**Guiding questions:**

*What task did you work on? How was the output created? (Many tasks do not have outputs that are saved externally, but think of any data frames you create as outputs.) What data manipulation, imputation, and analysis steps did you undertake? What did you find? What is your interpretation of your findings? Were you able to achieve your objective and/or desired output? Are there pending steps? What are the next steps for this project?*

**Date: 5/12/2023**

**Scripts: download\_shapefiles.R, buffers annexable**

**Output:**

**Notes:**

* Download block- and place-level shapefiles for 2008, 2009, 2011, 2012, 2015, 2016, 2018, 2019
* Upload 2007-2020 back to Sherlock
* Make 90pct threshold assignments for the following place IDs based on spatial analysis
  + 2008 (06-10 ACS), 2009 (07-11 ACS), ~~2010 (08-12 ACS)~~, 2011 (09-13 ACS), 2012 (10-14 ACS), 2015 (13-17 ACS), 2016 (14-18 ACS), ~~2017 (15-19 ACS),~~ 2018 (16-20 ACS), 2019 (17-21 ACS), 2020 (17-21 ACS)
* Buffers:
  + 2008 blocks in buffers of 2008 places
  + 2009 blocks in buffers of 2009 places
  + 2010 blocks in buffers of 2010 places
  + 2011 blocks in buffers of 2011 places
  + 2012 blocks in buffers of 2012 places
  + 2015 blocks in buffers of 2015 places
  + 2016 blocks in buffers of 2016 places
  + 2017 blocks in buffers of 2017 places
  + 2018 blocks in buffers of 2018 places
  + 2019 blocks in buffers of 2019 places

**Issues:**

**Date: 6/30**

**Script: TWO. buffers\_annexable.R; check\_annexable.R; download\_shapefiles.R**

**Output:**

Issues:

* Example for Tucson, AZ, is still wrong



Progress:

* Is it possible to generate place boundaries manually?
  + Tried this back in 2020/2021 with st\_dissolve and it didn’t work
  + It requires knowing the place IDs for every block in a given year, which doesn’t exist, and is the whole reason for manually assigning place IDs to blocks in inter-Censal years in the first place.
* Try two things:
  + 2019
    - download\_shapefiles.R
  + Validate against BAS data

Next step:

* Run analysis (research journal at THREE.R)

**Date: 6/27**

**Script: TWO.R (in Sherlock); TWO. buffers\_annexable.R**

**Output:**

Issues:

* Non-VRA places are just not good comparisons for VRA places, I think
  + VRA places are significantly less white than non-VRA places

Progress:

* Work on preliminary results:
  + Less likely to annex with time; no significant DID interaction
  + Annexation is generally associated with less % black, but moderated by post-VRA!

Next step:

* Run analysis (research journal at THREE.R)

**Date: 6/26**

**Script: TWO.R (in Sherlock); TWO. buffers\_annexable.R**

**Output:**

Issues:

* + - * Still wrong: by comparing the top left to bottom left, many more blocks are considered annexed than actually happened; but, comparing the top right to bottom left, we run the risk of not being able to identify annexed blocks.



* Solution: use contains and overlaps, but restrict to those blocks that have at least 90% area overlap

Progress:

* + Check 2014 blocks on 2020 places is in progress on Sherlock; (node 10414)
  + Check 2007 blocks on 2013 places is in progress on Sherlock; (node 35035)
  + Check 2000 blocks on 2007 places is in progress on Sherlock; (node 30680)

Next step:

* Identify annexing or not using BAS after all as an additional check

**Date: 6/21-6/22**

**Script: TWO.R (in Sherlock); TWO. buffers\_annexable.R**

**Output:**

Issues:

* + - Was working on a function to check identified annexations with whether or not they actually overlap with the place boundaries at t1. In the process of thinking through whether I should check annexed blocks using the block shapefile at t0 or t1, decided there would probably be a better way of identifying annexations just using this scheme:
      * “Fit” 2000 blocks (2010b) with 2000 place (2010b) shapefiles; “fit” 2000 blocks (2010b) with 2007 (2000b) place shapefiles and compare
      * Repeat for 2007 blocks (2000b) with 2007 (2000b) place shapefiles (done); 2007 (2000b) blocks with 2013 (2010b) place shapefiles
      * Repeat for 2014 blocks (2010b) with 2014 place (2010b) shapefiles; 2014 blocks (2010b) with 2020 (2020b) place shapefiles

Progress:

* + 2007 blocks on 2007 places is completed on Sherlock;
  + 2007 blocks on 2013 places is in progress on Sherlock; (node 48649)
    - * Completed 6/24
  + 2014 blocks on 2014 places and 2020 places is in progress on Sherlock; (node 6312)
    - * Also 60168 starting from MT
      * Completed 6/25
  + 2000 blocks on 2000 and 2007 places is in progress on laptop and Sherlock; (node 12455)
    - * Completed 6/25

Next steps:

**Date: 6/14 – 6/21**

**Script: Check Annexed.R and ONE.R**

**Output:**

Issues:

* + - The annexed data looks wrong—namely some blocks are categorized as annexed when they were not annexed. For example, in the image below, if Baytown annexed the blocks in yellow from 2007-2013, the shapefile for the place for 2014 should look as though the blocks had been incorporated. It does not appear that way, however.



At first, I identified that it could be inconsistent block boundaries between Decennial Censuses that was contributing to this issue, so I managed to fix it for Roberta City, GA, after using the crosswalk files:



This also entailed re-doing interpolated block data harmonized to 2010 boundaries for all (2000 and 2020).

However, even after crosswalking, the maps still looked wrong. After identifying a 3% Black dilution annexing place—Jonesboro, GA—I determined that the annexation map for 2014-2020 was wrong, in part because something looked strange about the 2007-2013 map. It turns out that there really are some places where blocks are not completely contained in a place!!

Wrong st\_contains() 

Non-complete blocks:



Progress:

* + Created interpolated 2007, 2013, and 2014 block-level data based on 2010 boundaries after using crosswalk files.
  + Figured out code to identify blocks producing the ‘right’ map as above.
  + Set up code in Sherlock for scaling up.

Next steps:

* + For blocks identified as annexed, check how much they overlap with the annexing place shapefile
  + See whether the intersection area is wholly within the block