

TECHNICAL REPORT:

100% Retabulation Audits: 2022 Primary and General Election Audit Data and Ballot Images from Leon County, Florida

Authors:

Lonna Rae Atkeson, Florida State University

Yimeng Li, Florida State University

Lisa A. Bryant, California State University, Fresno

Wendy L. Hansen, University of New Mexico

Austin Cutler, Florida State University

Carson BA Cary, Florida State University

Eli McKown-Dawson, Florida State University

Kenneth Mackie Fitchburg State University

Project Partners:

Mark Earley, Leon County Supervisor of Elections

Michael Morley, Florida State University

Citation: Atkeson, Lonna Rae. Yimeng Li, Lisa A Bryant, Wendy L. Hansen, Austin Cutler, Carson BA Cary, Eli McKown-Dawson, Kenneth Mackie. 2023. "100% Retabulation Audits: 2022 Primary and General Election Audit Data and Ballot Images from Leon County, Florida." LeRoy Collins Institute, Florida State University. Available at: https://2022voterdata.lci.fsu.edu/wp-content/uploads/2024/01/23-LCI-002-Audit-Technical-Report-V1_4.pdf.

Abstract: We developed a 2022 primary election audit and a general election data dashboard that includes images of ballots cast in Leon County as a pilot project to help identify obstacles and methods to place these types of data online for public review. We embedded these data on a website we built for this purpose. Additional pages on the website describe post-election audit processes, data produced by election officials to verify the election outcomes, interesting ballots found in the data, and information on how to use the dashboard. Obstacles we faced in presenting these types of data in an accessible format included protecting voter privacy and ballot secrecy, correcting ballot orientation, managing the quantity of the data presented, speed and performance in data presentation, developing the dashboard design, and creating a web design and useable layout. In this report, we discuss our design, the obstacles we faced, and the solutions we found.

Background on Post-election Audits

Free and fair elections are one of the pillars of American democracy, but ensuring a healthy democracy via a safe, secure, and efficient election process involves more than just making certain that the election process works well. The public must perceive that voting is easy and accessible, that their privacy is protected, and that results are determined fairly and accurately, without partisan bias or technological flaws.

Election auditing is a critical component of election security and integrity. Today, 45 states perform some sort of post-election audit, and two additional states have post-election audit pilot projects to ensure the accuracy of tabulators and the election outcomes.¹ Post-election audits are meant to be a public check on the voting system. Transparency around post-election audits helps to ensure a fair and accurate election process that maintains voter privacy while also helping to minimize the potential for fraud.² Their purpose is to ensure that only eligible voters participated in the election and that the equipment used to count ballots worked correctly and produced accurate results.³

At a time when some voters doubt the accuracy and veracity of elections, post-election audits can be a powerful tool to assure the public and especially stakeholders (candidates, election staff, etc.) that the results accurately reflect the will of the people. Therefore, our work seeks to expand election transparency and enhance election integrity through a pilot project that presents the data from the 2022 primary and general election post-election 100% retabulation audits in Leon County, Florida, including images of the ballots, in an accessible and public space. The audit shows that two independent tabulators accurately produce the same count votes, and therefore the same winners.

While the task seems simple, a great deal of variation exists in how states conduct post-election audits, and there are many ways to present the collected data. There is no standardized format across jurisdictions to present and inform voters of the outcome. Moreover, despite the theoretically public nature of post-election audits, less than half of the states make data from the post-election audits available in usable form for the public or researchers to examine.⁴ If data are not available for review, an audit cannot impact public confidence. In addition, there are many issues related to voter privacy, ballot secrecy, and vote buying that need to be considered when making ballot images and other audit data public. Our applied case study addresses these issues by developing a 2022 audit data dashboard that includes images of nearly all the ballots cast in both the primary and general elections in Leon County, Florida. The primary and general election data are embedded in two separate web pages on a website we built for this purpose. Additional pages on the website describe post-election data processes, present data produced by the state and county to verify the election outcomes, offer

¹NCSL, Post-election Audits, available at: <https://www.ncsl.org/research/elections-and-campaigns/post-election-audits635926066.aspx#state%20reqs>.

²See Huefner, Steven F., Daniel Tokaji, & Edward B. Foley. 2007. "From Registration to Recounts: The Election System of 5 Midwestern States," available at: <https://cpb-us-w2.wpmucdn.com/u.osu.edu/dist/b/90788/files/2021/05/From-Registration-to-Recounts.pdf>.

³See U.S. Election Assistance Commission. "Election Audits Across the United States." <https://www.eac.gov/election-officials/election-audits-across-united-states>.

⁴ See Jaffee, Jacob, Samuel Baltz, Fernanda Gonzalez, Kaiwen Guo, and Charles Stewart III. 2023. "Efficiency of Risk Limiting Audits," presented at the Southern Political Science Association's annual meeting, St. Pete, Florida.

images of “interesting” ballots found in the data, and provide information on how to use the dashboard.

You can find our dashboards and website [here](#).

It is worth noting that throughout the project, we faced many obstacles to presenting these types of data in an accessible format. Obstacles included protecting voter privacy and ballot secrecy, correcting ballot orientation for viewing the images, managing the quantity of the data presented, enhancing the speed and performance of data presentation in the dashboard itself, developing the dashboard design, and creating a web design with a useable layout. We describe in more detail below the obstacles we faced and our solutions in this report.

Data and Methods

Information about Leon County, Florida

Leon County is home to Tallahassee, the capital of Florida. It is located in the state’s Panhandle region. Leon County has roughly 203,000 active registered voters and is considered a medium-sized jurisdiction.⁵ It is a predominantly Democratic County with a majority (50.8%, 100,427) registered Democrat, about three in ten (29.7%, 58,783) registered Republican, about two in five (19.9%, 39,834) registered No Party Affiliation (NPA), and 1.8% (3,564) registered in some other non-major party.⁶

The 2022 statewide primary was held on Tuesday, August 23, with early voting available to all eligible voters from August 13-20. The general election was held on Tuesday, November 8, 2022, with early in-person voting available from October 29-November 5. Florida also has a no-excuse vote-by-mail policy that is very popular. Vote-by-mail ballots could be returned either by mail, in person at the Supervisor of Elections’ office, or at mail ballot drop boxes by 7 PM on Election Day.

The primary election in Florida is “closed,” meaning that only registered partisans can participate in their party’s primary election. However, in Leon County, city and county contests are nonpartisan and open to any eligible voter in the county. These facts led to three ballot types in the primary: 1) Democratic and nonpartisan contests; 2) Republican and nonpartisan contests, and 3) Nonpartisan contests only. Democrats made up 41,226 (62%%) voters and Republicans made up 19,297 (29%) voters, leaving 6,348 (9%) non-major party identifiers, including NPAs, for a total of 66,871 ballots cast. The smaller turnout for non-Democrats led to ballot privacy concerns among ballot images especially for the primary, which we discuss in detail below.

⁵See: <https://www.leonvotes.gov/> for recent data on voter registration.

⁶These data come from the Leon County Supervisor of Elections website, are time stamped August 1, 2023, and can be found in their current form here: https://www.leonvotes.gov/Portals/Leon/Documents/Data_Lists/Current%20Data/Snapshot/Active%20Voters%20by%20District.pdf

In Florida when a contest has only one candidate running, it is excluded from the ballot and the candidate wins by acclamation. This results in large variation in the number of contests across ballot types especially in the primary. Democrats had the longest and most interesting ballot, with a somewhat competitive statewide gubernatorial contest primarily between former Governor Charlie Crist and Agriculture Commissioner Nikki Fried. Also on the Democratic ballot were a statewide U.S. Senate contest, Florida Attorney General contest, and Florida Agriculture Commissioner contest, all of which were less competitive than the gubernatorial contest but still had multiple contenders.

Republicans had one statewide contest, for Agriculture Commissioner, but it was not particularly competitive – which likely helps to explain the low turnout for Republicans. In addition, there were two nonpartisan contests, one county judge contest and one county commissioner contest, in which all voters could participate. Down-ballot contests for which only some voters could vote included contests in House districts, County Commission districts, School Board districts, and City of Tallahassee mayor and City Council seats.

The shortest Democratic primary ballot included seven contests, while the longest included 12 contests. The shortest ballot for Republicans was four contests and the longest was eight. The shortest nonpartisan ballot was three contests and the longest was seven.

In the general election there were a total of 117,456 voters: 61,696 (53%) were Democrats, 37,860 (32%) were Republicans, and 17,900 (15%) were non-major party identifiers including NPAs.

Data

We obtained the data from the Leon County Supervisor of Elections for the 2022 primary and general elections in Leon County, Florida. We received the primary data on September 19, 2022, and the general election data on June 5, 2023. The data come from the auditing system used by the Leon County Supervisor of Elections – the ClearAudit system – and include:

- An image copy of both sides of the ballot for all ballots cast in Leon County, Florida. In the primary election, there were no contests on the back side of the ballot; in the general election, contests take up both sides of the ballot.
- The audit summary includes an aggregate comparison of the vote totals, undervotes, and overvotes for each candidate between the official vote count and the audit tally. The official vote count comes from Dominion tabulators; the audit tally comes from a ClearBallot tabulator.
- The audit cast vote records (CVR) contain detailed information for each ballot indicating whether a vote was cast in a contest, and if so, for which candidate(s), according to ClearAudit's determination based on the markings on the ballot. The audit CVR also includes information on precinct, vote mode, and language associated with each ballot.

- The audit oval confidence files include a confidence score or ranking for each oval within a contest (i.e., candidate or choice)⁷, grouped by vote mode (Election Day, Early Voting, or Vote-by-Mail) and ballot disposition or vote type (vote, undervote, overvote, or vote for an alternative candidate).

We supplement the audit data with other state- and county-level voting records, including the county and state canvas.

Data Processing

For our dashboard, we first cleaned the audit summary to show for each candidate: (1) the contest; (2) for the primary elections, also the party of the contest (Democratic Party, Republican Party, or nonpartisan races); (3) total ovals counted according to the official system (Dominion); (4) total ovals counted according to the audit system (ClearAudit); (5) the difference in ovals counted by the two systems; (6) total votes recorded according to the official system (Dominion); (7) total votes recorded according to the audit system (ClearAudit); (8) the difference in the votes recorded by the two systems; (9) total overvotes recorded by the audit system (ClearAudit); and (10) total undervotes recorded by the audit system (ClearAudit).

We also merged the audit CVR and the audit oval confidence files to show the following on our dashboard: (1) ClearAudit's determination for each oval (i.e., choice, meaning candidate) on each ballot based on the markings or the lack thereof (vote, undervote, overvote, or vote for an alternative candidate); and (2) the confidence ranking the ClearAudit system places on its determination by vote mode and determination. In the same table, we also show whether each ballot is an Election Day, early, or mail ballot and provide a link to the ballot image.

For our analysis of the audit data, we further append the precinct-level election results to the audit CVR and oval confidence files. Combining the audit data and the precinct-level election results enables us to identify potential ballots that explain the differences between the official and the audit counts.

Auditing Methodology

Leon County uses Dominion hardware and software for its official ballot tabulation. The software and hardware are certified by the State of Florida. For its official tabulation, the county uses Dominion-based products, including Dominion's Democracy Suite software for its in-person systems and Image Cast Central software for its mail ballot software.⁸

⁷We say choice because some of the items on the ballot were amendments or other nonoffice ballot questions.

⁸See Florida Secretary of State's page about vote tabulation certification here: <https://dos.myflorida.com/elections/voting-systems/about-voting-systems/> and a list of what each county used in 2022 here: <https://files.floridados.gov/media/705877/voting-systems-in-use-by-county-20220830.pdf>.

Leon County uses the ClearBallot ClearAudit for the 100% retabulation audit. All ballots are inserted into the ClearAudit tabulator at the Leon County Election Center. ClearAudit is a browser-based central count tabulation system. ClearAudit can tabulate ballots created by all major voting systems certified in the state of Florida, offering a truly independent retabulation. By independent we mean that there is no hardware or software in common between the official (Dominion) tabulation systems and audit (ClearBallot) tabulation systems.

The tabulators use different methods to identify votes. Dominion software searches for a vote by starting at the center of an oval and moving outward to its ends. ClearAudit software uses a larger zone fully encompassing the oval for each contest to search for and identify a vote for each candidate. By looking at a larger contest area around the oval, the ClearAudit system can identify stray marks and marks outside of the oval that could be a vote. For example, if someone circles the oval instead of filling it in, the ClearAudit system will see those marks and can better identify whether those are likely votes or not. ClearAudit's software also includes a vote visualization tool that potentially enables auditors to find uncounted or incorrectly counted ballots.

Once the election officially begins, Florida county Election Supervisors are allowed by statute to process and record mail ballots as they arrive.⁹ Immediately after these ballots are processed by the Dominion high-speed tabulator, they are taken to the ClearAudit tabulator and processed a second time. Ballots from early voting are counted by the Dominion ICE tabulators on site and are then transported to the election center at the end of each day of early voting and processed through the ClearAudit tabulator the next morning at the election center. Ballots from Election Day are transported to the election center on election night and are inserted into the same Dominion and then ClearAudit tabulator for counting the next day.

In Leon County, ballots are read by the Dominion system and then read by the ClearAudit tabulator over the course of the election. Ballots are tabulated within mode by batch and given consecutive numbers such that the first batch tabulated is designated by its mode (mail =VBM, Election Day=ED, and Early Vote=EV) and then its batch (001, 002, etc.). The first ballot tabulated within each batch starts the counting at 1. So, the first ballot of the first batch of vote-by-mail ballots counted by the ClearAudit tabulator would be VBM-001+100001, the second ballot of the first batch of vote by mail ballots would be VBM-001+100002, etc. The ballot images record this number in the file name, so the numbering scheme can later be used to locate the image of a particular ballot.

⁹Florida Statutes Section 101.68(2)(a).

Designing the Dashboard

The primary goal of our project was to make the audit data and ballot images accessible to the public for review. To accomplish this, we built a web page and a data dashboard for both the primary and general elections. The dashboard contains all the ClearAudit audit data and links to the ballot images.¹⁰

Our first attempt at making the dashboard was in the Tableau software and included only the primary election data. We spent a great deal of time and resources designing the dashboard and making it user-friendly. While the appearance of the dashboard was acceptable, it proved to be unusable due to the size of the data. We upgraded the dashboard from the free version of Tableau to the enterprise version of Tableau, but we ultimately ran into the same issues. Much of the data would not load at all, and when it did, it was incredibly slow.

After determining that Tableau was not a viable option, we considered Microsoft's Power BI and the R package Shiny, both of which are designed to make interactive apps for displaying data. After creating drafts of the primary dashboard in each program, we determined that Shiny was the best option (See Appendix A for Shiny Code), for two key reasons. First, the user experience on the dashboard using Shiny is better, as we have more control over the user interface. Second, since the Shiny dashboard uses code to create the app, we can reproduce the dashboard for multiple elections with relative ease, maintaining consistency with the dashboard appearance and interface as long as the data are structured the same way each time. With Power BI, each election would necessitate creating a new dashboard.

This choice does not come without its drawbacks. While the data loads significantly faster in the Shiny app compared to Tableau, with the Shiny app taking just under 45 seconds to load the primary election data as opposed to the minute-plus of the Tableau dashboard, Power BI loaded the data immediately. We did not attempt embedding the Power BI version of the dashboard into the website, but in our tests it was able to load the data nearly instantly. Furthermore, the Power BI dashboard was able to load all the data instantly, while the Shiny app dashboard loads the data in chunks of 10,000 rows of the data on each page. While this is minimally disruptive to the user's experience, we felt that it was worth that sacrifice for the other benefits of using the Shiny app, including the appearance of the dashboard and the overall usability and replicability of the dashboard. The Power BI dashboard was unable to include the Leon County precinct map, which was another reason to choose the Shiny app. Additionally, Power BI was overall more restrictive on how we were able to present the data.

¹⁰See our dashboard page here: <https://2022voterdata.lci.fsu.edu/dashboard/2022-general-election/>.

We were also under the impression that Shiny would be completely free due to R being entirely open source. While it is true that creating the dashboard itself was free, and the dashboard for the primary audit was able to be hosted for free, the size of the data in the general election audit required that we upgrade to a paid account with Shiny. While the primary election app was able to be uploaded to Shiny for free, there were limits on the amount of RAM our app could use for free, and we exceeded those limits for the general. That said, the cost of the account with Shiny is comparable to, although slightly greater than, the cost of a subscription to Power BI.

The Shiny app dashboard utilizes a combination of R code, specifically from the tidyverse, sf, DT, plotly, Shiny, Shinydashboard, and htmltools packages, CSS, and JavaScript. The tidyverse code was used to clean the data within the dashboard, such as changing column names for the two tables. Tidyverse code was also used with code from the Shiny package to filter the tables in response to the interactive elements on the dashboard, including the dropdown menus for party and contest, as well as the checkboxes for vote type and vote method. Code from the Shiny package was utilized to make the base elements of each dashboard, such as the dropdown menus, and the DT package was used to create the two tables. Some elements of the tables were not able to be changed with R code, such as the summing at the bottom of the first table, and JavaScript was used to add these elements. JavaScript was also used to ensure that the sizing and placement of the map would fit within the bounds of the dashboard. The sf package was used to load the shape file for the map of Leon County's precincts, and the plotly package was used to create the map. The map is static for the primary election to protect voter privacy in smaller precincts but is fully interactive in the general election dashboard.

All of the aesthetic elements of the dashboard were changed using CSS code. This includes the color, size, and shape of each element of the dashboard. The placement of each element of the dashboard was determined through a combination of Shiny, Shiny dashboard, and CSS code. One concern we had with making the dashboard was to ensure that it would be accessible to users on a wide range of devices. We determined the best dimensions of the dashboard through trial and error, testing the dashboard's appearance on larger 27-inch monitors and small 13-inch screens, to ensure that the dashboard would fit in a wide range of applications. The dashboard is not mobile phone-friendly due to the size and complexity of the data.

Describing the Dashboards

Figures 1 and 2 show screenshots of the dashboards for the primary and general elections.

Figure 1. Image of Primary Election Dashboard

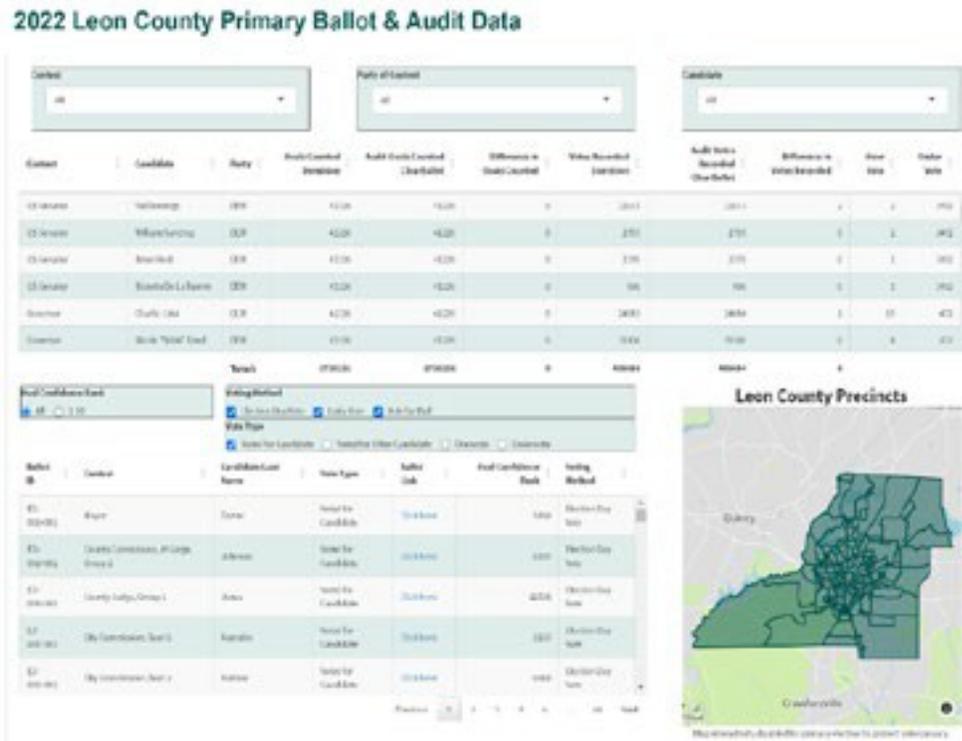
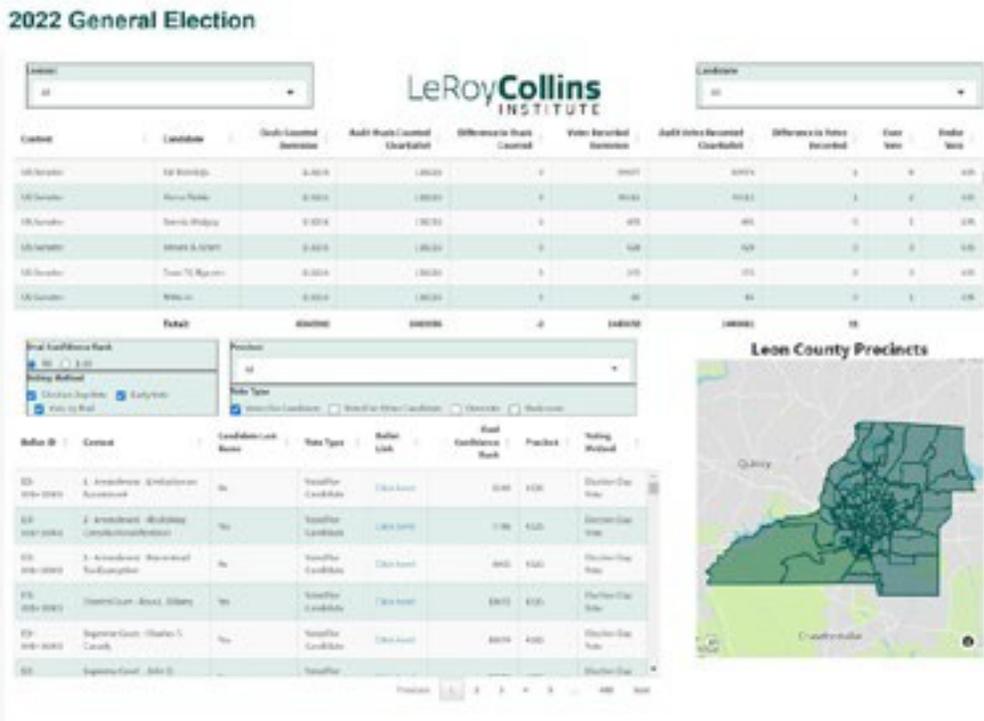


Figure 2. Image of General Election Dashboard



The 2022 Primary Election Dashboard

Focusing first on Figure 1, which presents the primary election data in table format, three filters appear at the top of the dashboard. The first filters the data and corresponding ballot image links by contest, the second by candidate, and the third by party.

Figure 2, which presents the general election, has only two filters – contest and candidate – because party does not make sense when we move to interparty competition. These filters manipulate data in both Table 1 and Table 2 of the dashboard, which have nearly identical categories or columns of data for both the primary and general elections. Table 1 shows the aggregated auditing data from the ClearAudit tabulator. The order of the contests in filter 1 is the same as is found on the ballot, therefore the Democratic US Senate contest appears at the top, followed by US House, then Governor, etc.

The ClearAudit audit system examines each oval, so the total count at the bottom of the table reflects that fact. For example, if we were to filter the contest for the Democratic nomination to the Senate contest, we would see four candidates with 41,226 ballots (voters). To determine the total number of ovals, the calculation is 41,226 ballots X 4 candidates, which equals 164,904 total ovals counted. Every oval gets a disposition code or count, which includes 1) vote for candidate, 2) vote for other candidate, 3) overvote, or 4) undervote. Thus, while there are 164,904 ovals recorded in the US Senate contest, there are only 41,226 eligible voters participating in this contest.

The third and fourth columns show the number of ovals counted for each candidate. This should equal the total number of eligible voters who participated in the election. These numbers should be the same across systems and subtract to zero – which they do, as shown in column 5. If these numbers were not the same, that would suggest that one tabulating system counted more ballots than the other and there could be questions about ballot chain of custody, ballot design (tick marks), or related issues.

Columns 6 and 7 show the number of votes counted for each candidate in a particular contest, first for the official tabulation and then for the audit tabulation. For example, Democratic Senate candidate Val Demings received 33,571 votes on the official count and 33,573 votes with the ClearAudit System. Column 8 shows the difference between the official vote count and the audit vote count. In the case of candidate Demings, two more votes were found on the ClearAudit system than the official first tabulation system. This suggests that ClearAudit converted two votes that were not counted and were likely considered undervotes in the first tabulation of votes for Demings.

Column 9 shows the overvotes and column 10 shows the number of undervotes as recorded by the audit system. Overvotes are when a voter votes for two or more candidates in a contest and therefore the votes go uncounted. Undervotes are when voters choose to skip a contest and not vote for any candidate. In the case of Val Demings, the data show that seven ballots included an overvote for Senate, and out of the 41,226 eligible voters 1,492 chose not to select any candidate for US Senator, and therefore undervoted.

All in all, when we look at the primary dashboard results across all the primary contests, we can see that there is only an 8-vote difference between the first tabulation and the second audit tabulation. That is 8 out of 1,736,232 ovals counted, or a 99.9985% match. In all cases in the primary election, differences are because the ClearAudit system found and counted one or two more votes than Dominion. It found one extra vote in each of the following contests: Governor, Democratic Commissioner of Agriculture, County Judge, County Commissioner At Large, County Commissioner District 5, School Board District 1, and mayor; and it found two extra votes in the Democratic US Senate race.

2022 General Election Dashboard

In the general election example shown in Figure 2, we can see that with nearly twice as many ballots, there is more variation across the two tabulation systems than we saw in the primary. The first table is identical in layout for both dashboards, but the general election only has two filters at the top (one for contest and another for candidate). Because this is not a primary contest with party ballots, we did not include a party filter.

In the general election, a total of 45 discrepancies in votes were recorded between Dominion and ClearAudit out of 6,060,088 ovals counted. This translates to a 99.9983% match. We see some cases where ClearAudit found one more vote to count than Dominion (e.g. votes for Governor DeSantis, Attorney General Moody, State Senator Simon in District 3, Supreme Court Justice Couriel, District Court Judges Kelsey, Long, and Rowe, County Commissioner Johnson, Mayor, and contests for all three amendments). There were also a few cases where ClearAudit found 2 or 3 more votes to count than did Dominion: both Commissioner of Agriculture candidates and Supreme Court retentions for Justices Canady, Grosshans, Labarga, and Polston. We also see a handful of cases where Dominion counted one more vote than ClearAudit (US Senate candidate Demings, US Representative Lawson, CFO candidate Hattersley, State Senator Ausley, State Representative Franklin, Supreme Court retentions for Justices Canady, Couriel, Labarga, and Polston, District Court retentions for Judges Bilbrey, Long, and Winokur, County Commissioner O'Keef, and School Board District 3 candidate Jones), and one case where Dominion counted two more votes (Mayor candidate Dozier).

Three contests (state House District 8 and County Commission Districts 2 and 5) show differences in the number of ovals counted. In the case of the state House District 8, two candidates had one fewer counted oval in the audit than in the official tabulation system. This is an odd result, and we are unsure how to interpret it at this time. Leon County ignores the differences because they do not change any outcomes, but they seem to suggest that there may be a capture problem that should be more closely examined. It does not appear in other races in which these contests are nested, so it appears likely to be an intraballot counting problem and not a missing ballot or chain of custody issue. Something similar is seen in County Commission Districts 2 and 5.

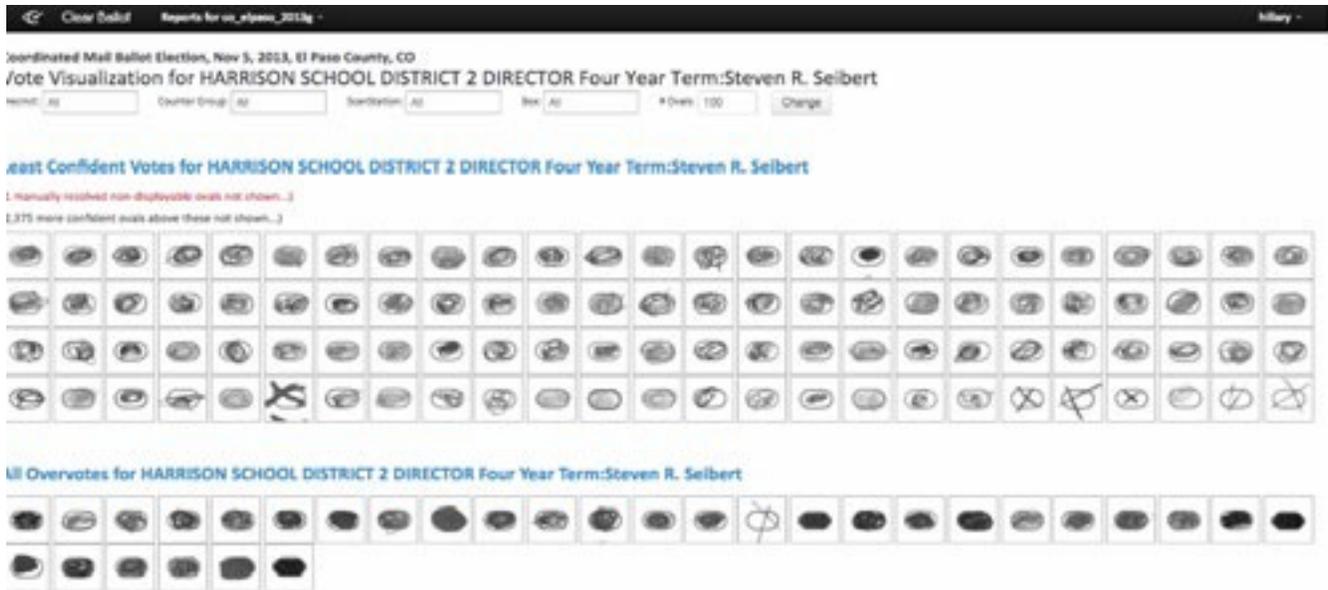
The US Senate contest consisted of six candidates, including a verified write-in candidate. We see that in Leon County, Val Demings won the most votes, a tally that was consistent across both vote tabulators. Each vote tabulator recorded the same number of ovals, indicating it read the same number of ballots. However, in terms of votes recorded, Val Demings received one less vote and Marco Rubio received one more vote in the ClearAudit tabulation system than in the official voting system. Because they recorded the same number of ballots it is not the case here that the ClearAudit tabulator found an additional vote, but instead the two tabulators appear to have counted one or more ballots differently. There is no way to know exactly which ballot or ballots those might be, though ClearAudit would suggest starting with the ballots with the lowest confidence ovals in these races (see more below under Finding Discrepant Ballots).

In the general election, unlike in the primary, we see several examples where it seems that a ballot was counted differently because it does not appear as if any new votes are found. The same number of ovals are counted in each contest, but they are allocated to different candidates. Besides the Senate contest, this also occurs in the US Congressional District 2, where there is one more vote for one candidate and one less vote for the other candidate. We see the same type of discrepancy for the Chief Financial Officer, state Senate District 3, state House District 3, on the retention question for judges (Canady, Couriel, Labarga, Polston, Long), and Mayor.

We also see that District Court Judge Bilbrey had two fewer votes and Judge Winokur has one fewer vote in the ClearAudit system than in Dominion. We also see one fewer vote counted in the County Commission District 5 and School Board District 3 contests in the audit count than in the Dominion count. Among the amendments, we find several instances where ClearAudit was able to capture one more vote.

Table 2 in each dashboard (the table at the bottom of the screen) displays the ballots and provides links to the ballot images. Ballots are displayed one at a time by clicking on the “Click here!” link under the column labeled “Ballot link.” As described earlier, the Ballot ID is the batch number followed by the ballot number in that batch. Ballots can be sorted using the filters at the top and the filters in the middle of the screen, including oval confidence, ballot mode or voting method (e.g. early, Election Day, or vote-by mail), and vote type, which is the oval disposition codes (e.g. vote for candidate, vote for other candidate, undervote, overvote). ClearAudit produces a confidence ranking for each oval on the ballot and in Figure 3 we show an example screen shot of the ClearAudit oval rankings. In the ClearAudit software, a user can hover over an oval and the ballot contest will pop up for review. The photo shows the 100 least confident ovals. The last oval is the least confident and the first oval is the 100th least confident oval. We captured the confidence rankings and included it in our data set. A user can click to filter the 20 least confident dispositions of any type or select all.

Figure 3. Image from ClearAudit Software Ranking Least Confident Votes



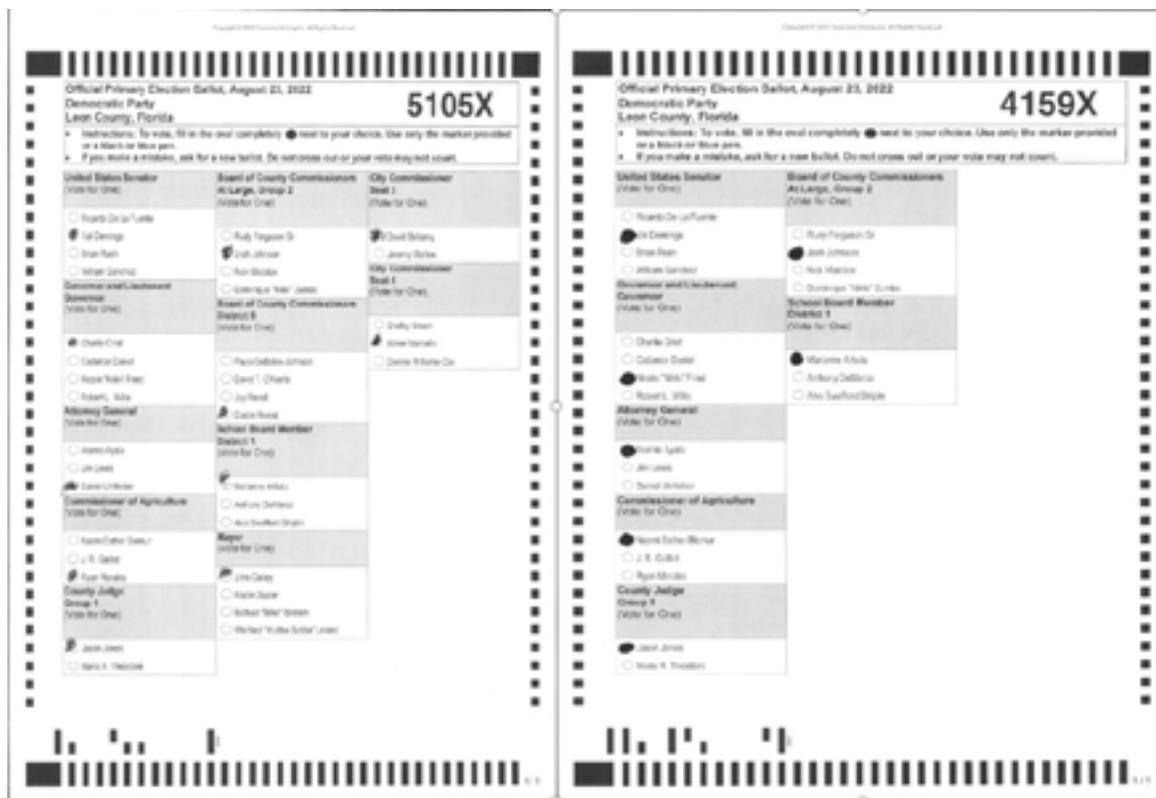
Finding Discrepant Ballots

We do not, of course, have the ClearAudit software, only the dataset it produced. Therefore, we must sort and search through ballot data and images to attempt to identify any discrepancy. The audit data in our dashboard contains this feature because it allows the user to identify the 20 least confident ovals by vote type (vote for a candidate, vote for an alternative candidate, undervotes, and overvotes). We limited this to 20 because after the first few ovals it becomes essentially impossible for the human eye to discern differences in oval quality. However, a user can click on the vote type column and obtain the full rankings from the first to the last ballot available.

In Figure 4 below, we show an image of the School Board District 1 contest from the primary election. This was a three-way race among candidates Marianne Arbula, Anthony DeMarco, and Alva Swafford Striplin. The image shows that the ClearAudit system found one more vote for Arbula than did Dominion. By comparing the official precinct results to the results of the ClearAudit system, we were able to identify in which precinct the ClearAudit system counted one more vote, and then used confidence data in that precinct to help identify the likely ballot that was not counted by Dominion but was counted by ClearAudit.

The ballot on the left represents the least confident oval in this contest in the precinct in which the discrepancy exists, while the ballot to the right represents the most confident oval. The ballot to the right has dark, neatly colored-in ovals, while the ovals on the left are somewhat poorly filled in, with the School Board race showing most of the coloring on the outside of the oval. In this case, the ballot on the left was not counted by the Dominion machine but the ClearAudit tabulator, which looks at the larger “contest zone” surrounding an oval, returned a vote for Arbula.

Figure 4. Example Ballots showing Oval Confidence



Using the Map for the General Election

One last feature we included in our dashboard is a precinct map for Leon County. Because of ballot privacy issues, we had to exclude an interactive map for ballot selection by precinct for the primary election, but the map does work for most precincts in the general election. We had to combine 21 smaller precincts into groups of two or three to protect voter privacy in the general election and comply with the 30-person rule.

Ballot Image Data Facts

On our website, we included several interesting ballot buckets for voters to review. These include ballots with no votes, ballots with signatures, other interesting ballots with marks, and ballots with overvotes for the gubernatorial primary, which has poor instructions — “Governor and Lieutenant Governor (Vote for One)” — creating confusion for a small set of voters.

In addition, our close look at the primary provided some interesting descriptive facts. For example, the cast vote record allowed us to explore how voters interacted with their ballot. Table 1 shows that around 81% voted in every contest on their ballots, and this was generally true for each party ballot and regardless of the number of the contests on the ballot. Of course, voters who voted only in the nonpartisan races had the fewest number of contests to consider, and this may be why they had a

slightly higher completion rate, on average, than either Democrats or Republicans.

Table 1. % Ballots Completed by Party, Leon County 2022 Primary

	Democrat	NPA	Republican	Total
100% Complete Ballot N	79.7 32,849	85.9 5,452	80.8 15,585	80.6 53,886
< 100% Complete Ballot N	20.3 8,377	14.1 896	19.2 3,712	19.4 12,985
Total	41,226	6,348	19,297	66,871

Table 2 shows that 49 voters (.07%) submitted a blank ballot. These voters presumably wanted the voter credit for turnout but had no candidate preferences. We also found that 644 (almost 1% of voters) came to the polls to vote for only one candidate. We have placed the ballots in which there are no votes under the category of interesting ballots for user examination.

Table 2. % Ballots Voting in No or Only 1 Contest, Leon County 2022 Primary

	Democrat	NPA	Republican	Total
Voted in 0 Contests N	0.03 13	0.2 13	0.12 23	0.07 49
Voted in 1 Contests N	0.49 200	2.17 138	1.59 306	0.96 644

Table 3 breaks down the undervotes and overvotes by office in the primary election. We see that the Democratic gubernatorial contest had the largest number of overvotes. This is likely due to the confusing instruction at the top of the ballot that says, "Governor and Lieutenant Governor Vote for One." Under the "Interesting Ballots" menu item we have placed a tab that allows a user to examine these overvoted ballots.

The data also show that the office of County Judge, which was a nonpartisan countywide contest between two candidates, has the largest number (7,307) of undervotes. Obviously, voters have little interest in this ballot item.

However, the largest percentage (12.5%) of undervotes was for the Commissioner of Agriculture contest on the Democratic ballot, which was somewhat surprising given that this is a statewide office. However, the race ended somewhat in scandal, which may explain the high undervote rate. The female candidate in the contest was Naomi Esther Blemur, a Black entrepreneur and a member of the Miami-Dade Democratic Executive Committee.¹¹ Jacques Gaillot had run for office before, but failed in both his election bids. Ryan

¹¹See Lemongello, Steve, Jeffrey Schweers, Leslie Postal, Stephen Hudak, Martin E. Comas, Ryan Gillespie, Monivette Cordeiro, Annie Martin and Agigail Hasbrook. 2022. Meet the Democratic Candidates Running in the August 23 Primary," Orlando Sentinel, August 14, 2022, available at: <https://www.orlandosentinel.com/2022/08/27/meet-the-candidates-in-the-florida-primary-election-tuesday/>

Morales was a business consultant and hemp farmer. Little was known about any of these candidates, but during the election social media posts from Blemur revealed that she was deeply religious and believed abortion was a sin; she also made what some considered to be homophobic posts.¹² This led to several prominent Democrats rescinding their endorsements, including Miami Mayor Daniella Levine Cava and state Senators Annette Taddeo and Shevrin Jones. Still, Blemur — the only female candidate — won the primary. Democratic primary voters typically would be unlikely to support a pro-life or openly homophobic candidate, so perhaps this last-minute confusion led many voters to simply skip this contest.

Table 3. Frequency of Overvotes and Undervotes in the Primary

Contest	Overvotes	Undervotes	Voter N	% Overvotes	% Undervotes
Senate	2	1492	41,226	0.005	3.62
Governor	20	472	41,226	0.05	1.14
Attorney General (Dem)	2	3,397	41,226	0.005	8.24
Comm. of Agriculture (GOP)	1	1,035	19,297	0.005	5.36
Comm. of Agriculture (Dem)	5	5,166	41,226	0.012	12.53
State 8 th House District(Dem)	1	707	14,808	0.007	4.77
County Judge	3	7,307	66,871	0.004	10.92
County Commisioner At Large	14	4,863	66,871	0.021	7.27
County Commisioner District 1	0	220	7,573	0.000	2.91
County Commisioner District 2	4	209	5,772	0.069	3.62
County Commisioner District 3	0	906	15,363	0.000	5.90
County Commisioner District 5	2	1,049	18,684	0.011	5.61
School Board District 1	1	1,596	19,600	0.005	8.14
School Board District 4	2	911	17,075	0.012	5.34
Mayor	1	782	39,026	0.003	2.00
County Commisioner Seat 3	3	1,390	39,026	0.008	3.56
County Commisioner Seat 5	3	2,636	39,026	0.008	6.75

¹²See Schweers, Jeffrey. 2022. "GOP runs the table with Cabinet races," Transparency USA, available at: <https://www.transparencysusa.org/fl/race/attorney-general-of-florida.sday/>

We are continuing to examine the data, especially for the general election (which we received more recently) for additional information.

Project Obstacles

As part of our dashboard development, we encountered a number of challenges. One was the size of the data sets we were building, especially the ballot image data – these items created problems with speed and, consequently, useability. We discussed this issue in our *Designing the Dashboard* section above.

As part of our dashboard, we also present copies of the ballot images so that they are available for public review. However, before we could present the ballot images, we had to resolve a number of different ballot obstacles here as well.

Ballot Orientation

The first problem we encountered was ballot orientation. There is no single correct orientation when county staff insert ballots into the audit tabulator. The tabulator can read ballots regardless of the orientation. They can be inserted with either the front or back of the ballot facing up, and the top or bottom of the ballot can be inserted first. That creates four possible ballot orientations. These include front-top, which is the orientation we want presented in our dashboard, and the other three ways: upside down (back-top or back-bottom) or flipped (front-bottom). Therefore, to make the ballots presentable to the public, we had to identify the ballot orientation and then flip, turn, or flip and turn a great many ballot images so they would be displayed in the preferred orientation.

Identifying and correcting the ballot orientation was achieved using Python's OpenCV image library. In essence, the bottom-left corner of the ballot was isolated and analyzed. If this region of the ballot contained any shape that had an area less than 5000 pixels, the ballot was considered upside down (bottom ballot), and therefore was flipped. If the bottom-left corner of the ballot contained a shape greater than 5000 pixels, the ballot was considered right-side up (top ballot). On a top ballot, the bottom-left and top-right corners each contained two similar boxes. However, the box on the top-right corner is significantly smaller in area than the box on the bottom-left corner. For a ballot that is upside down, this significantly smaller box would appear on the bottom-left rather than the top-right. This method identified if the smaller box was on the bottom-left or top-right of the ballot. Based on this, the ballot was reorientated.

The 5000-pixel heuristic was determined by surveying 1000 ballots, which included all three ballot types – “ED,” “EV,” and “VBM” – and calculating the max, min, avg, and median area in the Region of Interest (ROI) for all top ballots (right-side up) and bottom ballots (upside-down). Table 4 shows the statistics for this survey.

If jurisdictions do not have access to this technology, the low-tech solution would be to orient the ballots in the correct direction as they are processed into the audit tabulator.

Table 4. Ballot Survey Statistics

Top-Max	7519.5
Top-Min	6802.0
Top-Avg	7290.778538812785
Top-Median	7293.5
Bot-Max	3182.5
Bot-Min	0.0
Bot-Avg	723.1896551724138
Bot-Median	664.0

Ballot Privacy

Today the secret ballot is considered a fundamental aspect of a democratic election system, meant to protect voters from fear of intimidation, and is essential to the value of election integrity.¹³ Ballot secrecy is guaranteed across the country in state constitutions. Ballot privacy, however, is somewhat at risk when we place election outcome data online with ballot images. For example, in many jurisdictions aggregated vote totals are available from the county canvass by precinct or by precinct and vote mode, and these data can often be downloaded from county and state election websites. In primary elections, this might also be broken down further by party ballots. In the 2022 Leon County primary election, ballots were broken down by vote mode, party ballot, and precinct, potentially creating small cells of voters within combined categories.

To protect voter privacy, states often have arbitrary reporting laws that specify when precinct information must be hidden. Florida has a 30-voter threshold for reporting precinct data by type of ballot or counting group (counting groups could mean Democratic, Republican or Independent ballots during a primary, vote mode, or the interaction of both).¹⁴ If there are less than 30 voters in any of these conditions then the precinct totals must be reported as zero and are only counted in the aggregate totals.

We followed this rule, which resulted in a number of problems for placing the ballot images online, especially for the primary. The problem stems from the fact that the ballot image contains information about both its precinct and vote mode and, in the case of the primary election, party. First, there is the printed tag in the upper right-hand corner that indicates the precinct (see Figure 4 for 2 ballot examples), which is the four-digit number that also indicates the party by X (Democrat), Y (Republican), or Z (nonpartisan). Second, there are tick marks at the bottom of the ballot.

¹³Thomas M. Franck, 1992, "The Emerging Right to Democratic Governance," *The American Journal of International Law*, pp 46-91.

¹⁴See State Statute 98.09812a available at: http://www.leg.state.fl.us/statutes/index.cfm?App_mode=Display_Statute&URL=0000-0099/0098/0098.html

These also provide information on party, precinct, and language. Finally, combinations of ballot contests as they appear on the ballot can also be used to identify the voting precinct. While Florida does not include information about ballot language preferences in the voter registration file, some states do – which, given the very few number of non-English ballots here, would have triggered a privacy issue in such venues.

The ballot ID information led to substantial problems for maintaining voter privacy, particularly in the primary because we had to consider not only what data we were providing in our dashboard and ballot images, but also what data from the state and local canvass could be used to triangulate enough information to identify a voter, his or her ballot, and consequently his or her vote choices. Leon County is a predominantly Democratic County, and as such there were few Republican or nonparty voters. When ballots could be categorized by mode, precinct, and party, we found that more than half of precincts did not meet the 30-ballot statutory requirement.

To resolve these privacy issues for the primary election and comply with state law, we removed precinct and party information in the upper right-hand corner of the ballot the tick marks at the bottom of the ballot and disabled the Leon County precinct map. Finally, we had to withhold 43 ballots because the unique contest combinations on these ballots could reveal individual vote choices and violate voter privacy and state law.

In the general election, where there were more voters and no party-specific ballots to distinguish sets of voters, the problems were not as severe. Of 135 precincts, 21 had 30 or fewer voters for at least one vote mode. One additional precinct had more than 30 voters, but every Early-Voting ballot was for Crist and every Vote-by-Mail ballot was for Demings. We needed to combine another eight precincts with these small precincts for the redaction to work. This led us to 30 precincts that required us to remove the precinct numbers and barcodes. A total of 4,422 ballots out of 118,216 ballots were affected.

In addition, we had to be concerned with voters self-identifying by printing or signing their name on their ballot. Therefore, we had to develop methods to identify and redact personal identifiers.

Ballot Redaction Methodology

To maintain voter privacy, the following items had to be redacted from every primary ballot. For the front side of ballots this included:

1. Precinct and party information in the upper right-hand corner.
2. Tick marks at the bottom of the ballot.
3. For “EV” ballots, the “Checkin EV” tag.

For the back side of ballots this included:

1. Precinct and party information on the center of the ballot;
2. Tick marks at the bottom of the ballot.
3. For “EV” ballots, the “EV” tag.
4. For “VBM” ballots, the QR code and precinct number next to it.

Ballot Redaction Precinct Information

To redact precinct and party information in the upper-right hand corner for the primary and precinct information for a relatively small number of ballots in the general election, a region at the top right of the ballot region of interest (ROI) was cropped for further analysis. This further analysis included:

1. Identifying shapes and contours in the ROI;
2. Getting the contour with the greatest area, calling it best contour;
3. Extracting the top-left point of the best contour (x, y).

With the x and y values extracted from the contour with the largest area, we determined that there was a fixed distance in pixels from this contour to the precinct information that had to be redacted. We then calculated the location of the precinct information by adding the x and y values of the contour start point determined earlier with a fixed distance. This returned the starting point of the area that needed to be redacted. We then did a similar process to determine where the end of the redaction location was. A similar method was employed to redact the bars on the bottom-left corner of the ballot and the “Checkin EV” tag on the bottom-right side of the ballot.

This method needed to be employed because there were relatively large variations between ballots in the y-position where the precinct information resides, so the redaction location had to be dynamically calculated. For example, in some cases the top left point where the precinct information resides was at (1240, 210) and on other ballots it occurred at (1260, 180). Therefore, it was crucial to dynamically calculate redaction position to keep other aspects of the ballot intact.

For the back side of ballots, the process for redacting the bars and the “Checkin EV” tag was identical to that of the front side. However, the precinct information was displayed in a different region than on the front side. Fortunately, it was located in a relatively unimportant region of the ballot. Because of this, we were able to redact a large fixed region of the ballot that covered anywhere the precinct information might occur. This was also true for the QR code on the “VBM” ballots.

Running redaction operations on all ballots

Because we were working with a large number of ballots, speed was an issue. Fortunately, this task was a perfect candidate for Python's threading library, specifically because of the large number of files to write. Threading was employed by equally splitting up the list of ballots to redact into 11 different sets. Eleven threads were started and each was fed one of these sets. With no threading, running through a sample set of approximately 700 ballots took around 26 seconds. With threading, it took less than 3.5 seconds.

Converting back to PDF

The overall process also required that these images be converted back to PDFs. This was achieved using python's Pillow library.

Signature Identification

From the primary election, each ballot was manually examined by human eyes for voter names or signatures. However, in the general election, due to the sheer number of ballots, manual signature identification was not a viable option.

We used a machine learning model to autonomously identify signatures on ballots. The best way to do this was to use a pretrained model which already had many of the necessary weights and biases to properly label images. Using a pretrained model lowered the total required training time and reduced the amount of data required to train the model.

We used the You Only Look Once version 8 (YOLOv8) medium-sized model. To train the model, we collected signature and other voter writing data from ballots identified in the primaries. We supplemented these data with two signature datasets for a total of 138 signatures.

We randomly selected 1,500 general election ballots for a starter data set. We manually looked at these ballots to ensure that there was no voter writing already on them. We set aside 20% of the ballot dataset to have no signatures on them. This was done to reduce the number of false positives produced by the model. The signatures were then placed semi-randomly on each ballot and augmented to create more variation between signatures.

The ballots, with signatures, were then split into two different data sets: a training set (80% of total ballots) and a validation set (20% of total ballots).

The Machine Learning Model

Three main iterations of the model contributed to the final set of identified ballots. Model 1's dataset was largely created by following the procedure outlined in Data Creation. Model 1 and 2 were trained for 75 epochs, then stopped. YOLOv8 was then told to treat the last checkpoint as a pretrained model, which resets various model parameters – most notably, learning rate. It was observed that doing this in earlier trial runs of models produced much better results. Model iterations 2 and 3 were trained off additional true and false positives obtained from running model 1 through the entire general election set and categorizing its labels.

A comparable model was developed for the primary election ballots to validate the methodology's results, and performed very well.

Project Impact

This is an important pilot project that placed post-election audit data and ballot images online for public review. Our project provides important insights and valuable information for election administrators who are considering displaying post-election audit data, election results, and ballot images.

Along the way, we faced a large number of obstacles that election administrators would likely have to deal with, especially if they decided to put ballot images online. The size of the data sets we created were large and we had to find software to manage them in order to establish a good user interface. Problems around voter privacy especially in the primary, were ubiquitous. Data triangulation from the state and county canvass, the voter file, and the ballot data created potential privacy concerns. Ballot tick marks, contest arrangements on the ballots (i.e. ballot styles), and ballot IDs are all distinct ways to identify ballots, providing potential avenues to determine voter choices.

We also found that a small but not insignificant number of voters printed or signed their names directly on the ballot, also creating voter privacy problems. Given the possibility of vote-buying and laws covering voter privacy and ballot secrecy, our project identifies the scope of the problem and outlines possible solutions, including technological solutions on the back end and policy solutions on the front end.

In addition, we gained valuable pieces of descriptive information during the project about how voters interact with their ballots and how they use their ballots as an expressive tool. On our website, we included several interesting ballot buckets for voters to review. These include ballots with no votes, ballots with signatures, other interesting ballots with marks, and ballots with overvotes for the gubernatorial primary, which had poor directions creating confusion for a small set of voters.

Appendix A: Shiny Code

Primary Election Dashboard Code

```
# Load necessary Libraries and read data
library(tidyverse)
library(DT)
library(sf)
library(shiny)
library(shinydashboard)
library(plotly)
library(shinyWidgets)
library(htmltools)

table_1 <- read_csv('Comparison of votes cast_Clean - Copy sheet 1.csv')

table_1 %>%
  mutate(`Difference_Choice Recorded` = `Difference_Choice Recorded`*-1,
         `Difference_Ballots Counted` = `Difference_Ballots Counted`*-1) %>%
  select(Contest, Choice, Party, `Dominion_Ballots Counted`, `ClearBallot_Ballots Counted`,
         `Difference_Ballots Counted`, `Dominion_Choice Recorded`, `ClearBallot_Choice Recorde
d`,
         `Difference_Choice Recorded`, `Overvoted WithVote for this Choice`,
         `Undervoted WithoutVote for this Choice`) %>%
  rename('Ovals \n Counted \n Dominion' = `Dominion_Ballots Counted`,
        'Audit Ovals \n Counted \n ClearBallot' = `ClearBallot_Ballots Counted`,
        'Difference in \n Ovals \n Counted' = `Difference_Ballots Counted`,
        'Votes \n Recorded \n Dominion' = `Dominion_Choice Recorded`,
        'Audit Votes \n Recorded \n ClearBallot' = `ClearBallot_Choice Recorded`,
        'Difference in \n Votes Recorded' = `Difference_Choice Recorded`,
        'Over \n Vote' = `Overvoted WithVote for this Choice`,
        'Under \n Vote' = `Undervoted WithoutVote for this Choice`)
%>%
  mutate(Party = ifelse(is.na(Party)==TRUE, 'NPR', Party)) -> table_1

table_2 <- read_csv('table_2_dash_all.csv')

table_2 %>%
  rename('Ballot ID' = BallotID,
        'Choice' = cand_last,
        'Vote Type' = vote_type,
        'Ballot Link' = view_ballot,
        'Oval Confidence Rank' = oval_confidence_rank,
        'Voting Method' = vote_mode,
        'Precinct' = PrecinctID) %>%
  mutate(Party = ifelse(is.na(Party) == TRUE, 'NPR', Party)) -> table_2

map <- st_read('Election_Precincts_-_Leon_County', layer = 'Election_Precincts_-_Leon_County')

Sys.setenv('MAPBOX_TOKEN' = 'pk.eyJ1IjoieYXVzdGluLWN1dGx1ciIsImEiOiJjbGt0enpwZG4wMW5iM3NsaDAXNjBoMm5nIn0.oBeanbdPK0aurRNUJG7jIg')

ui <- fluidPage(
  tags$head(
    tags$style(HTML(
```

```

".content-wrapper { background-color: white; margin: 0}",
".container-fluid {font-size: 12px; padding: 0}'",
".custom-select-wrapper {margin:1px; background-color: #dfedeb; border: 1px solid black;
}]",
".custom-select {margin:1px; width: 100%; max-width: 350px; background-color: #dfedeb;}"
,
".stripe tbody tr:nth-child(even) { background-color: #dfedeb; }]",
'.shiny-input-checkboxgroup {margin:1px; background-color: #dfedeb; border: 1px solid black; justify-content: center}',
'.shiny-input-checkboxgroup.shiny-input-container-inline label~.shiny-options-group, .shiny-input-radiogroup.shiny-input-container-inline label~.shiny-options-group{margin-left: 1px}',
',
'.js-irs-0 {background: #dfedeb; border: 1px solid black; }',
'.js-irs-0 .irs-bar-edge {background: #dfedeb; border: 1px solid black; }',
'.box {margin:1px; border: 1px solid white; -webkit-box-shadow: none; -moz-box-shadow: none; box-shadow: none;}',
'.box-header { background-color: #f9f9f9; }',
'.selectize-input {margin:1px; width: 95%; }',
'.selectize-control { padding-left: 5%; }',
'.shiny-input-radiogroup {margin: 1px; background-color: #dfedeb; border: 1px solid black; justify-content: center}',
".plotly { max-width: 100%;margin:1px }]",
".content-body {margin:1px; padding: 0; }]",
".dataTable-wrapper { margin: 1px;}",
".row { margin-bottom: 0; }]",
'.col-special {margin-right:-15px}',
'.box-body {padding: 1px}',
'.total-table{margin:0; justify-content: center}',
'.link{color: #782F40; font-size: 25px; color: #782F40;font-weight: bold}',
'.link:hover{color: #00524d}'
)),
tags$script(
"$(document).ready(function(){
var mapPlots = document.getElementsByClassName('plotly');
if (mapPlots.length > 0) {
for (var i = 0; i < mapPlots.length; i++) {
mapPlots[i].addEventListener('plotly_relayout', function(eventdata){
Plotly.relayout(this, {autosize: true});
});
}
}
});"
)
),
dashboardPage(
dashboardHeader(disable = TRUE),
dashboardSidebar(disable = TRUE),
dashboardBody(
fluidRow(box(width = 6, solidHeader = TRUE,
column(width = 12,
div(HTML('<p> <a class = "link"
href="https://fsu.qualtrics.com/jfe/form/SV_e3xJRgF6JVDGluu">Click here to tell us ab
out your experience!</a></p>'))))),
fluidRow(
box(width = 4, solidHeader = TRUE,

```

```

        column(width = 12,
              div(class = "filters",
                  div(class = "filter-item custom-select-wrapper",
                      selectInput("Contest", label = "Contest",
                                  choices = c("All", unique(table_1$Contest)))
                  )
              )
        ),
        box(width = 4, solidHeader = TRUE,
            column(width = 12,
                  div(class = "filters",
                      div(class = "filter-item custom-select-wrapper",
                          selectInput("Party", label = "Party of Contest",
                                      choices = c("All",
                                                  'Republican Party (REP)',
                                                  'Democratic Party (DEM)',
                                                  'Nonpartisan Race (NPR)'))
                      )
            )
        ),
        box(width = 4, solidHeader = TRUE,
            column(width = 12,
                  div(class = "filters",
                      div(class = "filter-item custom-select-wrapper",
                          selectInput("Candidate", label = "Choice",
                                      choices = c("All", unique(table_1$Choice)))
                      )
            )
        ),
        fluidRow(
            box(width = 12,
                div(
                    DTOutput("my_table")
                )
            )
        ),
        fluidRow(
            box(width = 8, solidHeader = TRUE,
                fluidRow(
                    column(width = 4, class = 'col-special',
                        div(radioButtons("OvalCat", label = "Oval Confidence Rank",
                                        choices = c("All", '1-20'), inline = TRUE, selected = 'Al
1'))),
                    column(width = 8,
                        div(
                            checkboxGroupInput("VotingMethod", label = "Voting Method",
                                                selected = c('Early Vote', 'Vote by Mail',
                                                            'Election Day Vote'),
                                                choices = c(unique(table_2`Voting Method`)),
                                                inline = TRUE)),
                            div(

```



```

if (input$Candidate != 'All' && !is.null(input$Candidate)){
  filtered <- filter(filtered, Choice == input$Candidate)
}

filtered

})

sketch <- htmltools::tags$table(
  tableHeader(names(table_1)),
  tableFooter(rep("", ncol(table_1)))
)

output$my_table <- renderDT({
  datatable(filtered_table(), filter = 'none',
    escape = FALSE,
    options = list(scrollY = 200, dom = 'Bfrtip', lengthMenu = list(-1),
      searching = FALSE, info = FALSE, paging = FALSE,
      footerCallback = JS(
        "function(tfoot, data, start, end, display) {"
        "  var api = this.api();"
        "  for(var i = 2; i < data[0].length - 2; i++) {"
        "    var footerContent;"
        "    if (i === 2) {"
        "      footerContent = '<span style=\"font-size: 14px;\">Absolute Total:</span>';"
        "    } else {"
        "      footerContent = api.column(i, {page:'all'}).data().reduce(function(a,b){",
        "        if(isNaN(a)) return ' '; return Math.abs(a)+Math.abs(b);",
        "      });"
        "    }"
        "    $(api.column(i).footer()).html(footerContent);"
        "  }"
        "}"
      )),
    rownames = FALSE,
    selection = 'none',
    class = 'cell-border stripe',
    fillContainer = TRUE,
    container = sketch
  )
})

filtered_table_2 <- reactive({
  filtered <- table_2

  if (!is.null(input$VotingMethod) && length(input$VotingMethod) > 0) {
    filtered <- filter(filtered, `Voting Method` %in% input$VotingMethod)
  }

  if (is.null(input$VotingMethod) && length(input$VotingMethod) == 0){
    filtered <- filtered[NULL,]
  }
})

```

```

}

if (input$Contest != "All" && !is.null(input$Contest)) {
  filtered <- filter(filtered, Contest == input$Contest)
}

if (!is.null(input$VoteType) && length(input$VoteType) > 0) {
  filtered <- filter(filtered, `Vote Type` %in% input$VoteType)
}

if (is.null(input$VoteType) && length(input$VoteType) == 0){
  filtered <- filtered[NULL,]
}

if (input$OvalCat != "All" && input$OvalCat == '1-20') {
  filtered <- filter(filtered, `Oval Confidence Rank` <= 20)
}

if (input$Party != "All" && input$Party == 'Republican Party (REP)') {
  filtered <- filter(filtered, Party == 'REP')
}

if (input$Party != "All" && input$Party == 'Democratic Party (DEM)') {
  filtered <- filter(filtered, Party == 'DEM')
}

if (input$Party != "All" && input$Party == 'Nonpartisan Race (NPR)') {
  filtered <- filter(filtered, Party == 'NPR')
}

if (input$Candidate != 'All' && !is.null(input$Candidate) && input$Candidate != 'Ricardo D
e La Fuente'){
  filtered <- filter(filtered, Choice == word(input$Candidate, -1))
}

if (input$Candidate != 'All' && !is.null(input$Candidate) && input$Candidate == 'Ricardo D
e La Fuente'){
  filtered <- filter(filtered, Choice == 'De La Fuente')
}

select(filtered, -oval_cat, -Party, -Precinct)
})

output$my_table_2 <- renderDT({
  datatable(filtered_table_2(), filter = 'none',
    options = list(rowsGroups = list(0),
      scrolly = 250, scrollCollapse = TRUE, dom = 'Bfrtip', lengthMenu
= list(10000, -1),
      searching = FALSE, info = FALSE, serverSide = TRUE, deferRender =
TRUE, virtualScroll = TRUE),
    rownames = FALSE,
    selection = 'none',
    class = 'cell-border stripe',
    fillContainer = TRUE,
    escape = FALSE,

```

```

        callback = JS('table.page(1).draw(false);')
    ) %>% formatStyle(columns = 'Ballot Link',
                      target = 'row',
                      css = list("display:block; width:100%"))
})

output$map_plot <- renderPlotly({
  filtered_map_plot <- plot_mapbox() %>%
    add_sf(data = map, color = I("#00524d"),
           opacity = 1, hoverinfo = "text", showlegend = FALSE) %>%
    layout(mapbox = list(
      center = list(
        lat = 30.455000,
        lon = -84.253334
      ),
      zoom = 8
    ),
    dragmode = FALSE,
    margin = list(l = 0, r = 0, t = 0, b = 0),
    xaxis = list(showgrid = FALSE, zeroline = FALSE, showline = FALSE, showticklabels = FALSE),
    yaxis = list(showgrid = FALSE, zeroline = FALSE, showline = FALSE, showticklabels = FALSE)
  ) %>%
  config(displayModeBar = FALSE, staticPlot = TRUE)
})
}

# Run the Shiny app
shinyApp(ui, server)

```

General Election Dashboard Code

```

# Load necessary libraries and read data
library(tidyverse)
library(DT)
library(sf)
library(shiny)
library(shinydashboard)
library(plotly)
library(shinyWidgets)
library(htmltools)

table_1 <- read_csv('table_1_gen.csv')

table_1 %>%
  mutate(`Difference_Choice Recorded` = `Difference_Choice Recorded`*-1,
         `Difference_Ballots Counted` = `Difference_Ballots Counted`*-1) %>%
  select(Contest, Choice, `Dominion_Ballots Counted`, `ClearBallot_Ballots Counted`,
         `Difference_Ballots Counted`, `Dominion_Choice Recorded`, `ClearBallot_Choice Recorded`,
         `Difference_Choice Recorded`, `Overvoted WithVote for this Choice`,
         `Undervoted WithoutVote for this Choice`) %>%

```

```

rename('Ovals \n Counted \n Dominion'      = `Dominion_Ballots Counted`,
       'Audit Ovals \n Counted \n ClearBallot' = `ClearBallot_Ballots Counted`,
       'Difference in \n Ovals \n Counted' = `Difference_Ballots Counted`,
       'Votes \n Recorded \n Dominion'      = `Dominion_Choice Recorded`,
       'Audit Votes \n Recorded \n ClearBallot' = `ClearBallot_Choice Recorded`,
       'Difference in \n Votes Recorded'      = `Difference_Choice Recorded`,
       'Over \n Vote'                        = `Overvoted WithVote for this Choice`,
       'Under \n Vote'                      = `Undervoted WithoutVote for this Choice`)
-> table_1

table_2 <- readRDS('table_2_gen_dash_all.rds')

table_2 %>%
  rename('Ballot ID'          = BallotID,
         'Choice'            = cand_last,
         'Vote Type'         = vote_type,
         'Ballot Link'       = view_ballot,
         'Oval Confidence Rank' = oval_confidence_rank,
         'Voting Method'     = vote_mode,
         'Precinct'         = PrecinctID) -> table_2

st_read('Shape files - combined precincts', layer = 'Leon_ShapeFile') %>%
  mutate(PRECINCT = str_replace_all(PRECINCT, '/', ' & '),
         PRECINCT = case_when(nchar(PRECINCT) > 11 ~ str_replace(PRECINCT, ' &', ','),
                              TRUE ~ PRECINCT)) -> map

Sys.setenv('MAPBOX_TOKEN' = 'pk.eyJ1IjoiYXVzdGluLWN1dGx1ciIsImEiOiJjbGt0enpwZG4wMW5iM3NsADAxNjBoMm5nIn0.oBeanbdPK0aurRNUJG7jIg')

ui <- fluidPage(
  tags$div(id = "map"),
  verbatimTextOutput("selected_precinct"),
  tags$head(
    tags$style(HTML(
      ".content-wrapper { background-color: white; margin: 0}",
      ".container-fluid {font-size: 12px; padding: 0}",
      ".custom-select-wrapper {margin:1px; background-color: #dfedeb; border: 1px solid black;
    }",
      ".custom-select {margin:1px; width: 100%; max-width: 350px; background-color: #dfedeb;}"
    ),
    ".stripe tbody tr:nth-child(even) { background-color: #dfedeb; }",
    ".shiny-input-container:not(.shiny-input-container-inline){max-height: 43px}",
    ".shiny-input-checkboxgroup {margin:1px; background-color: #dfedeb; border: 1px solid black; justify-content: center}",
    ".shiny-input-checkboxgroup.shiny-input-container-inline label~.shiny-options-group, .shiny-input-radiogroup.shiny-input-container-inline label~.shiny-options-group{margin-left: 1px}",
    ".js-irs-0 {background: #dfedeb; border: 1px solid black; }",
    ".js-irs-0 .irs-bar-edge {background: #dfedeb; border: 1px solid black; }",
    ".box {margin:1px; border: 1px solid white; -webkit-box-shadow: none; -moz-box-shadow: none; box-shadow: none;}",
    ".box-header { background-color: #f9f9f9; }",
    ".selectize-control { padding-left: 7px; padding-right: 7px; margin-bottom:3px}",
    ".shiny-input-radiogroup {margin: 1px; background-color: #dfedeb; border: 1px solid black; justify-content: center}",
  )
)

```

```

".plotly { max-width: 100%;margin:1px }",
".content-body {margin:1px; padding: 0; }",
".dataTable-wrapper { margin: 1px;}",
".row { margin-bottom: 0; }",
'.col-special {margin-right:-15px}',
'.box-body {padding: 1px}',
'.total-table{margin:0; justify-content: center}',
'.link{color: #782F40; font-size: 25px; color: #782F40;font-weight: bold}',
'.link:hover{color: #00524d}'
)),
tags$script(
  "$(document).ready(function(){
var mapPlots = document.getElementsByClassName('plotly');
if (mapPlots.length > 0) {
  for (var i = 0; i < mapPlots.length; i++) {
    mapPlots[i].addEventListener('plotly_relayout', function(eventdata){
      Plotly.relayout(this, {autosize: true});
    });
  }
}
});"
)
),
dashboardPage(
  dashboardHeader(disable = TRUE),
  dashboardSidebar(disable = TRUE),
  dashboardBody(
    fluidRow(box(width = 6, solidHeader = TRUE,
      column(width = 12,
div(HTML('<p> <a class = "link"
  href="https://fsu.qualtrics.com/jfe/form/SV_b3nqC2VFkT1AGcC">Click here to tell us ab
out your experience!</a></p>')))),
    fluidRow(
      box(width = 4, solidHeader = TRUE,
        column(width = 12,
          div(class = "filters",
            div(class = "filter-item custom-select-wrapper",
              selectInput("Contest", label = "Contest",
                choices = c("All", unique(table_1$Contest)))
            )
          )
        )
      ),
      box(width = 4, solidHeader = TRUE,
        column(width = 12,
          div(style = "display: flex; justify-content: center; margin-top: 15px;",
            img(src = 'collins_logo.png', height = '55px')
          )
        )
      ),
      box(width = 4, solidHeader = TRUE,
        column(width = 12,
          div(class = "filters",
            div(class = "filter-item custom-select-wrapper",
              selectInput("Candidate", label = "Choice",

```

```

                                choices = c("All", unique(table_1$Choice)))
                            )
                        )
                    )
                ),
            fluidRow(
                box(width = 12,
                    div(
                        DTOutput("my_table")
                    )
                )
            ),
            fluidRow(
                box(width = 8, solidHeader = TRUE,
                    column(width = 12,
                        fluidRow(
                            column(width = 4, class = 'col-special',
                                div(radioButtons("OvalCat", label = "Oval Confidence Rank",
                                    choices = c("All", '1-20'), inline = TRUE, selected = 'A
11')),
                                div(
                                    checkboxGroupInput("VotingMethod", label = "Voting Method",
                                        selected = c('Early Vote', 'Vote by Mail',
                                            'Election Day Vote'),
                                        choices = c(unique(table_2`Voting Method`)),
                                        inline = TRUE))
                                ),
                            column(width = 8,
                                div(class = "filters",
                                    div(class = "filter-item custom-select-wrapper",
                                        selectInput("Precinct", label = "Precinct",
                                            choices = c("All", unique(table_2$Precinct)))
                                    )
                                )
                            ),
                            div(
                                checkboxGroupInput("VoteType", label = "Vote Type",
                                    choices = c('Voted for Choice',
                                        'Voted for Other Choice',
                                        'Overvote', 'Undervote'),
                                    selected = c('Voted for Choice'),
                                    inline = TRUE))
                            )
                        )
                    ),
                div(
                    DTOutput("my_table_2")
                ),
                column(width = 4,
                    box(width = 12, solidHeader = TRUE,
                        h3("Leon County Precincts", style = "text-align: center; font-size: 24px; font-weight: bold; margin: 0;"),
                        div(style = "display: flex; justify-content: center;",
                            div(style = "width: 100%; height: 100%; ",
                                plotlyOutput("map_plot")
                            )
                        )
                    )
                )
            )
        )
    )

```



```

    )),
    rownames = FALSE,
    selection = 'none',
    class = 'cell-border stripe',
    fillContainer = TRUE,
    container = sketch
  )
})

filtered_table_2 <- reactive({
  filtered <- table_2

  if (!is.null(input$VotingMethod) && length(input$VotingMethod) > 0) {
    filtered <- filter(filtered, `Voting Method` %in% input$VotingMethod)
  }

  if (is.null(input$VotingMethod) && length(input$VotingMethod) == 0){
    filtered <- filtered[NULL,]
  }

  if (input$Contest != "All" && !is.null(input$Contest)) {
    filtered <- filter(filtered, Contest == input$Contest)
  }

  if (!is.null(input$VoteType) && length(input$VoteType) > 0) {
    filtered <- filter(filtered, `Vote Type` %in% input$VoteType)
  }

  if (is.null(input$VoteType) && length(input$VoteType) == 0){
    filtered <- filtered[NULL,]
  }

  if (input$OvalCat != "All" && input$OvalCat == '1-20') {
    filtered <- filter(filtered, `Oval Confidence Rank` <= 20)
  }

  if (input$Precinct != 'All' && !is.null(input$Precinct)){
    filtered <- filter(filtered, Precinct == input$Precinct)
  }

  if (input$Candidate != 'All' && !is.null(input$Candidate)){
    filtered <- filter(filtered, Choice == word(input$Candidate, -1))
  }

  select(filtered, -oval_cat)
})

output$my_table_2 <- renderDT({
  datatable(filtered_table_2(), filter = 'none',
    options = list(rowsGroups = list(0),
      scrolly = 175, scrollCollapse = TRUE, dom = 'Bfrtip', lengthMenu
= list(5000, -1),
      searching = FALSE, info = FALSE, serverSide = TRUE, deferRender =
TRUE, virtualScroll = TRUE),

```

```

        rownames = FALSE,
        selection = 'none',
        class = 'cell-border stripe',
        fillContainer = TRUE,
        escape = FALSE,
        callback = JS('table.page(1).draw(false);')
    ) %>% formatStyle(columns = 'Ballot Link',
                     target = 'row',
                     css = list("display:block; width:100%"))
})

filtered_map_plot <- reactive({

  my_palette <- c("#D3D3D3", "#00524d")

  if (input$Precinct != 'All' && !is.null(input$Precinct)){
    selected_precinct <- input$Precinct
    map$color <- ifelse(map$PRECINCT == selected_precinct, I("#D3D3D3"), I("#00524d"))

    plot_mapbox(data = map, key = ~PRECINCT) %>%
      add_sf(data = map, color = ~color, colors = my_palette,
             opacity = 1, showlegend = FALSE, hoverinfo = 'text',
             hovertext = ~PRECINCT) %>%
      layout(mapbox = list(
        center = list(
          lat = 30.455000,
          lon = -84.253334
        ),
        zoom = 8.5
      ),
      margin = list(l = 0, r = 0, t = 0, b = 0)
    ) %>%
      config(displayModeBar = FALSE, scrollZoom = FALSE) %>%
      event_register('plotly_click')

  } else {
    map$color <- I("#00524d")

    plot_mapbox(data = map, key = ~PRECINCT) %>%
      add_sf(data = map, color = ~I(color), colors = my_palette,
             opacity = 1, showlegend = FALSE, hoverinfo = 'text',
             hovertext = ~PRECINCT) %>%
      layout(mapbox = list(
        center = list(
          lat = 30.455000,
          lon = -84.253334
        ),
        zoom = 8.5
      ),
      margin = list(l = 0, r = 0, t = 0, b = 0)
    ) %>%
      config(displayModeBar = FALSE, scrollZoom = FALSE) %>%
      event_register('plotly_click')
  }
}

```

```
})  
output$map_plot <- renderPlotly({  
  filtered_map_plot()  
})  
  
observeEvent(event_data("plotly_click"), {  
  event <- event_data("plotly_click")  
  if (!is.null(event)) {  
    clicked_precinct <- event$key  
    updateSelectInput(session, "Precinct", selected = clicked_precinct)  
  }  
})  
  
}  
  
# Run the Shiny app  
shinyApp(ui, server)
```

LeRoyCollins INSTITUTE

Established in 1988, the LeRoy Collins Institute is a nonpartisan, statewide policy organization which studies and promotes creative solutions to key private and public issues facing the people of Florida and the nation. The Institute is located in Tallahassee at Florida State University.

Named in honor of Florida Governor LeRoy Collins, the Institute is governed by a distinguished board of directors chaired by John Marks III. Other board members include executive, state, and local officials and senior professionals from across the state.

All Institute publications may be found on the Institute's website.

LeRoy Collins Institute Board of Directors:

Director Lonna Atkeson, Ph. D., Tallahassee
Chairman John Marks III, Tallahassee

Lester Abberger, Tallahassee
Jim Apthorp, Tallahassee
Roy Collins III, Tallahassee
Richard Crotty, Orlando
Bryan Desloge, Tallahassee
Pete Dunbar, Tallahassee
Rick Edmonds, St. Petersburg
Joel Embry, Jacksonville
Dykes Everett, Winter Park
Pegeen Hanrahan, Gainesville
Jim Ley, Sarasota

John Martinez, Orlando
Jane Menton, Tallahassee
Janet Owen, Orlando
David Rasmussen, Tallahassee
Don Slesnick, Coral Gables
Katy Sorenson, Miami
Hansel Tookes, Palm Beach
Nicole Washington, Miami
Alan Williams, Tallahassee
Carol Weissert, Tallahassee
Tim Chapin, Tallahassee