Gerrymandering Project

**02/09**

1. Decided to go with city council districts shapefiles Gustavo shared after learning that Census does not provide city-level data. Census MSAs (metropolitan areas) or places are different from cities.
2. Created a for loop that (1) reads each city’s council districts shapefile; (2) dissolves districts polygons; (3) removes unnecessary holes in the dissolved city boundary; and (4) assign it as a new object named “*cityname*\_city\_boundary.”
   1. i.e.) Before vs after removing unnecessary holes in New York
3. Created a for loop that (1) imports each city’s tract AND block-level demographics shapefiles; (2) filters tracts and blocks within the city boundary; (3) and assign them as new objects, named “*cityname*\_tracts\_boundary” and “*cityname*\_blocks\_boundary,” respectively.
   1. I have finished writing this loop, but intersection is not working yet because different levels of boundary files (city, tract) currently have different datum and projection. To make the intersection function work, I have to make sure all the boundary files have the same datum and projection.

**Next steps:** (1) ~~Fix Houston and Austin geometry issues;~~ (2) ~~Make sure all cities’ boundary files and Census datasets have the same datum and projection~~; (3) ~~add partisanship data~~.

**02/09**

1. Fixed Houston and Austin geometry issues. Running the ‘st\_union’ function on city council districts yielded the error message: “*Error in s2\_geography\_from\_wkb(x, oriented = oriented, check = check) : Evaluation error: Found # features with invalid spherical geometry*.”
   1. **Cause:** This error happened due to a change in the backend engine for unprojected coordiates from GEOS to s2. GEOS treats projected coordinates as planar (i.e. two points lie on a line of infinite max length) while s2 is more "correct" (two points lie on a great circle of circumference of 40 075 kilometers)
   2. **Two Solutions Available:**  sf\_use\_s2(FALSE) vs st\_make\_valid
      1. ‘sf::sf\_use\_s2(FALSE)’: Turns off the s2 processing
      2. ‘sf::st\_make\_valid()’: Make an invalid geometry valid
   3. I chose ‘st\_make\_valid’ because I believe there are some reasons the package has changed its geoprocessing approach to s2. Also, the second option is as simple as running the first one.
2. Made sure all cities’ boundary files and Census datasets have the same datum and projection. Cities’ boundary files had their own CRS, while Census datasets had the same CRS of “+proj=longlat +datum=NAD83 +no\_defs.” So I converted all cities’ boundaries’ CRS to that.
3. Created a for loop that (1) reads each state’s precinct returns shapefile; (2) resets the datum and projection; and (3) assigns it as a new object named “*statecode*\_state\_precinct\_returns.”
4. [In Progress] Create tract- and block-level spatially-weighted demographic and partisanship data.
   1. **Issues**: Codes take too long to run, especially block-level ones.

Next Steps: (1) find a way to optimize my codes – filter relevant counties first; (2) try QGIS and see how it works; (3) get more cities’ data ready.

**02/18**

1. Optimized several parts including:
   1. Replacing “st\_intersection” with a filter function, leaving tracts and blocks within the city boundary only.
2. At the end of every computationally intensive stage, added a function saving the output datasets into an RData object. That way, if a dataset somehow disappears from the environment, I can simply load the RData object, rather than rerunning the code.
3. After trying intersection a couple of times in R, moved to QGIS.
4. Installed the “Area Weighted Average” plug-in.

NEXT STEP: Find the most efficient way to produce the file.

**02/25**

1. Looked for possible ways to automate a repetitive task in QGIS
   1. Options:
      1. PyQGIS
      2. ModelBuilder
2. Get rid of geometry after intersection.

**03/02**

1. **Final data structure**
   1. **GEOID - unique block ID**
   2. **NAME - block name**
   3. **Asian\_p**
   4. **Black\_p**
   5. **Hispanic\_p**
   6. **White\_p**
   7. **Total\_p**
   8. **Sw\_g20predbid - spatially-weighted votes for Biden**
   9. **Sw\_g20prertru - spatially-weighted votes for Trump**
   10. **Geometry – block-level**
2. **Final data structure #2**
   1. **Council District ID**
   2. SW\_Asian\_p – block-level
   3. B
   4. H
   5. W
   6. T
   7. SW\_g20Biden – precinct-level
   8. SW\_g20Trump
   9. **Geometry – council district**

03/09

1. Created a for loop that (1) reads in each city’s intersection output, (2) standardizes the data structure, and (3) saves two files for each city: “city\_intersection\_weighted\_block”, “city\_intersection\_weighted\_council\_district.” Ultimately, each city will have three files below:
   1. block\_precinct\_council\_district\_intersection
   2. intersection\_weighted\_block
   3. intersection\_weighted\_council\_district
2. Finished 9 cities thus far. Issues with Houston, Philadelphia, Jacksonville, Fortworth should be fixed.

Next Steps: (1) fixing issues with the four cities above; (2)