

```
In [1]: # Importing major Libraries

# Analysis by rahul badola-
#contact- contact.rahulbadola@gmail.com
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

Basic Composition of Data

```
In [2]: #loading data into 'df' variable
```

```
df=pd.read_csv('LS_2.0.xls')
```

```
In [3]: # First 5 rows
```

```
df.head()
```

```
Out[3]:
```

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL	nCASES	AGE	CATEGORY	EDUCATION	ASS
0	Telangana	ADILABAD	SOYAM BAPU RAO	1	BJP	Lotus	MALE	52	52.0		ST	12th Pass	30,99,4 ~ 30 L
1	Telangana	ADILABAD	Godam Nagesh	0	TRS	Car	MALE	0	54.0		ST	Post Graduate	1,84,77,8 ~ 1 Cr
2	Telangana	ADILABAD	RATHOD RAMESH	0	INC	Hand	MALE	3	52.0		ST	12th Pass	3,64,91,0 ~ 3 Cr
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	NaN	NaN		NaN	NaN	
4	Uttar Pradesh	AGRA	Satyapal Singh Baghel	1	BJP	Lotus	MALE	5	58.0		SC	Doctorate	7,42,74,0 ~ 7 Cr

```
In [4]: # Exploring the column names
```

```
df.columns
```

```
Out[4]: Index(['STATE', 'CONSTITUENCY', 'NAME', 'WINNER', 'PARTY', 'SYMBOL', 'GENDER',
'CRIMINAL\nCASES', 'AGE', 'CATEGORY', 'EDUCATION', 'ASSETS',
'LIABILITIES', 'GENERAL\nVOTES', 'POSTAL\nVOTES', 'TOTAL\nVOTES',
'OVER TOTAL ELECTORS \nIN CONSTITUENCY',
'OVER TOTAL VOTES POLLED \nIN CONSTITUENCY', 'TOTAL ELECTORS'],
dtype='object')
```

```
In [5]: df.dtypes
```

```
Out[5]: STATE                                object
CONSTITUENCY                             object
NAME                                      object
WINNER                                   int64
PARTY                                    object
SYMBOL                                  object
GENDER                                  object
CRIMINAL\nCASES                          object
AGE                                     float64
CATEGORY                                object
EDUCATION                               object
ASSETS                                  object
LIABILITIES                             object
GENERAL\nVOTES                           int64
POSTAL\nVOTES                           int64
TOTAL\nVOTES                             int64
OVER TOTAL ELECTORS \nIN CONSTITUENCY    float64
OVER TOTAL VOTES POLLED \nIN CONSTITUENCY float64
TOTAL ELECTORS                           int64
dtype: object
```

```
In [6]: # Exploring the unique values in Criminal cases Column
```

```
df['CRIMINAL\nCASES'].unique()
```

```
Out[6]: array(['52', '0', '3', nan, '5', '1', '6', '2', '4', '9', '7', '40', '8',
'24', '14', '28', 'Not Available', '12', '11', '13', '15', '204',
'10', '22', '31', '42', '16', '240', '41', '18'], dtype=object)
```

```
In [7]: # Replacing 'Not Available' with 0
df['CRIMINAL\nCASES']=np.where(df['CRIMINAL\nCASES']=='Not Available',0,df['CRIMINAL\nCASES'])
```

```
In [8]: # Shape- Rows & Column
df.shape
```

Out[8]: (2263, 19)

```
In [9]: # Checking Information
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2263 entries, 0 to 2262
Data columns (total 19 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   STATE                                2263 non-null   object
1   CONSTITUENCY                        2263 non-null   object
2   NAME                                2263 non-null   object
3   WINNER                              2263 non-null   int64
4   PARTY                              2263 non-null   object
5   SYMBOL                              2018 non-null   object
6   GENDER                             2018 non-null   object
7   CRIMINAL                            2018 non-null   object
CASES
8   AGE                                2018 non-null   float64
9   CATEGORY                           2018 non-null   object
10  EDUCATION                           2018 non-null   object
11  ASSETS                              2018 non-null   object
12  LIABILITIES                          2018 non-null   object
13  GENERAL                             2263 non-null   int64
VOTES
14  POSTAL                             2263 non-null   int64
VOTES
15  TOTAL                             2263 non-null   int64
VOTES
16  OVER TOTAL ELECTORS                 2263 non-null   float64
IN CONSTITUENCY
17  OVER TOTAL VOTES POLLED             2263 non-null   float64
IN CONSTITUENCY
18  TOTAL ELECTORS                       2263 non-null   int64
dtypes: float64(3), int64(5), object(11)
memory usage: 336.0+ KB
```

```
In [28]: # Using Describe to check the discription and detailed info
df.describe()
```

Out[28]:

	WINNER	CRIMINAL\nCASES	AGE	GENERAL\nVOTES	POSTAL\nVOTES	TOTAL\nVOTES	OVER TOTAL ELECTORS \nIN CONSTITUENCY	OVER TOTAL VOTES POLLED \nIN CONSTITUENCY
count	2018.000000	2018.000000	2018.000000	2.018000e+03	2018.000000	2.018000e+03	2018.000000	2018.000000
mean	0.267096	1.453915	52.273538	2.911903e+05	1105.111001	2.922954e+05	17.596810	25.808492
std	0.442553	7.636973	11.869373	2.545964e+05	1661.283371	2.555874e+05	14.886247	21.403985
min	0.000000	0.000000	25.000000	1.339000e+03	0.000000	1.342000e+03	0.097941	1.000039
25%	0.000000	0.000000	43.250000	3.047625e+04	97.000000	3.074375e+04	1.953617	2.870669
50%	0.000000	0.000000	52.000000	2.846300e+05	463.000000	2.855250e+05	18.036861	27.749894
75%	1.000000	1.000000	61.000000	5.058620e+05	1545.500000	5.076175e+05	30.708115	44.349684
max	1.000000	240.000000	86.000000	1.066824e+06	19367.000000	1.068569e+06	51.951012	74.411856

```
In [10]: # Checking Null Values
df.isnull().sum()
```

```
Out[10]: STATE 0
          CONSTITUENCY 0
          NAME 0
          WINNER 0
          PARTY 0
          SYMBOL 245
          GENDER 245
          CRIMINAL\nCASES 245
          AGE 245
          CATEGORY 245
          EDUCATION 245
          ASSETS 245
          LIABILITIES 245
          GENERAL\nVOTES 0
          POSTAL\nVOTES 0
          TOTAL\nVOTES 0
          OVER TOTAL ELECTORS \nIN CONSTITUENCY 0
          OVER TOTAL VOTES POLLED \nIN CONSTITUENCY 0
          TOTAL ELECTORS 0
          dtype: int64
```

```
In [11]: # Exploring Columns containing Null values

df[df.SYMBOL.isna()==True]
```

Out[11]:

	STATE	CONSTITUENCY	NAME	WINNER	PARTY	SYMBOL	GENDER	CRIMINAL\nCASES	AGE	CATEGORY	EDUCATION	ASSE
3	Telangana	ADILABAD	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14	Gujarat	AHMEDABAD WEST	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
39	West Bengal	ALIPURDUARS	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
46	Uttarakhand	ALMORA	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
54	Andhra Pradesh	AMALAPURAM	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...
2225	Tamil Nadu	VIRUDHUNAGAR	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2230	Andhra Pradesh	VISAKHAPATNAM	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2235	Andhra Pradesh	VIZIANAGARAM	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2241	Telangana	WARANGAL	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2262	Telangana	ZAHIRABAD	NOTA	0	NOTA	NaN	NaN	NaN	NaN	NaN	NaN	NaN

245 rows × 19 columns

```
In [12]: # Dropping Columns containing Null values

df=df.dropna()
```

```
In [13]: # Rechecking Null Values

df.isnull().sum()
```

```
Out[13]: STATE 0
          CONSTITUENCY 0
          NAME 0
          WINNER 0
          PARTY 0
          SYMBOL 0
          GENDER 0
          CRIMINAL\nCASES 0
          AGE 0
          CATEGORY 0
          EDUCATION 0
          ASSETS 0
          LIABILITIES 0
          GENERAL\nVOTES 0
          POSTAL\nVOTES 0
          TOTAL\nVOTES 0
          OVER TOTAL ELECTORS \nIN CONSTITUENCY 0
          OVER TOTAL VOTES POLLED \nIN CONSTITUENCY 0
          TOTAL ELECTORS 0
          dtype: int64
```

```
In [14]: # Reseting Index of Data frame

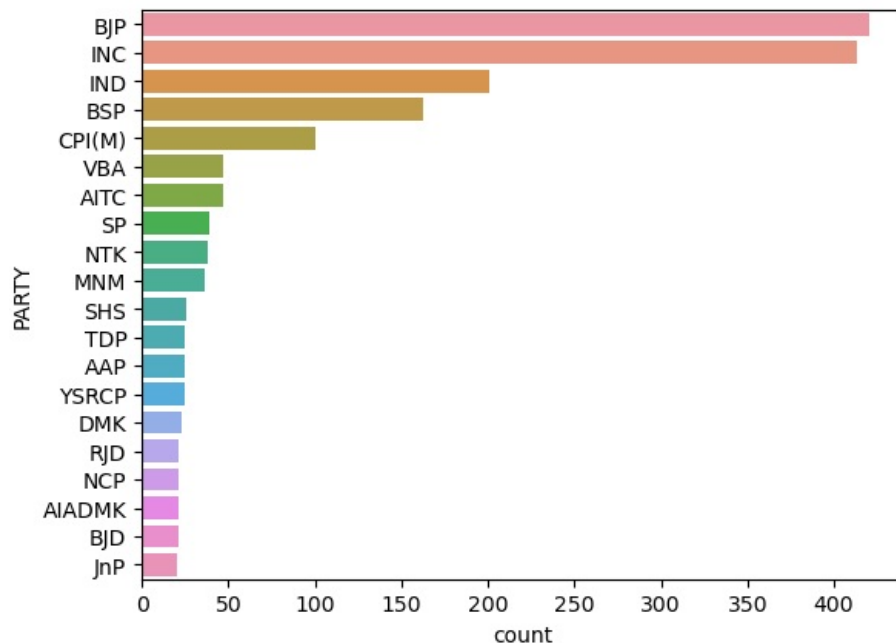
df.reset_index(drop=True,inplace=True)
```

EDA- Exploratory Data Analysis

Basic Analysis

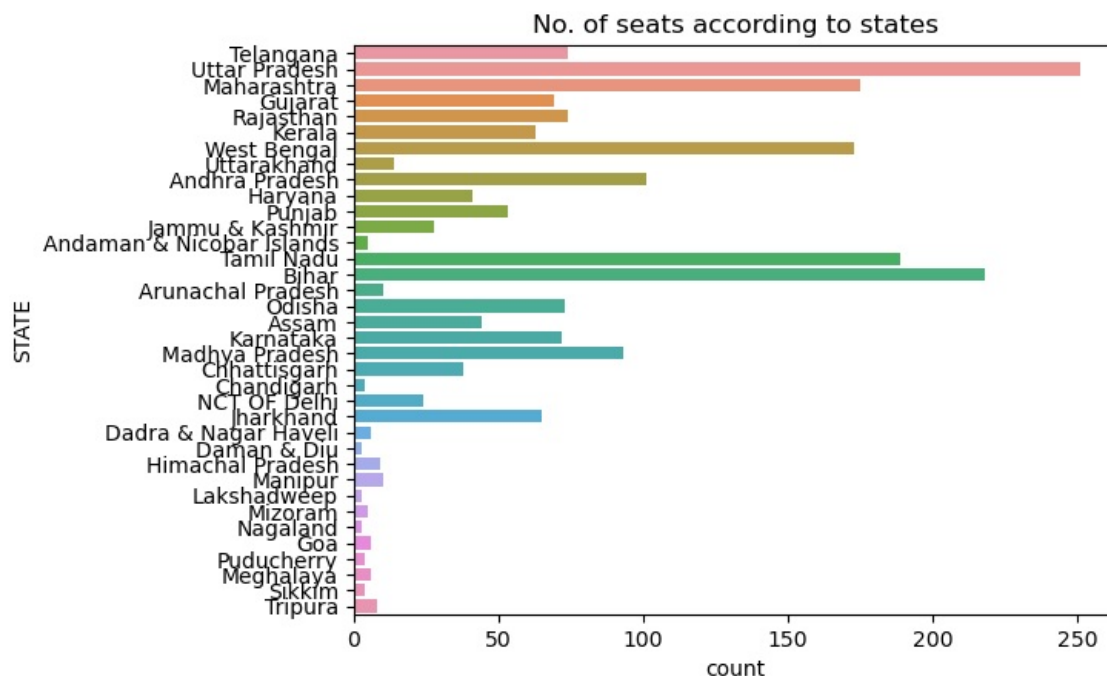
```
In [15]: # Cadidates in Each Party
sns.barplot(x=df.PARTY.value_counts().head(20),y=df.PARTY.value_counts().head(20).index)
```

```
Out[15]: <Axes: xlabel='count', ylabel='PARTY'>
```



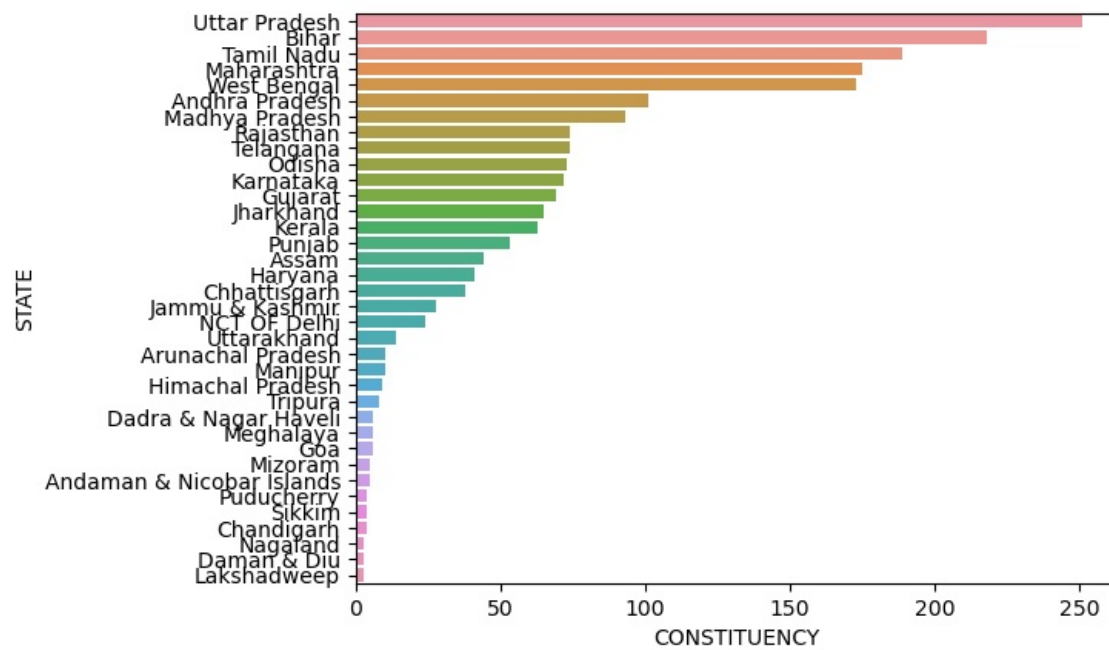
```
In [877]: # No. of seats according to states
sns.countplot(y=df.STATE)
plt.title('No. of seats according to states')
```

```
Out[877]: Text(0.5, 1.0, 'No. of seats according to states')
```



```
In [878]: # No. of Constituency in each state
cons=df.groupby('STATE')['CONSTITUENCY'].count().sort_values(ascending=False)
sns.barplot(x=cons,y=cons.index)
```

```
Out[878]: <Axes: xlabel='CONSTITUENCY', ylabel='STATE'>
```

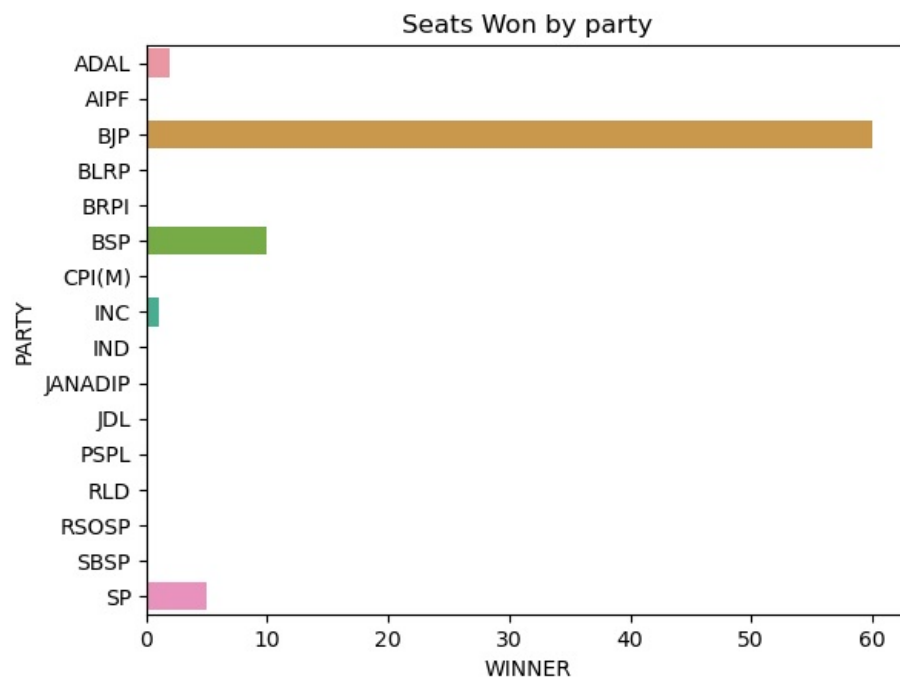


Seats & Winner Analysis

```
In [879]: # Winner in each states(Enter the name of state)
def winner_in_state():
    inp=input('Enter the name of State: ')
    state=df[df.STATE==inp]
    grp=state.groupby(['STATE','PARTY'])['WINNER'].sum()
    sns.barplot(x=grp,y=grp.index.get_level_values('PARTY'))
    plt.title('Seats Won by party')
```

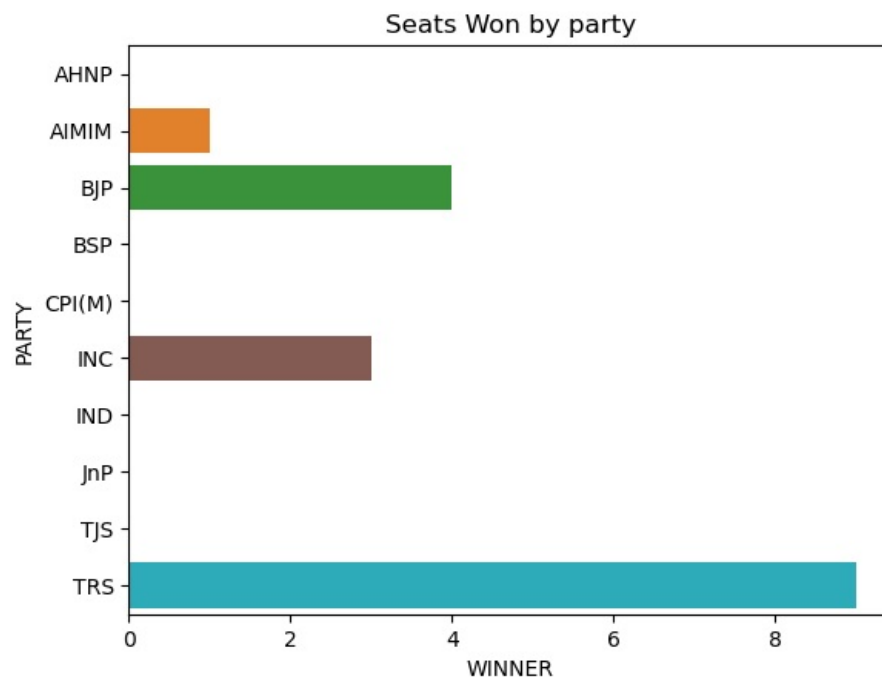
```
In [76]: winner_in_state()
```

Enter the name of State: Uttar Pradesh



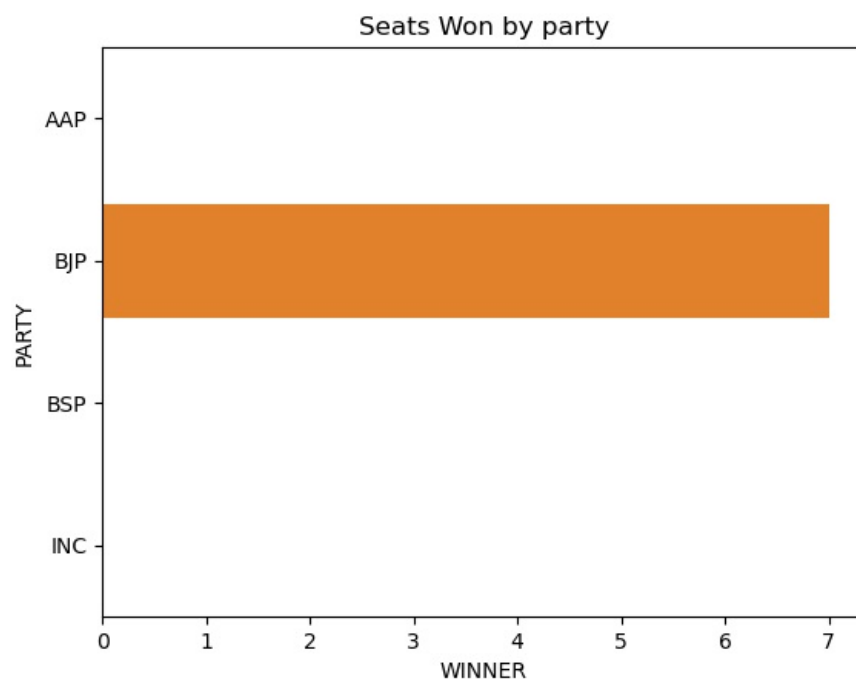
```
In [44]: winner_in_state()
```

Enter the name of State: Telangana



In [45]: `winner_in_state()`

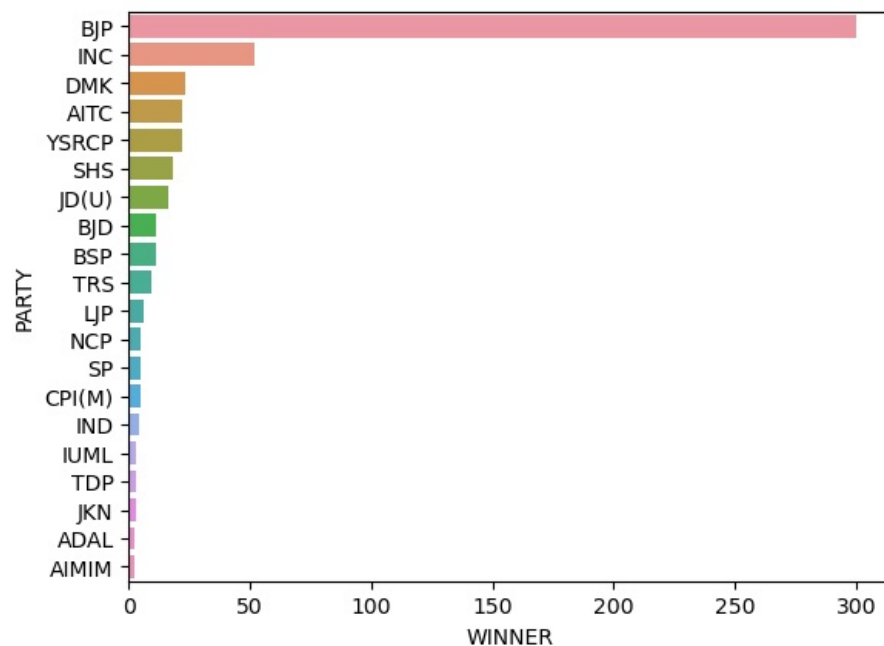
Enter the name of State: NCT OF Delhi



In [355]: `# No. of seats won by Parties`

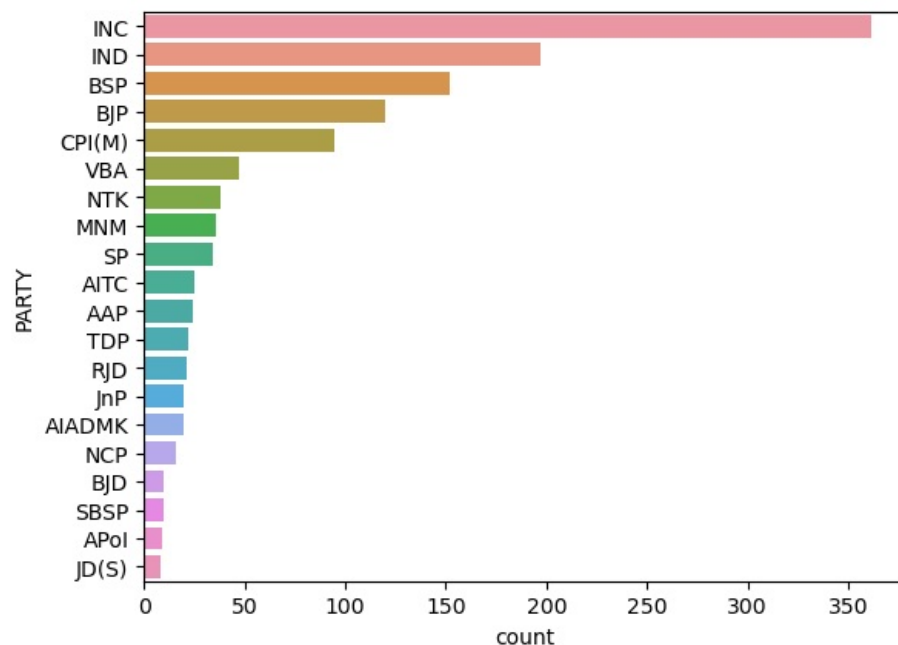
```
won=df[df.WINNER==1]
seats_won=won.groupby('PARTY')['WINNER'].sum().sort_values(ascending=False).head(20)
sns.barplot(x=seats_won,y=seats_won.index)
```

Out[355]: <Axes: xlabel='WINNER', ylabel='PARTY'>



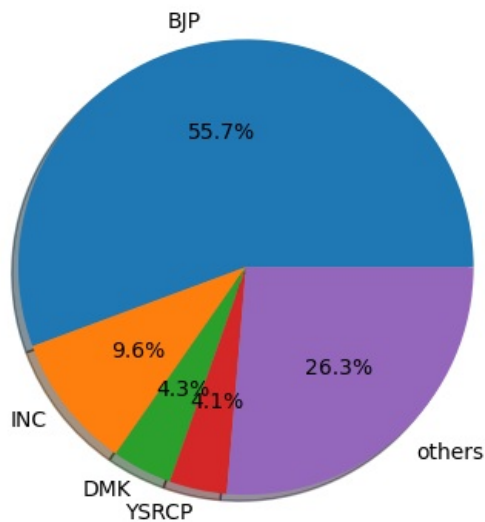
```
In [356]: # Seats Lose by party
lose=df[df.WINNER==0]
sns.barplot(x=lose.PARTY.value_counts().head(20),y=lose.PARTY.value_counts().head(20).index)
```

Out[356]: <Axes: xlabel='count', ylabel='PARTY'>



```
In [919]: # Percentage Seats Won
seats_won=pd.DataFrame({'seats_won':won.PARTY.value_counts()[4:]}.reset_index())
others_seats_won=pd.DataFrame({'PARTY':['others'],
                              'seats_won':won.PARTY.value_counts()[4:].sum()})
won_data=pd.concat([seats_won,others_seats_won])
plt.pie(won_data.seats_won,labels=won_data.PARTY,autopct='%1.1f%%',shadow=True)
```

```
Out[919]: ([<matplotlib.patches.Wedge at 0x159bd74e350>,
<matplotlib.patches.Wedge at 0x159bd8de650>,
<matplotlib.patches.Wedge at 0x159bd82f810>,
<matplotlib.patches.Wedge at 0x159bd8d2610>,
<matplotlib.patches.Wedge at 0x159bd901ad0>],
[Text(-0.19451975306684988, 1.0826643365636517, 'BJP'),
Text(-0.8699166283727463, -0.6732347730774112, 'INC'),
Text(-0.5030991928917576, -0.9782081588862681, 'DMK'),
Text(-0.23225539530863598, -1.0752011120483598, 'YSRCP'),
Text(0.744264571697094, -0.8099816339377957, 'others')],
[Text(-0.10610168349100901, 0.5905441835801736, '55.7%'),
Text(-0.47449997911240704, -0.36721896713313334, '9.6%'),
Text(-0.2744177415773223, -0.533568086665237, '4.3%'),
Text(-0.12668476107743779, -0.5864733338445598, '4.1%'),
Text(0.4059624936529604, -0.4418081639660703, '26.3%')])
```



Gender Analysis

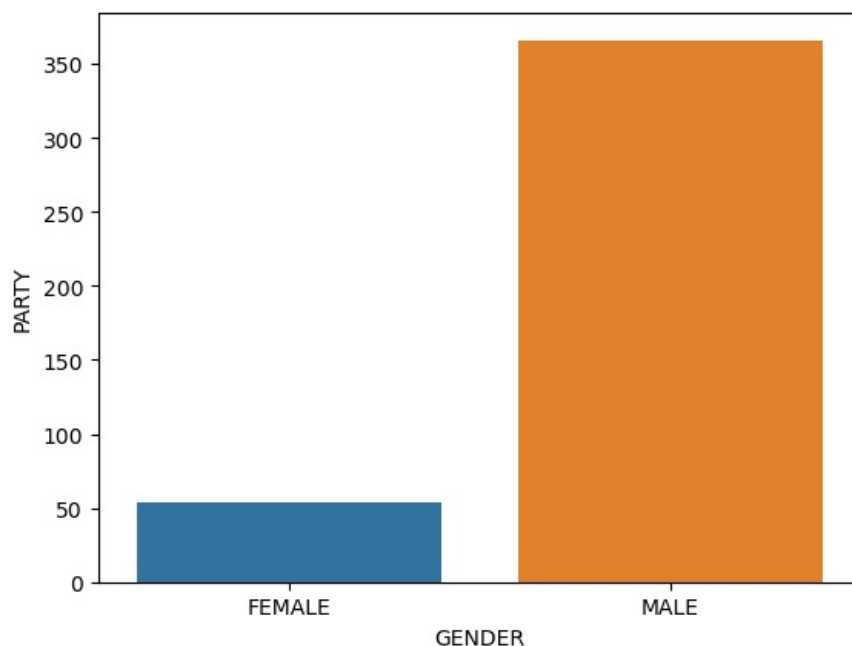
```
In [391]: # Distribution of Male & Female in each Party
```

```
prty=input('Enter Party Name: ')
a=df[df.PARTY==prty]
a=a.groupby('GENDER')['PARTY'].count()
sns.barplot(x=a.index,y=a)
```

Enter Party Name: BJP

<Axes: xlabel='GENDER', ylabel='PARTY'>

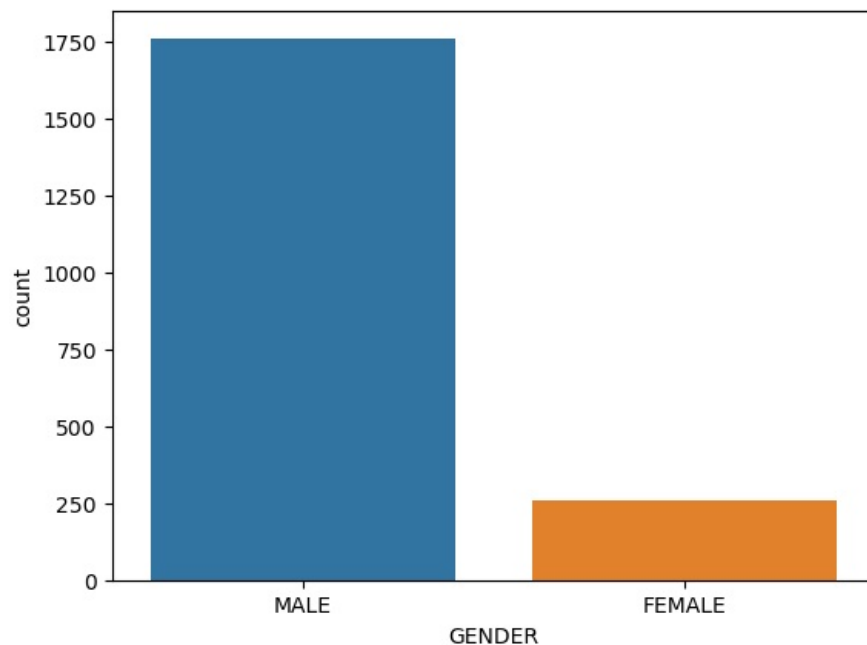
```
Out[391]:
```



```
In [360]: # Distribution of Male & Female
```

```
sns.countplot(x=df.GENDER)
```

```
Out[360]: <Axes: xlabel='GENDER', ylabel='count'>
```

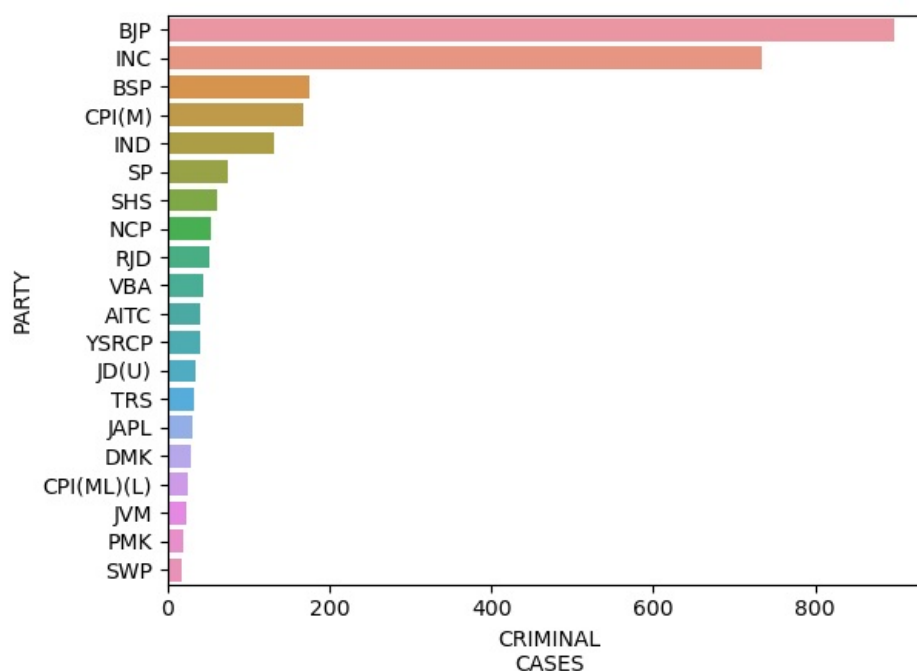



Criminal Cases Analysis

In [361]: # Highest to Lowest Criminal Cases on Party's candidates

```
df['CRIMINAL\nCASES'] = df['CRIMINAL\nCASES'].astype(int)
b = df.groupby('PARTY')['CRIMINAL\nCASES'].sum().sort_values(ascending=False).head(20)
sns.barplot(x=b, y=b.index)
```

Out[361]: <Axes: xlabel='CRIMINAL\nCASES', ylabel='PARTY'>



In [392]: # Detailed view of Criminal cases on parties with respect to seats won.

```
b = df.groupby('PARTY')['CRIMINAL\nCASES'].sum().sort_values(ascending=False)
win_party = won.PARTY.value_counts()
perc_seats_won = (win_party / win_party.sum()) * 100
cases = (b / b.sum()) * 100

data = pd.merge(pd.DataFrame({'seats_won': win_party}), pd.DataFrame({'Percentage_seats_won': perc_seats_won.round(2)}),
               data.rename(columns={'key_0': 'PARTY'}, inplace=True))

cases_data = pd.DataFrame({'no. of criminal cases': b, '% criminal cases': cases.round(2)}).reset_index()
real_data = pd.merge(data, cases_data, on='PARTY')
real_data
```

Out[392]:

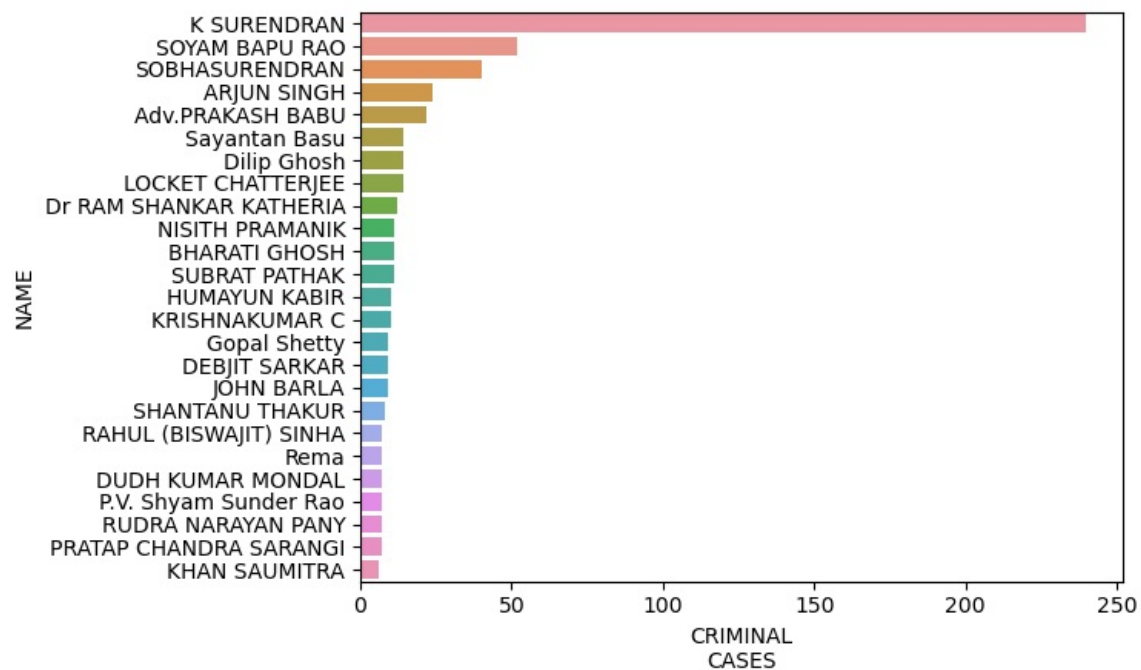
	PARTY	seats_won	Percentage_seats_won	no. of criminal cases	% criminal cases
0	BJP	300	55.66	898	30.61
1	INC	52	9.65	734	25.02
2	DMK	23	4.27	28	0.95
3	YSRCP	22	4.08	41	1.40
4	AITC	22	4.08	41	1.40
5	SHS	18	3.34	62	2.11
6	JD(U)	16	2.97	35	1.19
7	BJD	11	2.04	1	0.03
8	BSP	11	2.04	175	5.96
9	TRS	9	1.67	32	1.09
10	LJP	6	1.11	12	0.41
11	SP	5	0.93	74	2.52
12	NCP	5	0.93	54	1.84
13	CPI(M)	5	0.93	168	5.73
14	IND	4	0.74	131	4.46
15	JKN	3	0.56	1	0.03
16	TDP	3	0.56	7	0.24
17	IUML	3	0.56	3	0.10
18	ADAL	2	0.37	2	0.07
19	SAD	2	0.37	10	0.34
20	AIMIM	2	0.37	13	0.44
21	SKM	1	0.19	0	0.00
22	AIADMK	1	0.19	12	0.41
23	AAP	1	0.19	7	0.24
24	JMM	1	0.19	4	0.14
25	NPF	1	0.19	0	0.00
26	NDPP	1	0.19	0	0.00
27	RLTP	1	0.19	0	0.00
28	AIUDF	1	0.19	9	0.31
29	MNF	1	0.19	0	0.00
30	KEC(M)	1	0.19	0	0.00
31	RSP	1	0.19	5	0.17
32	JD(S)	1	0.19	3	0.10
33	AJSUP	1	0.19	2	0.07
34	VCK	1	0.19	3	0.10
35	NPEP	1	0.19	1	0.03

In [29]: *# Highest to Lowest criminal cases on a Particular party candidate(Enter the name of Party in order to see cand*

```
df['CRIMINAL\nCASES']=df['CRIMINAL\nCASES'].astype(int)
def criminal_cases():
    name=input('Enter the name of Party: ')
    new_df=df[df.PARTY==name]
    no_cases=new_df.groupby('NAME')['CRIMINAL\nCASES'].sum().sort_values(ascending=False).head(25)
    sns.barplot(x=no_cases,y=no_cases.index)
```

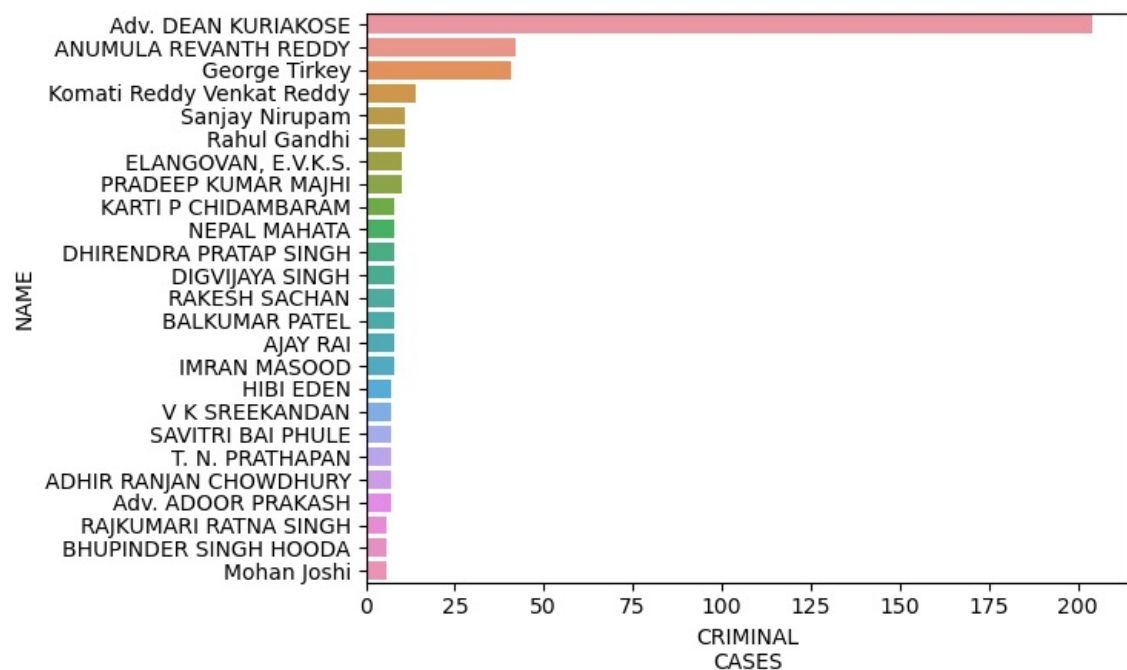
In [25]: criminal_cases()

Enter the name of Party: BJP



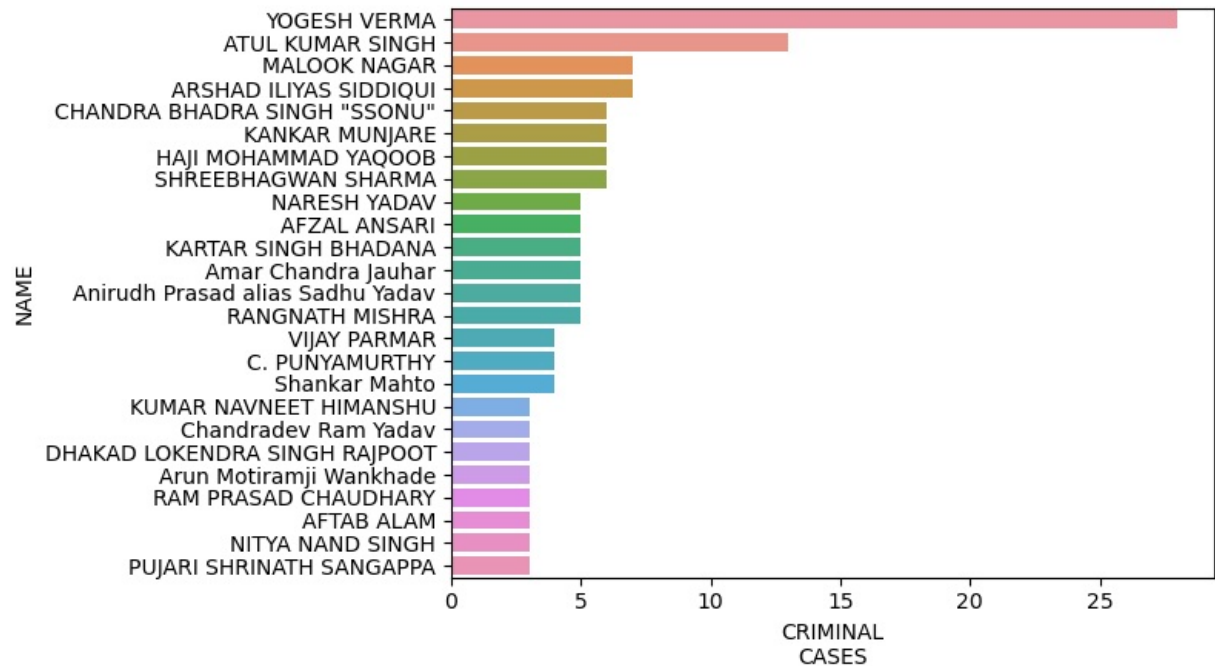
In [26]: `criminal_cases()`

Enter the name of Party: INC



In [27]: `criminal_cases()`

Enter the name of Party: BSP



Age Analysis

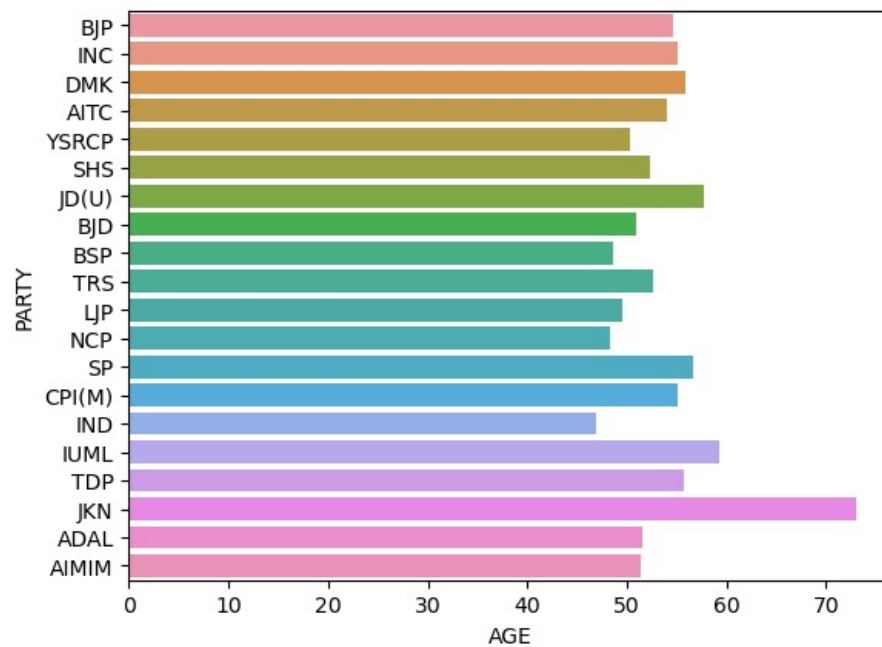
In [393]: # Average Age of Candidates in Each Party

```
age=df.groupby('PARTY')['AGE'].mean().reset_index()
age_sort=won.groupby('PARTY')['WINNER'].sum().sort_values(ascending=False).reset_index()
avg_age=pd.merge(age_sort,age,on='PARTY').head(20)

sns.barplot(x=avg_age.AGE,y=avg_age.PARTY)
avg_age
```

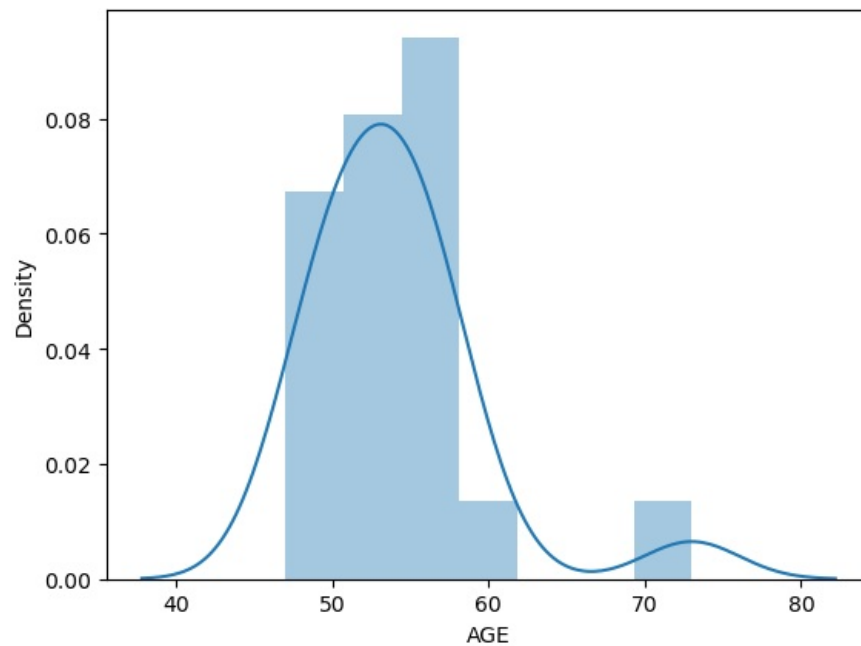
Out[393]:

	PARTY	WINNER	AGE
0	BJP	300	54.583333
1	INC	52	55.031477
2	DMK	23	55.826087
3	AITC	22	54.042553
4	YSRCP	22	50.320000
5	SHS	18	52.346154
6	JD(U)	16	57.777778
7	BJD	11	50.904762
8	BSP	11	48.588957
9	TRS	9	52.647059
10	LJP	6	49.500000
11	NCP	5	48.285714
12	SP	5	56.666667
13	CPI(M)	5	55.120000
14	IND	4	46.985075
15	IUML	3	59.333333
16	TDP	3	55.800000
17	JKN	3	73.000000
18	ADAL	2	51.500000
19	AIMIM	2	51.333333



```
In [394]: # Distribution of Age
sns.distplot(avg_age.AGE)
```

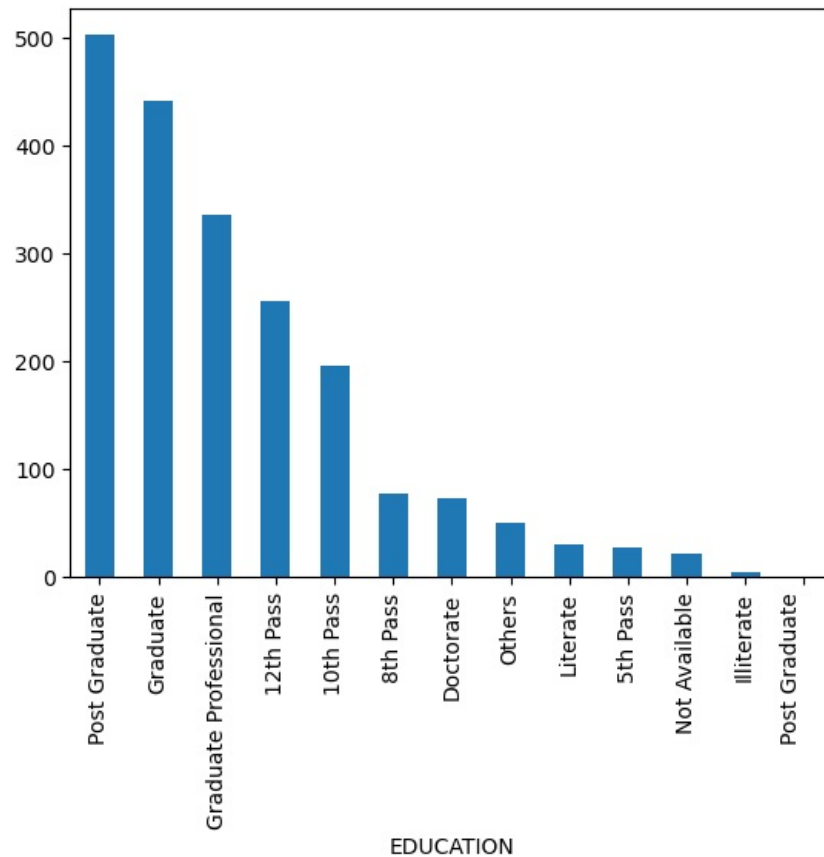
```
Out[394]: <Axes: xlabel='AGE', ylabel='Density'>
```



Education Analysis

```
In [395]: # Aggregate Education qualification
df.EDUCATION.value_counts().plot(kind='bar')
```

```
Out[395]: <Axes: xlabel='EDUCATION'>
```

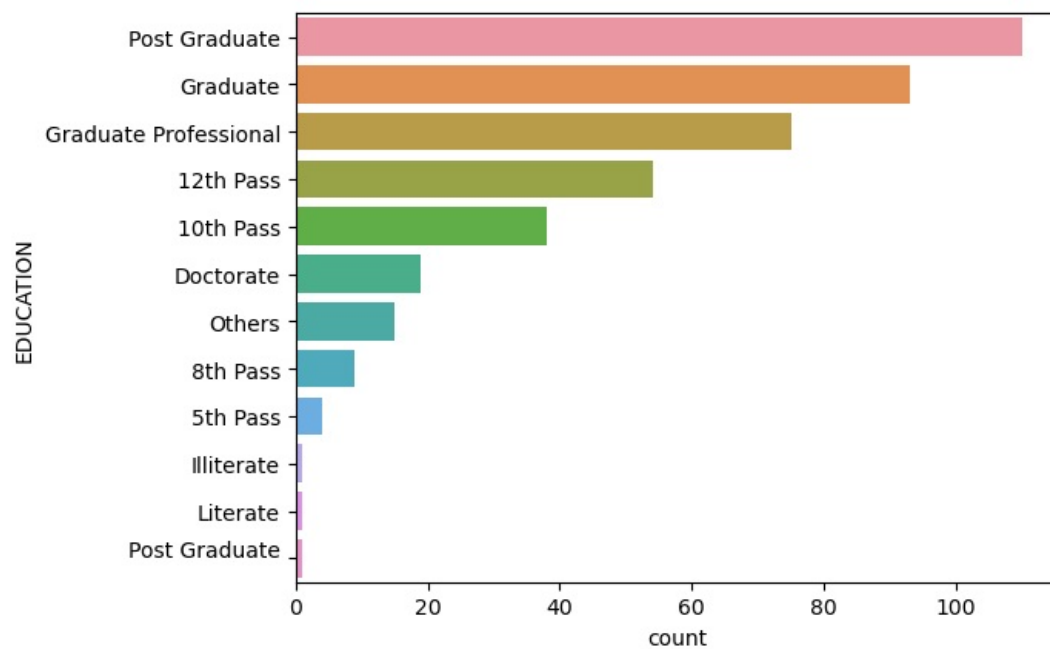


```
In [396.. # Distribution of Education qualification in a particular Party(Enter the name of the party)

def education():
    name=input('Enter the Party Name: ')
    party_sort=df[df.PARTY==name]
    sns.barplot(x=party_sort.EDUCATION.value_counts(),y=party_sort.EDUCATION.value_counts().index)
    print(party_sort.EDUCATION.value_counts())
```

```
In [397.. education()

Enter the Party Name: BJP
EDUCATION
Post Graduate      110
Graduate           93
Graduate Professional 75
12th Pass          54
10th Pass          38
Doctorate          19
Others             15
8th Pass           9
5th Pass           4
Illiterate         1
Literate           1
Post Graduate\n    1
Name: count, dtype: int64
```



In [398..] education()

Enter the Party Name: INC

EDUCATION

Post Graduate 114

Graduate 104

Graduate Professional 84

12th Pass 47

10th Pass 30

Doctorate 11

8th Pass 9

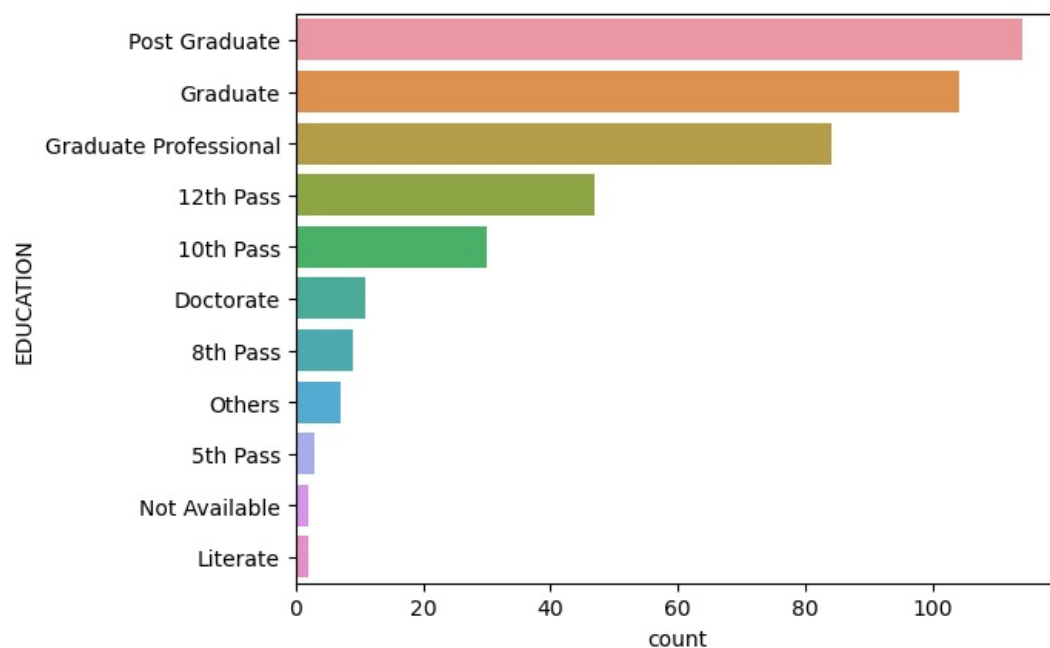
Others 7

5th Pass 3

Not Available 2

Literate 2

Name: count, dtype: int64



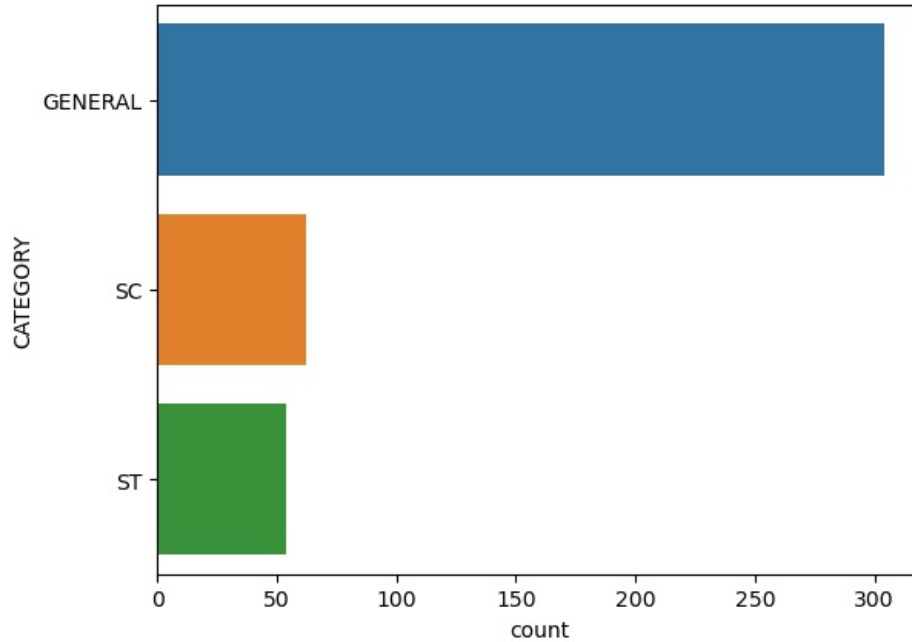
Category(cast) Analysis

In [399..] # Candidates Category in each party

```
In [400... def category():
name=input('Enter the Party Name: ')
cat_sort=df[df.PARTY==name]
sns.barplot(x=cat_sort.CATEGORY.value_counts(),y=cat_sort.CATEGORY.value_counts().index)
print(cat_sort.CATEGORY.value_counts())
```

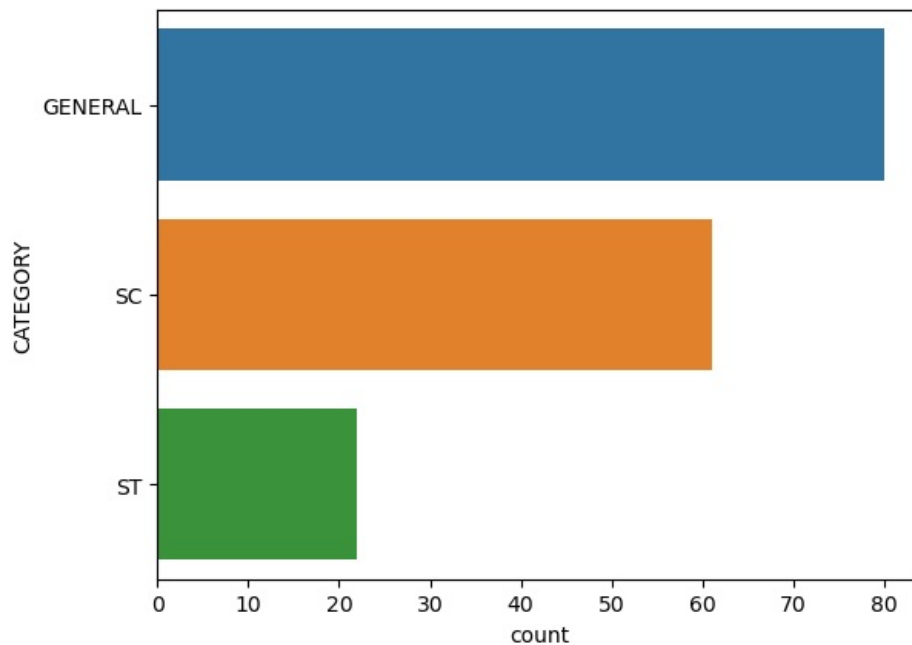
```
In [401... category()
```

```
Enter the Party Name: BJP
CATEGORY
GENERAL    304
SC          62
ST          54
Name: count, dtype: int64
```



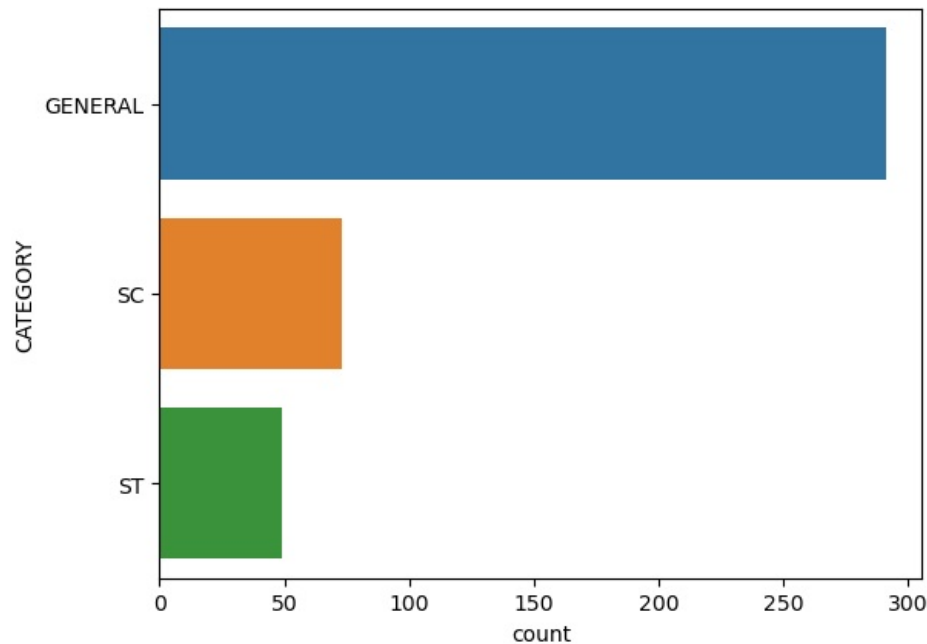
```
In [402... category()
```

```
Enter the Party Name: BSP
CATEGORY
GENERAL    80
SC          61
ST          22
Name: count, dtype: int64
```



```
In [195... category()
```

```
Enter the Party Name: INC
CATEGORY
GENERAL    291
SC          73
ST          49
Name: count, dtype: int64
```

Assets Analysis

```
In [ ]: # Assets of Party's candidate
```

```
In [880]: df.ASSETS.reset_index(drop=True,inplace=True)
```

```
In [885]: asset=df.ASSETS
```

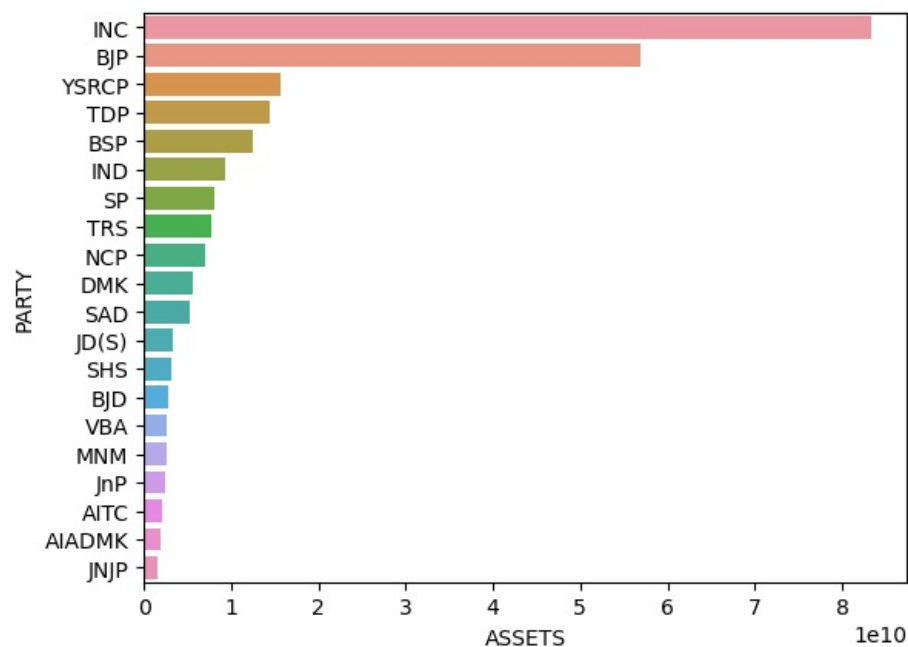
```
In [886]: asset_rupee = []
for i in asset:
    part = i.split('Rs ')
    if len(part) > 1:
        rah = part[1].split('\n')[0].replace(',','')
        asset_rupee.append(int(rah))
    else:
        asset_rupee.append(None)

# Convert 'asset_rupee' list to a DataFrame and assign it to the 'ASSETS' column
df['ASSETS'] = asset_rupee
```

```
In [889]: df.ASSETS=asset_rupee
```

```
In [890]: total_asset=df.groupby('PARTY')['ASSETS'].sum().sort_values(ascending=False).head(20)
sns.barplot(x=total_asset,y=total_asset.index)
```

```
Out[890]: <Axes: xlabel='ASSETS', ylabel='PARTY'>
```

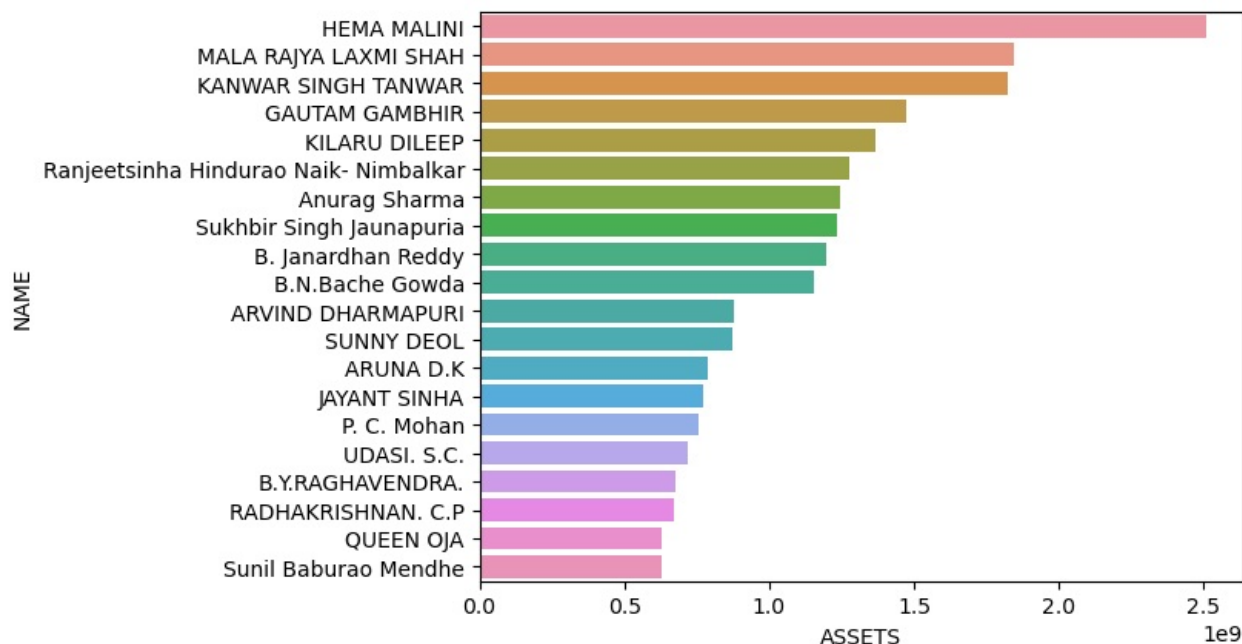


In []:

```
In [833... # Enter The name of the party to see the Assets
def asset():
    party=input('Enter the name of Party: ')
    new_df=df[df.PARTY==party]
    ab=new_df.groupby('NAME')['ASSETS'].sum().sort_values(ascending=False).head(20)
    sns.barplot(x=ab,y=ab.index)
```

In [834... asset()

Enter the name of Party: BJP



Liabilities Analysis

```
In [835... # Liabilities of Party's candidate
```

```
In [836... liab=df.LIABILITIES
```

```
In [837... liab[0]
```

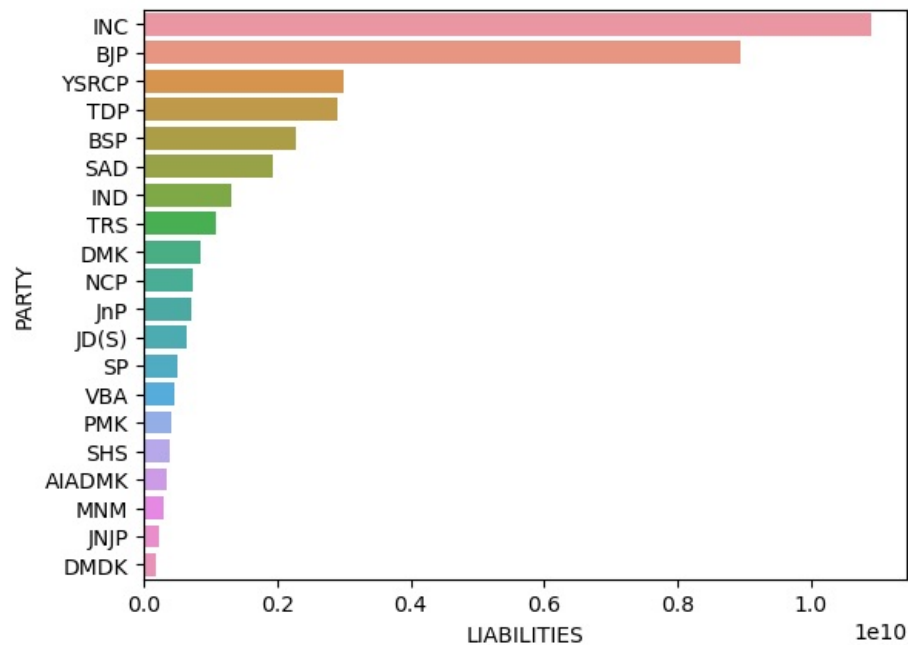
Out[837]: 'Rs 2,31,450\n ~ 2 Lacs+'

```
In [838... liab_rupee=[]
for i in liab:
    parts=i.split('Rs ')
    if len(parts)>1:
        lb = parts[1].split('\n')[0].replace(',','')
        liab_rupee.append(int(lb))
    else:
        liab_rupee.append(None)
```

```
In [839... df.LIABILITIES=liab_rupee
```

```
In [840... total_liab=df.groupby('PARTY')['LIABILITIES'].sum().sort_values(ascending=False).head(20)
sns.barplot(x=total_liab,y=total_liab.index)
```

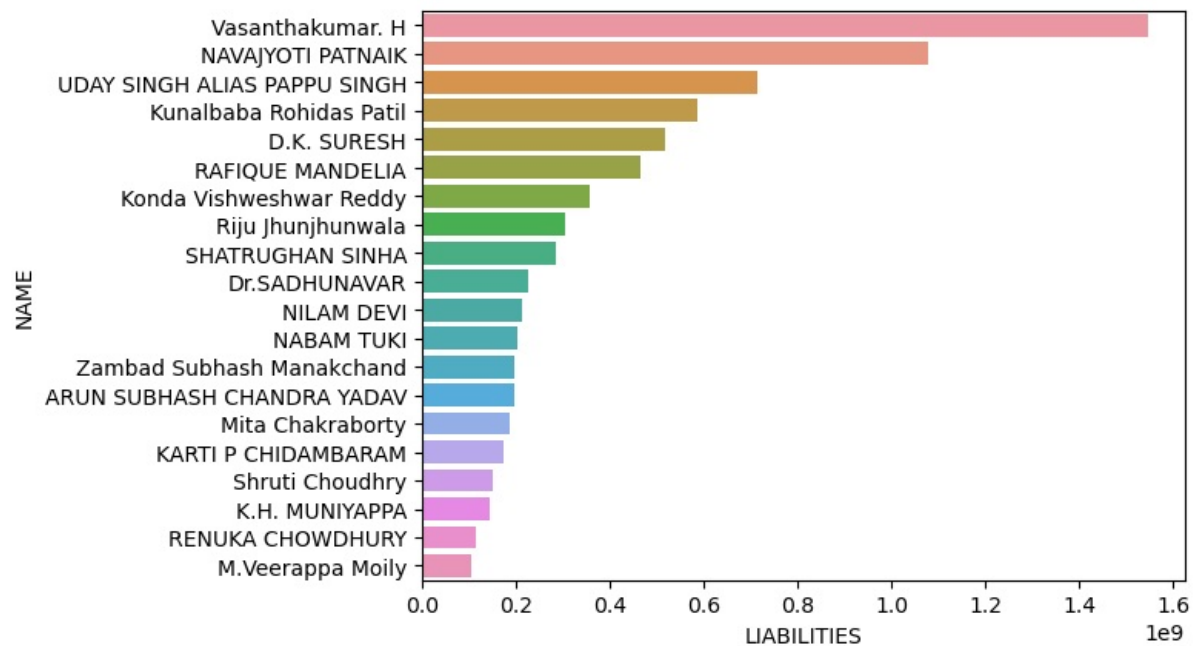
Out[840]: <Axes: xlabel='LIABILITIES', ylabel='PARTY'>



```
In [841... # Enter The name of the party to see the LIABILITIES
def liab():
    party=input('Enter the name of Party: ')
    new_df=df[df.PARTY==party]
    ab=new_df.groupby('NAME')['LIABILITIES'].sum().sort_values(ascending=False).head(20)
    sns.barplot(x=ab,y=ab.index)
```

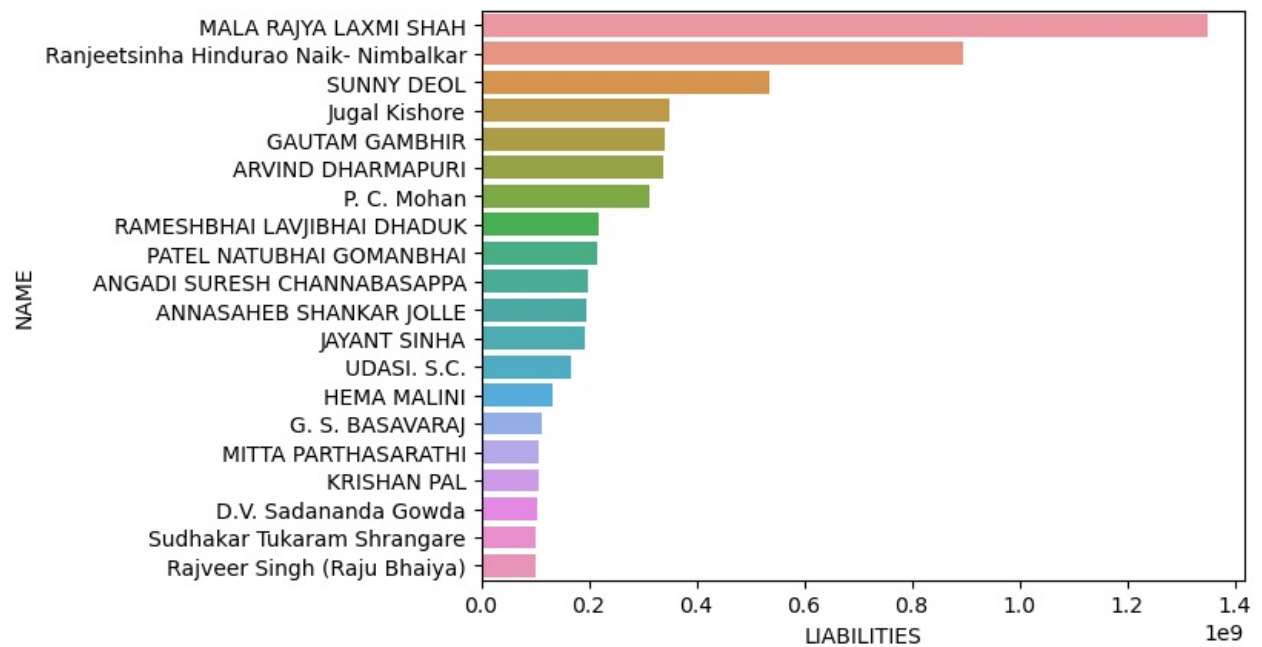
In [842... liab()

Enter the name of Party: INC



In [734... liab()

Enter the name of Party: BJP



General & Postal & Total votes Analysis

In [737...] *# General & Postal & Total votes*

```
def votes():
    party=input('Enter Party name Here: ')
    vote=df[df.PARTY==party]

    print(f'Total General Votes got by {party}: ',vote['GENERAL\nVOTES'].sum())
    print(f'Postal Votes got by {party}: ',vote['POSTAL\nVOTES'].sum())
    print(f'Postal Votes got by {party}: ',vote['TOTAL\nVOTES'].sum())
```

In [738...] votes()

```
Enter Party name Here: BJP
General Votes got by BJP: 227923509
Postal Votes got by BJP: 1015047
Postal Votes got by BJP: 228938556
```

In [747...] votes()

```
Enter Party name Here: BSP
General Votes got by BSP: 20749113
Postal Votes got by BSP: 59081
Postal Votes got by BSP: 20808194
```

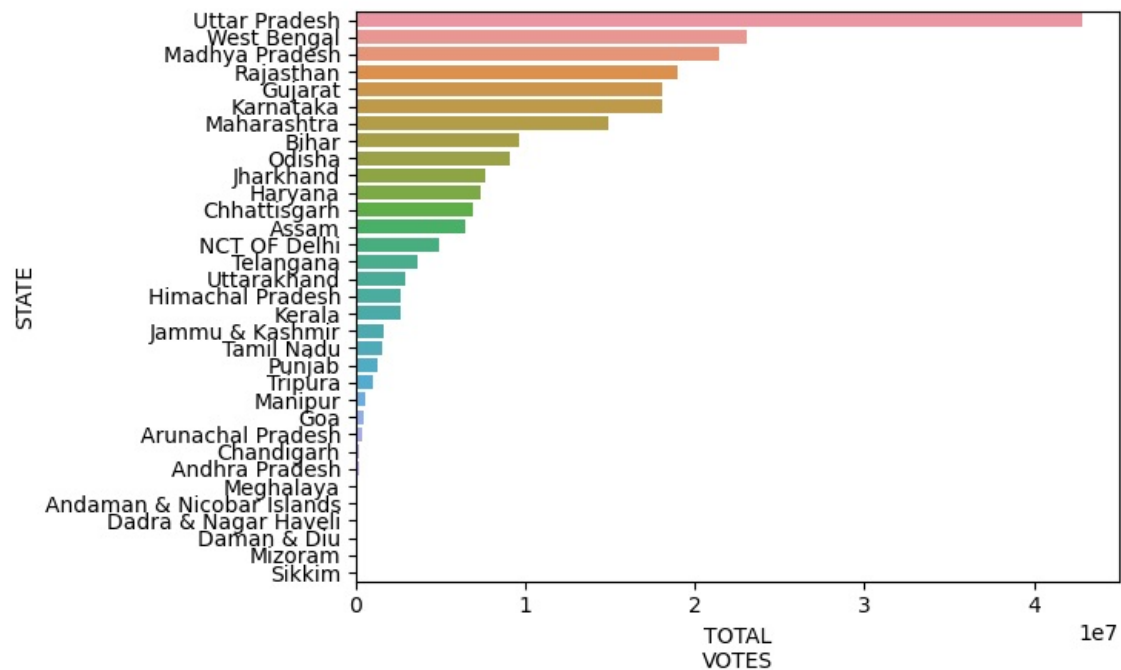
In [742...] *# Votes got by each party from each State*

```
def vote_state():
    party=input('Enter Party name Here: ')
    vote=df[df.PARTY==party]

    vote_st=vote.groupby('STATE')['TOTAL\nVOTES'].sum().sort_values(ascending=False)
    sns.barplot(x=vote_st,y=vote_st.index)
```

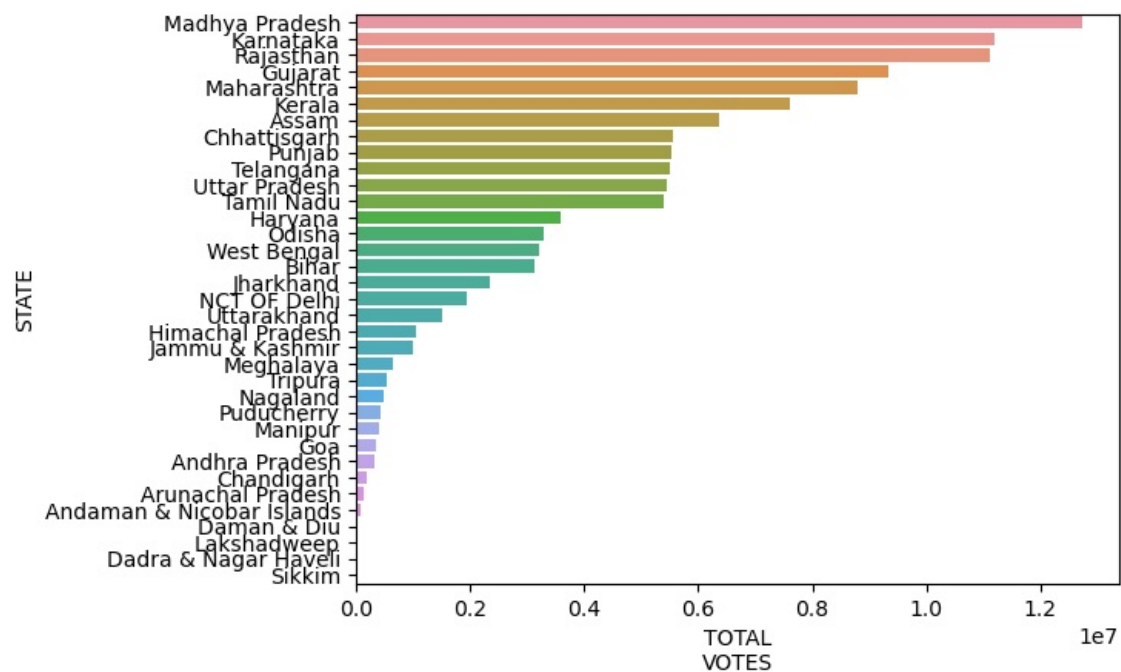
In [743...] vote_state()

```
Enter Party name Here: BJP
```



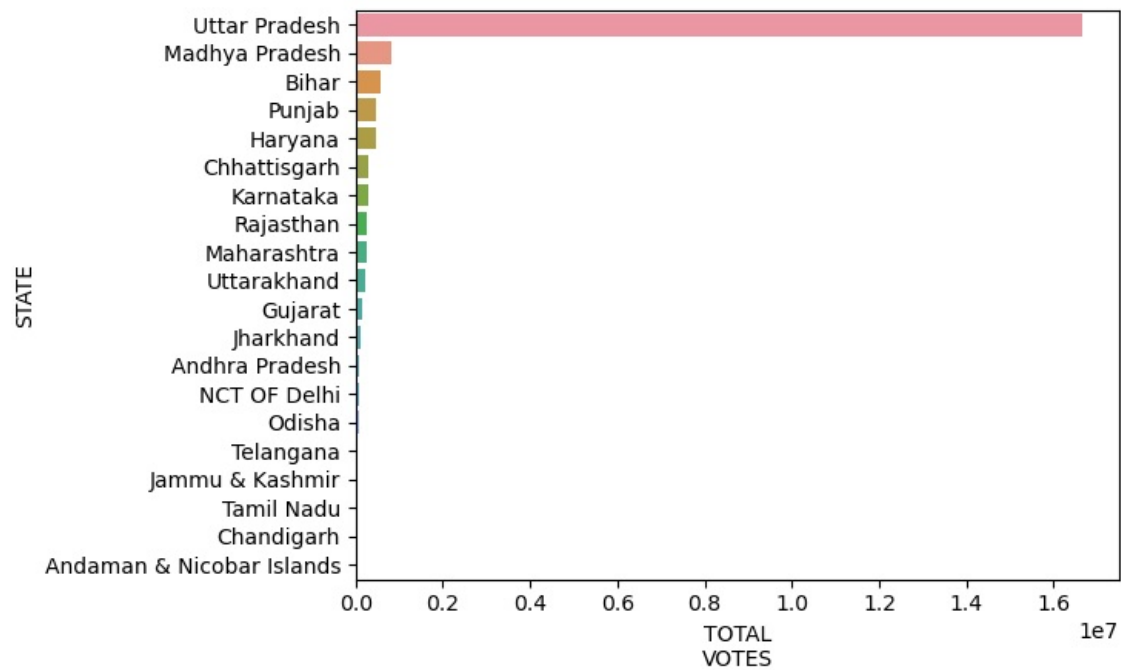
In [744... vote_state()

Enter Party name Here: INC



In [745... vote_state()

Enter Party name Here: BSP



Electors Analysis

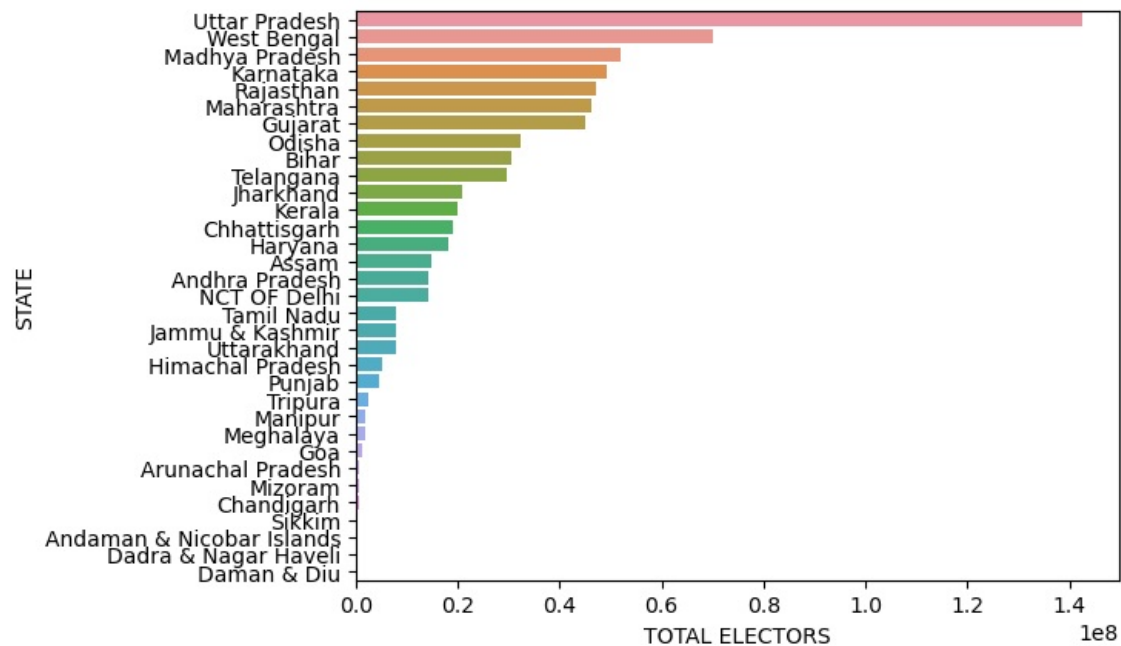
In [749..] # Total Electors in a particualr Party(Enter the party name to see their total Electors)

```
def electors():
    party=input('Enter Party name Here: ')
    vote=df[df.PARTY==party]

    vote_st=vote.groupby('STATE')['TOTAL ELECTORS'].sum().sort_values(ascending=False)
    sns.barplot(x=vote_st,y=vote_st.index)
```

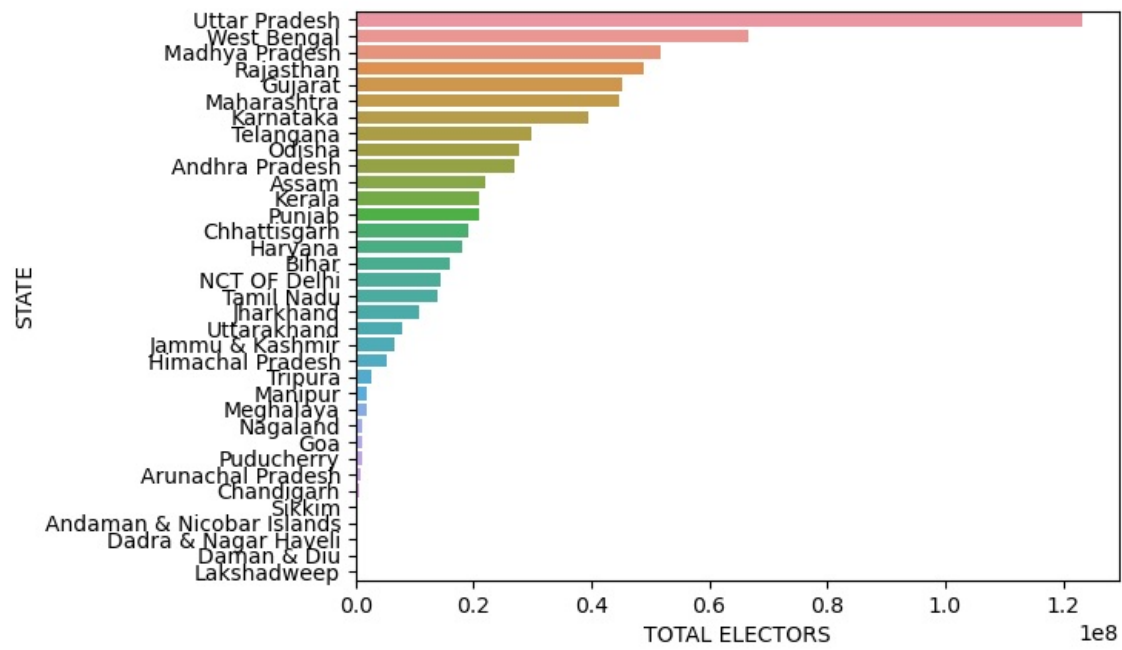
In [750..] electors()

Enter Party name Here: BJP



In [751..] electors()

Enter Party name Here: INC



```
In [ ]: # Analysis by rahul badola-
        #contact- contact.rahulbadola@gmail.com
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js