Data Cleaning Process for Markov Chain Analysis

Anna Marie Vagnozzi

This document details the process used to generate the necessary data files for the Markov chain analysis used to assess 2021 election maps for partisan bias. All files referenced can be found in a GitHub repository at `https://github.com/vagnozzia408/gerrymandering_public` in the `redistricting_2021/` subdirectory. All references to files in this document will be relative to this subdirectory.

1 Required Software

All operations described in this document are performed in ArcGIS Pro 2.8.3, Excel, Python 3, Jupyter Notebook, and Notepad++.

2 Required Data

2.1 District Maps

Districting plans are most commonly shared as *block assignment files* that assign each census block to a single district. Such files can often be obtained from state legislative pages or from groups (such as the League of Women Voters) that may be proposing a districting plan. Shapefiles for census blocks can be downloaded from the U.S. Census Bureau at `https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html`.

To generate a *district shapefile*, locate the block assignment file (`.csv` or `.xlsx`) and the census block shapefile (`.shp`), then perform the following steps in ArcGIS Pro.

1. Add the census block shapefile and the block assignment file to a current project.
2. Use the Join function to join the block assignment file to the shapefile using the GEOID field.
3. Use the `Dissolve` geoprocessing tool to dissolve features by the district ID field.
4. Use the `Feature Class to Shapefile` geoprocessing tool to export the resulting feature class as a shapefile to the appropriate directory.

2.2 Voting Data

In South Carolina, voting data by precinct can be found publicly at `https://www.scvotes.gov/`, the website of the South Carolina Election Commission. For the Markov chain analysis, a proxy for Republican/Democratic voter preferences must be chosen; a detailed description of how to choose

an appropriate proxy can be found in the master's thesis associated with this project[1].

John Ruoff provided a cleaned `.xlsx` version of relevant voting data by precinct as reported by SCVotes. The precincts correspond to those in place at the time of the 2020 General Election. The U.S. Senate race was used as a proxy for voter preference in this analysis, so for each precinct, number of votes cast for the Democratic Senate candidate and the Republican Senate candidate were reported.

2.3 Precinct Shapefile

A *precinct shapefile* must be obtained that corresponds to the General Election that will be used as a proxy for voter preferences, as voting data must have a one-to-one match with each precinct.

Precinct shapefiles can be obtained through correspondence with the South Carolina Revenue and Fiscal Affairs Office; however, these maps are typically prone to error and difficult to clean. A more reasonable choice for a starting precinct map is to use a shapefile of *Census Voting Districts (VTDs)*. Though it may not offer an exact correspondence to the precincts listed by SCVotes for a given election, it usually has fewer errors and can be more easily cleaned. This VTD shapefile was provided by John Ruoff and can be found in the GitHub repository referenced at the top of this document.

The majority of this document pertains how to clean the precinct shapefile in order to generate the input file for the Markov chain code.

3 Precinct Cleaning Process

This process begins with the VTD Shapefile provided by John:
`sc_maps/census_VTDs_2021/Voting_District_2021-09-22.shp`

Before proceeding, create a geodatabase to store all intermediate ArcGIS data.

3.1 Create a Map Bounding Box

First get the VTD shapefile loaded into ArcGIS Pro.

1. Export the VTD shapefile as a Feature Class to have full access to the range of ArcGIS functionality. It's a good idea to keep track of map versions if you ever need to go back a previous step in the process, so save the Feature Class in your geodatabase with a version number (`v0`).

2. (Optional) Delete unnecessary fields of data that will not be utilized in the analysis. This tends to speeds up ArcGIS functionality. (E.g. In the 2021 analysis, I kept only the fields ID, VTD, COUNTY, STATE, NAME, POPULATION, Shape_Length, and Shape_Area.)

A polygon must be created around the existing map in ArcGIS Pro to manage precincts along the outer border of the state. To create this bounding box, first create a copy of the `v0` map as a Feature Class. Then, in this copied map, perform the following:

[1] Vagnozzi, Anna Marie, "Detecting Partisan Gerrymandering through Mathematical Analysis: A Case Study of South Carolina" (2020). *All Theses.* 3347. Available at `https://tigerprints.clemson.edu/all_theses/3347`.

1. Use the `Merge` tool under the `Edit` tab to merge all features into a single polygon. The result will be a single-polygon map of South Carolina.

2. Use ArcGIS to create a new polygon with a hole in the middle that fully contains the state polygon. The shape of the outer polygon is not important.

3. Use the `Align Edge` tool under the `Edit` tab to snap the edge of the hole (inner border of the new polygon) with the outside of the state polygon.

4. Delete the state polygon in the center. This will leave a single polygon with a SC-shaped hole in the middle.

5. Use the `Merge` *geoprocessing tool* (separate from the button on the `Edit` tab) to merge this copied map Feature Class to the original `v0` map. Save the resulting map as `v1`. This map's attribute table will have one record per precinct plus one for the outer state polygon.

3.2 Resolve Geographic Errors

Associated files for this step in the process can be found in the GitHub repository under the sub-directory `1_resolve_map_discrepancies/`.

To check for **gaps and overlaps** between precincts:

1. Run the `Polygon to Line` geoprocessing tool on the `v1` map.

2. In the resulting `LEFT_FID` field, check to see if any of the cells equal -1. There should only be one record with this value: the polygon for the outer state.

3. If more than one record has a `LEFT_FID` field with -1, it means that the associated polygon has a border that is present at a gap or an overlap between precincts. Gaps and overlaps must be resolved before proceeding. If the number of errors is small, this can be done by manually adjusting precinct boundaries using the `Edit` tab. More sophisticated processes may be needed if there is a large number of errors.

Multipart precincts occur if a precinct is made up of two or more non-contiguous polygons. (Point-contiguity is considered non-contiguous in this context.) There must be no multipart precincts to run the Markov chain analysis. To check for and resolve multipart precincts:

1. Run the `Multipart to Singlepart` geoprocessing tool on the `v1` map. This will generate a new Feature Class.

2. Check the resulting number of records in the new Feature Class against the number of records in the `v1` map. If the new Feature Class has a greater number of records, there is at least one precinct made up of multiple polygons (resulting in more than one record for that precinct).

3. To identify where these occur, use the `Table to Excel` tool to export the attribute table of the new Feature Class generated by `Multipart to Singlepart`.

4. In Excel, highlight duplicate values of the `VTD` field (or other unique identifier) to locate the multipart precincts.

5. Create a copy of the `v1` map and save it as `v2`. This map will contained resolved multipart precincts. (See `Multipart_to_Singlepart_r2.xlsx`.)

6. Referencing the Excel sheet to locate the multipart polygons, use the tools under the `Edit` tab in ArcGIS to appropriately edit polygon vertices and adjust precinct boundaries in the `v2` map.

Reasonable judgment calls will need to be made regarding how to adjust the boundaries in a manner that does not drastically impact compactness, precinct neighbors, or other relevant variables. A record of changes made to the 2021 SC precinct map are detailed below for reference.

- **VTD 45011000012 (Snelling):** Removed small area contained within the neighboring SRS precinct.
- **VTD 45015000038 (Sangaree 1):** Removed small sliver contained in the neighboring Discovery precinct to the south.
- **VTD 45015000040 (Sangaree 3):** Removed sliver to the east beside the neighboring Seventy Eight precinct.
- **VTD 45021000002 (Alma Mill):** Removed self-intersection in the southwest region of the precinct.
- **VTD 45035000047 (Newington):** Removed tiny piece in southeast corner of neighboring Newington 2 precinct.
- **VTD 45041000063 (West Florence 1):** Removed tiny piece in south-southeast region by the south bordering precinct.
- **VTD 45059000006 (Laurens 6):** Removed one small piece in the western region. The larger piece on the northern region was too large to justify removal, so vertices were edited to connect the two regions.
- **VTD 45063000043 (Pine Ridge 1):** Removed sliver to the south of the south-bordering precinct, Pine Ridge 2. Also edited corresponding border of Gaston 2.
- **VTD 45063000055 (Cayce 2A):** Removed small piece on northern region.
- **VTD 45071000036 (Prosperity City):** Removed tiny piece to the north.
- **VTD 45075000118 (Suburban 8):** Removed three small pieces in the southeast region.
- **VTD 45077000101 (Stone Church):** Removed small piece contained in neighboring Friendship precinct.
- **VTD 45077000102 (University):** Removed small piece at southeast region.
- **VTD 45089000011 (Hemingway):** Two roughly equally sized regions were contiguous only at a point, so the vertices were edited to more fully connect the two regions.
- **VTD 45091000056 (Anderson Road):** Removed sliver to the north of the north-bordering precinct, Celanese.
- **VTD 45091000079 (Hopewell):** Removed a sliver far to the southeast of the precinct.
- **VTD 45091000086 (Springdale):** Removed a sliver far to the north of the precinct.

Once all geographic errors have been resolved, re-run the two geoprocessing tools to ensure that no new errors have been created. `Polygon to Line` should result in only one record with `LEFT_FID=-1`, and `Multipart to Singlepart` should result in the same number of records as the `v2` map, indicating no multipart precincts.

For the 2021 data, this process resulted in a `v2` map that contained 2,269 polygons (2,268 precincts plus the outer state polygon).

3.3 Resolving VTD—Precinct Discrepancies

Associated files for this step in the process can be found in the GitHub repository under the subdirectory `1_resolve_map_discrepancies/`.

Now you can resolve discrepancies between the VTD precincts and the voting data precincts. (This will allow you to join the voting data to the shapefile.) Below is the most straightforward way I have found to do this.

1. Use the `Table to Excel` geoprocessing tool in ArcGIS Pro to export the attribute table for the `v2` map. Sort this table of shapefile precinct data in Excel by County, Precinct Name. (See first sheet of `Precinct_Check_r1.xlsx`.)

2. Create a new sheet and add the table of voting data. Sort the voting data in Excel by County, Precint Name. (See the second sheet of `Precinct_Check_r1.xlsx`.)

3. Create a new sheet. Put these two sorted lists side-by-side. (See the third sheet titled "Discrepancies" of `Precinct_Check_r1.xlsx`.)

4. In the Discrepancies sheet with the side-by-side data, create a check column, e.g. `=IF([shapefile precinct name]=[vote data precinct name],"ok","CHECK")`. Also create a column to document any fixes.

5. Create a new version of the map (`v3`) where edits will be made.

6. Fix any precinct naming errors (e.g. "Mt." vs. "Mount", "No." vs. "#", etc. or obvious misspellings). Name adjustments may be made either in the Excel sheet of voting data or the shapefile polygons themselves (see note below).

7. Determine locations of any precinct splits or merges and make the appropriate changes to the shapefile by editing precinct boundaries using the `Edit` tab in ArcGIS. Splits and merges can be identified if a precinct is present on one list and not the other. If splitting a precinct, population counts will need to be estimated for each new resulting precinct; one way to do this is by splitting population proportionally by number of registered voters in the voting precinct.

8. After making all adjustments, re-run `Polygon to Line` and `Multipart to Singlepart` on `v3` to ensure that no new errors have occurred as a result of editing precinct boundaries.

The *South Carolina Code of Laws* keeps a detailed record of precinct changes that may be helpful to reference during this process[2]. This will help not only in identifying precinct name changes, but

[2] `https://law.justia.com/codes/south-carolina/2020/title-7/chapter-7/`

also any merges or splits that have happened. It may also be helpful to have a precinct map from an earlier point in time, which could provide a guide for how to adjust precinct boundaries if a merge or split has occurred; this could be obtained through Revenue and Fiscal Affairs.

This process will result in the `v3` precinct shapefile having a one-to-one correspondence with the voting data, which allows for voting data to be added to the shapefile attribute table. In the 2021 process, this resulted in a shapefile with 2,264 polygons (2,263 precincts and 1 outer state).

3.4 Dissolving Precinct Holes

The Markov chain analysis requires that the precinct map must not contain holes; in other words, no precinct may be fully contained inside another precinct.

1. Save a new version (`v4`) of the precinct map; this version will contain precincts without holes.
2. Use the `Merge` tool under the `Edit` tab in ArcGIS to merge hole precincts to their surrounding precinct. When doing so, take note of how ArcGIS handles merging the record fields and make adjustments as necessary. (E.g., populations should be added to form the new merged record, but other fields may need to be handled differently.)

In the 2021 process, three hole precincts were merged to their surrounding precincts.

- **East McColl (VTD 45069000011)** was merged to **McColl (VTD 45069000010)** and renamed *McColl/East McColl*.
- **Prosperity City (VTD 45071000036)** was merged to **Prosperity Outside (VTD 45071000051)** and renamed *Prosperity Outside/Prosperity City*.
- **Whitmire City (VTD 45071000049)** was merged to **Whitmire Outside (VTD 4507000050)** and renamed *Whitmire Outside/Whitmire City*.

This process resulted in 2,261 polygons (2,260 precincts and 1 outer state) in version `v4`.

3.5 Merging Vote Data to Shapefile

Associated files for this step in the process can be found in the GitHub repository under the sub-directory `1_resolve_map_discrepancies/`.

Merging precincts may result in duplicate values of the formerly unique identifier field, `VTD`. To resolve this and make `VTD` a unique field again:

1. Use the `Table to Excel` tool to export the attribute table for `v4` of the map. (See `Precinct_Check_r2.xlsx`.)
2. In Excel, highlight duplicate values within the `VTD` column.
3. For each duplicate `VTD`, rename the field for one of the duplicated records in the Feature Class (`v4`).

`VTD` reassignment for the 2021 precinct map were as follows:

- **Yeaman's Club** changed from VTD 45015000068 to **45015000102**.

- **Jordanville** changed from VTD 45051000112 to **45051000230**.
- **Hickory Hill** changed from VTD 45051000163 to **45051000231**.
- **Joyner Swamp** changed from VTD 45051000166 to **45051000232**.
- **Spring Branch** changed from VTD 45051000168 to **45051000233**.
- **Taylorsville** changed from VTD 45051000186 to **45051000234**.

Once changes have been made to the `v4` shapefile and the `VTD` field is unique, it is time to merge the vote data to the shapefile.

1. Use the `Table to Excel` tool to re-export the attribute table. In Excel, sort the table by County, Precinct Name. (You can verify that the `VTD` field is now unique by using conditional formatting to highlight duplicate values. See the first sheet of `Precinct_Check_r3.xlsx`.)
2. Create a new sheet in the Excel table, and copy and paste the voting data from its source. Sort the table by County Name, Precinct Name. (See the second sheet of `Precinct_Check_r3.xlsx`. Voting data were obtained from John Ruoff.)
3. Create a new sheet, then paste the shapefile precinct data and the voting data side-by-side. (See the third sheet titled "Combined" of `Precinct_Check_r3.xlsx`.)
4. Create a "check" column to verify that the data are lined up correctly, e.g. `=IF([shapefile precinct name]=[vote data precinct name],"ok","CHECK")`. Make adjustments in the spreadsheet as necessary to ensure that precincts are properly matched.
5. Once the data are matched for each precinct, create a new unique identifier (in the 2021 analysis, this is called `PSN` for "*precinct serial number*"). This unique identifier should be indexed from $0, 1, \ldots, n-1$ for n precincts, and the outer state should be assigned `PSN=-1`.
6. Save the matched sheet as a new Excel sheet (see `VTD_to_PSN_Join.xlsx`).
7. Add this new Excel sheet to ArcGIS, then join it to the Feature Class layer by `VTD` using the `Join` tool.
8. Export a copy of the Feature Class as `v5` of the map so the joined data are now a permanent part of the attribute table for the Feature Class.

This map (`v5`) should now have the following fields, which will be used in the Markov chain input file. Field names should be renamed as necessary to match the field names below.

- **PSN:** Unique ID for each precinct, indexed $0, 1, \ldots, n-1$ for n precincts with the outer state having `PSN=-1`. (In the 2021 data, `PSN` values range from $0, 1, \ldots, 2259$ for $2,260$ precincts.)
- **county:** This field should have a string with the county in which a precinct resides. This will likely have been joined to the precinct shapefile from the vote data.
- **pop:** Population for a given precinct. If using Census VTDs as the starting precinct map, the shapefile should contain these population counts.

- **voteA:** Number of Democratic votes cast in the proxy election in a given precinct. (The Democratic party is used as the reference party.)
- **voteB:** Number of Republican votes cast in the proxy election in a given precinct.

3.6 Calculating Precinct Areas

ArcGIS will generate a default field for polygon areas and perimeters, but they are in ambiguous units, so it is helpful to have columns that are in identifiable units. (This helps later with compactness calculations.)

1. In `v5` of the map, add a *double* field titled `Pct_Area`.
2. Right-click on the field header for `Pct_Area` and select `Calculate Geometry`.
3. Set the following parameters:
   - Property: Area
   - Area Unit: km^2 *(or unit of choice)*
   - Coordinate System: Current Map
4. Add another *double* field titled `Pct_Perim`.
5. Right-click on the field header for `Pct_Perim` and select `Calculate Geometry`.
6. Set the following parameters:
   - Property: Perimeter length
   - Length Unit: km *(or unit of choice that matches area units)*
   - Coordinate System: Current Map

If any subsequent changes to the map geography are made, be sure to recalculate these fields using the steps above.

4 Generating Neighbor Lists

The Markov chain code requires a precinct adjacency list in which each precinct's neighbors (and their corresponding shared perimeter lengths) are listed in clockwise order. This is one of the most time-consuming parts of the data cleaning process. Before beginning, make sure that you have the most recent version (`v5`) of the South Carolina precinct Feature Class added to ArcGIS.

4.1 Generating the Precinct Lines Feature

1. Use the `Polygon to Line` geoprocessing tool to generate a Feature Class in which polygon boundaries appear as line features. This extracts relevant precinct neighbor data. Use the following parameters:
   - Input Features: `SC_Precinct_Map_v5`
   - Output Feature Class: `SC_Precincts_as_Lines_LR`
   - Check *Identify and store polygon neighboring information*

2. The above process will add SC_Precincts_as_Lines_LR as a Feature Class to your ArcGIS project. Open its attribute table and sort by the fields LEFT_FID then RIGHT_FID.

3. Remove the record that corresponds to the boundary of the outer state polygon, which should have LEFT_FID=-1. The line feature that makes up this boundary is not needed.

4. In the attribute table, add *long integer* fields SRC_PSN and NBR_PSN.

5. Join the field OBJECTID from v5 of the precinct map to the LEFT_FID field in the lines feature.

6. Calculate the SRC_PSN field using the joined layer's PSN field, then remove the join.

7. Join the field OBJECTID from v5 of the precinct map to the RIGHT_FID field in the lines feature.

8. Calculate the NBR_PSN field using the joined layer's PSN field, then remove the join.

9. Sort the lines feature attribute table by SRC_PSN then NBR_PSN.

10. Export a copy of the lines feature class (SC_Precincts_as_Lines_LR) and save it as SC_Precincts_as_Lines_RL.

11. In the copied lines feature, switch the field names of SRC_PSN and NBR_PSN. You will need to do this in two separate saves (e.g. rename the fields oldSRC_PSN and newSRC_PSN, save, then rename them NBR_PSN and SRC_PSN and save again) or the changes will not stick.

12. Sort the copied lines feature by SRC_PSN then NBR_PSN.

13. Use the Merge geoprocessing tool to combine the two lines features using the following parameters:

    - Input Datasets: SC_Precincts_as_Lines_LR and SC_Precincts_as_Lines_RL
    - Output Dataset: SC_Precincts_as_Lines_All

14. Sort the attribute table of the combined lines feature, SC_Precincts_as_Lines_All, by SRC_PSN then NBR_PSN.

The resulting Feature Class should be saved in and easily accessible from your working geodatabase. Each record of this Feature Class's attribute table contains a line segment that makes up a part of a precinct boundary, along with the unique PSN of the source precinct and the neighboring precinct that shares a boundary for that line segment.

4.2 Generating the Precincts Points Feature

1. Use the Feature Vertices To Points geoprocessing tool to generate a Feature Class made up of all the vertices from v5 of the map. Use the following parameters:

    - Input Features: SC_Precinct_Map_v5
    - Output Feature Class: SC_Precincts_as_Points
    - Point Type: All vertices

2. In the attribute table for the points feature generated by the previous step, add a new *long integer* field titled `CLCKWS_IDX` and calculate it using the `OBJECTID` field. Because vertices are stored in clockwise order, sorting `CLCKWS_IDX` in ascending order will give the clockwise order of the points making up a precinct boundary.

3. In the points feature attribute table, delete all fields except `OBJECTID`, `PSN`, `NAME` (precinct name), and `CLCKWS_IDX`. This ensures faster geoprocessing.

4. Sort the attribute table by `PSN` then `CLCKWS_IDX`.

4.3 Generating Input File for neighbors.py Script

Associated files for this step in the process can be found in the GitHub repository under the subdirectory `2_neighbors/`.

The Python script `neighbors.py` will create a list of polygon neighbors and their shared perimeters in clockwise order by `PSN`. It requires an input file generated as follows.

1. Use the `Tabulate Intersection` geoprocessing tool with the following parameters to combine relevant data from the lines feature and points feature. *This process may take a while.*

   - Input Zone Features: `SC_Precincts_as_Lines_All`
   - Zone Fields: `SRC_PSN`, `NBR_PSN`, `Shape_Length`
   - Input Class Features: `SC_Precincts_as_Points`
   - Output Table: `SC_Precincts_Point_Line_Intersection`
   - Class Fields: `PSN`, `CLCKWS_IDX`

2. In the resulting table, `SC_Precincts_Point_Line_Intersection`, delete the fields `PNT_COUNT` and `PERCENTAGE`. The table will likely have millions of records, so it is normal for this to take some time.

3. Sort the table by `SRC_PSN`, `NBR_PSN`, `CLCKWS_IDX` in ascending order.

4. Use the `Table to Table` tool to save `SC_Precincts_Point_Line_Intersection` as a `.csv` file. Be sure to enter the `.csv` file extension when typing the name of the output data in the `Table to Table` parameters. This resulting file will serve as the input for `neighbors.py`. (See `SC2021_clockwise_data.csv`.)

5. Open the `.csv` file in Notepad++ (or another text editor) and ensure that the file uses Unix (LF) end-of-line conversion and UTF-8 encoding.

6. Follow the instructions in the comments for the `neighbors.py` script to make any necessary adjustments before running the script from the command line.

The result will be a `.csv` output file that contains, for each precinct, a list of neighboring PSNs and a list of corresponding shared perimeter lengths. (See `SC2021_clockwise_neighbors.csv`.)

5 Assigning Precincts to Districts

Associated files for this step in the process can be found in the GitHub repository under the subdirectory `3_precinct_assignment/`.

To begin, have the most recent version of the precinct shapefile (`v5`) and the relevant district shapefile loaded into ArcGIS. (See note in Section 2.1 at the beginning of this document for instructions on how to generate the district shapefile if a districting plan is provided as a block assignment file.)

The Markov chain analysis requires that each precinct be assigned to only one district, which is often not the case for an actual districting plan. The process detailed below assigns each precinct to the district that contains the largest proportion of its area.

1. Add the most recent precinct Feature Class (`v5`) and the relevant district Feature Class to the current ArcGIS project.
2. Use the `Intersect` geoprocessing tool to overlap the precinct and district maps. When choosing the input features, be sure to list the precinct map *first* and the district map second. Leave all other parameters as their defaults.
3. Use the `Table to Excel` geoprocessing tool to export the attribute table of the resulting feature class as an `.xlsx` file.
4. In the Excel sheet, highlight duplicate values of `PSN`. To keep only the record for a `PSN` that has the largest area:
    - Custom sort the sheet by `PSN` (ascending) then `Area` (descending). *Note: This should be one of the last two columns, not the area field calculated in Section 3.6.*
    - Use the `Remove Duplicate Values` feature under the `Data` tab in Excel to remove duplicate `PSN` values. This will keep the first row of any duplicate `PSN` values and delete the rest, so sorted this way, it will keep the district overlap with the largest area.
5. Verify that the number of records now corresponds to the number of precincts, then remove all columns except `PSN` and `District Number`. Save the Excel file. (See the Excel files beginning with `Intersect_` in the GitHub subdirectory.)
6. In ArcGIS, create a new field to store the assigned district number. (E.g. The field name `CongD_LWV` could be used to specify the Congressional district assignment from the proposed map by the League of Women Voters.)
7. Add the Excel file with the precinct-to-district assignment to ArcGIS.
8. Use the `Join` feature to join the table to the precinct map by `PSN`.
9. Calculate the new district number field using the `District Number` from the joined table, then remove the join. Repeat for all precinct-to-district assignments.

This process may result in some districts that are non-contiguous or contain holes. Spot-check the precinct assignment for places where this happens (this can usually be accomplished by setting the Symbology of the precinct map to color precincts based on the corresponding district field) and manually adjust precinct assignment in the attribute table using your best judgment. A record of

all precinct reassignments to districts for the maps analyzed in 2021 is detailed below.

Official Proposed Plan: Congressional

- No changes needed.

Official Proposed Plan: Congressional (House Draft)

- No changes needed.

Official Proposed Plan: Senate

- **PSN 1520** (LEESVILLE): Changed from District 23 to 25.
- **PSN 1552** (RIDGE ROAD): Changed from District 25 to 10.

Official Proposed Plan: House Draft #1

- **PSN 406** (Yellow House): Changed from District 103 to 99.
- **PSN 443** (Deer Park 1B): Changed from District 113 to 92.
- **PSN 461** (James Island 19): Changed from District 115 to 119.
- **PSN 464** (James Island 20): Changed from District 115 to 119.
- **PSN 471** (James Island 8A): Changed from District 115 to 119.
- **PSN 472** (James Island 8B): Changed from District 115 to 119.
- **PSN 473** (James Island 9): Changed from District 115 to 119.
- **PSN 481** (Ladson): Changed from District 113 to 94.
- **PSN 482** (Lincolnville): Changed from District 113 to 94.
- **PSN 637** (Chester Ward 4): Changed from District 43 to 41.
- **PSN 910** (Ebenezer No. 3): Changed from District 63 to 60.
- **PSN 979** (MURRELL'S INLET NO. 3): Changed from District 103 to 108.
- **PSN 1052** (GREENVILLE 28): Changed from District 22 to 23.
- **PSN 1127** (TANGLEWOOD): Changed from District 25 to 10.
- **PSN 1319** (SOCASTEE #1): Changed from District 68 to 61.
- **PSN 1330** (TILLY SWAMP): Changed from District 105 to 56.
- **PSN 1347** (OKATIE): Changed from District 122 to 120.
- **PSN 1747** (Georges Creek): Changed from District 4 to 5.
- **PSN 1800** (Brandon 1): Changed from District 70 to 75.
- **PSN 1873** (Pontiac 2): Changed from District 78 to 70.

- **PSN 2060** (CHERRYVALE): Changed from District 50 to 67.
- **PSN 2180** (Delphia): Changed from District 29 to 49.

Official Proposed Plan: House Draft #2

- **PSN 348** (Harbour Lake): Changed from District 99 to 15.
- **PSN 406** (Yellow House): Changed from District 103 to 99.
- **PSN 461** (James Island 19): Changed from District 115 to 119.
- **PSN 464** (James Island 20): Changed from District 115 to 119.
- **PSN 471** (James Island 8A): Changed from District 115 to 119.
- **PSN 472** (James Island 8B): Changed from District 115 to 119.
- **PSN 473** (James Island 9): Changed from District 115 to 119.
- **PSN 637** (Chester Ward 4): Changed from District 43 to 41.
- **PSN 910** (Ebenezer No. 3): Changed from District 63 to 60.
- **PSN 979** (MURRELL'S INLET NO. 3): Changed from District 103 to 108.
- **PSN 1052** (GREENVILLE 28): Changed from District 22 to 23.
- **PSN 1127** (TANGLEWOOD): Changed from District 25 to 10.
- **PSN 1319** (SOCASTEE #1): Changed from District 68 to 61.
- **PSN 1330** (TILLY SWAMP): Changed from District 105 to 56.
- **PSN 1347** (OKATIE): Changed from District 122 to 120.
- **PSN 1747** (Georges Creek): Changed from District 4 to 5.
- **PSN 1800** (Brandon 1): Changed from District 70 to 75.
- **PSN 1872** (Pontiac 1): Changed from District 70 to 78.
- **PSN 2060** (CHERRYVALE): Changed from District 64 to 67.
- **PSN 2180** (Delphia): Changed from District 29 to 49.

Proposed League Map: Congressional

- No changes needed.

Proposed League Map: Senate

- **PSN 801** (Carolina): Changed from District 38 to 44.

Proposed League Map: House

- **PSN 1347** (Okatie): Changed from District 122 to 118.
- **PSN 1457** (Sandy Pointe): Changed from district 120 to 118.

Once all precinct-to-district assignments have been made to `v5`, save the Feature Class.

6 Generating Markov Chain Input Files

Associated files for this step in the process can be found in the GitHub repository under the subdirectory `4_data_merge/`.

First prepare the final version of the precinct map.

- Export a copy of the Feature Class as `v6`.
- Remove the polygon for the outer state (`PSN=-1`).
- (Optional) You can also export the `v6` Feature Class as a shapefile so it can be shared beyond the geodatabase.

The final 2021 precinct map used in the 2021 analysis can be found in `sc_maps/precincts_2020/`.

To export the map data to be used in the generation of the Markov chain input file:

1. Use the `Table to Table` tool to export the attribute table of the `v6` map as a `.csv` file. (See `SC2021_map_data.csv`.)
2. Clean the `.csv` file in Excel so only the following fields remain in this order and with the following names:
   - `PSN` — unique precinct identifier
   - `unshared` (add this column) — a column that contains zero for each record
   - `area` — the area of the precinct calculated in the field `Pct_Area`
   - `pop` — the population in the precinct (obtained from VTD shapefile)
   - `voteA` — number of precinct-level Democratic votes
   - `voteB` — number of precinct-level Republican votes
   - `[District Number]` (multiple fields) — the names for these fields may vary; there will be as many fields as there are districts you wish to analyze
   - `county` — string designating the county name

   Save the cleaned file. (See `SC2021_map_data_cleaned.csv`.)

Now combine the neighbors data from Section 4.3 (`SC2021_clockwise_neighbors.csv`) with the map data generated above (`SC2021_map_data_cleaned.csv`) using the Python script `data_merge.py`. Each resulting input file will correspond to exactly one districting plan.

1. Verify that both input `.csv` files have Unix (LF) end-of-line conversion and that they use UTF-8 encording. (This can be changed quickly in Notepad++.)
2. Open `data_merge.py` in a text editor and address any items marked as `# TO DO`.
3. Save `data_merge.py`.
4. Using the neighbors data and map data as inputs, run the script in the command line to generate the `.csv` input files.

The last step is to convert the `.csv` input files into appropriately formatted `.txt` input files. For each input file:

1. Open a blank Excel sheet.
2. Using the `Get Data` option under the `Data` tab in Excel, import a `.csv` input file generated by `data_merge.py`. Use the default settings.
3. Use `Ctrl+A` to copy all data and paste into a blank `.txt` file. Close the Excel file without saving.
4. In the `.txt` file where you pasted the data, which should be tab-delimited, delete the `PSN` column header to leave a blank column header for the first column.
5. Add two new lines at the top of the `.txt` file that say `precinctlistv02` and `2260` (or the number of precincts in your map), respectively.
6. Set the EOL conversion to be Unix (LF) and verify that the file has UTF-8 encording.
7. Save the `.txt` file. This is the input file that will be fed into the `chain.cpp` code.

The input files generated for the 2021 analysis can be found in the subdirectory `input_files/`.

7 Calculating Compactness Bounds

Associated files for this step in the process can be found in the GitHub repository under the subdirectory `5_compactness/`.

One constraint that can be set for the chain code is a compactness threshold; this is typically chosen to be a compactness metric just slightly above the compactness measure for the starting districting plan. To calculate these compactness thresholds:

1. From the `v6` map (or most current version), use the `Dissolve` geoprocessing tool in ArcGIS with the following paramters:
   - Input Features:`SC_Precincts_2021_v6`
   - Output Feature Class: `MCDistricts_[District_Plan]`
   - Dissolve Field(s): `[District_Field]`
   - Statistics Field(s): Field area [`Pct_Area`]
   - Statistic Type: Sum
   - Uncheck *Create multipart features*

   Repeat this process for any districting plan you wish to analyze.
2. In the resulting attribute table, add the fields `DistArea` and `DistPerim`, then use `Calculate Geometry` to compute each district area and perimeter, respectively. Use the same units as those used to calculate precinct area in Section 3.6.
3. Verify that `SUM_Pct_Area` and `DistArea` are approximately the same.

4. Export the resulting attribute table as an Excel file using the `Table to Excel` geoprocessing tool. (See the Excel files whose filenames begin with `MCDistricts_`.)

5. Use the `compactness_thresholds.xlsx` Excel file to see how compactness thresholds were chosen for the "L1 compactness" measure, also known as *Inverse Polsby-Popper*. Each threshold was set to be 2.5% higher than the districting plan with the worst compactness score. (More details about the compactness metrics used can be found in the associated master's thesis.)

8 Running and Parsing Output

The readme file included with the Markov chain code contains documentation on how to compile and run the code for a particular map using the input files generated as described in this document. A full list of commands can also be viewed by typing `chain`.

As an example, consider the proposed 2021 South Carolina Congressional map. To run the analysis using the *median-mean* metric as a measure of partisan bias with an allowable population error of $\pm 5\%$ and an Inverse Polsby-Popper threshold of 46, the command line would be as follows. This would generate 2^{30} sample maps.

```
chain -f InputSC_CongDraft.txt -n 30 -N 7 --median_mean --poperror=.05
--L1-compactness=46
```

To write the output to a text file for analysis, add `> output.txt` to the end of the line. A Jupyter Notebook file can be found in the GitHub repository under `mc_output/` to assist with analyzing output.