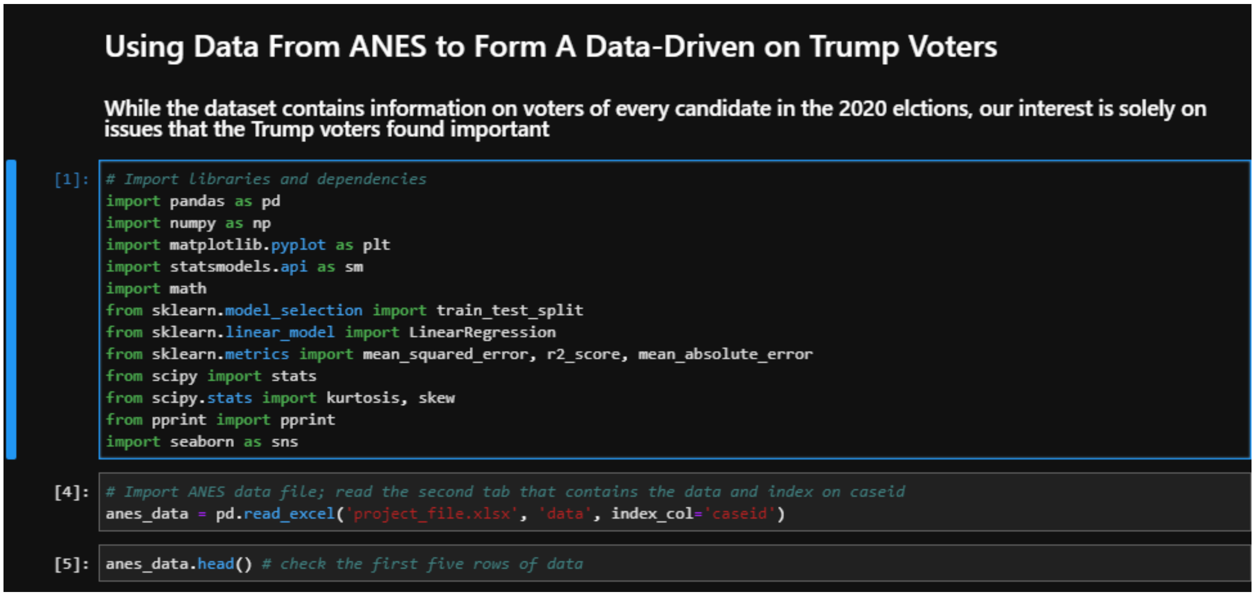


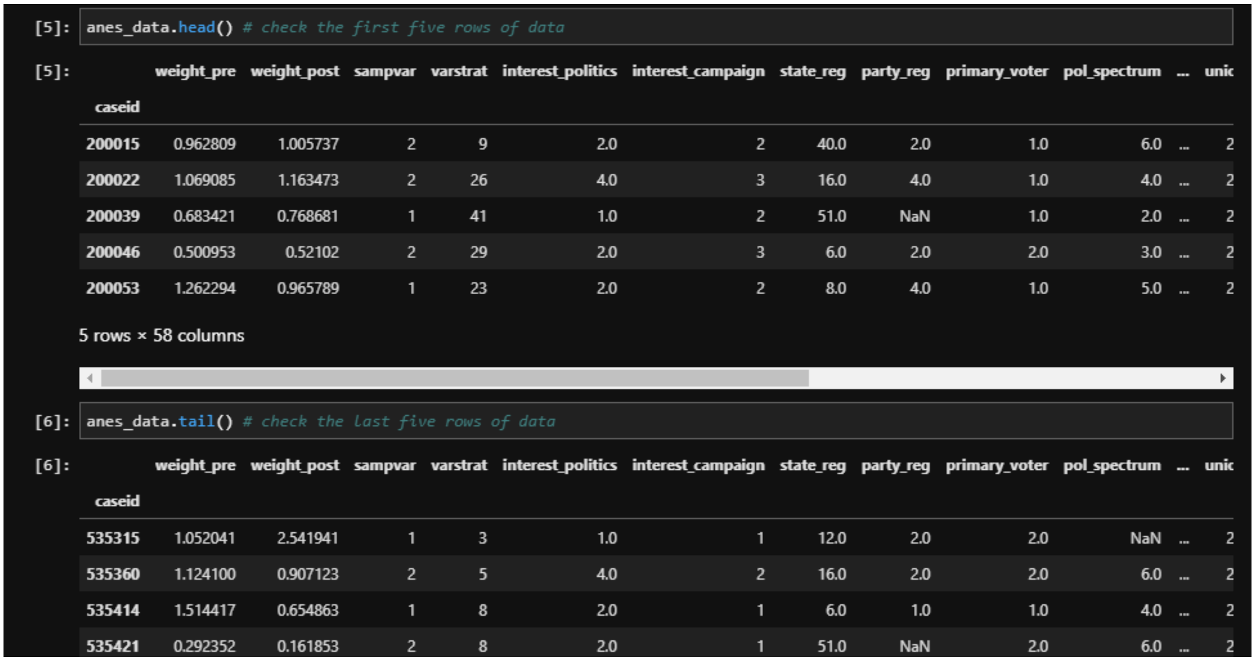
Using Data From ANES to Form A Data-Driven on Trump Voters

While the dataset contains information on voters of every candidate in the 2020 elctions, our interest is solely on issues that the Trump voters found important

[1]:
```
# Import libraries and dependencies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import math
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from scipy import stats
from scipy.stats import kurtosis, skew
from pprint import pprint
import seaborn as sns
```

[4]:
```
# Import ANES data file; read the second tab that contains the data and index on caseid
anes_data = pd.read_excel('project_file.xlsx', 'data', index_col='caseid')
```

[5]:
```
anes_data.head() # check the first five rows of data
```



[5]:
```
anes_data.head() # check the first five rows of data
```

[5]:

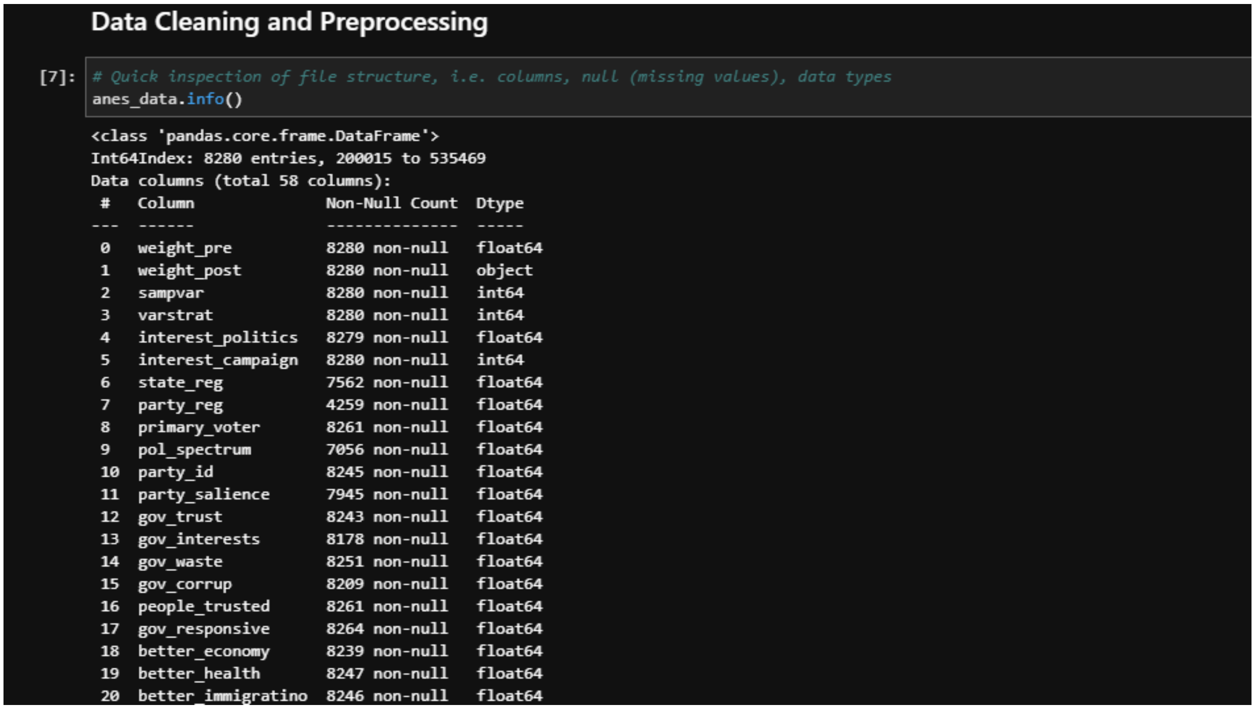
	weight_pre	weight_post	sampvar	varstrat	interest_politics	interest_campaign	state_reg	party_reg	primary_voter	pol_spectrum	...	unic
caseid												
200015	0.962809	1.005737	2	9	2.0	2	40.0	2.0	1.0	6.0	...	2
200022	1.069085	1.163473	2	26	4.0	3	16.0	4.0	1.0	4.0	...	2
200039	0.683421	0.768681	1	41	1.0	2	51.0	NaN	1.0	2.0	...	2
200046	0.500953	0.52102	2	29	2.0	3	6.0	2.0	2.0	3.0	...	2
200053	1.262294	0.965789	1	23	2.0	2	8.0	4.0	1.0	5.0	...	2

5 rows × 58 columns

[6]:
```
anes_data.tail() # check the last five rows of data
```

[6]:

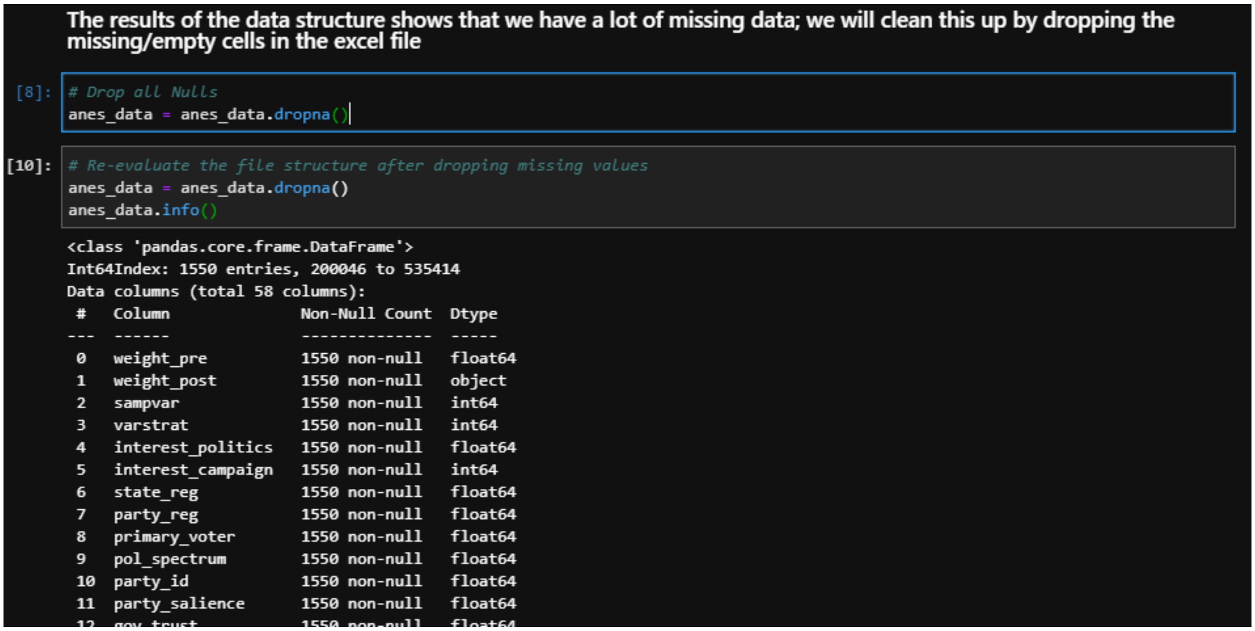
	weight_pre	weight_post	sampvar	varstrat	interest_politics	interest_campaign	state_reg	party_reg	primary_voter	pol_spectrum	...	unic
caseid												
535315	1.052041	2.541941	1	3	1.0	1	12.0	2.0	2.0	NaN	...	2
535360	1.124100	0.907123	2	5	4.0	2	16.0	2.0	2.0	6.0	...	2
535414	1.514417	0.654863	1	8	2.0	1	6.0	1.0	1.0	4.0	...	2
535421	0.292352	0.161853	2	8	2.0	1	51.0	NaN	2.0	6.0	...	2



Data Cleaning and Preprocessing

[7]:
```
# Quick inspection of file structure, i.e. columns, null (missing values), data types
anes_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8280 entries, 200015 to 535469
Data columns (total 58 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   weight_pre          8280 non-null   float64
 1   weight_post         8280 non-null   object
 2   sampvar             8280 non-null   int64
 3   varstrat            8280 non-null   int64
 4   interest_politics   8279 non-null   float64
 5   interest_campaign   8280 non-null   int64
 6   state_reg           7562 non-null   float64
 7   party_reg           4259 non-null   float64
 8   primary_voter       8261 non-null   float64
 9   pol_spectrum        7056 non-null   float64
 10  party_id            8245 non-null   float64
 11  party_salience      7945 non-null   float64
 12  gov_trust           8243 non-null   float64
 13  gov_interests       8178 non-null   float64
 14  gov_waste           8251 non-null   float64
 15  gov_corrup          8209 non-null   float64
 16  people_trusted      8261 non-null   float64
 17  gov_responsive      8264 non-null   float64
 18  better_economy      8239 non-null   float64
 19  better_health       8247 non-null   float64
 20  better_immigratino  8246 non-null   float64
```

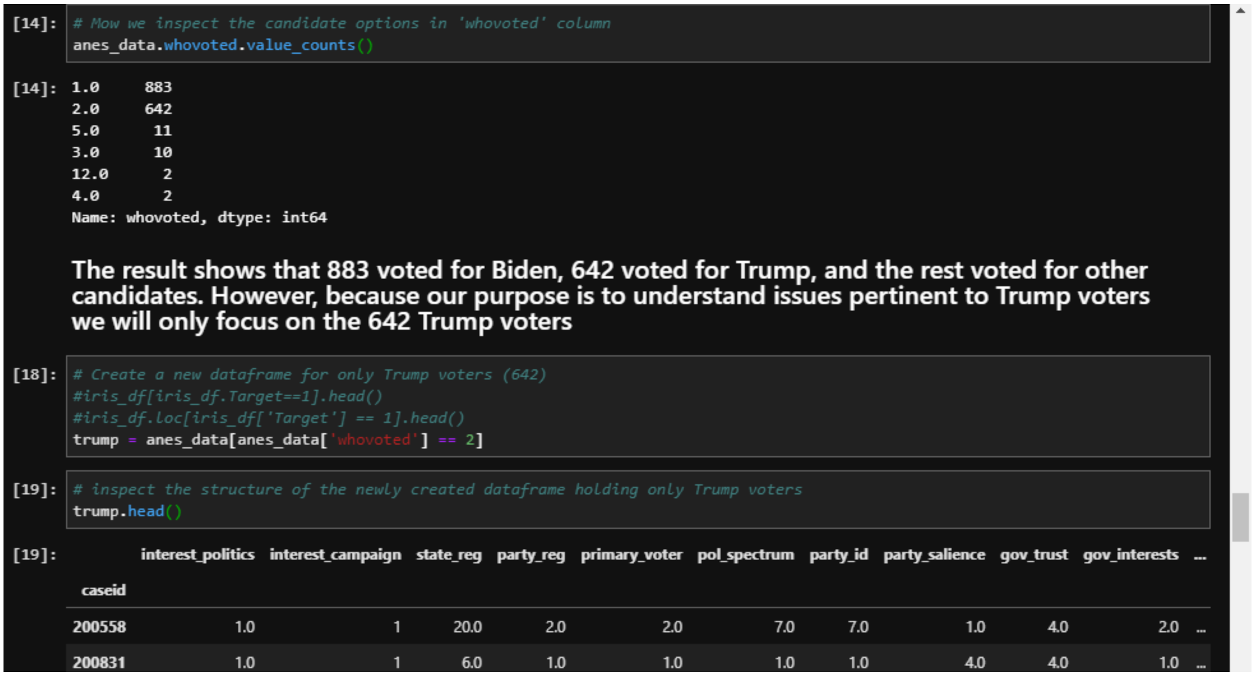


The results of the data structure shows that we have a lot of missing data; we will clean this up by dropping the missing/empty cells in the excel file

[8]:
```
# Drop all Nulls
anes_data = anes_data.dropna()
```

[10]:
```
# Re-evaluate the file structure after dropping missing values
anes_data = anes_data.dropna()
anes_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1550 entries, 200046 to 535414
Data columns (total 58 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   weight_pre          1550 non-null   float64
 1   weight_post         1550 non-null   object
 2   sampvar             1550 non-null   int64
 3   varstrat            1550 non-null   int64
 4   interest_politics   1550 non-null   float64
 5   interest_campaign   1550 non-null   int64
 6   state_reg           1550 non-null   float64
 7   party_reg           1550 non-null   float64
 8   primary_voter       1550 non-null   float64
 9   pol_spectrum        1550 non-null   float64
 10  party_id            1550 non-null   float64
 11  party_salience      1550 non-null   float64
 12  gov_trust           1550 non-null   float64
```



```
[14]: # Mow we inspect the candidate options in 'whovoted' column
      anes_data.whovoted.value_counts()

[14]: 1.0     883
      2.0     642
      5.0      11
      3.0      10
      12.0      2
      4.0       2
      Name: whovoted, dtype: int64
```

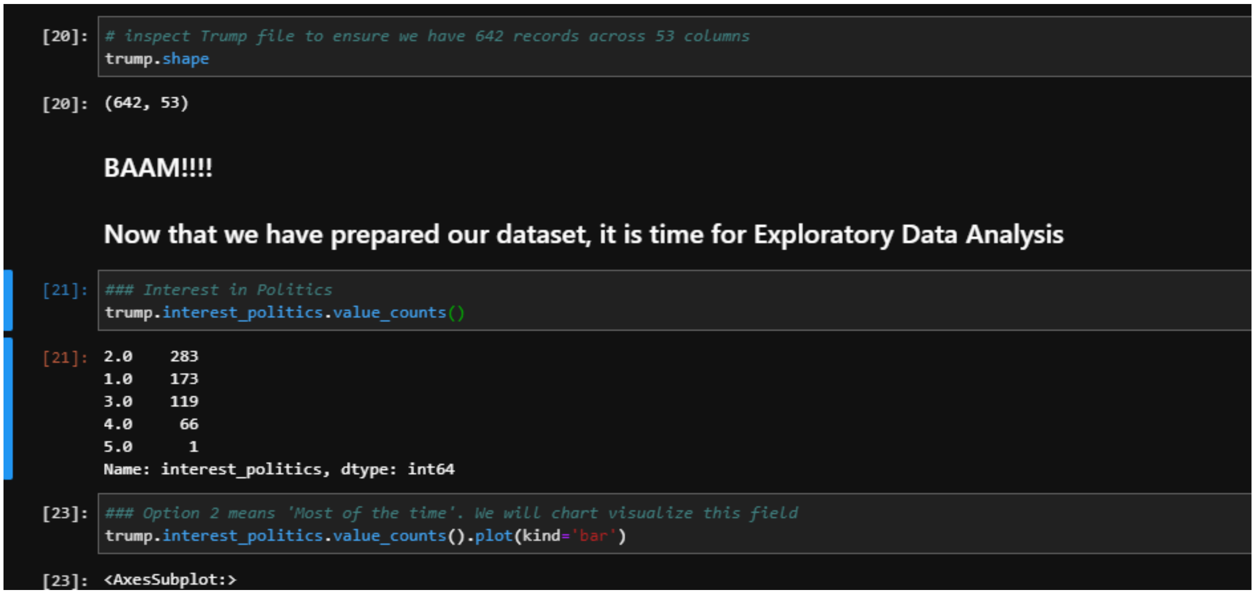
The result shows that 883 voted for Biden, 642 voted for Trump, and the rest voted for other candidates. However, because our purpose is to understand issues pertinent to Trump voters we will only focus on the 642 Trump voters

```
[18]: # Create a new dataframe for only Trump voters (642)
      #iris_df[iris_df.Target==1].head()
      #iris_df.loc[iris_df['Target'] == 1].head()
      trump = anes_data[anes_data['whovoted'] == 2]

[19]: # inspect the structure of the newly created dataframe holding only Trump voters
      trump.head()
```

[19]:

	interest_politics	interest_campaign	state_reg	party_reg	primary_voter	pol_spectrum	party_id	party_salience	gov_trust	gov_interests	...
caseid											
200558	1.0	1	20.0	2.0	2.0	7.0	7.0	1.0	4.0	2.0	...
200831	1.0	1	6.0	1.0	1.0	1.0	1.0	4.0	4.0	1.0	...



```
[20]: # inspect Trump file to ensure we have 642 records across 53 columns
      trump.shape

[20]: (642, 53)
```

BAAM!!!!

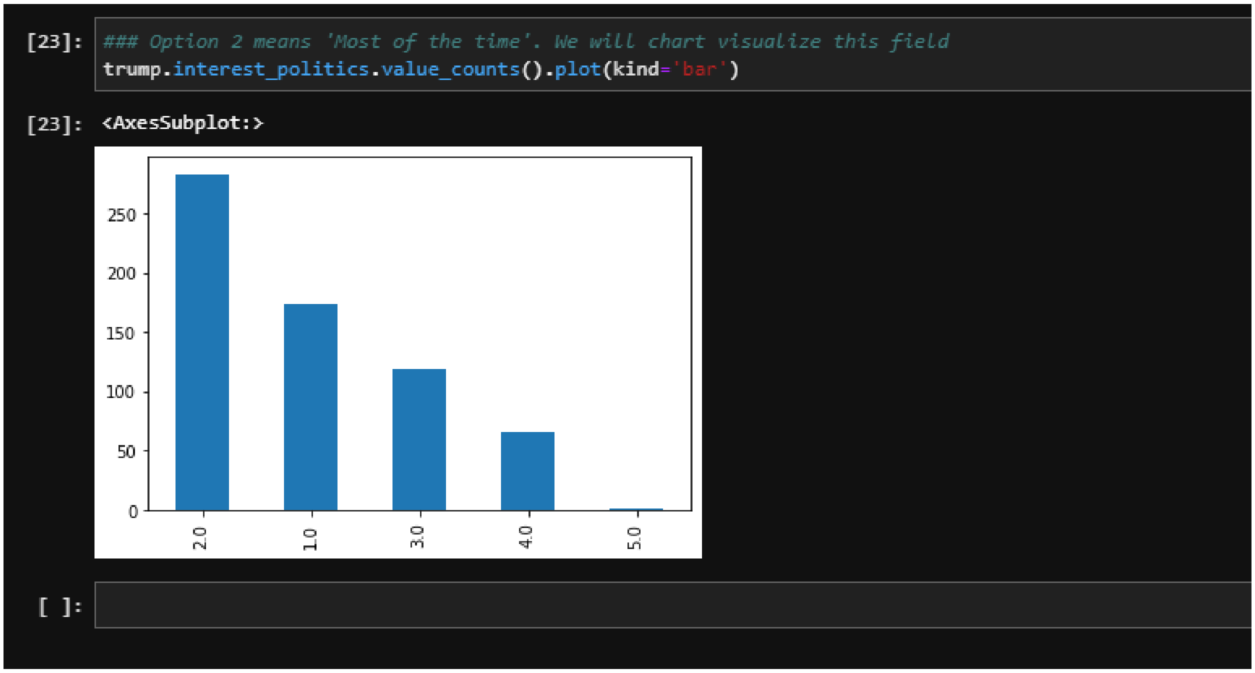
Now that we have prepared our dataset, it is time for Exploratory Data Analysis

```
[21]: ### Interest in Politics
      trump.interest_politics.value_counts()

[21]: 2.0    283
      1.0    173
      3.0    119
      4.0     66
      5.0      1
      Name: interest_politics, dtype: int64

[23]: ### Option 2 means 'Most of the time'. We will chart visualize this field
      trump.interest_politics.value_counts().plot(kind='bar')

[23]: <AxesSubplot:>
```



```
[23]: ### Option 2 means 'Most of the time'. We will chart visualize this field
      trump.interest_politics.value_counts().plot(kind='bar')
```

```
[23]: <AxesSubplot:>
```

```
[ ]:
```