**Ryan Fallon Assignment 3: Integer Programming Example---Algorithmic Redistricting**

**Data Sources (30 points)**
I used New Jersey's county-level demographic data, including total population figures, sourced from the U.S. Census Bureau (https://www.census.gov/quickfacts/fact/table/NJ_COUNTY_NAME) and GIS data for county boundaries (https://njogis-newjersey.opendata.arcgis.com/datasets/newjersey::county-boundaries-of-nj/explore). For adjacency, the county adjacency data from census archives (e.g., https://www2.census.gov/geo/docs/reference/county_adjacency.txt) was used to ensure accurate mapping of neighboring counties. The use of July 1, 2023 population data was critical to align with the most recent census information available to me. A potential next step would be incorporating data from historical election results to evaluate the political distribution of districts, although this was not part of the current implementation.

**Specification (Objective function and Constraints) (30 points)**
My main objective function was designed to maximize the compactness of the districts while ensuring they met population balance criteria. Each county was assigned to one district, aiming for equal representation (one-person-one-vote) with allowable deviations to account for demographic variances. Adjacency constraints ensured that each county assigned to a district was contiguous with at least one other county in the same district. This approach naturally supported the creation of compact districts. While racial or political balance was not a primary focus, future iterations could include constraints to maintain or analyze racial balance (e.g., percentage white alone versus other races) and voter distribution to align with proportional representation. I couldn’t get redistrict.py to find an optimal solution with 12 districts, so I moved forward with 6 districs such that each larger population segment gets 2 votes proportionally.

**Programming (30 points)**
The project was implemented using Python's PuLP, which facilitated formulating and solving the integer programming problem. The code accounted for the set partitioning problem where each county was assigned to a single district. The constraints were built to handle adjacency checks, population balance, and compactness. The results were visually represented with the help of GeoPandas and Matplotlib, ensuring that all findings were well-documented in my GitHub repository.

**Solution (30 points)**
The resulting map displayed six districts with populations ranging from 1,489,131 to 1,627,040 people, meeting the one-person-one-vote standard with less than 5% deviation. All counties were assigned to districts in a manner that balanced population and adhered to adjacency constraints. While the solution was mathematically optimal and passed validation for contiguity and population balance, there is room to explore secondary goals, such as political representation and racial balance, in future iterations. These additional

considerations would provide a more comprehensive view of the districts' fairness and equity. Project yields the following results:

```
Atlantic is assigned to district 2
Bergen is assigned to district 4
Burlington is assigned to district 1
Camden is assigned to district 1
Cape May is assigned to district 2
Cumberland is assigned to district 2
Essex is assigned to district 0
Gloucester is assigned to district 2
Hudson is assigned to district 3
Hunterdon is assigned to district 1
Mercer is assigned to district 1
Middlesex is assigned to district 5
Monmouth is assigned to district 5
Morris is assigned to district 0
Ocean is assigned to district 2
Passaic is assigned to district 4
Salem is assigned to district 2
Somerset is assigned to district 3
Sussex is assigned to district 0
Union is assigned to district 3
Warren is assigned to district 0
Status: Optimal
```

With the following populations by district:

```
District 0: 1622924 people
District 1: 1508217 people
District 2: 1555107 people
District 3: 1627040 people
District 4: 1489131 people
District 5: 1506422 people
```

**Maps and Discussion (30 points)**

The output map provided a clear visualization of the districts, color-coded by population to highlight the distribution across the state. This visualization confirmed that the algorithm effectively created compact and contiguous districts. When comparing this solution with manual redistricting or other redistricting tools like Districtr, my solution stands out for its strict adherence to population balance and geographic compactness, reducing potential gerrymandering concerns. While the map aligns with the one-person-one-vote principle (Evenwel v. Abbott, 2016), additional feedback from stakeholders, such as community organizations or political analysts, would help ensure the plan's fairness extends beyond just numeric optimization.