

# **Exhibit 1**

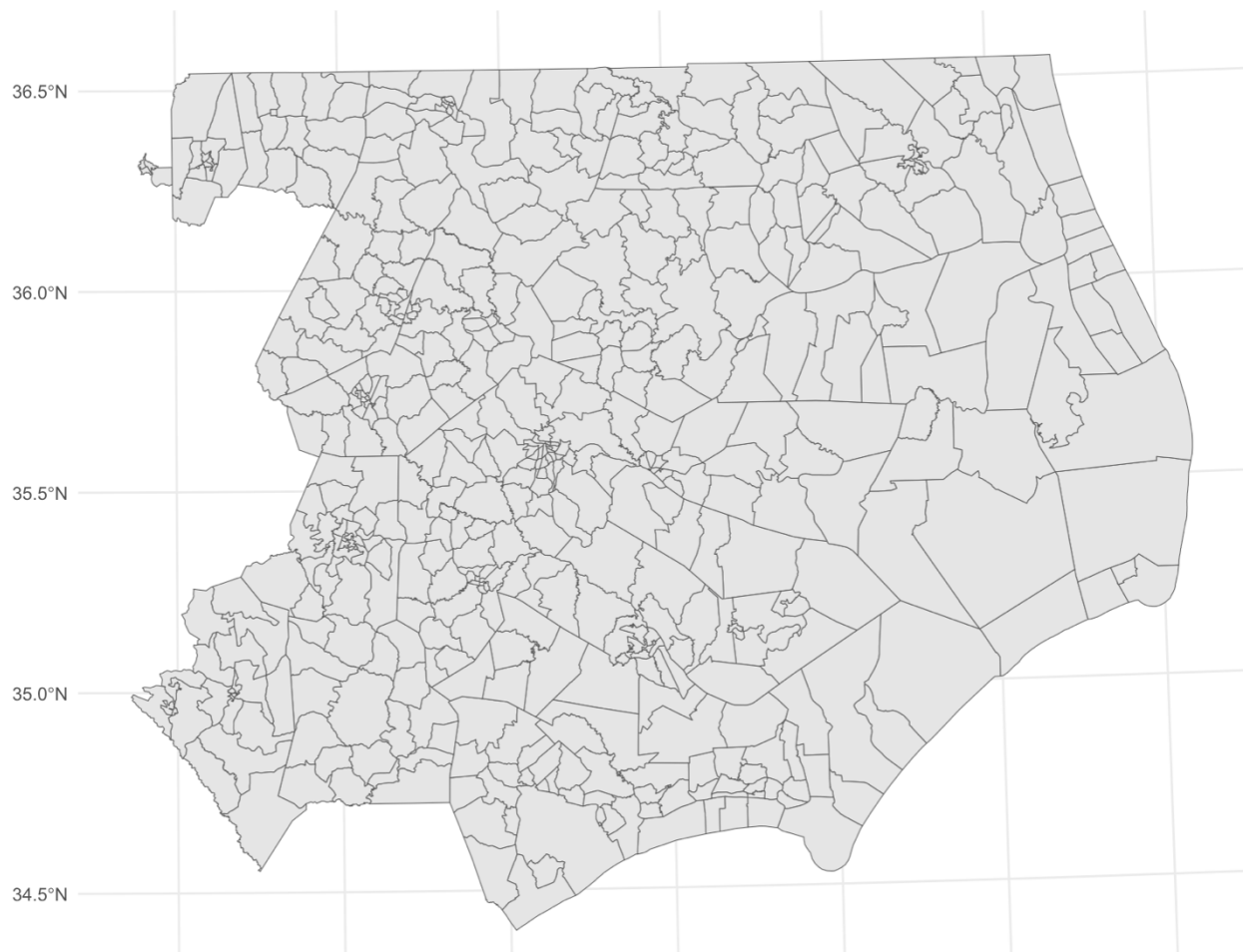
Dr. Michael Barber  
Code Replication of  
Dr. Rodden's November 18<sup>th</sup>, 2025  
Reply Report

Dr. Rodden conducts a new set of simulations in the report he submitted on November 18, 2025. I have looked at the backup data and code associated with that report and have found several errors that make the simulations incorrect and unreliable. I document these issues below.

Dr. Rodden begins by reading a shapefile into the software program “R” called export.shp.

```
13  
14 vtlds.sims <- read_sf("export.shp")  
15 |
```

This shapefile contains the geographic information of the precincts that are within District 1 and 3 in the 2023 and 2025 maps. A map of this shapefile is displayed below. This map appears correct to me aside from one small issue, which is that the map does not contain the portion of a split precinct in Granville County. However, this would not make a material difference because this precinct portion contains only 1,252 people.



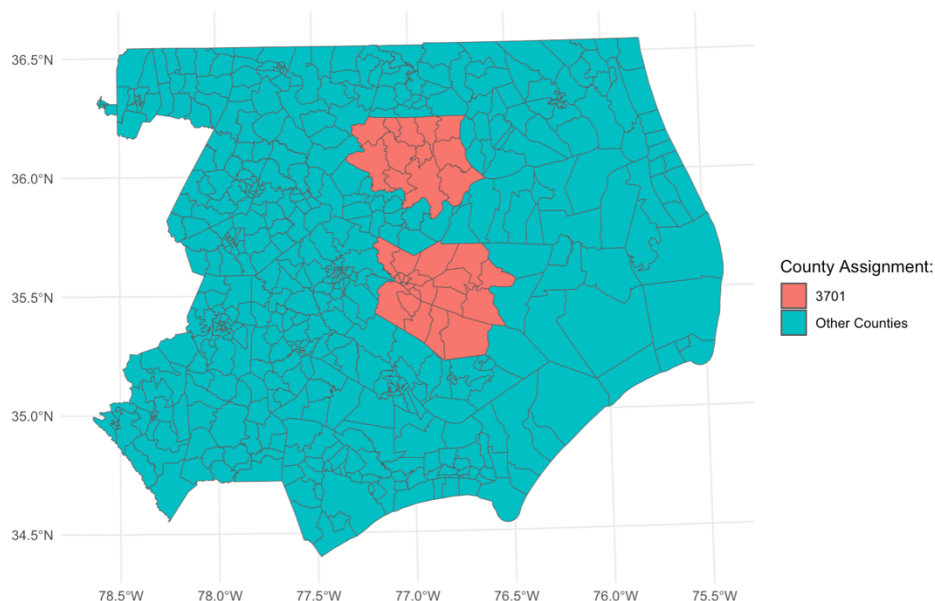
The larger issue is the assignment of counties in this file. The redistricting algorithm that Dr. Rodden uses (*redist*) requires a parameter that tells the algorithm which county each precinct is located in. The algorithm uses this parameter as it draws districts to try and minimize the number of split counties. This encourages the algorithm to create districts that follow the state's traditional redistricting criteria.

There are 33 counties that are either wholly or partially within Districts 1 and 3 in the 2023 and 2025 maps. However, the file that Dr. Rodden uses only contains 18 counties. The code below shows each county's code (e.g. 3701 is the first county in the file) and the number of precincts contained in each of these counties (county 3701 contains 33 precincts). One can simply count up the number of counties here and see that there are only 18 counties in the dataset. They proceed in numeric order from 3701 to 3719, but there is no county 3715.

```
> table(vtds.sims$COUNTY)
```

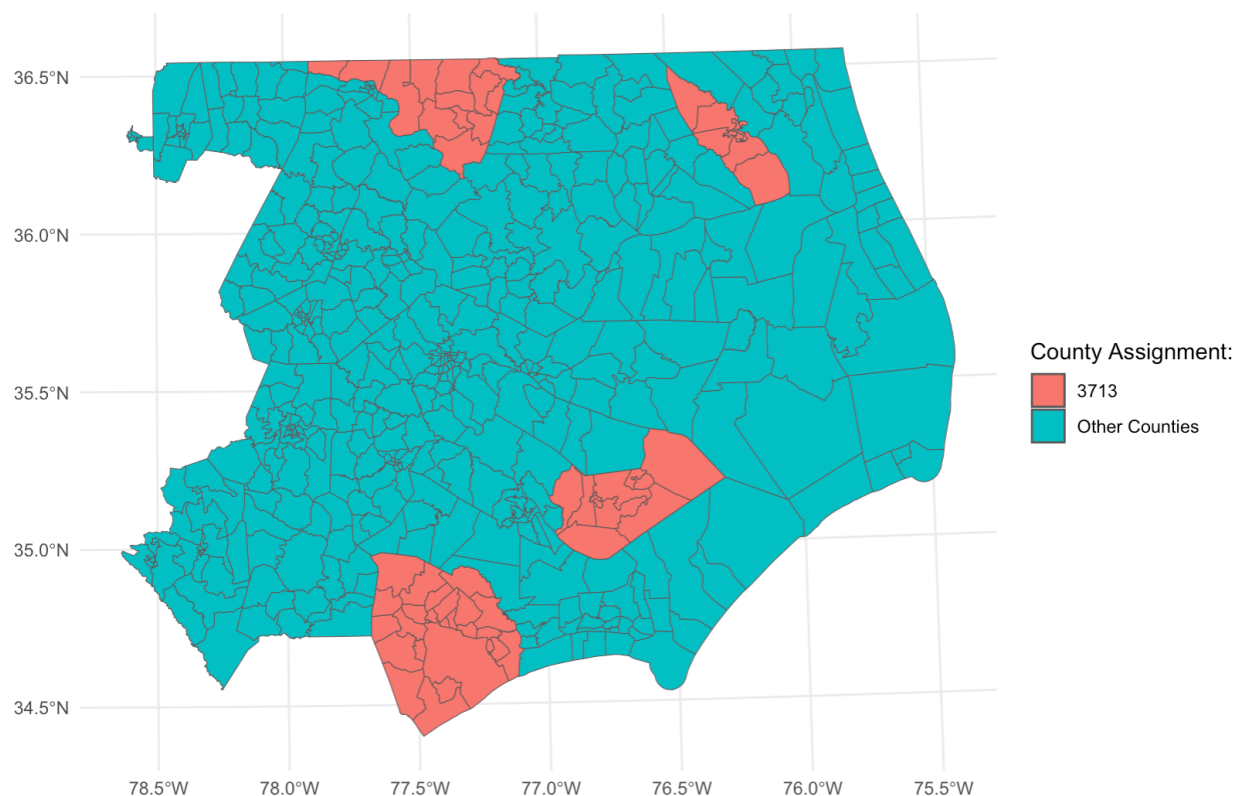
3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711	3712	3713	3714	3716	3717	3718	3719
33	3	28	27	27	40	20	23	20	29	13	24	56	47	18	6	32	52

How can it be that the precincts appear to cover the correct geographic space in the map above, and yet only contain 18 counties? The answer to this question is that the vtds.sims shapefile incorrectly groups multiple counties together and identifies them as single counties. An example is mapped below. The map below shows what the file identifies as county "3701". As can be seen in the map, the file incorrectly groups Bertie and Beaufort County together into the same county, 3701. This is obviously not correct.





This is not the only instance of this type of error. The map below shows the counties that are collectively coded as county “3713”. The file Dr. Rodden uses incorrectly considers Northampton, Pasquotank, Pamlico, and Onslow counties as one large, discontinuous county.



The implications of this error are profound. Because the file incorrectly identified county-precinct assignments, the algorithm cannot minimize county divisions correctly. As a result, the districts that are created by the algorithm are not reliable and do not reflect the actual criteria of minimizing county divisions. When the redistricting algorithm is run, it still generates districts, but while doing so it recognizes this problem and reports to the user that the county variable does not contain contiguous counties. It produces a warning message, which I have reproduced below from Dr. Rodden’s replication code.

```
Warning message:  
Counties were not contiguous; expect additional splits.
```

Despite these issues, the algorithm will still create districting plans (just incorrect ones), and Dr. Rodden instructs the algorithm to create 50,000 maps with 2 districts in each plan in this region of the state. On top of this county assignment error, there are a number of other errors that render the simulations invalid.

The kind of algorithm that Dr. Rodden is using is called a Sequential Monte Carlo (SMC) algorithm. SMC algorithms generate simulated redistricting plans by creating maps in a series of “splits.” At each step, the algorithm evaluates many partial plans, assigns them importance weights, and resamples the most promising ones so that the process continues to explore the feasible map space while producing a representative sample of maps. A well-functioning SMC sampler maintains a diverse collection of potential maps and ends with a large number of unique completed plans. In this setting, convergence refers to the algorithm’s ability to properly explore the distribution of legally valid redistricting plans rather than becoming “trapped” in a small region of the space. When the sampler does not converge properly, the final maps are highly repetitive and do not represent the wider distribution of feasible districtings.

A key indicator of healthy performance is resampling “efficiency”. This quantity measures the proportion of potential maps that survive the resampling step with meaningful weight. High efficiency indicates that the algorithm maintains diversity among candidate maps and is able to move through the universe of possible maps. A very low value indicates that almost all candidate maps are being dropped, so only a tiny number of maps move forward. When this happens, the sampler collapses onto a very small set of map configurations. This usually occurs when the constraints are too tight relative to the map’s geography, or when equal population requirements make it difficult to find viable plans (it is customary when using this algorithm to allow for a small amount of population deviation between districts).

The diagnostic statistics in Dr. Rodden’s SMC run indicate that the algorithm did not explore the universe of possible maps in a statistically accurate way. The algorithm produces a warning, “Less than 5% resampling efficiency” and encourages the user to take steps to correct this error. This is a sign of severe degeneracy and indicates that the resampling efficiency was below the accepted range. When resampling efficiency collapses, the algorithm essentially duplicates the same small set of plans over and over again, which prevents any real exploration. A screenshot of the warning that appears when I run the code provided by Dr. Rodden appears below.

```
Warning message:
Less than 5% resampling efficiency.
• Increase the number of samples.
• Consider weakening or removing constraints.
i If sampling efficiency drops precipitously in the final iterations,
  population balance is likely causing a bottleneck. Try increasing
  `pop_temper` by 0.01.
i If sampling efficiency declines steadily across iterations, adjusting
  `seq_alpha` upward may help a bit.
```

The *redist* algorithm allows users to access other diagnostics that indicate if the algorithm ran correctly. The acceptance rates reported in these diagnostics are also extremely low. Values around one percent indicate that almost all proposed plans were rejected. This is usually caused by constraints that are too strict for the underlying geography. When both acceptance rates and resampling efficiency fall to these levels, SMC algorithms no longer behave accurately. The screenshot below shows the warning produced by running Dr. Rodden’s code. The “\*”s at the far right of the “Resample” row and the corresponding note at the bottom indicate problems with “Bottlenecks”, or places where the algorithm is getting “stuck” and cannot properly explore the space of potential maps.

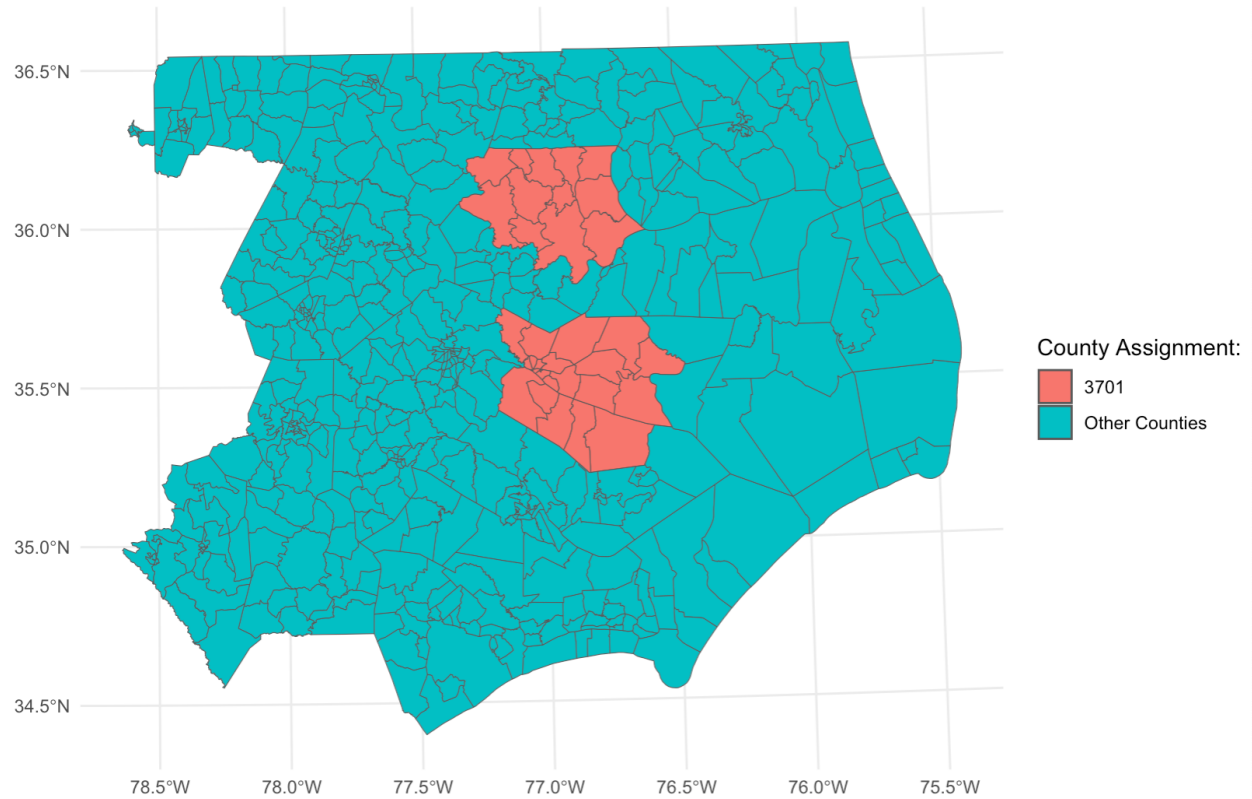
Sampling diagnostics for SMC run 10 of 10 (5,000 samples)						
	Eff. samples (%)	Acc. rate	Log wgt. sd	Max. unique	Est. k	
Split 1	610 (12.2%)	1.9%	2	3,150 (100%)	3	
Resample	7 (0.1%)	NA%	2	702 ( 22%)	NA	*

• Watch out for low effective samples, very low acceptance rates (less than 1%), large std. devs. of the log weights (more than 3 or so), and low numbers of unique plans. R-hat values for summary statistics should be between 1 and 1.05.  
 • (\*) Bottlenecks found: Consider weakening or removing constraints, or increasing the population tolerance. If the acceptance rate drops quickly in the final splits, try increasing `pop\_temper` by 0.01. If the weight variance (Log wgt. sd) increases steadily or is particularly large for the "Resample" step, consider increasing `seq\_alpha`. To visualize what geographic areas may be causing problems, try running the following code. Highlighted areas are those that may be causing the bottleneck.

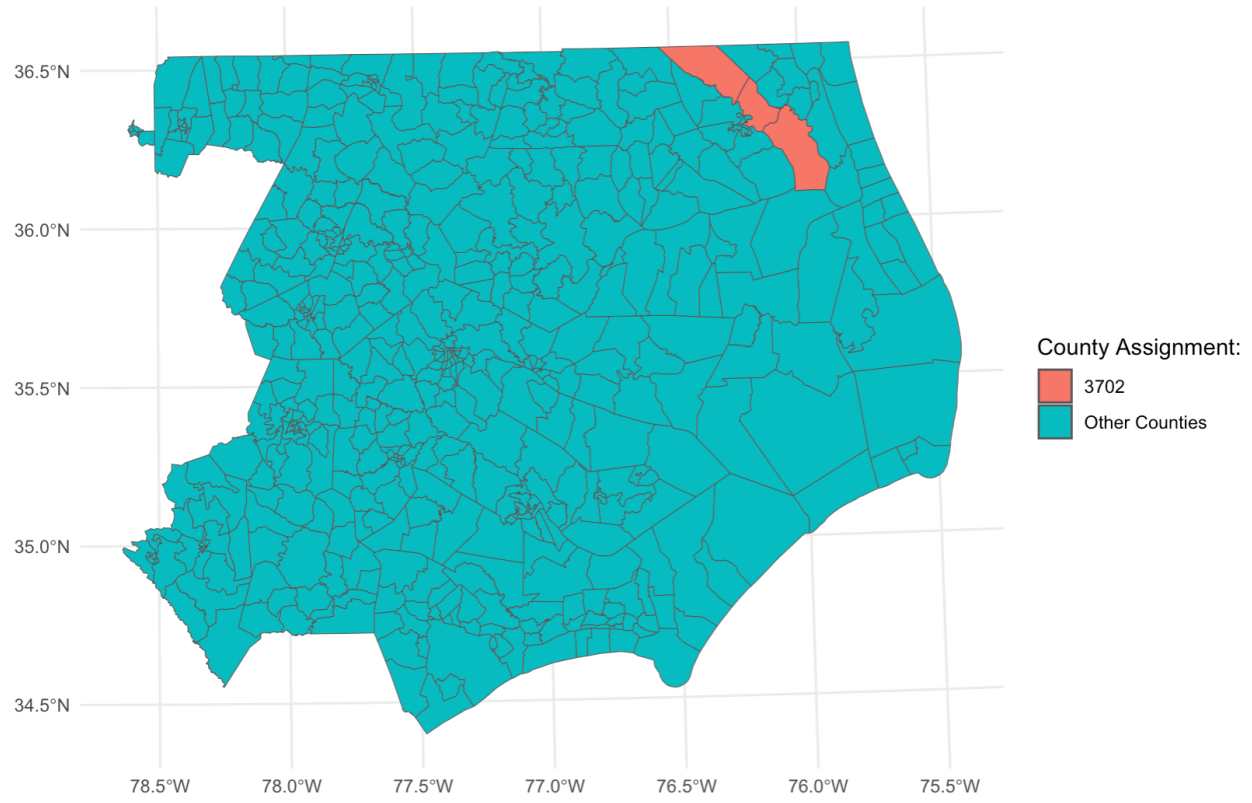
Taken together, these features indicate that the simulation output is not a reliable representation of the universe of legally valid redistricting plans. The sampler failed to achieve convergence, suffered from bottlenecks, and returned a set of maps that lack meaningful diversity. Beyond this, the algorithm was seeking to minimize county splits using a list of counties that is not accurate. Any statistical conclusions drawn from this ensemble would therefore rest on simulations that did not function as intended.

## County Assignments in the vtlds.sims Dataset

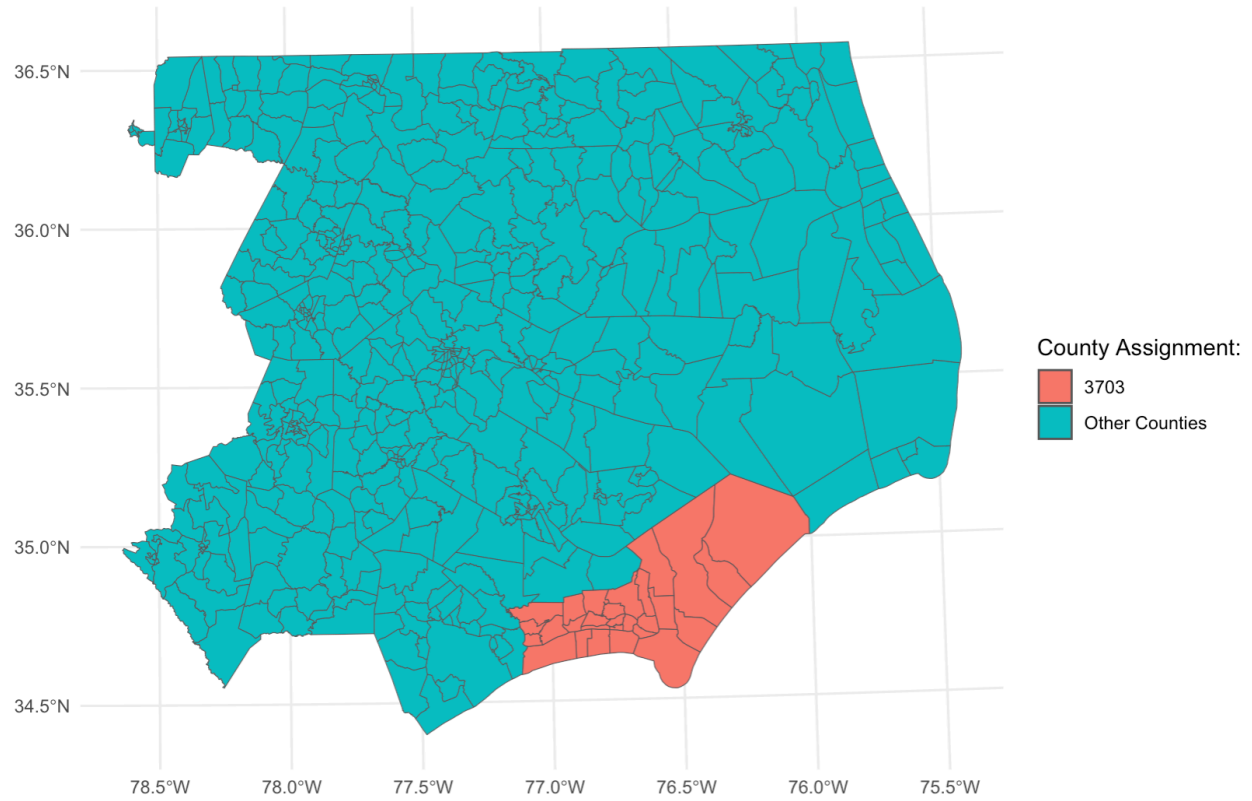
County “3101” contains Bertie and Beaufort counties.



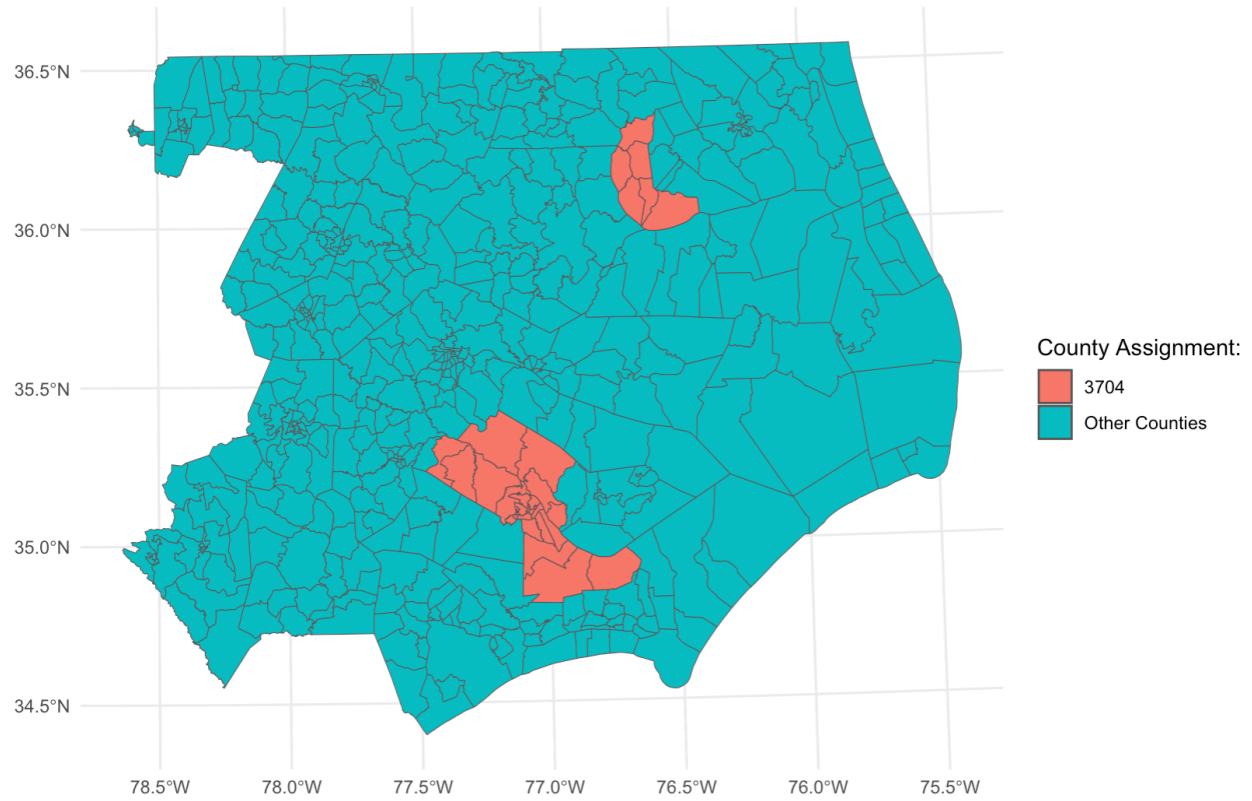
County “3702” includes Camden County



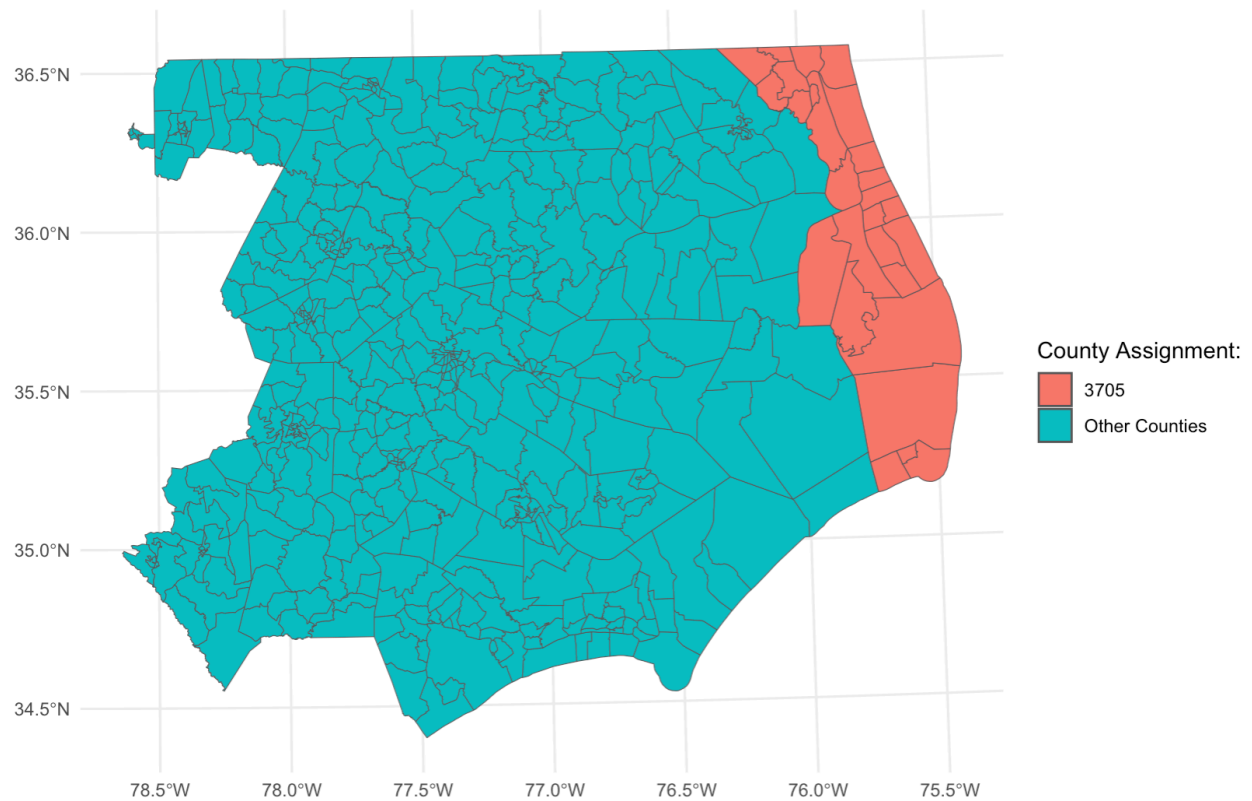
County “3703” includes Carteret County



County “3704” includes Chowan and Craven counties

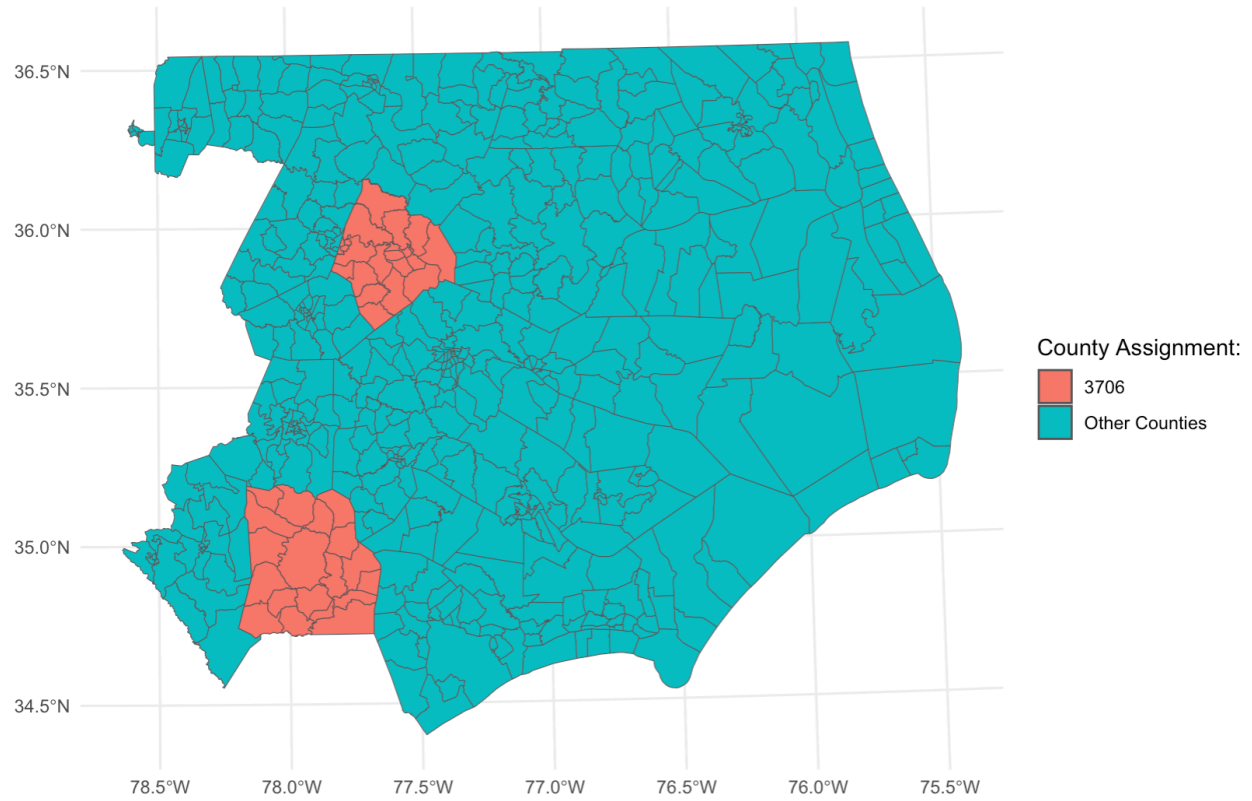


County “3705” includes Dare and Currituck counties.

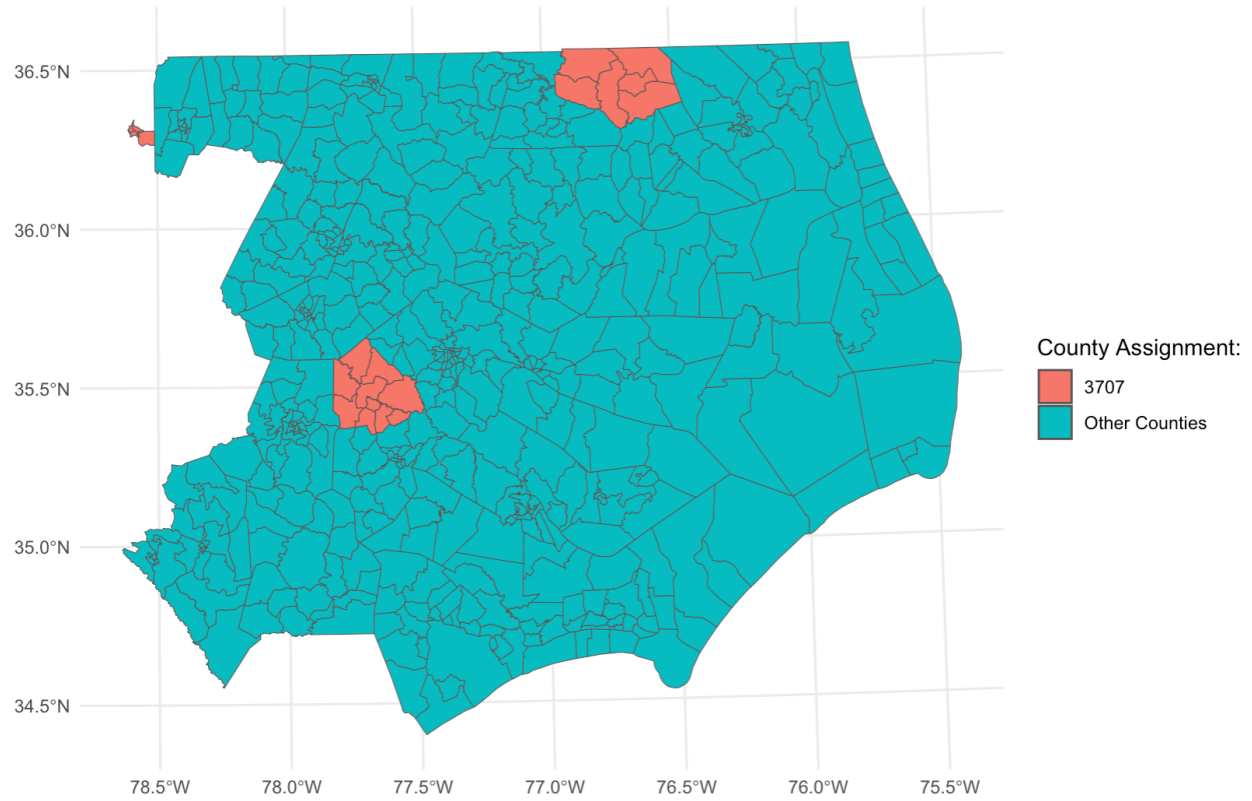




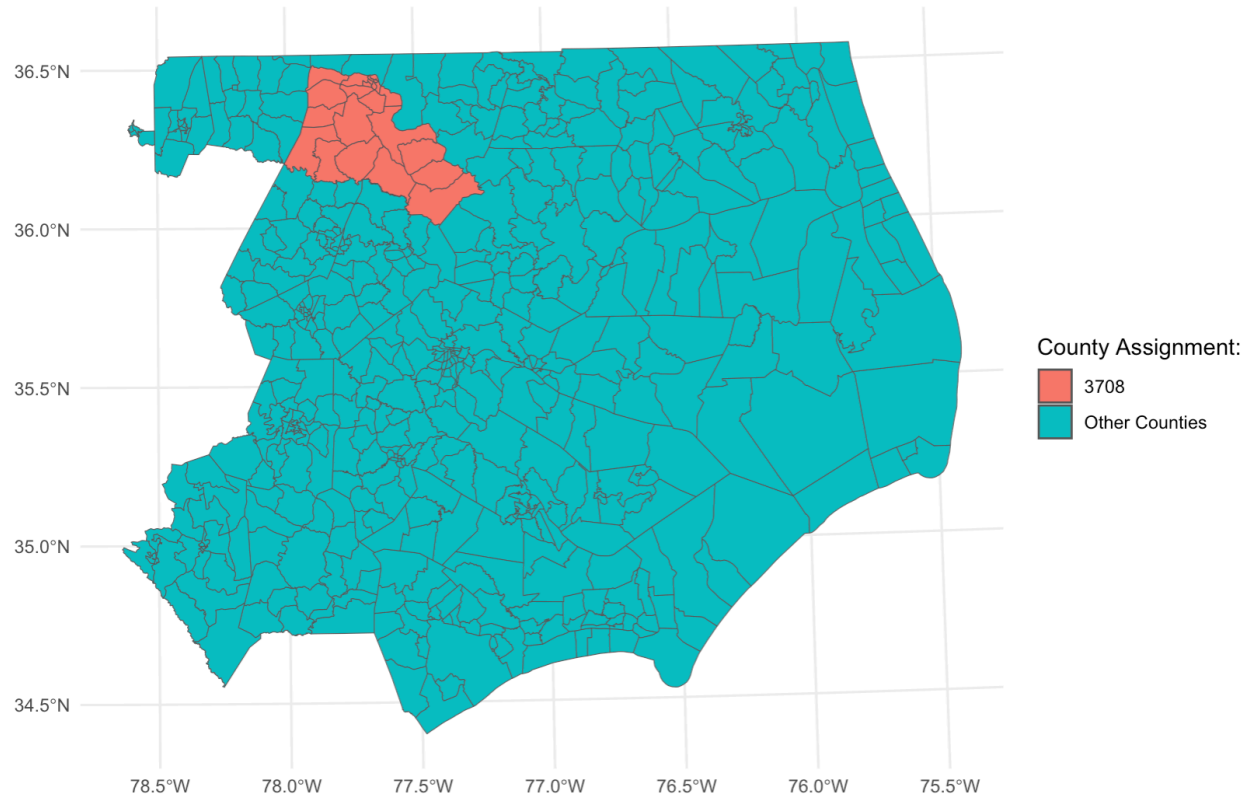
County “3706” includes Duplin and Edgecombe counties



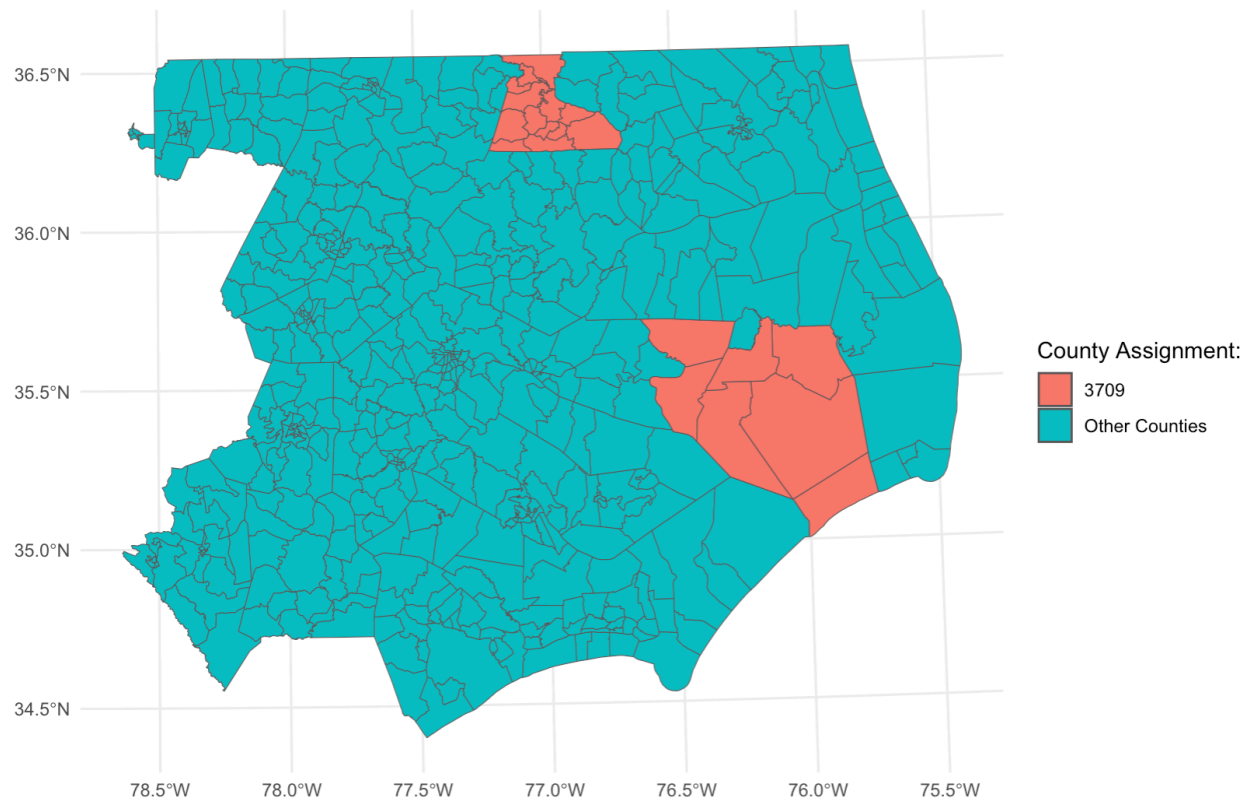
County “3707” includes Granville, Gates, and Greene counties



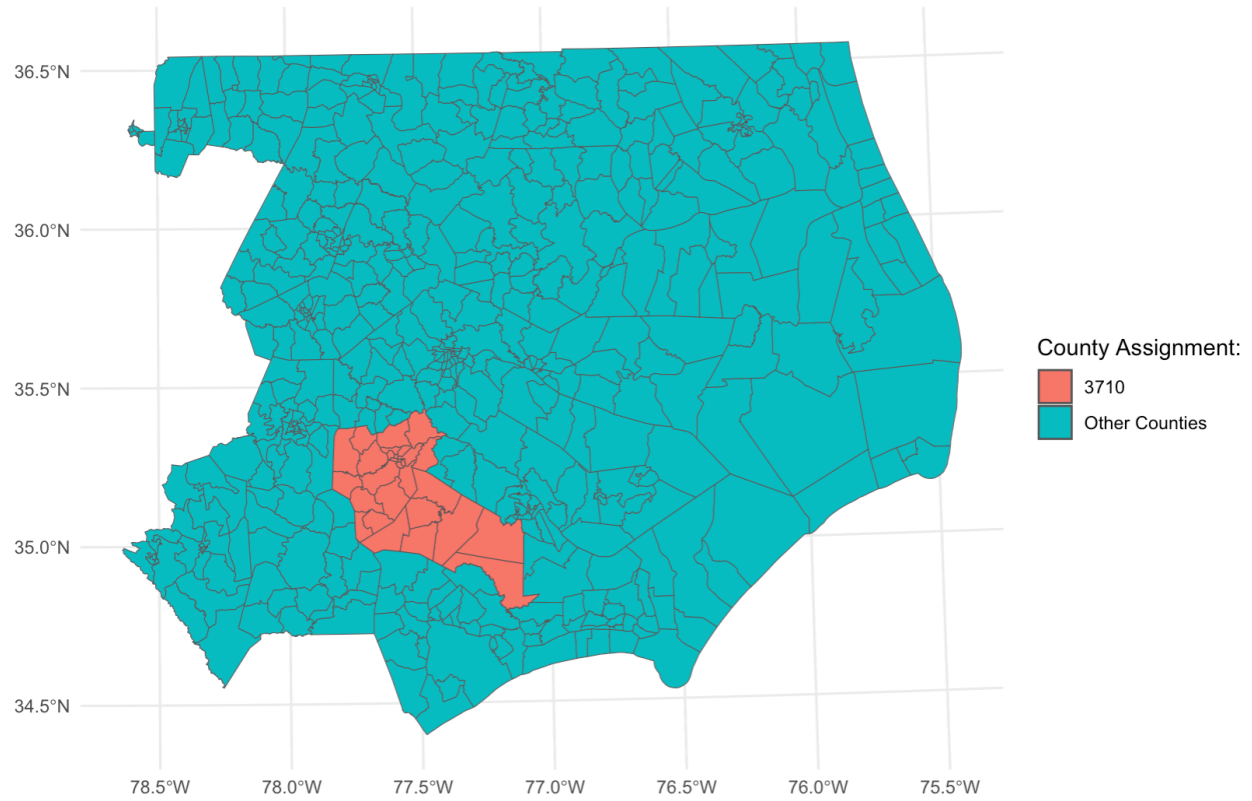
County “3708” contains Halifax County



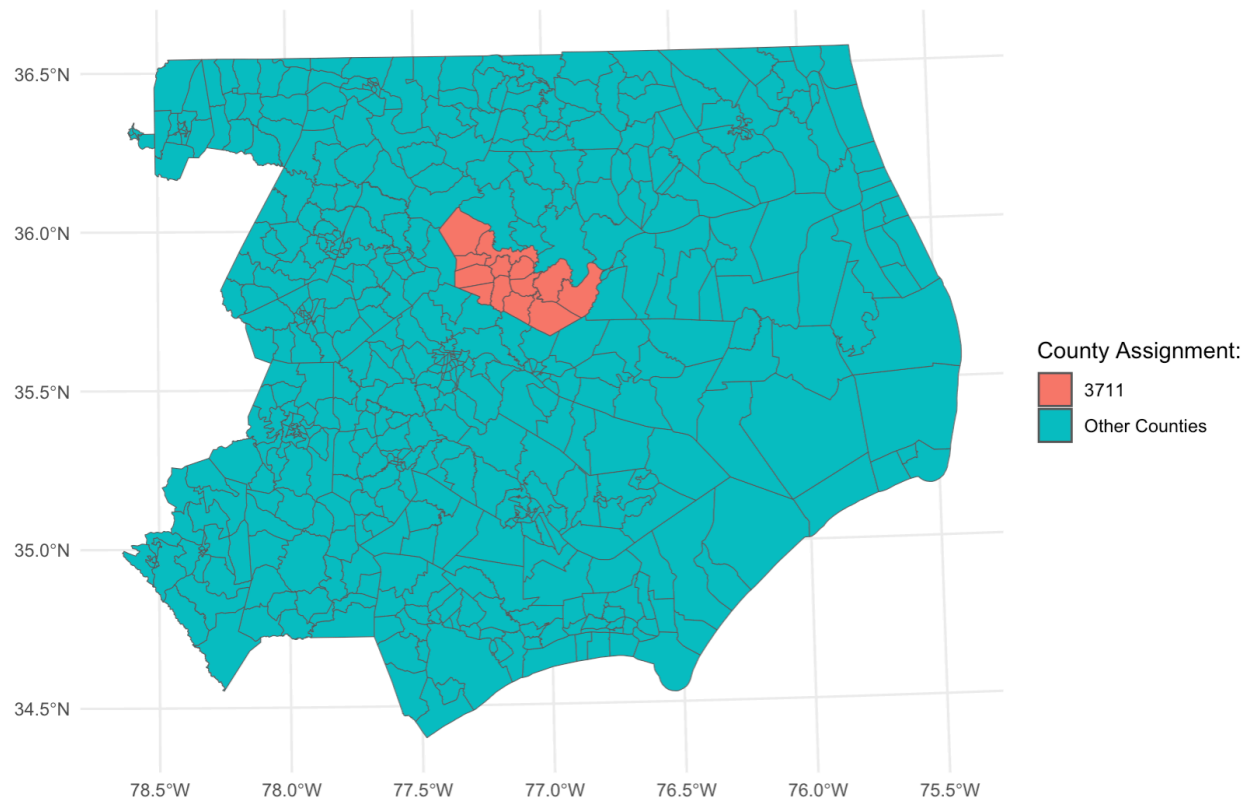
County “3709” contains Hertford and Hyde counties



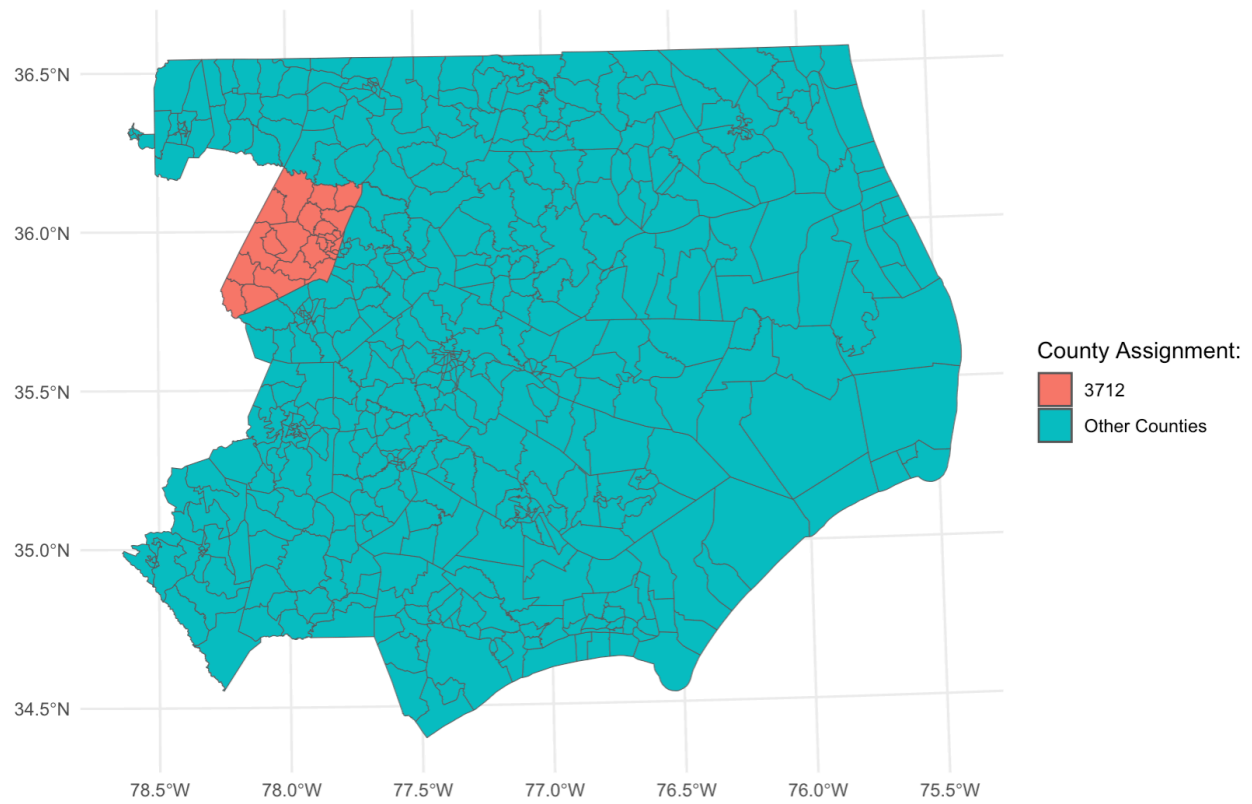
County “3710” contains Jones and Lenoir counties



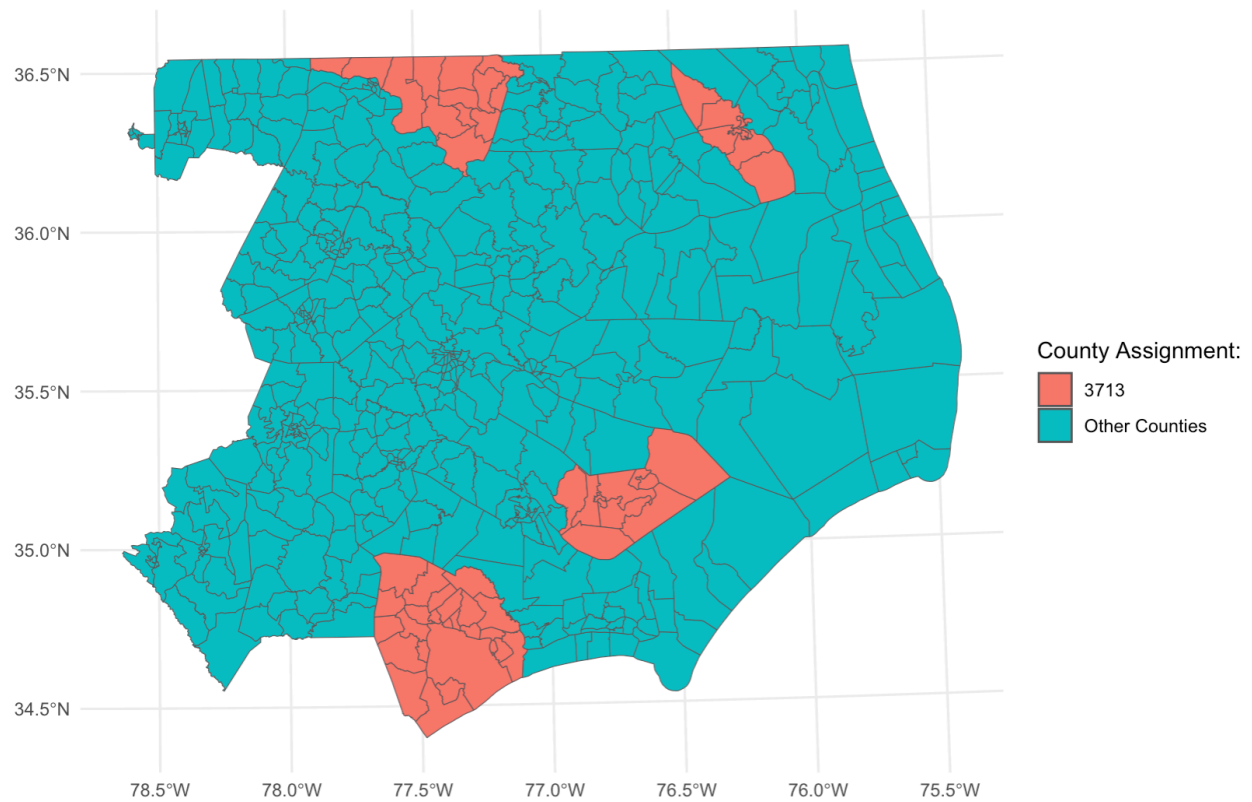
County “3711” contains Martin County



County “3712” includes Nash County

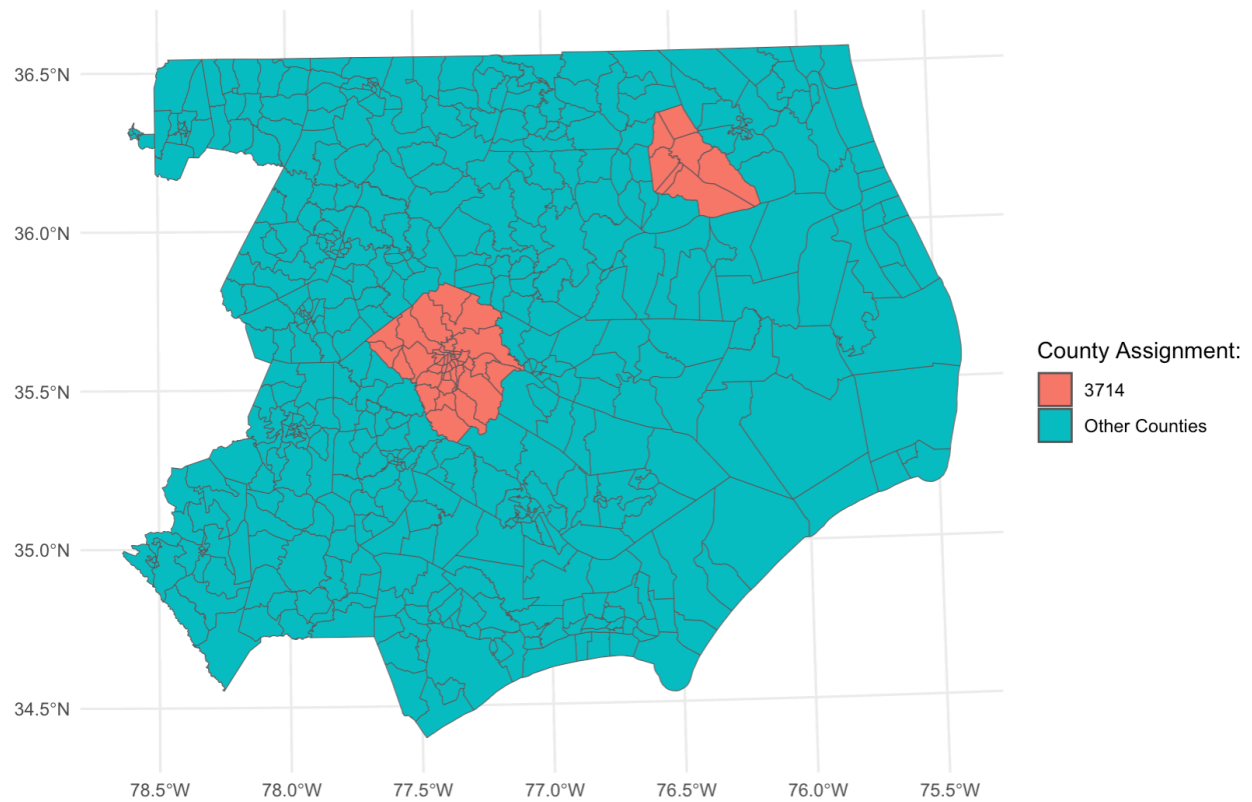


County “3713” includes Northampton, Pasquotank, Pamlico, and Onslow counties

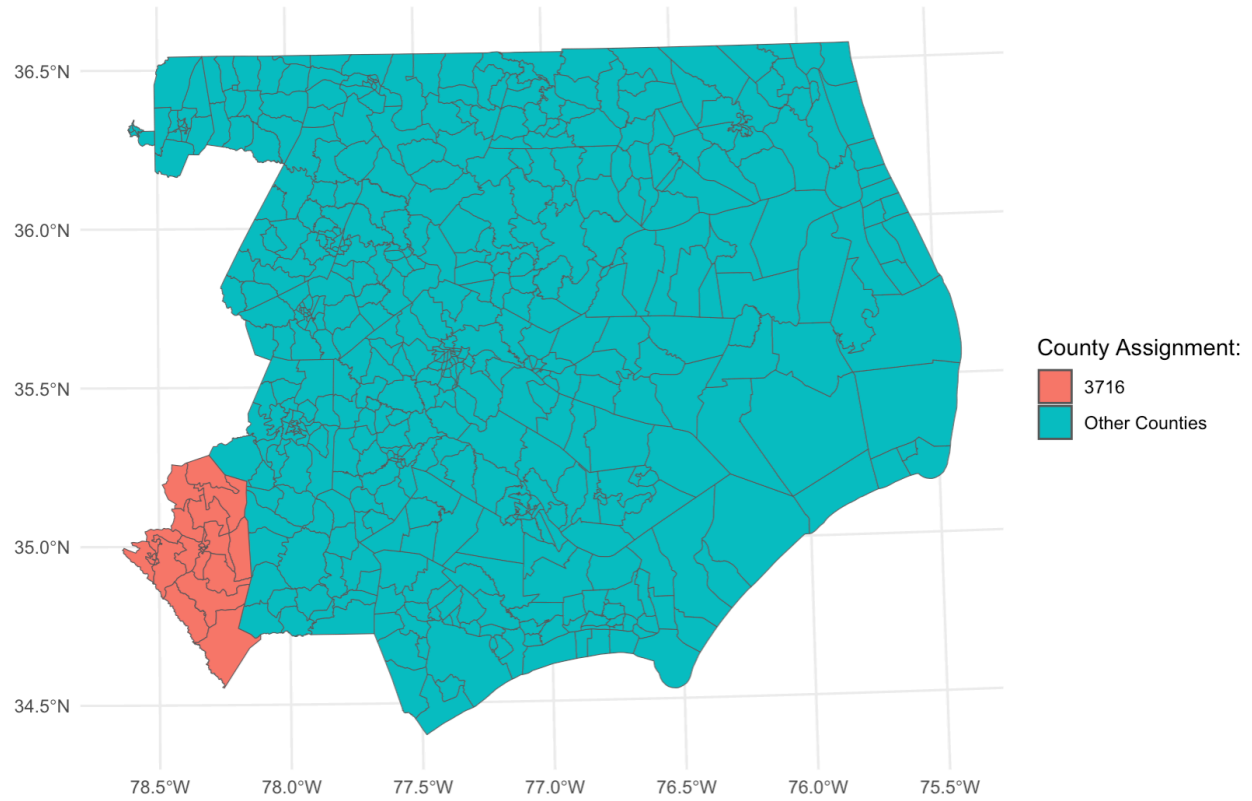




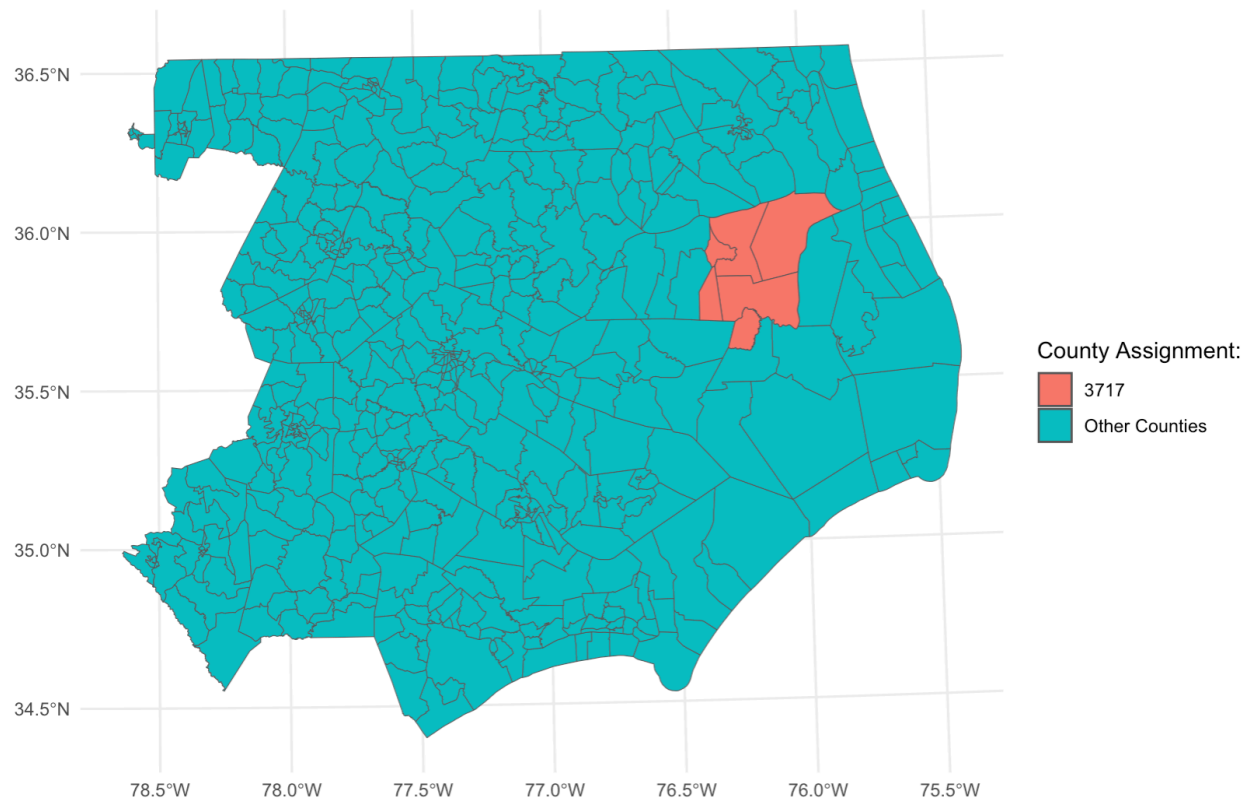
County “3714” includes Pitt and Perquimans counties



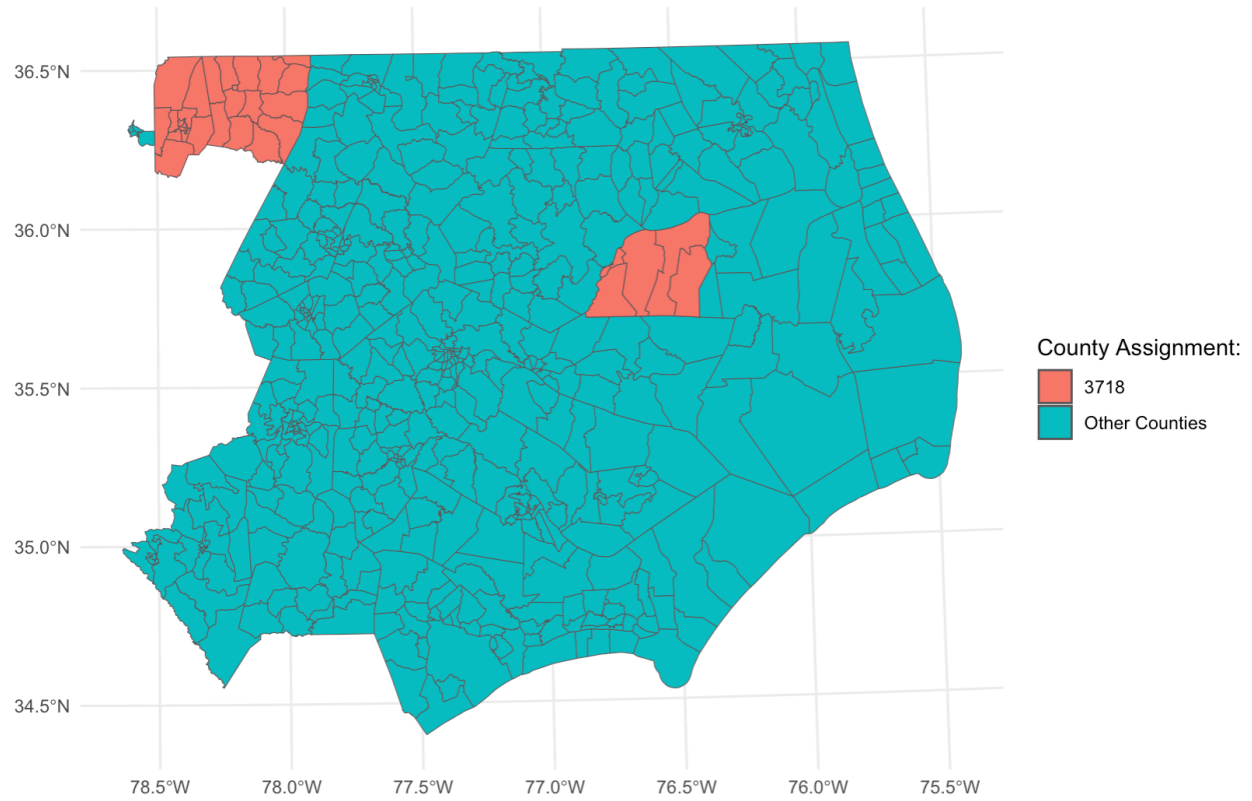
There is no County “3415” in the dataset. County “3416” includes Sampson County



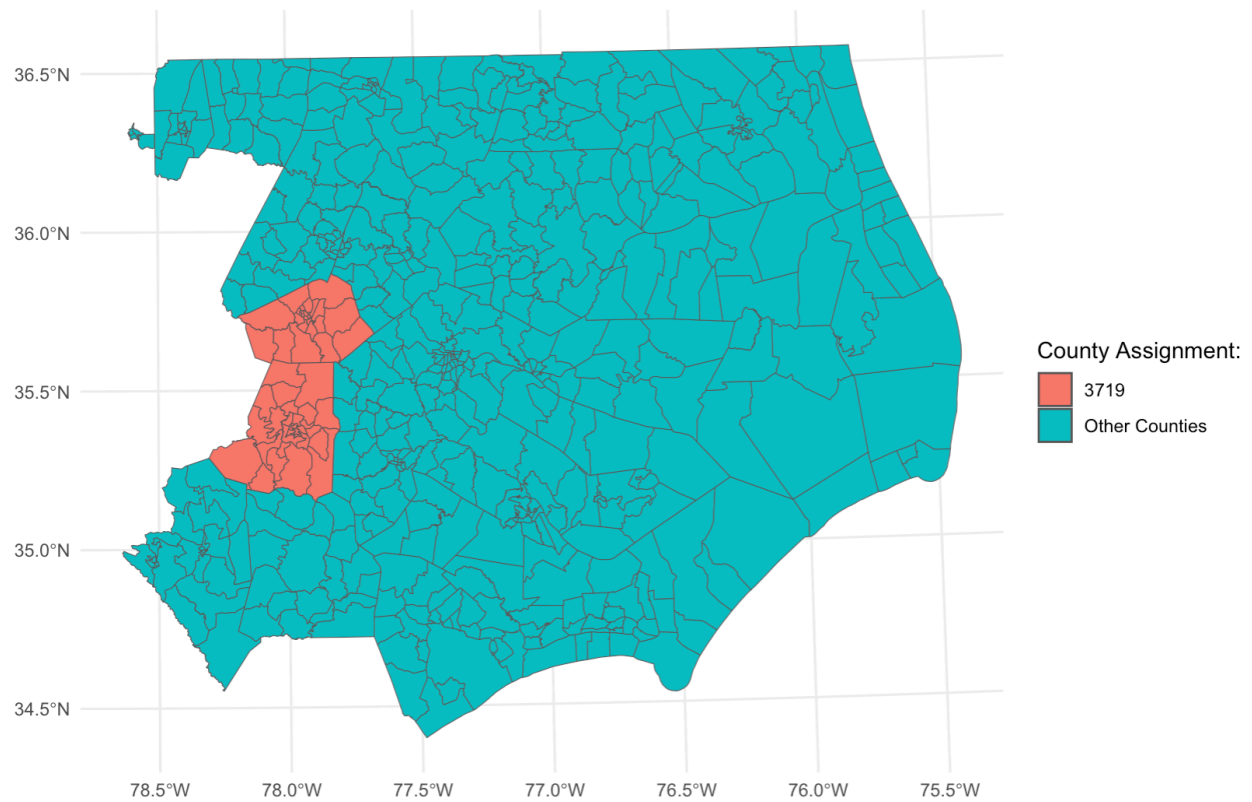
County “3717” includes Tyrrell County



County “3718” includes Vance, Warren, and Washington counties



County “3719” includes Wilson and Wayne counties



## Replication Code Run:

The code below is what I ran using the “sims\_two\_districts.R” file provided with Dr. Rodden’s backup data that accompanied his November 18, 2025 report.

```
library(sf)
library(tidyverse)
library(centr)
library(lmtest)

vtds.sims <- read_sf("export.shp")

vtds.sims <- vtds.sims %>%
  rowwise() %>%
  mutate(dem_votes = sum(c_across(starts_with("E_") & ends_with("_Dem"))),
    na.rm = T),
  rep_votes = sum(c_across(starts_with("E_") & ends_with("_Rep"))),
    na.rm = T)

names(vtds.sims)
table(vtds.sims$COUNTY)

#The shapefile provided did not map.
#This code makes no changes to the underlying file, but corrects the geometry
to map accurately.
st_geometry(vtds.sims) <- st_zm(st_geometry(vtds.sims), drop = TRUE, what =
"ZM")

ggplot(vtds.sims) + geom_sf() + theme_minimal()

ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3701, "3701", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3713, "3713", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")

nsims <- 50000
ndists <- 2
compactness <- .5
target_pop <- 745671

# smc simulation map setup.
map_smc <- redist_map(vtds.sims,
  total_pop = T_20_CENS_,
  pop_tol = .001,
  ndists = ndists
)

# run simulations
sims_smc <- redist_smc(map_smc,
  nsims = nsims / 10,
  verbose = TRUE,
  runs = 10,
  counties = COUNTY,
```

```


compactness = compactness
)
summary(sims_smc)

ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3701, "3701", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3702, "3702", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3703, "3703", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3704, "3704", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3705, "3705", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3706, "3706", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3707, "3707", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3708, "3708", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3709, "3709", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3710, "3710", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3711, "3711", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3712, "3712", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3713, "3713", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3714, "3714", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
#There is no county = 3715 in the data
# ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3715, "3715",
"Other Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3716, "3716", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3717, "3717", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3718, "3718", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")
ggplot(vtds.sims) + geom_sf(aes(fill = ifelse(COUNTY == 3719, "3719", "Other
Counties")))) + theme_minimal() + labs(fill = "County Assignment:")

```

I, Dr. Michael Barber, acting in accordance with 28 U.S.C. 1746, Federal Rule of Civil Procedure 26(a)(2)(B), and Federal Rules of Evidence 702 and 703, hereby declare that the foregoing is true and accurate to the best of my knowledge.

Dated: November 19, 2025



---

Michael Barber