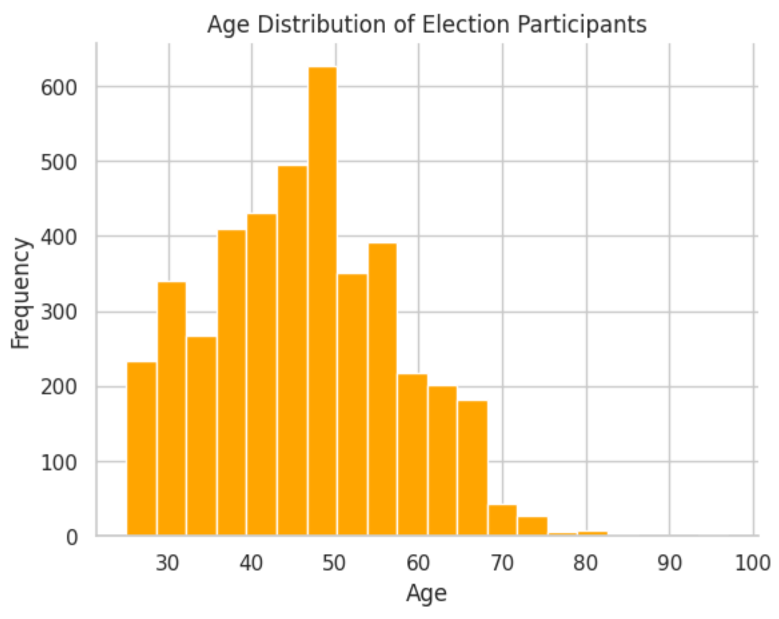
```
<ipython-input-392-8b3ae3e5109e>:14: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map instead.
  election_results = election_results.applymap(lambda x: x.lower() if isinstance(x, str) else x)
```

```
election_results.head()
```

	year	state	ac_number	ac_name	candidate_name	sex	age	category	party	symbol	general	postal	total	percentage_votes_polled	total_el
0	2014	andhra pradesh	1	ichchapuram	ganapa vanajakshi	female	31.0	general	ind	ring	613	4.0	617	0.357841	
1	2014	andhra pradesh	1	ichchapuram	nota	NaN	NaN	NaN	nota	nota	3872	8.0	3880	2.250280	
2	2014	andhra pradesh	1	ichchapuram	eswara rao kolli	male	50.0	general	inc	hand	2100	38.0	2138	1.239974	
3	2014	andhra pradesh	1	ichchapuram	dasari raju	male	42.0	general	jnp	glass tumbler	10940	183.0	11123	6.450996	
4	2014	andhra	1	ichchapuram	jannala survavara	male	67.0	general	bjp	lotus	1656	170.0	1826	1.059023	

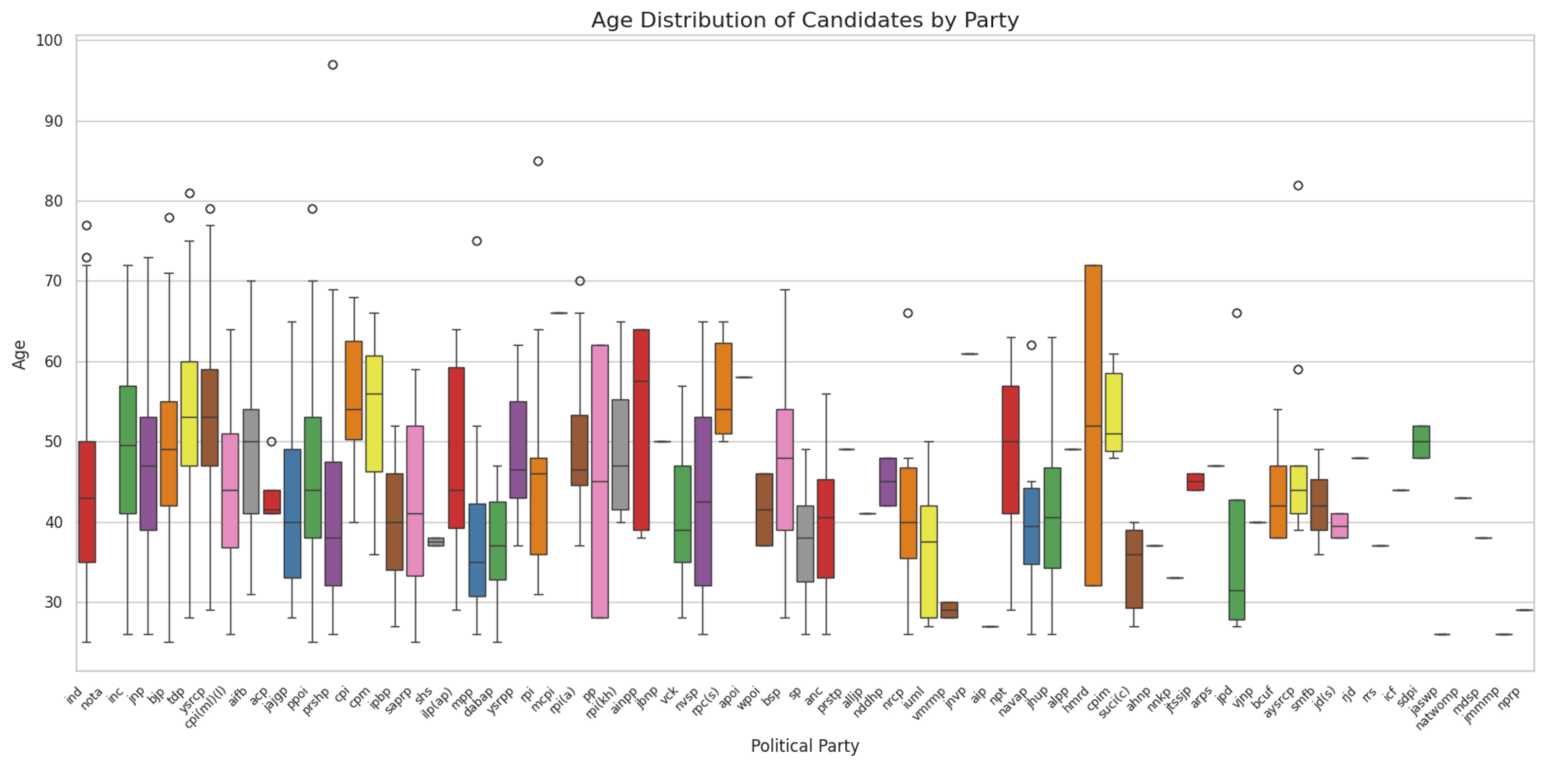
Next steps: Generate code with `election_results`   View recommended plots   New interactive sheet

```
# Age distribution of Election Participants using barplot
election_results['age'].plot(kind='hist', bins=20, color='orange')

plt.gca().spines[['top', 'right']].set_visible(False)
plt.title('Age Distribution of Election Participants')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



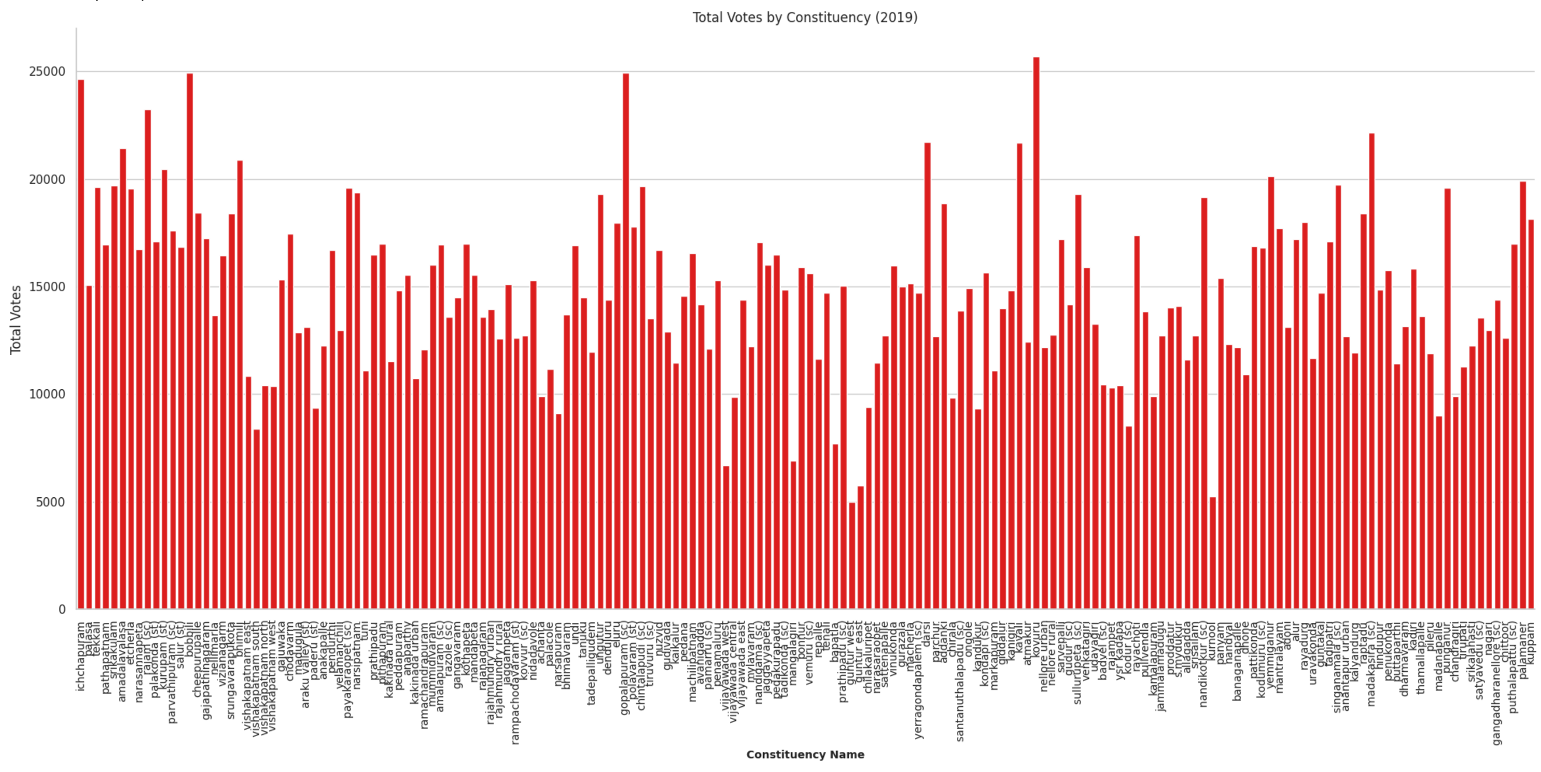
```
# Age distribution of candidates by party using boxplot.
sns.set_theme(style="whitegrid", palette="muted")

plt.figure(figsize=(16, 8))
sns.boxplot(x='party', y='age', data=election_results, palette="Set1")
plt.xticks(rotation=45, ha='right', fontsize=9)
plt.title('Age Distribution of Candidates by Party', fontsize=16)
plt.xlabel('Political Party', fontsize=12)
plt.ylabel('Age', fontsize=12)
plt.tight_layout()
plt.show()
```

```
<ipython-input-395-c76d0cb3a65a>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the

  sns.boxplot(x='party', y='age', data=election_results, palette="Set1")
```

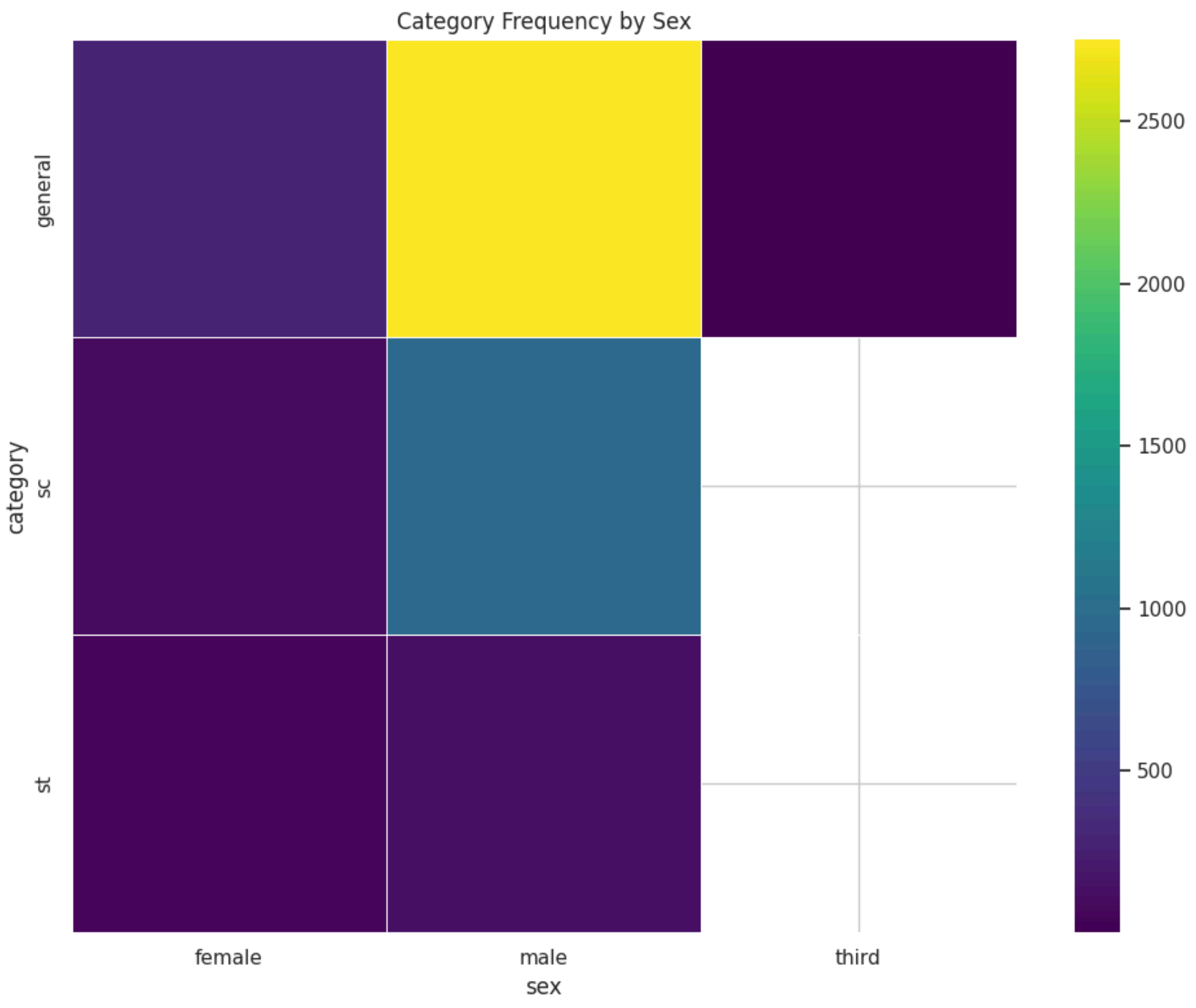


```python
# Total Votes by each Constituency in 2019 using barplot

# Filter the DataFrame to include only 2019 data
election_results_2019 = election_results[election_results['year'] == 2019]

plt.figure(figsize=(20, 10))
sns.barplot(
    x='ac_name',
    y='total',
    data=election_results_2019,
    color='red',
    ci=None
)

plt.xticks(rotation=90, fontsize=10)
plt.title('Total Votes by Constituency (2019)')
plt.xlabel('Constituency Name', fontsize=10, fontweight='bold')
plt.ylabel('Total Votes')
sns.despine()
plt.tight_layout()
plt.show()
```

```
<ipython-input-370-7ecd11eb0578>:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
```



```python
# Candidates category frequency by sex using a Co-relation heat map
plt.figure(figsize=(10, 8))
df_2dhist = pd.DataFrame({
    x_label: grp['category'].value_counts()
    for x_label, grp in election_results.groupby('sex')
})
sns.heatmap(df_2dhist, cmap='viridis', linewidths=0.5)
plt.title('Category Frequency by Sex')
plt.xlabel('sex')
plt.ylabel('category')
plt.tight_layout()
plt.show()
```



```
# unique names of constituencies in election dataset and map shapefile
unique_ac_names_election = election_results['ac_name'].unique()
unique_ac_names_andhra_map = andhra_map['assem_name'].unique()
```

```
print(unique_ac_names_election)
```

```
['ichchapuram' 'palasa' 'tekkali' 'pathapatnam' 'srikakulam'
 'amadalavalasa' 'etcherla' 'narasannapeta' 'rajam (sc)' 'palakonda (st)'
 'kurupam (st)' 'parvathipuram (sc)' 'salur (st)' 'bobbili'
 'cheepurupalle' 'gajapathinagaram' 'nellimarla' 'vizianagarm'
 'srungavarapukota' 'bhimili' 'vishakapatnam east' 'vishakapatnam south'
 'vishakapatnam north' 'vishakapatnam west' 'gajuwaka' 'chodavarm'
 'madugula' 'araku valley (st)' 'paderu (st)' 'anakapalle' 'pendurthi'
 'yelamanchili' 'payakaraopet (sc)' 'narsipatnam' 'tuni' 'prathipadu'
 'pithapuram' 'kakinada rural' 'peddapuram' 'anaparthy' 'kakinada urban'
 'ramachandrapuram' 'mummidivaram' 'amalapuram (sc)' 'razole (sc)'
 'gangavaram' 'kothapeta' 'mandapeta' 'rajanagaram' 'rajahmundry urban'
 'rajahmundry rural' 'jaggampeta' 'rampachodavaram (st)' 'kovvur (sc)'
 'nidadavole' 'achanta' 'palacole' 'narsapuram' 'bhimavaram' 'undi'
 'tanuku' 'tadepalligudem' 'ungutur' 'denduluru' 'eluru'
 'gopalapuram (sc)' 'polavaram (st)' 'chintalapudi (sc)' 'tiruvuru (sc)'
 'nuzvid' 'gudivada' 'kaikalur' 'pedana' 'machilipatnam' 'avanigadda'
 'pamarru (sc)' 'penamaluru' 'vijayawada west' 'vijayawada central'
 'vijayawada east' 'mylavaram' 'nandigama (sc)' 'jaggayyapeta'
 'pedakurapadu' 'tadikonda (sc)' 'mangalagiri' 'ponnur' 'vemuru (sc)'
 'repalle' 'tenali' 'bapatla' 'prathipadu (sc)' 'guntur west'
 'guntur east' 'chilakaluripet' 'narasaraopet' 'sattenapalle' 'vinukonda'
 'gurazala' 'macherla' 'yerragondapalem (sc)' 'darsi' 'parchur' 'addanki'
 'chirala' 'santanuthalapadu (sc)' 'ongole' 'kandukur' 'kondapi (sc)'
 'markapuram' 'giddalur' 'kanigiri' 'kavali' 'atmakur' 'kovuru'
 'nellore urban' 'nellore rural' 'sarvepalli' 'gudur (sc)'
 'sullurupeta (sc)' 'venkatagiri' 'udayagiri' 'badvel (sc)' 'rajampet'
 'ysr kadapa' 'kodur (sc)' 'rayachoti' 'pulivendla' 'kamalapuram'
 'jammalamadugu' 'proddatur' 's.mydukur' 'allagadda' 'srisailam'
 'nandikotkur (sc)' 'kurnool' 'panyam' 'nandyal' 'banaganapalle' 'dhone'
 'pattikonda' 'kodumuru (sc)' 'yemmiganur' 'mantralayam' 'adoni' 'alur'
 'rayadurg' 'uravakonda' 'guntakal' 'tadipatri' 'singanamala (sc)'
 'anantapur urban' 'kalyandurg' 'raptadu' 'madakasira (sc)' 'hindupur'
 'penukonda' 'puttaparthi' 'dharmavaram' 'kadiri' 'thamallapalle' 'pileru'
 'madanapalle' 'punganur' 'chandragiri' 'tirupati' 'srikalahasti'
 'satyavedu (sc)' 'nagari' 'gangadharanellore (sc)' 'chittoor'
 'puthalapattu (sc)' 'palamaner' 'kuppam']
```

```
print(unique_ac_names_andhra_map)
```

```
['ichchapuram' 'palasa' 'tekkali' 'pathapatnam' 'srikakulam'
 'amadalavalasa' 'etcherla' 'narasannapeta' 'rajam (sc)' 'palakonda (st)'
 'kurupam (st)' 'parvathipuram (sc)' 'salur (st)' 'bobbili'
 'cheepurupalle' 'gajapathinagaram' 'nellimarla' 'vizianagarm'
 'srungavarapukota' 'bhimili' 'vishakapatnam east' 'vishakapatnam south'
 'vishakapatnam north' 'vishakapatnam west' 'gajuwaka' 'chodavarm'
 'madugula' 'araku valley (st)' 'paderu (st)' 'anakapalle' 'pendurthi'
 'yelamanchili' 'payakaraopet (sc)' 'narsipatnam' 'tuni' 'prathipadu'
 'pithapuram' 'kakinada rural' 'peddapuram' 'anaparthy' 'kakinada urban'
 'ramachandrapuram' 'mummidivaram' 'amalapuram (sc)' 'razole (sc)'
 'gannavaram (sc)' 'kothapeta' 'mandapeta' 'rajanagaram'
 'rajahmundry urban' 'rajahmundry rural' 'jaggampeta'
 'rampachodavaram (st)' 'kovvur (sc)' 'nidadavole' 'achanta' 'palacole'
 'narsapuram' 'bhimavaram' 'undi' 'tanuku' 'tadepalligudem' 'ungutur'
 'denduluru' 'eluru' 'gopalapuram (sc)' 'polavaram (st)'
 'chintalapudi (sc)' 'tiruvuru (sc)' 'nuzvid' 'gangavaram' 'gudivada'
 'kaikalur' 'pedana' 'machilipatnam' 'avanigadda' 'pamarru (sc)'
 'penamaluru' 'vijayawada west' 'vijayawada central' 'vijayawada east'
 'mylavaram' 'nandigama (sc)' 'jaggayyapeta' 'pedakurapadu'
 'tadikonda (sc)' 'mangalagiri' 'ponnur' 'vemuru (sc)' 'repalle' 'tenali'
 'bapatla' 'prathipadu (sc)' 'guntur west' 'guntur east' 'chilakaluripet'
 'narasaraopet' 'sattenapalle' 'vinukonda' 'gurazala' 'macherla'
 'yerragondapalem (sc)' 'darsi' 'parchur' 'addanki' 'chirala'
 'santanuthalapadu (sc)' 'ongole' 'kandukur' 'kondapi (sc)' 'markapuram'
 'giddalur' 'kanigiri' 'kavali' 'atmakur' 'kovuru' 'nellore urban'
 'nellore rural' 'sarvepalli' 'gudur (sc)' 'sullurupeta (sc)'
 'venkatagiri' 'udayagiri' 'badvel (sc)' 'rajampet' 'ysr kadapa'
 'kodur (sc)' 'rayachoti' 'pulivendla' 'kamalapuram' 'jammalamadugu'
 'proddatur' 's.mydukur' 'allagadda' 'srisailam' 'nandikotkur (sc)'
 'kurnool' 'panyam' 'nandyal' 'banaganapalle' 'dhone' 'pattikonda'
 'kodumuru (sc)' 'yemmiganur' 'mantralayam' 'adoni' 'alur' 'rayadurg'
 'uravakonda' 'guntakal' 'tadipatri' 'singanamala (sc)' 'anantapur urban'
 'kalyandurg' 'raptadu' 'madakasira (sc)' 'hindupur' 'penukonda'
 'puttaparthi' 'dharmavaram' 'kadiri' 'thamallapalle' 'pileru'
 'madanapalle' 'punganur' 'chandragiri' 'tirupati' 'srikalahasti'
 'satyavedu (sc)' 'nagari' 'gangadharanellore (sc)' 'chittoor'
 'puthalapattu (sc)' 'palamaner' 'kuppam']
```

```
# mismatched constituency names from the dataset and map shapefile
mismatched_ac_names_election = set(unique_ac_names_election) - set(unique_ac_names_andhra_map)
mismatched_ac_names_andhra_map = set(unique_ac_names_andhra_map) - set(unique_ac_names_election)
```

```
print(sorted(mismatched_ac_names_andhra_map))
```

```
['gannavaram (sc)']
```

```
print(sorted(mismatched_ac_names_election))
```

```
[]
```

```
# Filter election_results for the year 2014
election_results_2014 = election_results[election_results['year'] == 2014]

election_results_2014.to_csv('election_results_2014.csv', index=False)
```

```
# Filter election_results for the year 2019
election_results_2019 = election_results[election_results['year'] == 2019]

election_results_2019.to_csv('election_results_2019.csv', index=False)
```

```
import matplotlib.patches as mpatches
```

```
winning_party = (
   election_results_2019.groupby('ac_name')
    .apply(lambda x: x.loc[x['percentage_votes_polled'].idxmax()])
    .reset_index(drop=True)
)

# Create a Dictionary for Data Filling
party_dict = pd.Series(winning_party['party'].values, index=winning_party['ac_name']).fillna('error').to_dict()

# Fill Data in GeoDataFrame
andhra_map['party'] = andhra_map['assem_name'].map(party_dict)

# Map Party Names to Colors
party_colors = {
    'tdp': 'yellow',
    'ysrcp': 'blue',
    'bjp': 'orange',
    'inc': 'green',
    'jnp': 'red',
    'ind': 'grey',
    'error': 'black' # error if no data found or shows error
}
andhra_map['color'] = andhra_map['party'].map(party_colors).fillna('black')
```

```
<ipython-input-381-45677cd11873>:3: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a
  .apply(lambda x: x.loc[x['percentage_votes_polled'].idxmax()])
```

```
# constituency wise winning party
print(party_dict)
```

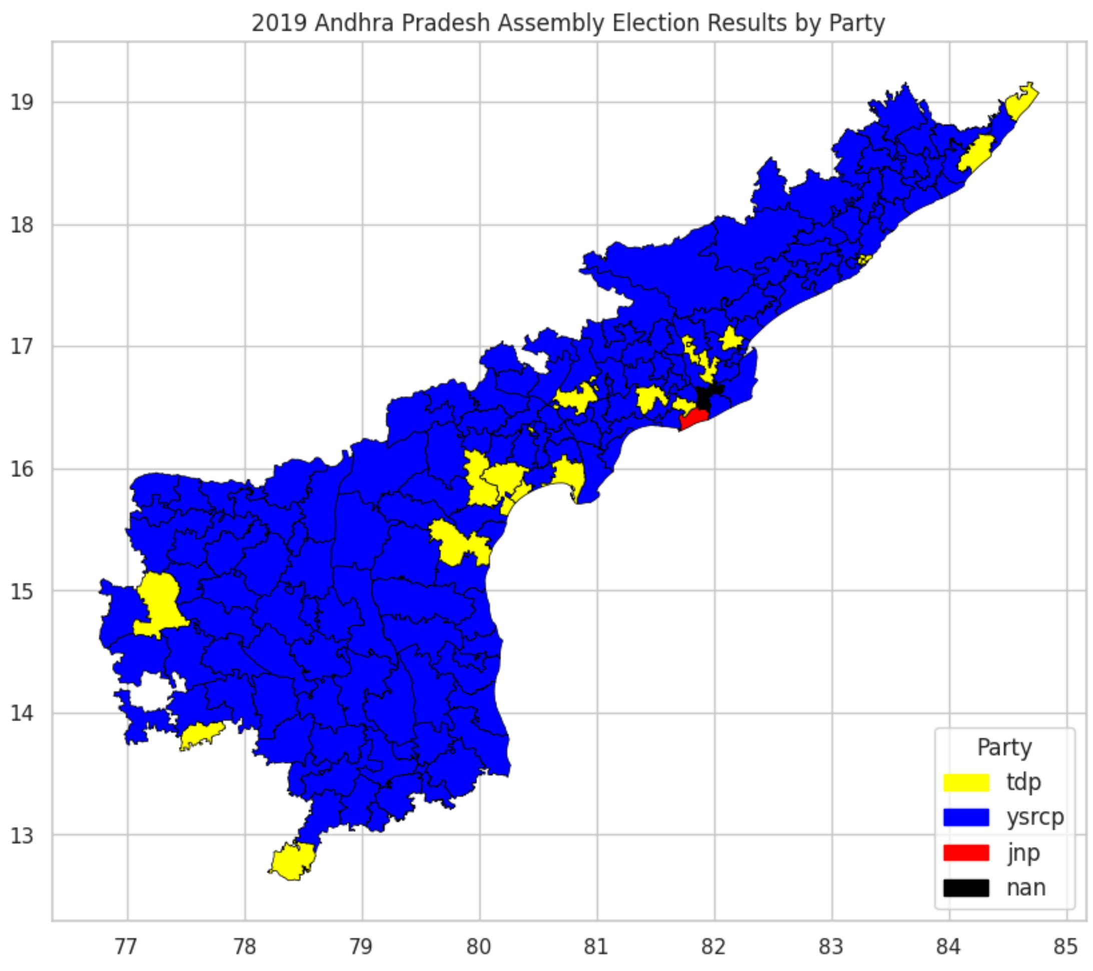
```
{'achanta': 'ysrcp', 'addanki': 'tdp', 'adoni': 'ysrcp', 'allagadda': 'ysrcp', 'alur': 'ysrcp', 'amadalavalasa': 'ysrcp', 'amalapuram (sc)': 'ysrcp',
```

```
andhra_map.head()
```

	objectid	assem_name	shape_leng	type	district	naaa	st_area(shape)	st_length(shape)	geometry	party	color
0	1	ichchapuram	5.001582e+08	gen	srikakulam	120	0.042896	1.410057	POLYGON ((84.57492 18.84095, 84.57491 18.84105...	tdp	yellow
1	2	palasa	5.105838e+08	gen	srikakulam	121	0.043754	1.743430	POLYGON ((84.47398 18.98993, 84.47401 18.9901,...	ysrcp	blue
2	3	tekkali	6.512106e+08	gen	srikakulam	122	0.055752	1.598741	POLYGON ((84.31164 18.73493, 84.31164 18.73493...	tdp	yellow
									POLYGON ((84.31164 18.73493,		

Next steps: Generate code with andhra_map | View recommended plots | New interactive sheet

```
# Map of 2019 Andhra Pradesh Assembly Election Results by party
fig, ax = plt.subplots(1, 1, figsize=(10, 10))
andhra_map.plot(ax=ax, color=andhra_map['color'], edgecolor='black', linewidth=0.5)

unique_parties = andhra_map[['party', 'color']].drop_duplicates()
legend_handles = [
    mpatches.Patch(color=row['color'], label=row['party']) for _, row in unique_parties.iterrows()
]

ax.legend(handles=legend_handles, title="Party", loc="lower right", fontsize='medium')
plt.title("2019 Andhra Pradesh Assembly Election Results by Party")
plt.show()

# one NaN (error) which shows black in the map
```



```
# Load election results data
df = pd.read_csv("/content/drive/MyDrive/AP elections/election_results_2019.csv")

# Find the winning party in each constituency
winners = df.loc[df.groupby("ac_name")["total"].idxmax(), "party"]

# Count constituencies won by each party
party_wins = winners.value_counts()

print("Total constituencies won by each party:")
print(party_wins)
```

```
Total constituencies won by each party:
party
ysrcp    150
tdp       23
jnp        1
Name: count, dtype: int64
```