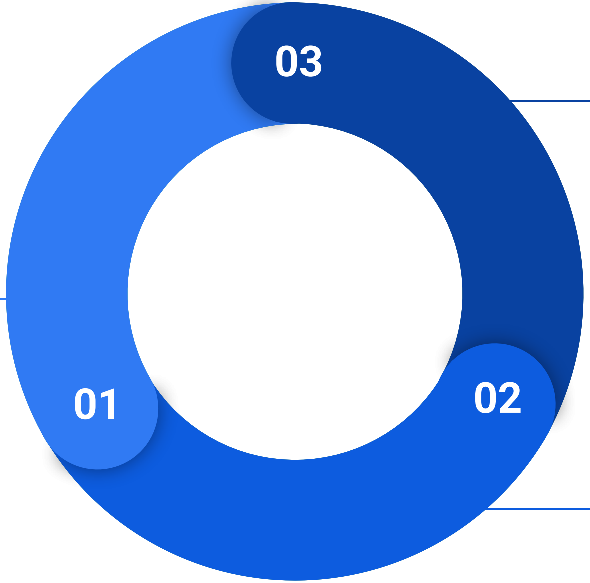
Advanced Geospatial Analysis for Presidential Election Results Integrity in Anambra State

ACSP - Senior Data Analyst - Stage Eight

INTRODUCTION
Primary Dataset Overview



Conclusion and Relevance

Assignment Requirements
1. Enhanced Dataset Preparation
2. Advanced Neighbor Identification
3. Sophisticated Outlier Score Calculation
4. Temporal and Demographic Comparative Analysis
5. Interactive Visualization and Reporting

1. INTRODUCTION

- **Dataset Overview**: The dataset is implemented for this task is [ANAMBRA_crosschecked.csv]
- **Analysis Objective:** Detect outlier polling units to identify potential electoral irregularities

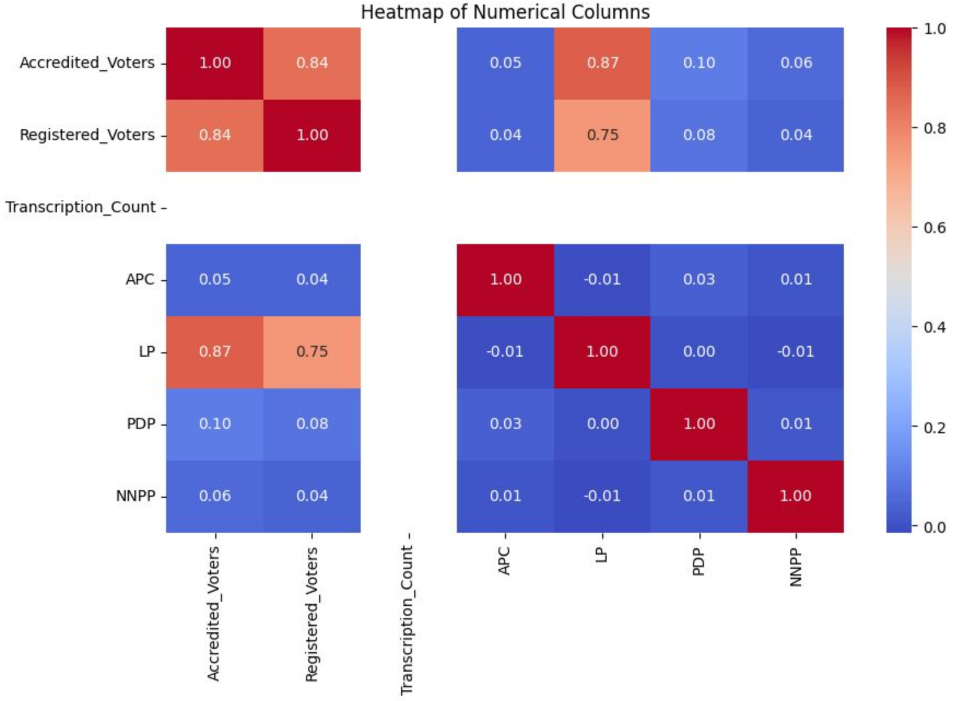
	Accredited_Voters	Registered_Voters	Transcription_Count	APC	LP	PDP	NNPP
count	3679.000000	3679.000000	3679.0	3679.000000	3679.000000	3679.000000	3679.000000
mean	115.236477	450.597445	-1.0	1.260669	103.258766	2.413971	0.556401
std	79.122946	333.768034	0.0	7.910370	77.716562	11.511426	5.703382
min	0.000000	1.000000	-1.0	0.000000	0.000000	0.000000	0.000000
25%	56.000000	203.000000	-1.0	0.000000	45.000000	0.000000	0.000000
50%	103.000000	397.000000	-1.0	0.000000	91.000000	1.000000	0.000000
75%	159.000000	640.500000	-1.0	1.000000	144.500000	2.000000	0.000000
max	582.000000	3770.000000	-1.0	350.000000	574.000000	465.000000	251.000000

1. INTRODUCTION

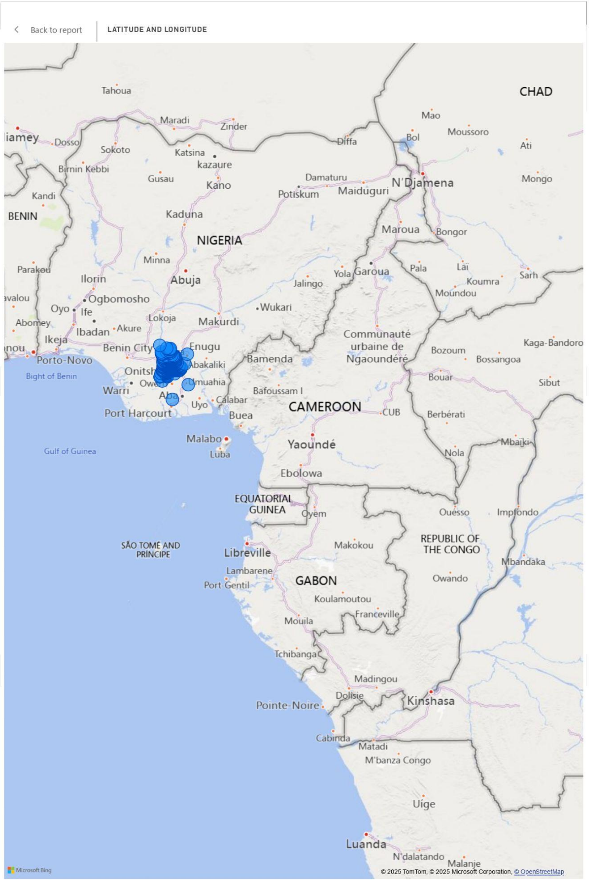
- **Data Loading and Cleaning:**:
  - The dataset was loaded into a Pandas DataFrame
  - No duplicate entries or missing values were found in the dataset.

- **Dataset Shape:** The dataset consists of 3679 rows and 19 columns.

- **Descriptive Statistics:** As shown in the table above;
  - The number of **Accredited Voters** ranges from **0 to 582**, with an average of 115, indicating differences in voter participation across polling units.
  - The **Labour Party (LP)** has the **highest mean votes (103)**, far exceeding other parties
  - **low median values** (0 votes each), indicating that many polling units recorded no votes for APC; NNPP; PDP
- The heatmap shows a high positive correlation (0.84) between Accredited Voters and Registered Voters



1. Enhanced Dataset Preparation:



Geocoding Polling Units

- ❖ **Objective:** Use geocoding techniques to ensure reliable geospatial data
- ❖ **Approach**
  - ➢ **Data Standardization**: Converted column names to lowercase for consistency.
  - ➢ **Geocoding API Integration**: Used Google Maps Geocoding API for location queries.
  - ➢ **Caching Strategy**: Implemented caching to avoid redundant API requests and optimize performance.
  - ➢ **Batch Processing:** Processed data in batches of 100 to prevent API rate limits.
  - ➢ **Manual Adjustments**: Incorporated manually verified coordinates for specific locations.
- ❖ **Implementation Highlights**
  - ➢ **API Query Format:** Combined Polling Unit Name, Ward, LGA, and State to construct search queries.
  - ➢ **Error Handling:** Logged failed requests and implemented fallback mechanisms.
  - ➢ **Automated Saving:** Stored results in a CSV file after each batch to ensure progress tracking.

1. Enhanced Dataset Preparation:

```python
# Google Geocoding API Key
API_KEY = "AIzaSyB4jKVUQN14ExqzVE3IjbKGIDHF-EBLkBo"

# Dictionary to cache API responses and avoid duplicate requests
cache = {}

# Function to get latitude and longitude with caching
def get_lat_lon(state, lga, ward, pu_name, pu_code):
    """
    Queries Google Geocoding API to get latitude & longitude for a polling unit.
    Uses caching to avoid redundant requests.
    """
    pu_identifier = pu_name if pd.notna(pu_name) else pu_code
    query = f"{pu_identifier}, {ward}, {lga}, {state}, Nigeria"

    # Check cache to avoid duplicate API requests
    if query in cache:
        return cache[query]

    url = f"https://maps.googleapis.com/maps/api/geocode/json?address={query}&key={API_KEY}"

    try:
        response = requests.get(url)
        response.raise_for_status()
        data = response.json()

        if data['status'] == 'OK':
            lat = data['results'][0]['geometry']['location']['lat']
            lon = data['results'][0]['geometry']['location']['lng']
            cache[query] = (lat, lon)  # Store result in cache
            return lat, lon
        else:
            print(f"Geocoding failed for {query}: {data['status']}")
            return None, None
    except Exception as e:
        print(f"Error fetching coordinates for {query}: {e}")
        return None, None


# Add Latitude and Longitude columns with progress tracking
batch_size = 100  # Save every 100 rows
output_file = "polling_units_with_coordinates.csv"

#.loc[] did not update correctly in a loop. using .at[] to fix it
for i in range(0, len(df), batch_size):
    print(f"Processing rows {i} to {i + batch_size}...")

    for j in range(i, min(i + batch_size, len(df))):
        lat, lon = get_lat_lon(df.at[j, 'state'], df.at[j, 'lga'], df.at[j, 'ward'], df.at[j, '
        df.at[j, 'latitude'] = lat
        df.at[j, 'longitude'] = lon

    # Save progress after each batch
    df.to_csv(output_file, index=False)
    time.sleep(1)  # Prevent API rate limiting
```

Geocoding Polling Units

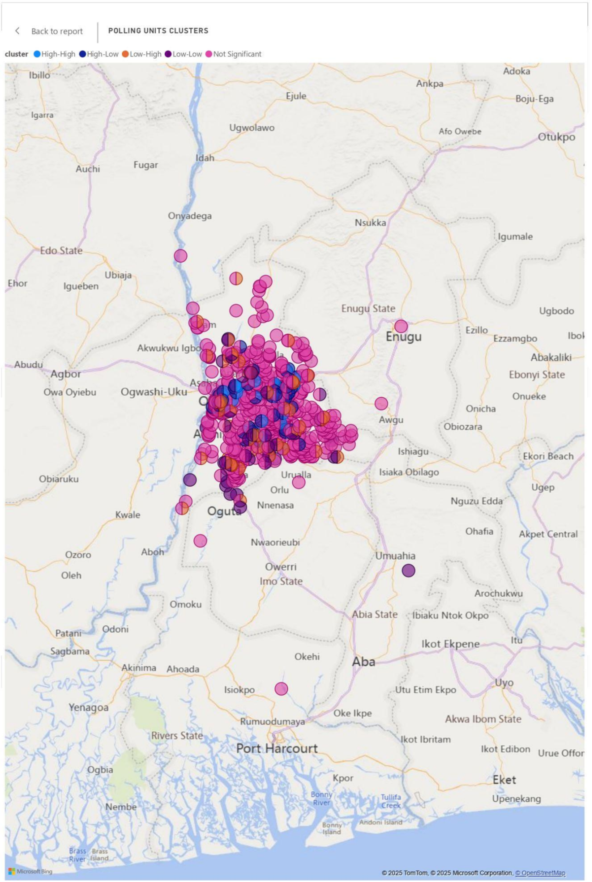
- **Challenges & Solutions**
  - **Two Missing or Ambiguous Locations:** Used known reference points from google map and manual corrections with python code.
  - **API Rate Limits**: Added time delays between requests to avoid throttling.
- The result is given in the map above, with geolocation point to Anambra State for each polling units
- The code snippet below was used to generate the latitude and longitude for each polling units.

2. Advanced Neighbor Identification

Geospatial Clustering of Polling Units

- **Objective**
  - Utilize DBSCAN (Density-Based Spatial Clustering of Applications with Noise) for dynamic cluster detection.

- **Methodology**
  - **Data Preprocessing**
    - Extracted latitude and longitude from the dataset.
    - Standardized coordinates using **StandardScaler** to improve clustering efficiency.
  - **Clustering Approach**
    - Applied **DBSCAN** with:
      - **Epsilon (ε) = 0.1** (defines neighborhood distance).
      - **Min_samples = 5** (minimum points required to form a cluster).
    - Identified 116 distinct clusters, with 196 noise points (-1 label).
    - **Cluster Labeling**
      - Classified clusters into:
        - **High-High (hotspots)**
        - **Low-Low (coldspots)**



2. Advanced Neighbor Identification

Geospatial Clustering of Polling Units

- **High-Low (outliers)**
- **Low-High (outliers)**
- **Not Significant** → Areas with no strong spatial clustering or pattern

❖ Major clusters include:

➢ **Cluster 65:** 671 polling units

➢ **Cluster 38:** 314 polling units

➢ **Cluster 63**: 217 polling units

❖ Key Insights

➢ Densely populated clusters indicate potential areas requiring electoral logistics optimization.

➢ Outliers (High-Low, Low-High) as displayed in the map suggest locations that need further investigation for data accuracy or special electoral considerations.

```
#loading the geo dataframe
df = pd.read_csv("/content/Geo_Anambra_data.csv")

# Extract latitude and longitude
coords = df[['latitude', 'longitude']].values

# Standardize features
scaler = StandardScaler()
coords_scaled = scaler.fit_transform(coords)

# Apply DBSCAN for clustering
epsilon = 0.1  # Adjust based on desired proximity (in normalized scale)
min_samples = 5  # Minimum points to form a cluster
clustering = DBSCAN(eps=epsilon, min_samples=min_samples, metric='euclidean').fit(coords_scaled)

df['cluster'] = clustering.labels_

# Debugging: Check number of clusters
num_clusters = len(set(clustering.labels_)) - (1 if -1 in clustering.labels_ else 0)
print(f"Identified clusters: {num_clusters}")
print(df['cluster'].value_counts())
```

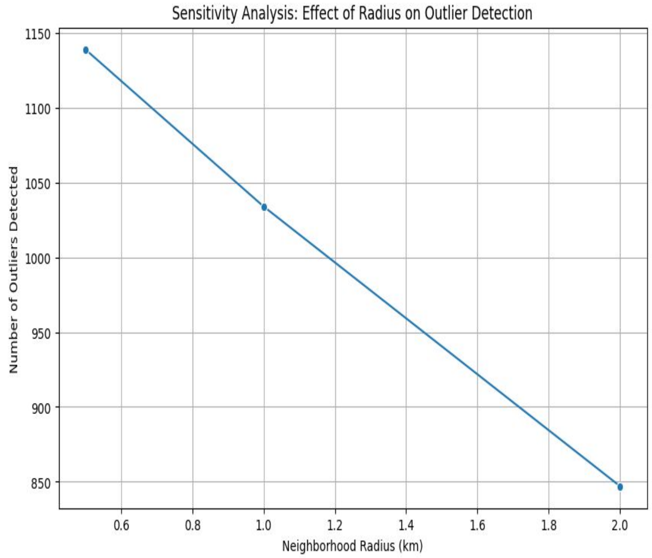
2. Advanced Neighbor Identification

Sensitivity Analysis: Effect of Neighborhood Radius on Outlier Detection

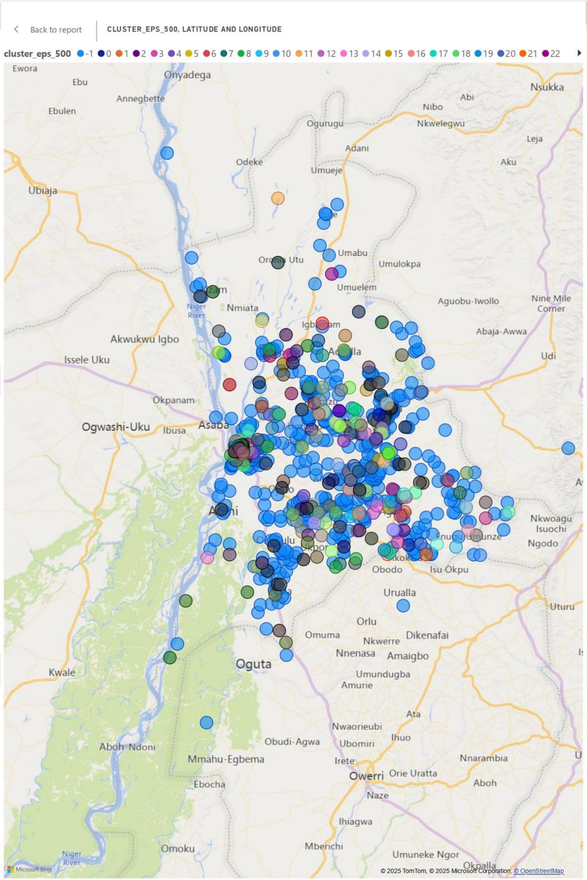
- **Objective**
  - valuate how varying neighborhood radii impacts outlier detection in polling unit clustering

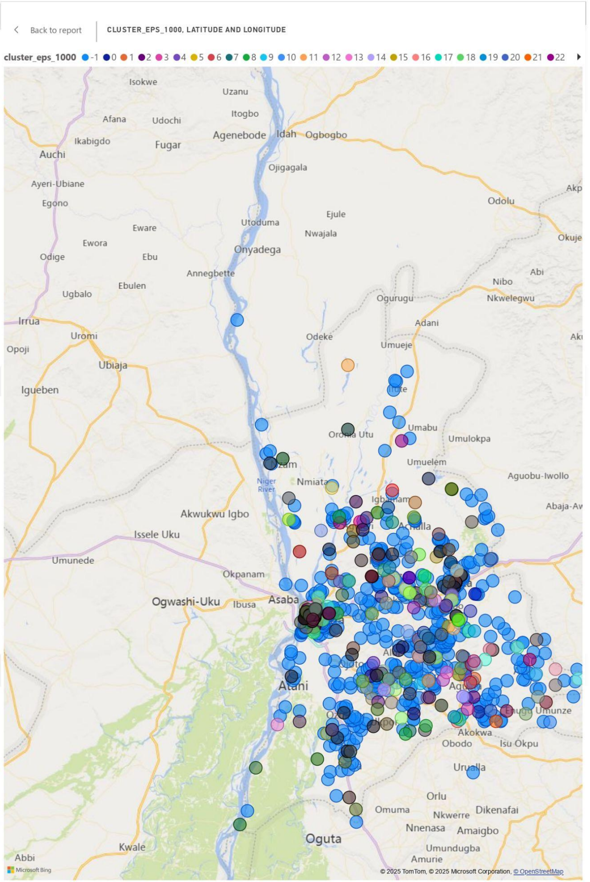
- **Methodology**
  - Standardized latitude and longitude coordinates for uniform scaling.
  - Applied DBSCAN clustering with three neighborhood radii:
    - Applied **DBSCAN** with:
      - **ε = 0.005 (500m)**
      - **ε = 0.01 (1km)**
      - **ε = 0.02 (2km)**
  - Counted outliers (noise points labeled -1) for each radius.

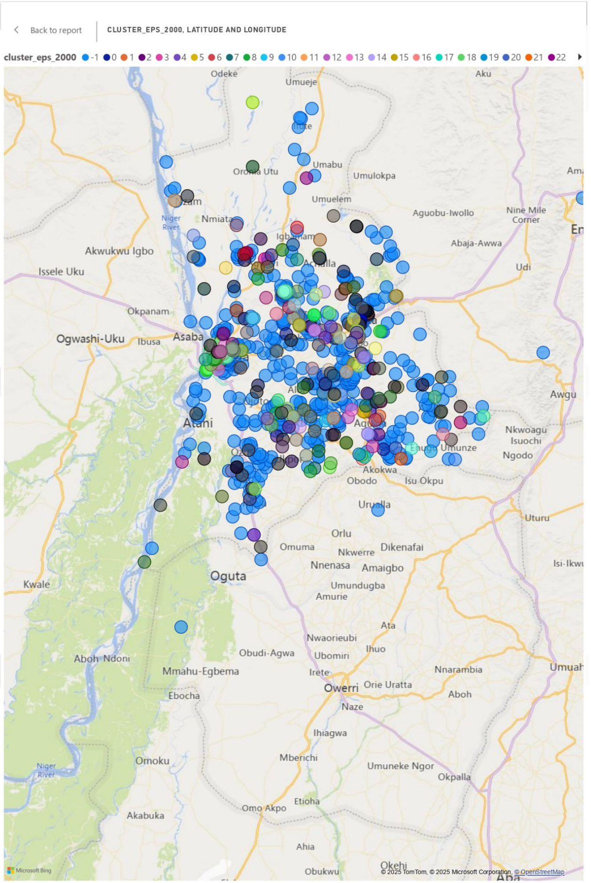
- Line plotted trend
  - As the radius increased, the number of outliers decreased:
    - Smaller radius → More outliers (more restrictive clustering)
    - Larger radius → Fewer outliers (clusters absorb more points).



2. Advanced Neighbor Identification







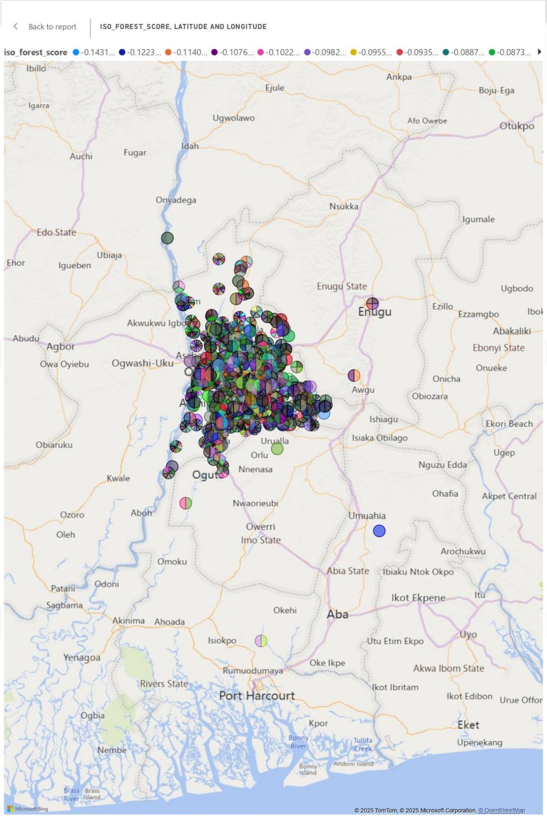
3. Sophisticated Outlier Score Calculation

Local Moran's I Outlier Score Calculation

- ❖ **Objective**
  - ➢ Compute outlier scores for polling units using Local Moran's I to detect spatial pattern

- ❖ **Methodology**
  - ➢ **Convert DataFrame to GeoDataFrame for spatial analysis.**
  - ➢ **Standardize numerical variable (e.g., accredited voters) using Z-score normalization.**
  - ➢ **Construct K-nearest neighbors (K=6)** spatial weights matrix to define local spatial relationships.
  - ➢ Compute Local Moran's I to assess localized spatial autocorrelation.

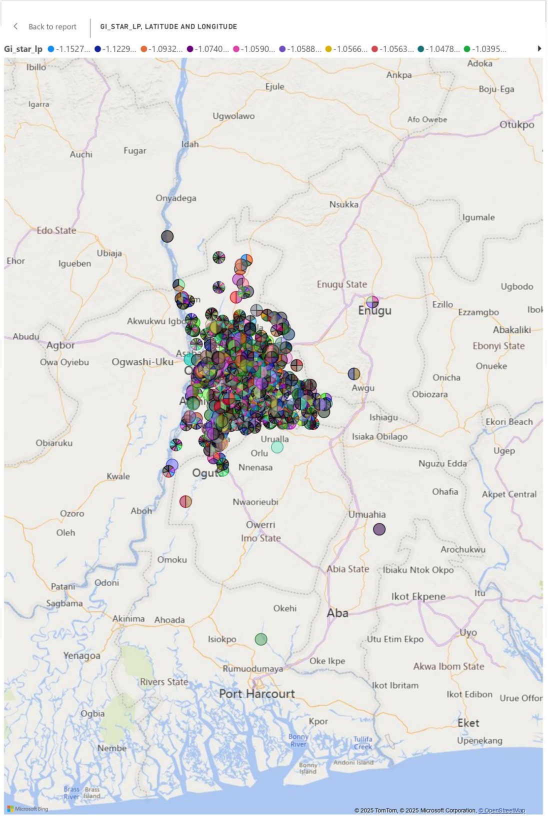
- ❖ Key Fi**ndings from the map:**
  - ➢ Sum of Outlier Scores: 592.84 (Total Local Moran's I sum)
  - ➢ Spatial autocorrelation in the map revealed significant clustering and outlier patterns across polling units.



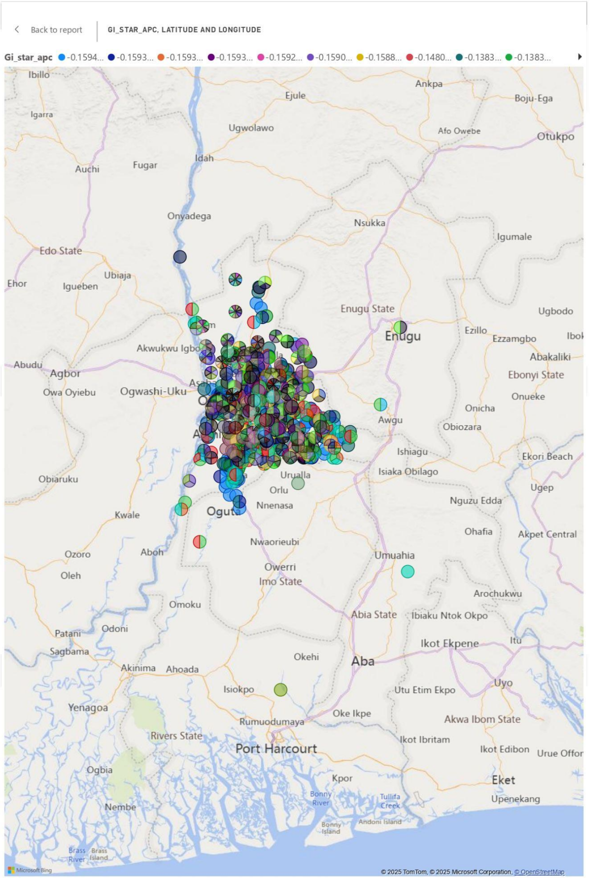
3. Sophisticated Outlier Score Calculation

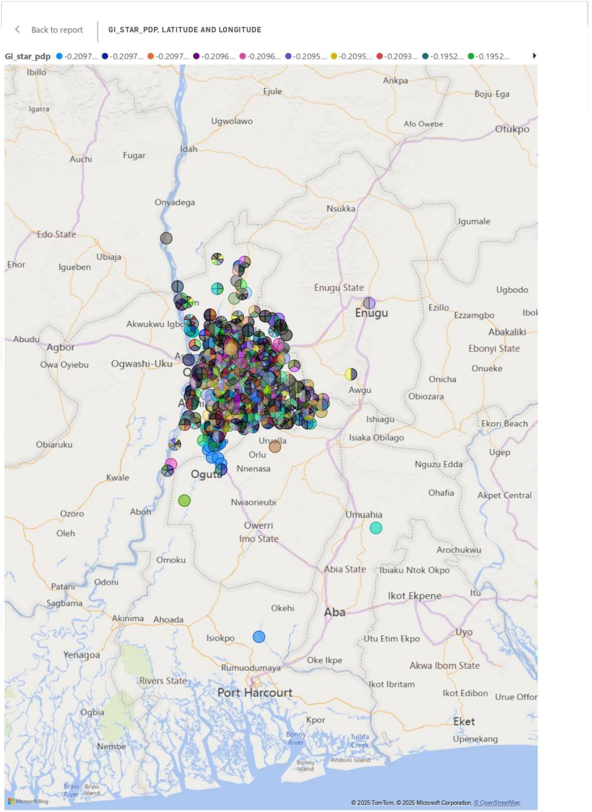
Getis-Ord Gi Outlier Score Calculation

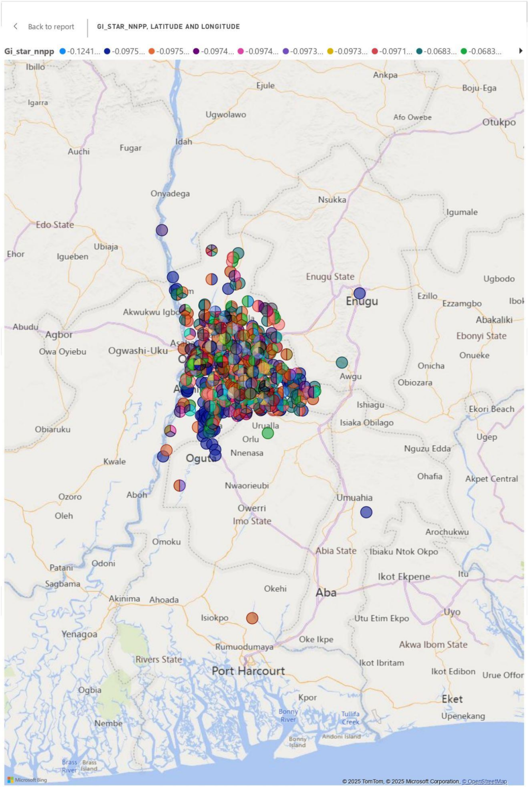
- ❖ **Objective**
  - ➢ Identify areas with statistically significant hotspots (high vote concentration) and coldspots (low vote concentration).
- ❖ **Methodology**
  - ➢ **Selected Vote Columns: "apc", "lp", "pdp", "nnpp" for analysis**
  - ➢ **Converted vote data to float format to ensure compatibility with statistical computations.**
  - ➢ **Constructed K-nearest neighbors (K=6)** spatial weights matrix for spatial context.
  - ➢ **Applied Getis-Ord Gi* to compute:**
    - ■ **Gi* Z-score** → Measures spatial clustering of high or low values.
    - ■ **p-value** → Determines statistical significance of clustering.
- ❖ Key Fi**ndings from the map:**
  - ➢ Significant hotspots and coldspots identified across different parties ,map(APC, LP, PDP, NNPP).
  - ➢ Patterns reveal spatial voting trends, which can inform electoral strategy and resource allocation.



3. Sophisticated Outlier Score Calculation



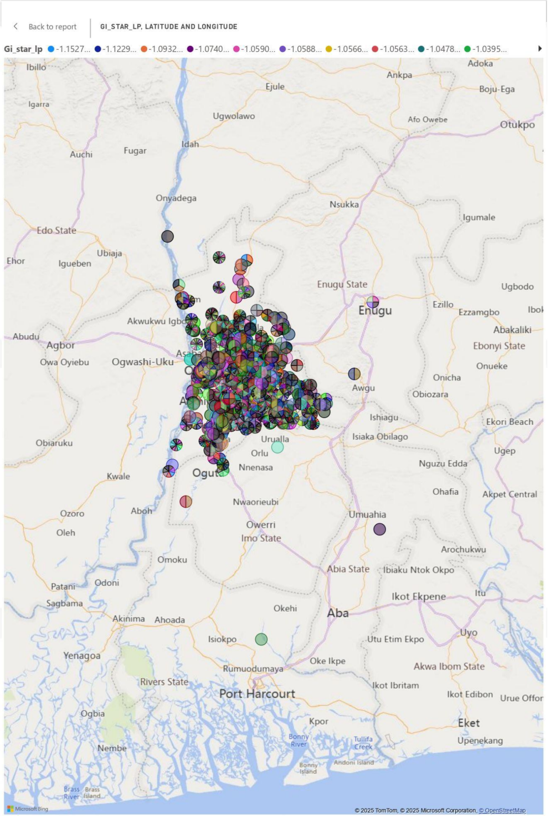




3. Sophisticated Outlier Score Calculation

Iso Forest Anomaly detection

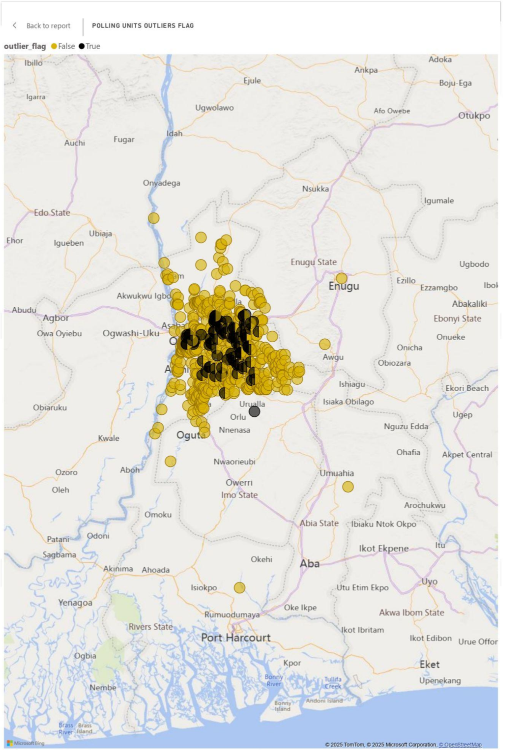
- ❖ **Objective**
  - ➢ Identify anomalous polling units using robust spatial statistical methods.
- ❖ **Methodology**
  - ➢ Selected relevant numerical features including latitude, longitude, vote counts, and spatial clustering metrics.
  - ➢ **Configured Isolation Forest with:**
    - ■ 100 estimators
    - ■ 5% contamination level (assumes ~5% anomalies)
    - ■ Random state = 42 for reproducibility
- ❖ Key F**indings from the map:**
  - ➢ Successfully assigned outlier scores to each polling unit.
  - ➢ Identified potential anomalies based on significantly low scores.



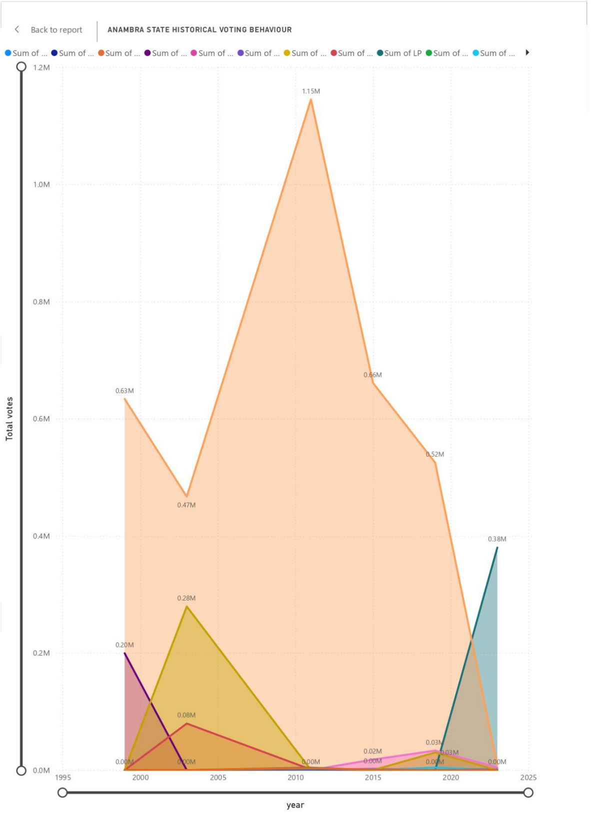
3. Sophisticated Outlier Score Calculation

Cross-Validation of Geospatial Outlier Detection

- ❖ **Objective**
  - ➢ Enhance the robustness of anomaly detection by combining multiple geospatial techniques.
- ❖ **Methodology**
  - ➢ Standardized anomaly scores from:
    - ■ Local Moran's I (Spatial autocorrelation)
    - ■ Gi (Getis-Ord)* (Hotspot detection)
    - ■ Isolation Forest (Machine learning-based anomaly detection)
  - ➢ Min-Max Scaling applied to normalize scores for comparability
  - ➢ **Aggregated Scores**: Computed a Combined Outlier Score by averaging the normalized values
  - ➢ Defined an outlier threshold (95th percentile) to flag extreme anomalies
- ❖ Key Fi**ndings from the map:**
  - ➢ The map Identified high-confidence anomalies where multiple techniques agreed..
  - ➢ Reduced false positives by ensuring only consistently detected outliers were flagged.



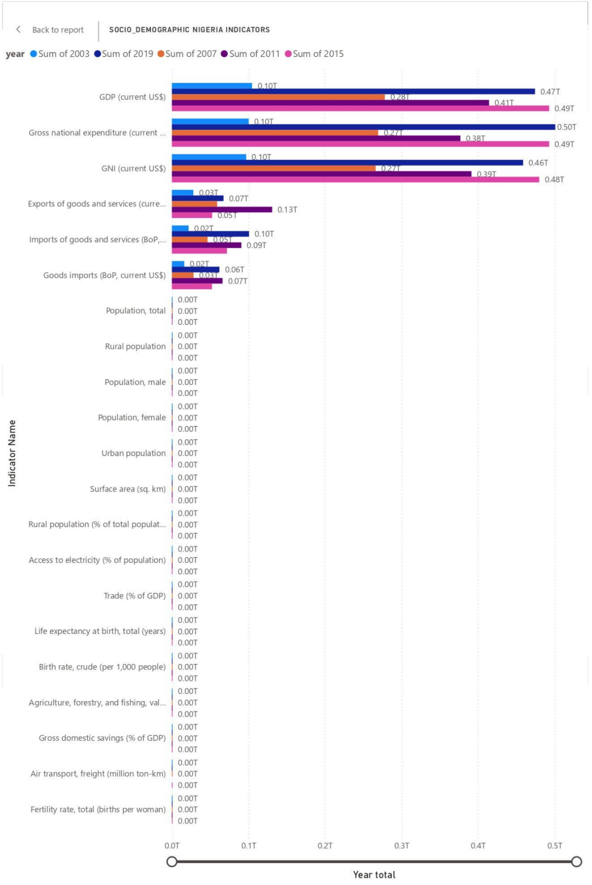
4. Temporal and Demographic Comparative analysis



Historical comparison on 1999 to 2023 Anambra presidential Election results by parties.

- **Dataset:** The dataset was sourced from Kaggle,
  - The presidential dataset were filtered to only Anambra state and merged alongside corresponding aggregates from 2023.
- **Key Findings:**
  - **Peak Turnout Around 2011**: Voting totals reached their highest level around 2011, with PDP party having the highest vote above 1.1 million.
  - **Subsequent Decline:** After 2011, there is a noticeable drop in total votes, suggesting either reduced voter turnout, voter apathy, or shifts in party popularity
  - **Dominant Party Performance:** PDP consistently led in vote share through the mid-2000s to early 2010s, before experiencing a decline
  - **Emergence of New Parties:** LP show a rise in votes in later years, indicating evolving political preferences and increased competition.
  - **Shifting Alliances/Preferences: T**he gradual changes in vote shares suggest that voter allegiance may be fluid, with parties gaining or losing ground over time.

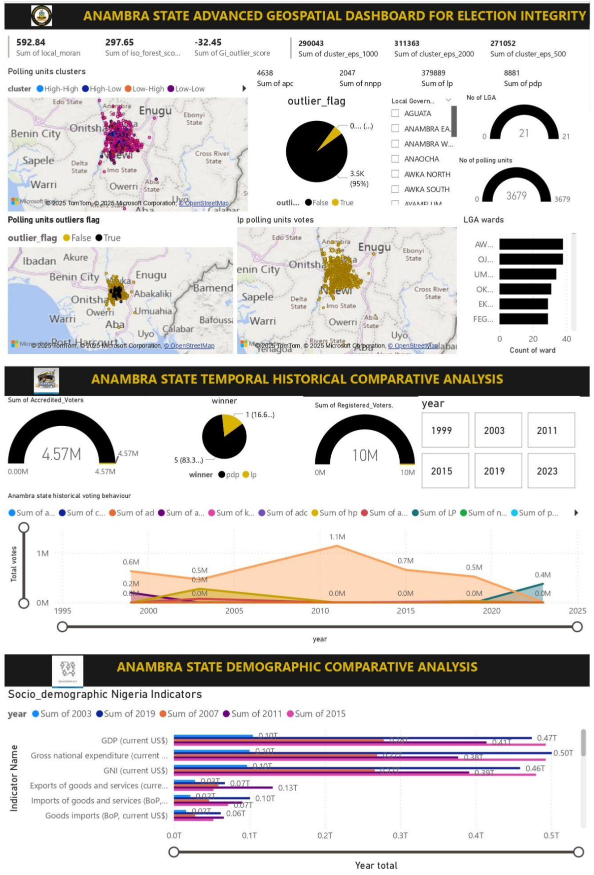
4. Temporal and Demographic Comparative analysis



Nigeria 2003 to 2019 Demographic Data.

- **Dataset:** The dataset was sourced from Github,
- **Key Findings:**
  - Rising GDP and GNI suggest increased economic activity, potentially impacting voter choices through policy preferences, job creation, or social welfare improvements.
  - Growing urban populations and a corresponding decrease in the rural share point to changing voter distributions.
  - Notable rises in both imports and exports indicate deeper integration into global trade, potentially shaping voter preferences on trade policies and economic reforms.
  - Improvements in life expectancy and shifts in fertility rates reflect changing socio-demographic conditions that may alter the electorate's policy priorities (e.g., healthcare, education).
  - These socio-economic shifts can create new or evolving voting blocs—urban youth, middle-class professionals—leading to different party alignments and potentially unexpected electoral results.

5. Interactive Visualization and Reporting

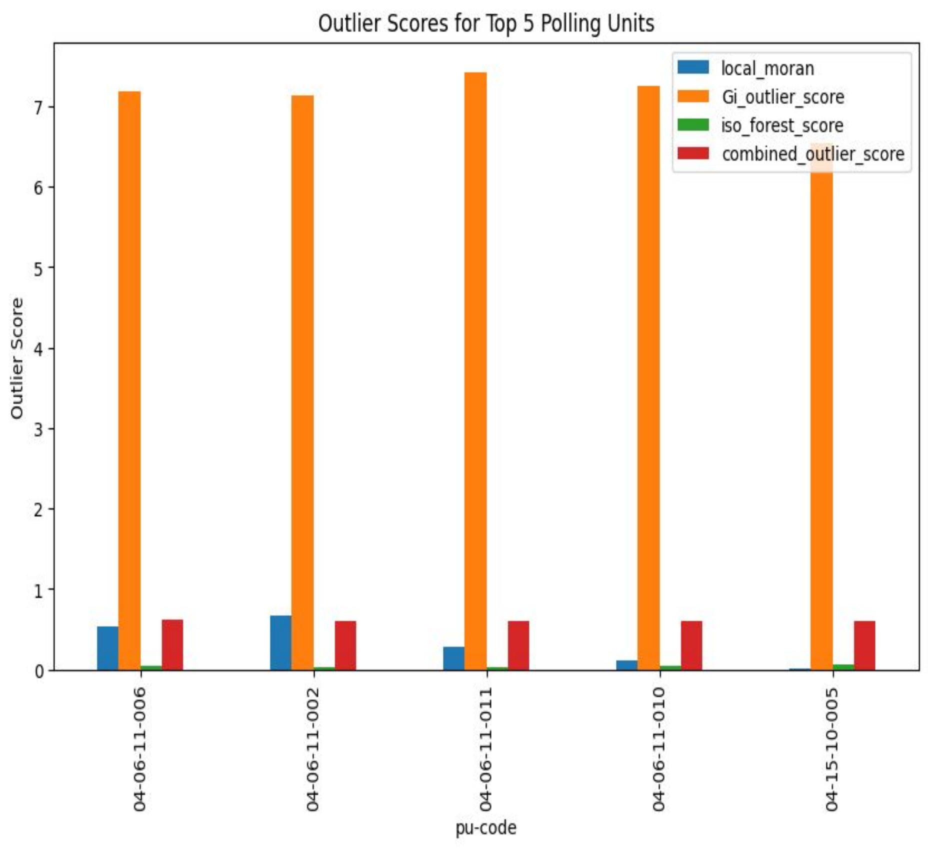


❖ **Dashboard Key Findings:**

- ➢ Geospatial Outlier Distribution
  - ■ The map highlights specific polling units flagged as outliers, indicating potential irregularities that warrant closer investigation.
- ➢ Regional Hotspots: Out of the 21 LGA in the dataset, Certain LGAs show a higher concentration of flagged polling units, suggesting spatial clustering of unusual voting behaviors.
- ➢ Historical Voting Trends
  - ■ Peak in Voter Turnout:
  - ■ ‘Shifting Party Dynamics:.
- ➢ Socio-Demographic Indicators
  - ■ Economic Growth: GDP, GNI, and government spending trends are generally upward, implying an expanding economy that could shape voter priorities.
  - ■ Population & Urbanization: Growing urban populations may correlate with different voting patterns, highlighting the importance of demographic shifts in explaining electoral outcomes.

Conclusion

- **Multiple Outlier Flags:** Through combined_outliers score, the top 5 polling units flag as outliers, have been identified in the bar chat.
- **Temporal Insights:** Historical voting data revealed fluctuations in turnout and party dominance over different election cycles, highlighting the dynamic nature of voter behavior.
- **Actionable Insights:** These findings enable stakeholders—election bodies, policymakers, and civil society—to target interventions in flagged regions, improve transparency, and ensure more credible electoral outcomes in future elections.



End of Report

[Link to Jupyter notebook]
[Link to Temporal Dataset]
[Link of the Demographic Dataset]