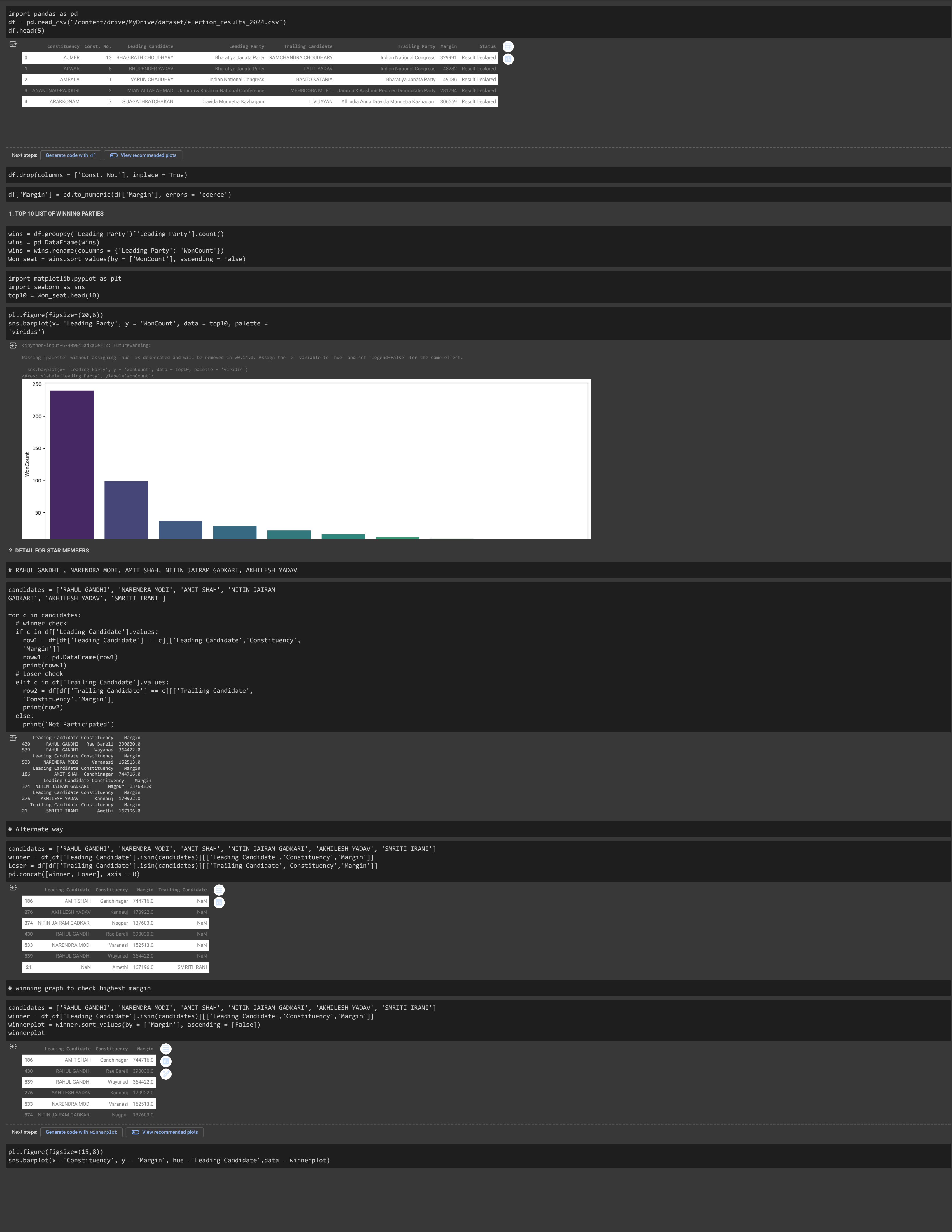


```
import pandas as pd
df = pd.read_csv("/content/drive/MyDrive/dataset/election_results_2024.csv")
df.head(5)
```

	Constituency	Const. No.	Leading Candidate	Leading Party	Trailing Candidate	Trailing Party	Margin	Status
0	AJMER	13	BHAGIRATH CHOUDHARY	Bharatiya Janata Party	RAMCHANDRA CHOUDHARY	Indian National Congress	329991	Result Declared
1	ALWAR	8	BHUPENDER YADAV	Bharatiya Janata Party	LALIT YADAV	Indian National Congress	48282	Result Declared
2	AMBALA	1	VARUN CHAUDHRY	Indian National Congress	BANTO KATARIA	Bharatiya Janata Party	49036	Result Declared
3	ANANTNAG-RAJOURI	3	MIAN ALTAF AHMAD	Jammu & Kashmir National Conference	MEHBOOBA MUFTI	Jammu & Kashmir Peoples Democratic Party	281794	Result Declared
4	ARAKKONAM	7	S JAGATHRATCHAKAN	Dravida Munnetra Kazhagam	L VIJAYAN	All India Anna Dravida Munnetra Kazhagam	306559	Result Declared

```
df.drop(columns = ['Const. No.'], inplace = True)
```

```
df['Margin'] = pd.to_numeric(df['Margin'], errors = 'coerce')
```

1. TOP 10 LIST OF WINNING PARTIES

```
wins = df.groupby('Leading Party')['Leading Party'].count()
wins = pd.DataFrame(wins)
wins = wins.rename(columns = {'Leading Party': 'WonCount'})
Won_seat = wins.sort_values(by = ['WonCount'], ascending = False)
```

```
import matplotlib.pyplot as plt
import seaborn as sns
top10 = Won_seat.head(10)
```

```
plt.figure(figsize=(20,6))
sns.barplot(x= 'Leading Party', y = 'WonCount', data = top10, palette =
'viridis')
```

```
<ipython-input-6-409845ad2a6e>:2: 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 same effect.

  sns.barplot(x= 'Leading Party', y = 'WonCount', data = top10, palette = 'viridis')
<Axes: xlabel='Leading Party', ylabel='WonCount'>
```

2. DETAIL FOR STAR MEMBERS

```
# RAHUL GANDHI , NARENDRA MODI, AMIT SHAH, NITIN JAIRAM GADKARI, AKHILESH YADAV
```

```
candidates = ['RAHUL GANDHI', 'NARENDRA MODI', 'AMIT SHAH', 'NITIN JAIRAM
GADKARI', 'AKHILESH YADAV', 'SMRITI IRANI']

for c in candidates:
  # winner check
  if c in df['Leading Candidate'].values:
    row1 = df[df['Leading Candidate'] == c][['Leading Candidate','Constituency',
    'Margin']]
    roww1 = pd.DataFrame(row1)
    print(roww1)
  # Loser check
  elif c in df['Trailing Candidate'].values:
    row2 = df[df['Trailing Candidate'] == c][['Trailing Candidate',
    'Constituency','Margin']]
    print(row2)
  else:
    print('Not Participated')
```

```
    Leading Candidate Constituency    Margin
430      RAHUL GANDHI   Rae Bareli  390030.0
539      RAHUL GANDHI      Wayanad  364422.0
    Leading Candidate Constituency    Margin
533     NARENDRA MODI     Varanasi  152513.0
    Leading Candidate Constituency    Margin
186         AMIT SHAH   Gandhinagar  744716.0
        Leading Candidate Constituency    Margin
374  NITIN JAIRAM GADKARI       Nagpur  137603.0
    Leading Candidate Constituency    Margin
276    AKHILESH YADAV      Kannauj  170922.0
   Trailing Candidate Constituency    Margin
21       SMRITI IRANI       Amethi  167196.0
```

```
# Alternate way
```

```
candidates = ['RAHUL GANDHI', 'NARENDRA MODI', 'AMIT SHAH', 'NITIN JAIRAM GADKARI', 'AKHILESH YADAV', 'SMRITI IRANI']
winner = df[df['Leading Candidate'].isin(candidates)][['Leading Candidate','Constituency','Margin']]
Loser = df[df['Trailing Candidate'].isin(candidates)][['Trailing Candidate','Constituency','Margin']]
pd.concat([winner, Loser], axis = 0)
```

	Leading Candidate	Constituency	Margin	Trailing Candidate
186	AMIT SHAH	Gandhinagar	744716.0	NaN
276	AKHILESH YADAV	Kannauj	170922.0	NaN
374	NITIN JAIRAM GADKARI	Nagpur	137603.0	NaN
430	RAHUL GANDHI	Rae Bareli	390030.0	NaN
533	NARENDRA MODI	Varanasi	152513.0	NaN
539	RAHUL GANDHI	Wayanad	364422.0	NaN
21	NaN	Amethi	167196.0	SMRITI IRANI

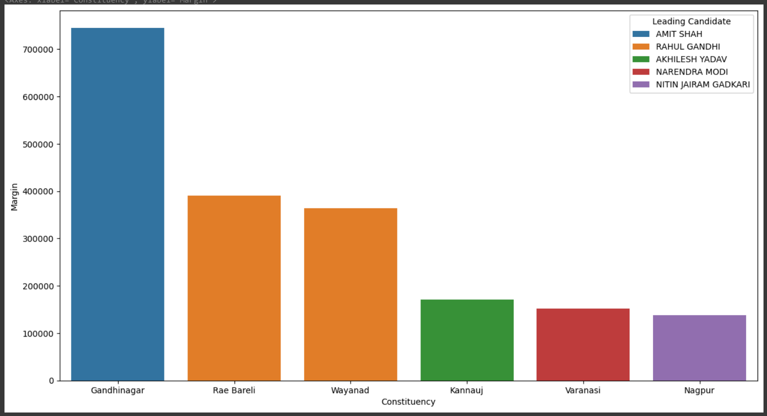
```
# winning graph to check highest margin
```

```
candidates = ['RAHUL GANDHI', 'NARENDRA MODI', 'AMIT SHAH', 'NITIN JAIRAM GADKARI', 'AKHILESH YADAV', 'SMRITI IRANI']
winner = df[df['Leading Candidate'].isin(candidates)][['Leading Candidate','Constituency','Margin']]
winnerplot = winner.sort_values(by = ['Margin'], ascending = [False])
winnerplot
```

	Leading Candidate	Constituency	Margin
186	AMIT SHAH	Gandhinagar	744716.0
430	RAHUL GANDHI	Rae Bareli	390030.0
539	RAHUL GANDHI	Wayanad	364422.0
276	AKHILESH YADAV	Kannauj	170922.0
533	NARENDRA MODI	Varanasi	152513.0
374	NITIN JAIRAM GADKARI	Nagpur	137603.0

```
plt.figure(figsize=(15,8))
sns.barplot(x ='Constituency', y = 'Margin', hue ='Leading Candidate',data = winnerplot)
```

```
<Axes: xlabel='Constituency', ylabel='Margin'>
```



3. Highest and lowest margin victory candidate

```
# Candidate Name | Margin -> cond: Name from leading list
```

```
df['Margin'] = pd.to_numeric(df['Margin'], errors = 'coerce')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 543 entries, 0 to 542
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Constituency       543 non-null    object 
 1   Leading Candidate  543 non-null    object 
 2   Leading Party      543 non-null    object 
 3   Trailing Candidate 542 non-null    object 
 4   Trailing Party     542 non-null    object 
 5   Margin             542 non-null    float64
 6   Status             543 non-null    object 
dtypes: float64(1), object(6)
memory usage: 29.8+ KB
```

```
Victory = df[['Leading Candidate', 'Leading Party', 'Margin']]
Victory = Victory.sort_values(by=['Margin'], ascending=[False])
print(Victory)
```

```
                 Leading Candidate            Leading Party     Margin
221                 SHANKAR LALWANI    Bharatiya Janata Party  1175092.0
157                 RAKIBUL HUSSAIN  Indian National Congress  1012476.0
525           SHIVRAJ SINGH CHOUHAN    Bharatiya Janata Party   821408.0
384                       C R PATIL    Bharatiya Janata Party   773551.0
186                       AMIT SHAH    Bharatiya Janata Party   744716.0
..                              ...                       ...        ...
226             RAO RAJENDRA SINGH    Bharatiya Janata Party     1615.0
236         RABINDRA NARAYAN BEHERA    Bharatiya Janata Party     1587.0
40                ADV ADOOR PRAKASH  Indian National Congress      684.0
357         RAVINDRA DATTARAM WAIKAR                 Shiv Sena       48.0
493  MUKESHKUMAR CHANDRAKAANT DALAL    Bharatiya Janata Party        NaN

[543 rows x 3 columns]
```

```
Ask = str(input('Want to see Top or Bottom list?: T/B'))
if(Ask == 'T'):
  a= int(input('Enter the top value to display'))
  print(Victory.head(a))
else:
  b = int(input('Enter the bottom value to display'))
  print(Victory.tail(b))
```

```
Want to see Top or Bottom list?: T/BB
Enter the bottom value to display5
                 Leading Candidate            Leading Party  Margin
226             RAO RAJENDRA SINGH    Bharatiya Janata Party  1615.0
236         RABINDRA NARAYAN BEHERA    Bharatiya Janata Party  1587.0
40                ADV ADOOR PRAKASH  Indian National Congress   684.0
357         RAVINDRA DATTARAM WAIKAR                 Shiv Sena    48.0
493  MUKESHKUMAR CHANDRAKAANT DALAL    Bharatiya Janata Party     NaN
```

```
# This is formatted as code
```

4. Vote % distribution by party

```
# Leading Party details
```

```
# Total Votes by party
# sum(Votes) filtered by leading and trailing party
# graph: x: sum(votes by party) y: Margin column
```

```
leadingvotes = df.groupby('Leading Party')['Margin'].sum()
leadingvotes = pd.DataFrame(leadingvotes)
leadingvotes = leadingvotes.sort_values(by = 'Margin', ascending = False)
```

```
def vote(x, leadingvotes, margin_col):
  per = x / leadingvotes[margin_col].sum()
  per = per*100
  return(per)
```

```
leadingvotes['Perct_Vote'] = leadingvotes['Margin'].apply(lambda x: vote(x, leadingvotes,'Margin'))
leadingvotes = leadingvotes.reset_index()
```

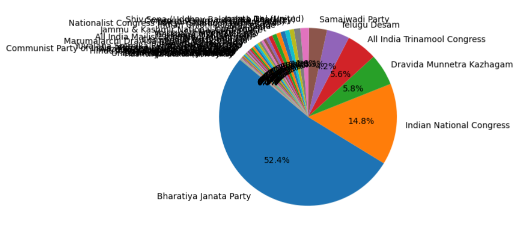
```
lead_top10 = leadingvotes.head(10).reset_index()
lead_top10
```

	Leading Party	Margin	Perct_Vote
0	Bharatiya Janata Party	46081536.0	52.369104
1	Indian National Congress	13005922.0	14.780507
2	Dravida Munnetra Kazhagam	5081582.0	5.774935
3	All India Trinamool Congress	4910963.0	5.581036
4	Telugu Desam	3660927.0	4.160440
5	Samajwadi Party	2869566.0	3.261102
6	Janata Dal (United)	1386413.0	1.575581
7	Shiv Sena (Uddhav Balasaheb Thackrey)	1015430.0	1.153980
8	Communist Party of India (Marxist)	746237.0	0.848057
9	Lok Janshakti Party(Ram Vilas)	720603.0	0.818925

Next steps: Generate code with `lead_top10` View recommended plots

```
plt.pie(leadingvotes['Margin'], labels=leadingvotes['Leading Party'], autopct='%1.1f%%', startangle=140)
```

```
([<matplotlib.patches.Wedge at 0x7a347b9b6c80>,
  <matplotlib.patches.Wedge at 0x7a347b9b7670>,
  <matplotlib.patches.Wedge at 0x7a347b03b070>,
  <matplotlib.patches.Wedge at 0x7a347b03bb20>,
  <matplotlib.patches.Wedge at 0x7a347b0826b0>,
  <matplotlib.patches.Wedge at 0x7a347bc180a0>,
  <matplotlib.patches.Wedge at 0x7a347af96aa0>,
  <matplotlib.patches.Wedge at 0x7a347af97130>,
  <matplotlib.patches.Wedge at 0x7a347dcf2fe0>,
  <matplotlib.patches.Wedge at 0x7a347dd85e10>,
  <matplotlib.patches.Wedge at 0x7a347b9b7310>,
  <matplotlib.patches.Wedge at 0x7a347b7faf80>,
  <matplotlib.patches.Wedge at 0x7a347b7fbd00>,
  <matplotlib.patches.Wedge at 0x7a347b7f8490>,
  <matplotlib.patches.Wedge at 0x7a347b7f8940>,
  <matplotlib.patches.Wedge at 0x7a347b7f90f0>,
  <matplotlib.patches.Wedge at 0x7a347b7f9a50>,
  <matplotlib.patches.Wedge at 0x7a347b7fa410>,
  <matplotlib.patches.Wedge at 0x7a347b7fb5e0>,
  <matplotlib.patches.Wedge at 0x7a347b7fa020>,
  <matplotlib.patches.Wedge at 0x7a347b97b7f0>,
  <matplotlib.patches.Wedge at 0x7a347b97bd60>,
  <matplotlib.patches.Wedge at 0x7a347b979300>,
  <matplotlib.patches.Wedge at 0x7a347b979b10>,
  <matplotlib.patches.Wedge at 0x7a347b979f60>,
  <matplotlib.patches.Wedge at 0x7a347b978a60>,
  <matplotlib.patches.Wedge at 0x7a347b978880>,
  <matplotlib.patches.Wedge at 0x7a347b97a5f0>,
  <matplotlib.patches.Wedge at 0x7a347b97bee0>,
  <matplotlib.patches.Wedge at 0x7a347b978d00>,
  <matplotlib.patches.Wedge at 0x7a347b8f9e70>,
  <matplotlib.patches.Wedge at 0x7a347b8fa1d0>,
  <matplotlib.patches.Wedge at 0x7a347b8faa10>,
  <matplotlib.patches.Wedge at 0x7a347b8face0>,
  <matplotlib.patches.Wedge at 0x7a347b8f9120>,
  <matplotlib.patches.Wedge at 0x7a347b8f9630>,
  <matplotlib.patches.Wedge at 0x7a347b8f8d60>,
  <matplotlib.patches.Wedge at 0x7a347b8f8670>,
  <matplotlib.patches.Wedge at 0x7a347b8fbb50>,
  <matplotlib.patches.Wedge at 0x7a347ba76ef0>,
  <matplotlib.patches.Wedge at 0x7a347ba77640>,
  <matplotlib.patches.Wedge at 0x7a347ba77c40>],
 [Text(-0.6424502253777664, -0.8928928871438373, 'Bharatiya Janata Party'),
  Text(1.0960349320954124, -0.0933135982941661, 'Indian National Congress'),
  Text(0.9314919643182213, 0.5850835157570085, 'Dravida Munnetra Kazhagam'),
  Text(0.6685057374489538, 0.8735559964866767, 'All India Trinamool Congress'),
  Text(0.3742562575134179, 1.0343752963562358, 'Telugu Desam'),
  Text(0.12513946205698365, 1.0928586894178447, 'Samajwadi Party'),
  Text(-0.04172282580950722, 1.0992084451124224, 'Janata Dal (United)'),
  Text(-0.13571303625155437, 1.091596066222018, 'Shiv Sena (Uddhav Balasaheb Thackrey)'),
  Text(-0.20405628890078875, 1.080907503424802, 'Communist Party of India (Marxist)'),
  Text(-0.26035756175670727, 1.0687440947374178, 'Lok Janshakti Party(Ram Vilas)'),
  Text(-0.31482879200047126, 1.0539842654079443, 'Nationalist Congress Party – Sharadchandra Pawar'),
  Text(-0.36780729372802384, 1.0366859672439226, 'Indian Union Muslim League'),
  Text(-0.4192536203566675, 1.016969223633551, 'Shiv Sena'),
  Text(-0.467041391334078, 0.995927878302806, 'Independent'),
  Text(-0.5057669747399651, 0.9768314937912187, 'Jammu & Kashmir National Conference'),
  Text(-0.5376723372127914, 0.9596397541766047, 'Janasena Party'),
  Text(-0.5656866223418561, 0.9433973952176582, 'Voice of the People Party'),
  Text(-0.5904472457776683, 0.9281013144875757, 'Jharkhand Mukti Morcha'),
  Text(-0.6142765604692338, 0.9125044149252582, 'Janata Dal (Secular)'),
  Text(-0.6366983227170335, 0.8970034815146017, 'All India Majlis-E-Ittehadul Muslimeen'),
  Text(-0.6581300811156953, 0.8813993398741846, 'Rashtriya Janata Dal'),
  Text(-0.6790806516002242, 0.8653608892376723, 'Communist Party of India'),
  Text(-0.698916792840968, 0.8494205770317174, 'Marumalarchi Dravida Munnetra Kazhagam'),
  Text(-0.715763173835823, 0.8352742537517062, 'Bharat Adivasi Party'),
  Text(-0.7298121045324322, 0.823027515990773, 'Aam Aadmi Party'),
  Text(-0.7429363729463676, 0.811200065183242, 'Asom Gana Parishad'),
  Text(-0.7551953084472228, 0.7998000038130184, 'Yuvajana Sramika Rythu Congress Party'),
  Text(-0.7665646851530886, 0.7889097435550826, 'Rashtriya Lok Dal'),
  Text(-0.7769529547672085, 0.778681004056542, 'Viduthalai Chiruthaigal Katch'),
  Text(-0.7863456468441316, 0.7691947241689092, 'Communist Party of India (Marxist-Leninist) (Liberation)'),
  Text(-0.7950043719863646, 0.7602420986255406, 'Aazad Samaj Party (Kanshi Ram)'),
  Text(-0.8031490025399447, 0.7516326760586531, 'Revolutionary Socialist Party'),
  Text(-0.8098818887368812, 0.7443731096002745, 'Hindustani Awam Morcha (Secular)'),
  Text(-0.8148883247919444, 0.7388890431707449, 'Kerala Congress'),
  Text(-0.8193592244093184, 0.7339281036827521, 'Nationalist Congress Party'),
  Text(-0.8236337062627429, 0.7291279160119286, 'AJSU Party'),
  Text(-0.8278295355930732, 0.7243605870005332, 'Sikkim Krantikari Morcha'),
  Text(-0.8316741938440565, 0.7199430778150722, 'Zoram People’s Movement'),
  Text(-0.834747700907996, 0.7163771882387205, 'United People’s Party, Liberal'),
  Text(-0.8373315744240184, 0.7133553353501989, 'Shiromani Akali Dal'),
  Text(-0.8396671398511325, 0.7106047384124448, 'Rashtriya Loktantrik Party'),
  Text(-0.8416942239600473, 0.7082025369569739, 'Apna Dal (Soneylal)')],
 [Text(-0.3504273956605998, -0.48703248389663845, '52.4%'),
  Text(0.597837235688406, -0.05089832634227241, '14.8%'),
  Text(0.5080865259917571, 0.3191364631401864, '5.8%'),
  Text(0.36463949315397476, 0.4764850889927327, '5.6%'),
  Text(0.20413977682550064, 0.564204707103401, '4.2%'),
  Text(0.06825788839471834, 0.5961047396824607, '3.3%'),
  Text(-0.022757904987003936, 0.599568242788594, '1.6%'),
  Text(-0.07402529250084783, 0.5954160361211007, '1.2%'),
  Text(-0.11130343030952113, 0.589585910958982, '0.8%'),
  Text(-0.1420132155036585, 0.5829513244022279, '0.8%'),
  Text(-0.171724795636620, 0.5749005084043333, '0.8%'),
  Text(-0.2006221602152857, 0.5654650730421396, '0.8%'),
  Text(-0.22868379292181862, 0.5547104856183005, '0.8%'),
  Text(-0.25474984981858795, 0.5432333881651669, '0.7%'),
  Text(-0.27587289531270826, 0.5328171784315737, '0.5%'),
  Text(-0.2932758202978862, 0.5234398659145116, '0.5%'),
  Text(-0.3085563394591942, 0.5145803973914499, '0.4%'),
  Text(-0.322062134060546, 0.5062370806295867, '0.4%'),
  Text(-0.3350599420741275, 0.49772968086832264, '0.4%'),
  Text(-0.347289994209291, 0.4892746262806918, '0.4%'),
  Text(-0.3589800442449247, 0.48076327629500976, '0.4%'),
  Text(-0.3704076281455768, 0.47201503049327576, '0.4%'),
  Text(-0.38122734154961885, 0.463320314744573, '0.4%'),
  Text(-0.3904162766377216, 0.4556041384100215, '0.3%'),
  Text(-0.39807932974496296, 0.4489240996313308, '0.3%'),
  Text(-0.40523802160710953, 0.4424727628272228, '0.3%'),
  Text(-0.4119247136984851, 0.43625454753437365, '0.2%'),
  Text(-0.4181261919016847, 0.43031440557549955, '0.2%'),
  Text(-0.42379252078211366, 0.4247350931217501, '0.2%'),
  Text(-0.4289158073695262, 0.41956075863758685, '0.2%'),
  Text(-0.43363874835619887, 0.4146775083412039, '0.2%'),
  Text(-0.43808127411269704, 0.4099814596683562, '0.2%'),
  Text(-0.44175375749284423, 0.4060216961456042, '0.1%'),
  Text(-0.44448454079560, 0.4030303871840426, '0.1%'),
  Text(-0.4469232133141736, 0.400324420190592, '0.1%'),
  Text(-0.449254748870587, 0.39770613600650645, '0.1%'),
  Text(-0.45154338305076713, 0.39510577472756353, '0.1%'),
  Text(-0.4536404693694853, 0.39269622426276657, '0.1%'),
  Text(-0.4553169277679977, 0.39075119358475663, '0.1%'),
  Text(-0.4567263133221918, 0.38910291019101756, '0.1%'),
  Text(-0.4580002581006176, 0.3876025845886062, '0.0%'),
  Text(-0.45910594034184393, 0.38629229288562206, '0.0%')])
```



```
# Trailing Party details
```

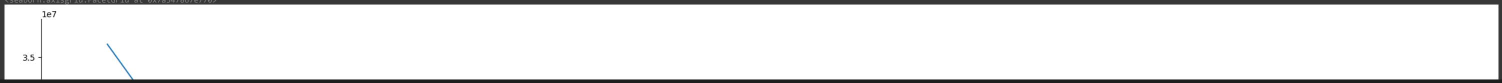
```
Trailing_votes = df.groupby('Trailing Party')['Margin'].sum().reset_index()
Trailing_votes = Trailing_votes.sort_values(by = 'Margin', ascending = False)

def trail(x,Trailing_votes,Margin_col):
  t = x/Trailing_votes['Margin'].sum()
  t = t*100
  return(t)

Trailing_votes['Tperc'] = Trailing_votes['Margin'].apply(lambda x : trail(x,Trailing_votes,'Margin'))
Trailplot = Trailing_votes.head(8)
```

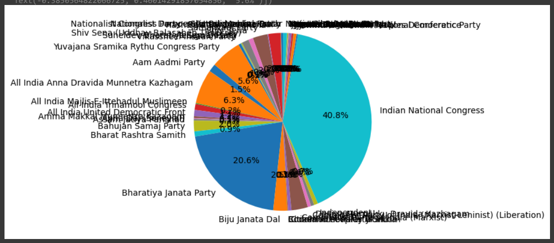
```
sns.relplot(x = 'Trailing Party', y = 'Margin', data = Trailplot, kind = 'line', height = 10, aspect = 2.5)
```

```
<seaborn.axisgrid.FacetGrid at 0x7a347867e770>
```



```
plt.pie(Trailing_votes['Margin'], labels = Trailing_votes['Trailing Party'], autopct = '%1.1f%%', startangle = 140)
```

```
([<matplotlib.patches.Wedge at 0x7a3478b30940>,
  <matplotlib.patches.Wedge at 0x7a3478b30820>,
  <matplotlib.patches.Wedge at 0x7a3478b31630>,
  <matplotlib.patches.Wedge at 0x7a3478b31cc0>,
  <matplotlib.patches.Wedge at 0x7a3478b32350>,
  <matplotlib.patches.Wedge at 0x7a3478b329e0>,
  <matplotlib.patches.Wedge at 0x7a3478b33070>,
  <matplotlib.patches.Wedge at 0x7a3478b33700>,
  <matplotlib.patches.Wedge at 0x7a3478b33d90>,
  <matplotlib.patches.Wedge at 0x7a3478b68460>,
  <matplotlib.patches.Wedge at 0x7a3478b30910>,
  <matplotlib.patches.Wedge at 0x7a3478b69150>,
  <matplotlib.patches.Wedge at 0x7a3478b697e0>,
  <matplotlib.patches.Wedge at 0x7a3478b69e70>,
  <matplotlib.patches.Wedge at 0x7a3478b6a500>,
  <matplotlib.patches.Wedge at 0x7a3478b6ab90>,
  <matplotlib.patches.Wedge at 0x7a3478b6b220>,
  <matplotlib.patches.Wedge at 0x7a3478b6b8b0>,
  <matplotlib.patches.Wedge at 0x7a3478b6bf40>,
  <matplotlib.patches.Wedge at 0x7a34789a4610>,
  <matplotlib.patches.Wedge at 0x7a34789a4ca0>,
  <matplotlib.patches.Wedge at 0x7a34789a5330>,
  <matplotlib.patches.Wedge at 0x7a34789a59c0>,
  <matplotlib.patches.Wedge at 0x7a34789a6050>,
  <matplotlib.patches.Wedge at 0x7a34789a66e0>,
  <matplotlib.patches.Wedge at 0x7a34789a6d70>,
  <matplotlib.patches.Wedge at 0x7a34789a7400>,
  <matplotlib.patches.Wedge at 0x7a34789a7a90>,
  <matplotlib.patches.Wedge at 0x7a34789e0160>,
  <matplotlib.patches.Wedge at 0x7a34789e07f0>,
  <matplotlib.patches.Wedge at 0x7a34789e0e80>,
  <matplotlib.patches.Wedge at 0x7a34789e1510>,
  <matplotlib.patches.Wedge at 0x7a34789e1ba0>,
  <matplotlib.patches.Wedge at 0x7a34789e2230>,
  <matplotlib.patches.Wedge at 0x7a34789e28c0>,
  <matplotlib.patches.Wedge at 0x7a34789e2f50>,
  <matplotlib.patches.Wedge at 0x7a34789e35e0>,
  <matplotlib.patches.Wedge at 0x7a34789e3c70>,
  <matplotlib.patches.Wedge at 0x7a3478a24340>,
  <matplotlib.patches.Wedge at 0x7a3478a249d0>,
  <matplotlib.patches.Wedge at 0x7a3478a25060>,
  <matplotlib.patches.Wedge at 0x7a3478a256f0>],
 [Text(-0.8740146635506384, 0.6679059573760849, 'Aam Aadmi Party'),
  Text(-1.0088966045029775, 0.4383236719849419, 'All India Anna Dravida Munnetra Kazhagam'),
  Text(-1.0762899504672534, 0.22715620731821864, 'All India Majlis-E-Ittehadul Muslimeen'),
  Text(-1.0842758312115466, 0.18532652764920032, 'All India Trinamool Congress'),
  Text(-1.0945888907403336, 0.1089732089453465, 'All India United Democratic Front'),
  Text(-1.0984480195486162, 0.05841188534641866, 'Amma Makkal Munnettra Kazagam'),
  Text(-1.0991824059123891, 0.04240328445594613, 'Apna Dal (Soneylal)'),
  Text(-1.0996839650452563, 0.026366209859286227, 'Assam Jatiya Parishad'),
  Text(-1.0987708772177822, -0.05198614602051076, 'Bahujan Samaj Party'),
  Text(-1.0895841416249195, -0.15101787417219065, 'Bharat Rashtra Samith'),
  Text(-0.7542307853303122, -0.8007096367972729, 'Bharatiya Janata Party'),
  Text(-0.03001297708272475, -1.099590478863214, 'Biju Janata Dal'),
  Text(0.05980470761006565б, -1.09837306820027, 'Bodoland Peoples Front'),
  Text(0.06499654477484235, -1.0980780706158064, 'Citizen Action Party-Sikkim'),
  Text(0.08898321391906507, -1.0963949961763935, 'Communist Party of India'),
  Text(0.2116028382045818, -1.0794555288958252, 'Communist Party of India (Marxist)'),
  Text(0.3320922541854173, -1.048672844461059, 'Communist Party of India (Marxist-Leninist) (Liberation)'),
  Text(0.3713832218953673, -1.035410306349428, 'Desiya Murpokku Dravida Kazhagam'),
  Text(0.41324732031979955, -1.0194246672748826, 'Independent'),
  Text(1.0924027467122939, 0.1290590522800947, 'Indian National Congress'),
  Text(0.18087823347829574, 1.0850267575750245, 'Jammu & Kashmir National Conference'),
  Text(0.15470520909001, 1.0890667097475786, 'Jammu & Kashmir Peoples Democratic Party'),
  Text(0.13473922527941218, 1.091716694555462, 'Janata Dal (Secular)'),
  Text(0.12231107617582947, 1.0931788511697025, 'Janata Dal (United)'),
  Text(0.09821584723960176, 1.0956065203123826, 'Jharkhand Mukti Morcha'),
  Text(0.08146627176847869, 1.096979145911236, 'Kerala Congress (M)'),
  Text(0.07537279815516444, 1.097414662421758, 'Mizo National Front'),
  Text(0.06934943135791663, 1.0978117581672797, 'Naga Peoples Front'),
  Text(0.059914460639317144, 1.0983670868185644, "National People's Party"),
  Text(0.029136350964705546, 1.0996140564090937, 'Nationalist Congress Party'),
  Text(-0.00764813429295091, 1.099973411515859, 'Nationalist Congress Party – Sharadchandra Pawar'),
  Text(-0.021729383503419013, 1.0997853581006438, 'Nationalist Democratic Progressive Party'),
  Text(-0.024567541914729738, 1.099725618454107, 'Pattali Makkal Katch'),
  Text(-0.10233813735385465, 1.0952291566804382, 'Rashtriya Janata Dal'),
  Text(-0.18396301817117852, 1.0845080027115295, 'Rashtriya Samaj Paksha'),
  Text(-0.27987100490045624, 1.0638008369126286, 'Samajwadi Party'),
  Text(-0.39633650488185385, 1.0261176223503992, 'Shiv Sena'),
  Text(-0.46443187185036017, 0.9971474496831303, 'Shiv Sena (Uddhav Balasaheb Thackrey)'),
  Text(-0.5099943757851546, 0.9746310772120448, 'Suheldev Bharatiya Samaj Party'),
  Text(-0.517829205138829, 0.9704910686375681, 'Telugu Desam'),
  Text(-0.5337772024732043, 0.9618117789463176, 'Vikassheel Insaan Party'),
  Text(-0.7059368840455663, 0.8435953507126724, 'Yuvajana Sramika Rythu Congress Party')],
 [Text(-0.47673527102762087, 0.3643123403869554, '1.5%'),
  Text(-0.5503072388198058, 0.23908563926451373, '6.3%'),
  Text(-0.5870672457094108, 0.12390338580993743, '0.2%'),
  Text(-0.5914231806608435, 0.1010871968995638, '1.1%'),
  Text(-0.5970484858583638, 0.05943993215200717, '1.2%'),
  Text(-0.599153465208336, 0.031861028370773814, '0.3%'),
  Text(-0.5995540395885759, 0.023129064248697883, '0.1%'),
  Text(-0.5998276172974124, 0.014381569014156123, '0.3%'),
  Text(-0.5993295693915175, -0.028356079647551317, '2.0%'),
  Text(-0.5943186227045015, -0.08237338591210398, '0.9%'),
  Text(-0.4113986101801702, -0.43675071098033064, '20.6%'),
  Text(-0.016370714772395316, -0.5997766248344802, '2.5%'),
  Text(0.03262074960549036, -0.5991125826546926, '0.1%'),
  Text(0.035452660786277644, -0.5989516748813489, '0.1%'),
  Text(0.04853629850130821, -0.5980336342780328, '0.6%'),
  Text(0.11541972992977187, -0.5887939248522682, '3.0%'),
  Text(0.18114122955568215, -0.572003369706032, '0.6%'),
  Text(0.20257266648838215, -0.5647692580087789, '0.6%'),
  Text(0.2254076292653452, -0.5560498185135723, '0.7%'),
  Text(0.5958560436612511, 0.07039584669823347, '40.8%'),
  Text(0.09866085462452494, 0.5918327768591043, '0.2%'),
  Text(0.08438465950364182, 0.5940363871350428, '0.5%'),
  Text(0.07349412287967937, 0.5954818333938883, '0.0%'),
  Text(0.06671513245954333, 0.5962793733652922, '0.3%'),
  Text(0.053572280312510046, 0.5976035565340269, '0.4%'),
  Text(0.04443614823735201, 0.5983522614061286, '0.1%'),
  Text(0.04111243535736241, 0.5985898158664134, '0.1%'),
  Text(0.03782696255886361, 0.5988064135457888, '0.1%'),
  Text(0.03268061489417298, 0.5991093200828533, '0.2%'),
  Text(0.015892555071657568, 0.5997894853140511, '0.7%'),
  Text(-0.004171709614336859, 0.5999854971904685, '0.3%'),
  Text(-0.011852391001864915, 0.5998829226003511, '0.1%'),
  Text(-0.0134004774080344, 0.5998503373386038, '0.0%'),
  Text(-0.05582080219301162, 0.5973977218256935, '2.2%'),
  Text(-0.10034346445700647, 0.5915498196608342, '0.2%'),
  Text(-0.1526569117638852, 0.5802550019523428, '2.7%'),
  Text(-0.2161835481173748, 0.5597005212820358, '0.9%'),
  Text(-0.2533264755547419, 0.543898608918071, '1.3%'),
  Text(-0.2781787504282661, 0.5316169512065698, '0.2%'),
  Text(-0.2824522937120885, 0.5293587647114006, '0.1%'),
  Text(-0.29115120134902045, 0.5246246066979914, '0.5%'),
  Text(-0.3850564822066725, 0.46014291857054856, '5.6%')])
```



ENTER THE CONSTITUEMCY TO GET CANDIDATES DETAILS

```
df['Constituency'] = df['Constituency'].astype(str)
df['Constituency'] = df['Constituency'].str.upper()
```

```
df[df['Constituency'] == 'AJMER'][['Leading Candidate','Leading Party',
'Margin']]
```

	Leading Candidate	Leading Party	Margin
0	BHAGIRATH CHOUDHARY	Bharatiya Janata Party	329991.0

```
Ask = str(input('Enter constituency name:__'))
Ask = Ask.upper()
if(Ask in df['Constituency'].values):
```