Geospatial Analysis and Outlier Detection in Election Data

Introduction

Ensuring the integrity and transparency of electoral processes is paramount in maintaining public trust in democratic institutions. One critical aspect of election integrity is the analysis of polling data to identify irregularities that might suggest anomalies or potential fraud. In this report, I detail an outlier election detection analysis I carried out using data from the 2023 Presidential Elections in Yobe state, Nigeria

Methodology

I utilized Google sheets and python for my analysis. Google sheets was used for data cleaning and co-ordinates sourcing while python was used for outlier score calculation, sorting and visualization. Each step is detailed below:

Data Pre-processing and Cleaning:

- The election dataset had information on polling units and accrued votes by parties but did not have geographic coordinates. To obtain the longitude and latitude, I utilized the Awesome Geocoder Table add-on in Google sheets. All rows had their coordinates returned except one which I manually imputed by typing the address on google maps, locating the pin and copying the returned coordinates.
- Next, I prepared the dataset for analysis by checking for nulls, formatting datatypes and checking for duplicates, particularly on the polling unit column (PU-Code) since this had to be unique.
- I created a new column 'Voter Consistency' using an IF function to check that accredited votes were not more than the registered votes as this would be an obvious red flag of voter fraud. If only 100 people are registered to vote in a particular polling unit and on election day 200 accredited votes are recorded, it means there's something fishy.
- Finally, to conclude my data cleaning I dropped the last column 'Result_file' to give the dataset a cleaner look because the column contained a long line of urls and was also not needed for my analysis.

Finding Neighbours Using KD-Tree:

- A KD-Tree was constructed with python code in Jupyter notebook using the geographic coordinates of the polling units.
- For each polling unit, neighbouring units within a 1 km radius were identified using the KD-Tree.
- These neighbours were used to calculate the average votes for comparison.

```
[1]: import pandas as pd
     import numpy as np
     from scipy.spatial import KDTree

     # Defining the file path to my CSV file
     file_path = 'Yobe election sheet.csv'

     # Loading polling units data from CSV
     data = pd.read_csv(file_path, encoding='latin1')
```

```
# Function to find neighbours using KD-Tree
def find_neighbours_optimized(data, radius_km=1):
    # Filtering out rows with NaN values in Latitude or Longitude
    data = data.dropna(subset=['Latitude', 'Longitude'])

    coordinates = data[['Latitude', 'Longitude']].values
    tree = KDTree(coordinates)
    neighbours = {}

    for i, row in data.iterrows():
        unit_id = row['PU-Code']
        unit_location = (row['Latitude'], row['Longitude'])
        indices = tree.query_ball_point(unit_location, radius_km / 6371)  # Earth's radius in km
        neighbours[unit_id] = data.iloc[indices]['PU-Code'].tolist()
        neighbours[unit_id].remove(unit_id)  # Remove self from neighbours

    return neighbours
```

Calculating Outlier Scores:

- Outlier scores were calculated for each polling unit and each party by comparing the unit's votes against the average votes of its neighbours.
- The absolute difference between the unit's votes and the average neighbour votes was used as the outlier score.

```python
def calculate_outlier_scores(data, neighbours):
    parties = ['APC', 'LP', 'PDP', 'NNPP']
    outlier_scores = []

    for i, row in data.iterrows():
        unit_id = row['PU-Code']

        # Check if unit_id is in neighbours (to handle cases where some units may not have neighbours due to filtering)
        if unit_id in neighbours:
            neighbour_ids = neighbours[unit_id]
            neighbour_votes = data[data['PU-Code'].isin(neighbour_ids)]

            for party in parties:
                unit_votes = row[party]
                if not neighbour_votes.empty:
                    avg_neighbour_votes = neighbour_votes[party].mean()
                    outlier_score = abs(unit_votes - avg_neighbour_votes)
                else:
                    outlier_score = np.nan

                outlier_scores.append({
                    'PU-Code': unit_id,
                    'party': party,
                    'outlier_score': outlier_score,
                    'neighbours': neighbour_ids
                })
```

Data Integration:

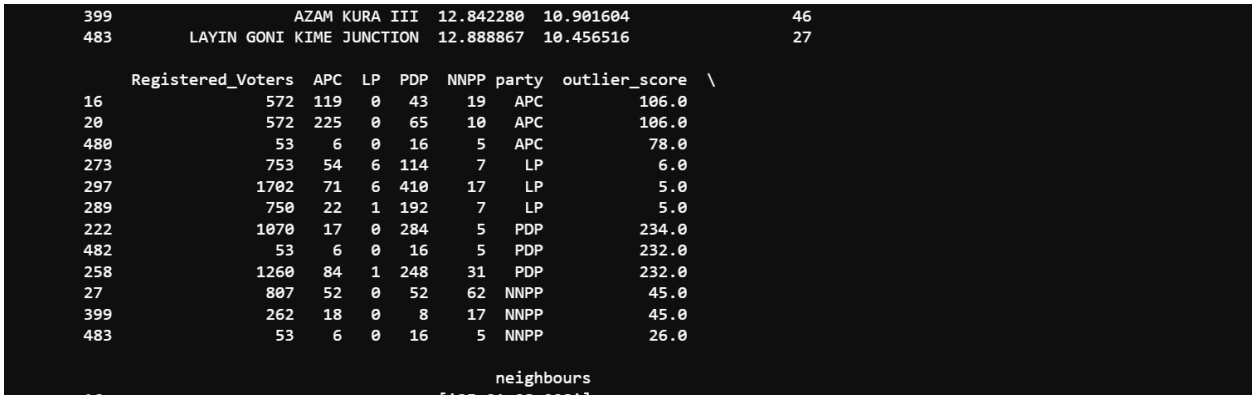
- The outlier scores were merged back into the original dataset for further analysis and visualization.

Summary of Findings

Below is a sorted list of the top three outlier scores for each party. PDP had the highest outlier score overall with 234.0 while LP had the lowest observed outlier score of 6.0

- **APC**:
    1. PU-Code: 35-01-08-007, Outlier Score: 106.0
    2. PU-Code: 35-01-08-008, Outlier Score: 106.0
    3. PU-Code: 35-13-10-014, Outlier Score: 78.0
- **LP**:
    1. PU-Code: 35-14-01-019, Outlier Score: 6.0
    2. PU-Code: 35-14-03-015, Outlier Score: 5.0
    3. PU-Code: 35-14-03-001, Outlier Score: 5.0
- **PDP**:
    1. PU-Code: 35-12-10-006, Outlier Score: 234.0
    2. PU-Code: 35-13-10-014, Outlier Score: 232.0
    3. PU-Code: 35-13-10-001, Outlier Score: 232.0
- **NNPP**:
    1. PU-Code: 35-01-09-008, Outlier Score: 45.0
    2. PU-Code: 35-01-09-018, Outlier Score: 45.0

3. PU-Code: 35-13-10-014, Outlier Score: 26.0



399		AZAM KURA III	12.842280	10.901604					46
483		LAYIN GONI KIME JUNCTION	12.888867	10.456516					27
	Registered_Voters	APC	LP	PDP	NNPP	party	outlier_score	\	
16	572	119	0	43	19	APC	106.0		
20	572	225	0	65	10	APC	106.0		
480	53	6	0	16	5	APC	78.0		
273	753	54	6	114	7	LP	6.0		
297	1702	71	6	410	17	LP	5.0		
289	750	22	1	192	7	LP	5.0		
222	1070	17	0	284	5	PDP	234.0		
482	53	6	0	16	5	PDP	232.0		
258	1260	84	1	248	31	PDP	232.0		
27	807	52	0	52	62	NNPP	45.0		
399	262	18	0	8	17	NNPP	45.0		
483	53	6	0	16	5	NNPP	26.0		
							neighbours		

Detailed Examples of the Top 3 Outliers

Example 1: APC

- **PU-Code**: 35-01-08-007
  - **Outlier Score**: 106.0
  - **Neighbouring Units**: ['35-01-08-008']
  - **Explanation**: The APC votes at this unit were significantly higher than the average votes of its neighbour indicating a potential outlier.

Example 2: LP

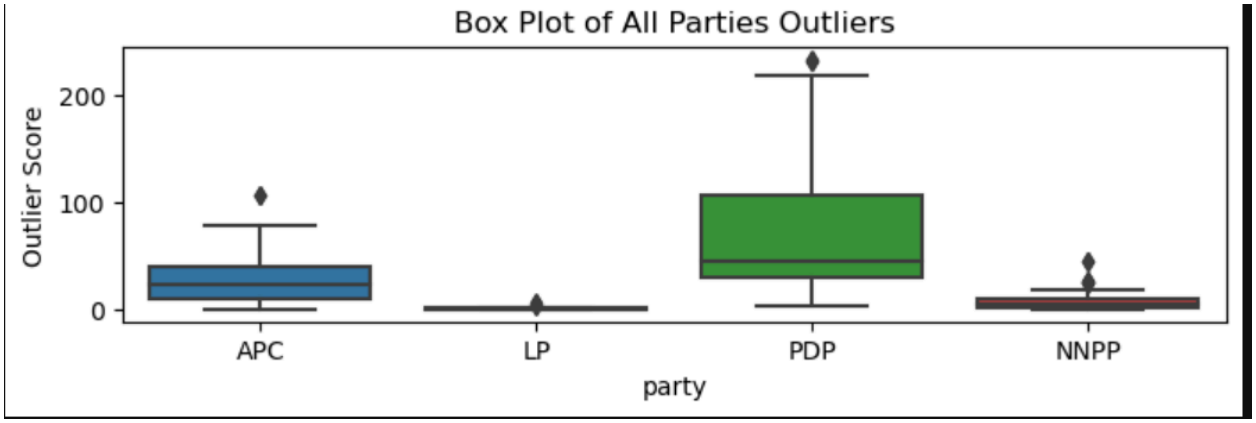
- **PU-Code**: 35-14-01-019
  - **Outlier Score**: 6.0
  - **Neighbouring Units**: ['35-14-01-042', '35-14-01-039', '35-14-01-001']
  - **Explanation**: The LP votes at this unit were slightly higher than the average votes of its neighbours indicating a minor outlier.

Example 3: PDP

- **PU-Code**: 35-12-10-006
  - **Outlier Score**: 234.0
  - **Neighbouring Units**: ['35-12-10-020', '35-12-10-014']
  - **Explanation**: The PDP votes at this unit were significantly higher than the average votes of its neighbours indicating a potential outlier.

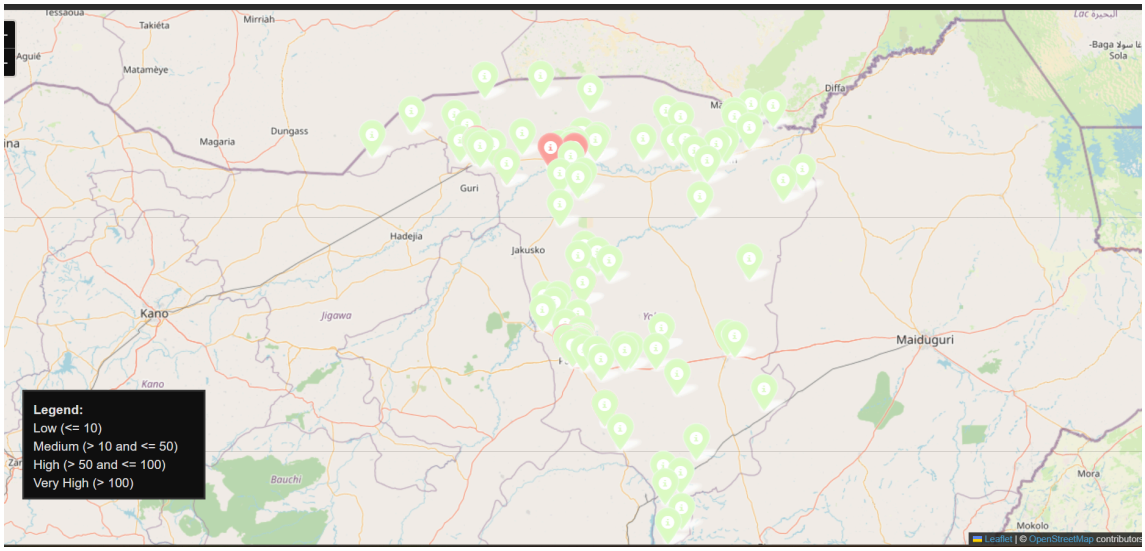
Visualizations

To provide a better understanding of the spatial distribution of the outliers, a box plot showing the outliers in each party is shown.



A map visualization was also created using Folium. The map includes markers for each polling unit, with colors indicating the outlier scores:

- **Green**: Low outlier score (<= 10)
- **Yellow**: Medium outlier score (> 10 and <= 50)
- **Orange**: High outlier score (> 50 and <= 100)
- **Red**: Very high outlier score (> 100)



Conclusion

Summary of Findings

The geospatial analysis and outlier detection methodology applied to the election data identified several polling units with significant outlier scores. These outliers suggest potential irregularities in the voting patterns for the different parties. The top outliers were identified and visualized on a map, providing a clear spatial representation of the areas with notable deviations in vote counts.

Key Insights

- Polling units with high outlier scores, especially those significantly deviating from their neighbours, warrant further investigation to ensure election integrity.
- The visualization aids in quickly identifying and addressing areas with potential irregularities.

Recommendations

- Further investigations should be conducted on the identified outliers to determine the cause of the deviations.
- Continuous monitoring and validation techniques to maintain the integrity of future elections should be implemented

This analysis provides a foundation for improving election data transparency and reliability and it is my hope that it contributes to the body of knowledge around fair and accurate electoral processes.

Thank you.