Election and Voting Database Project

Technical Report

Group members:

- Megan Cogguillo
- Patrick Saitta
- Joshua Sohan

Overview

For our project, we created a database containing data on the results of the 2016 and 2018 House, Senate, and Presidential elections as well as demographic and economic characteristics of eligible voters and their communities during those years. We extracted data on:

- Senate election returns by county
- House election returns by district
- Presidential election returns by state
- Unemployment data by state
- Demographic data by congressional district

We then cleaned/transformed the data, and used python to load our clean output files as tables into a SQL database (postgres). While the data contains differing levels of granularity (e.g. district, county, state), all results could easily be aggregated at the state level via a query of the database.

Data Extraction & Transformation

Senate Data

Description:

- We used a dataset (csv file) available from the MIT Election Lab projection with state-level returns for Senate elections from 1976 to 2018: https://doi.org/10.7910/DVN/PEJ5QU
- The dataset is structured such that each state has multiple rows: one or more for each candidate. While we initially wanted to transform the data to show the percentage of the vote that Republican candidates, Democrat candidates, and Other candidates received in each state (so each state would have a single row) as well as the winner for each state, it became clear that this might not be a good idea because of the differences in the ballot for each state:
    - States have varying numbers of candidates and parties on the ballot. For instance, while Hawaii had two candidates on the ballot (one Democrat, one Republican), New Jersey had eight candidates of eight different parties

  - A candidate can run as a member of multiple parties. For instance, Kirsten Gillibrand ran as both a Democrat and as a member of the Working Families Party
  - Typically, each state would only have one Senate race in a single year, but states may occasionally have two senate races on the ballot at once due to a special election. Minnesota and Mississippi both had two Senate seats on the ballot in 2018
  - Elections work differently in different states. For instance, while most states will only have one Republican and one Democrat on the ballot in the general election, California had two Democrats on the ballot, as a result of their nonpartisan primary system where the top two finishers advance to the general election regardless of party
  - Candidates might appear on the ballot in such a way that requires supplemental context to understand the results. For instance, Bernie Sanders is a registered Independent--and appears on the ballot as an Independent--but caucuses with the Democrats in the Senate. His victory is considered a win for Democrats, even though he is not actually a Democrat

Clean-up Process:

- Read csv file into a pandas DataFrame
- Dropped all rows where year was not 2016 or 2018 (these were the only years we were interested in)
- Dropped the columns that were not necessary for our database
- Renamed columns to have clearer names and create consistency across our datasets
- Created a new column to calculate the share of the vote that each candidate received
- Changed all parties other than “Democrat” and “Republican” to “Other” to allow for simpler analysis
- Exported the resulting DataFrame into a csv file

House Data

Description:

- We used a dataset (csv file) available from the MIT Election Lab projection with constituency-level (i.e. House district) returns from House elections from 1976 to 2018: https://doi.org/10.7910/DVN/IG0UN2
- The dataset is structured such that each House district has multiple rows: one or more for each candidate. Similar to the Senate data, the ballot can look very different across districts, so we chose to keep this underlying structure

Clean-up Process:

- Read csv file into a pandas DataFrame
- Dropped all rows where year was not 2016 or 2018 (these were the only years we were interested in)
- Dropped the columns that were not necessary for our database

- Renamed columns to have clearer names and create consistency across our datasets
- Created a new column with a unique identifier for each district formatted as “state abbreviation - district number” (e.g. CA-10). This is also required extracting the district number from the district column (initially formatted as “District #”) via a split function
- Created a new column to calculate the share of the vote that each candidate received
- Changed all parties other than “Democrat” and “Republican” to “Other” to allow for simpler analysis
- Exported the resulting DataFrame into a csv file

<u>President Data</u>

Description:

- We retrieved the president data from https://electionlab.mit.edu/data
- We found this data insightful because it provided information from the last 40 years dating back to 1976

Clean Up Process:

- We began to clean data by reviewing the CSV file we downloaded.
- The CSV contained information that was not needed. Therefore we began deleting columns and information that was not necessary such as presidential election data prior to 2016
- We changed our column names to format our dataframe for SQL
- We decided to inquire what was the percentage of votes the candidates received in each state by dividing the number of votes the candidate received by the total amount of votes received
- Once we have created the data frame we saved this data to a CSV file

<u>2018 Census data by District</u>

Description:

- We used a set of census data (in four separate .xlsx files) available from the US Census with demographic data about the US population on a Congressional District level.
- We cleaned up each of the files separately in their own notebooks (age_, sap_, edu_, and race_2018_cleanup). Then, we combined the data from the four resulting .csv files into one file (combined2018.csv), as shown in the combined_2018_cleanup.ipynb file. Clean-up Process (for each of the four files):
    - Read xlsx file into jupyter notebook (e.g. age_2018.xlsx) and create DataFrame
    - Get rid of first few rows of junk
    - Drop columns of data that are not needed
    - Rename columns from names like “Unnamed: xx” to appropriate names
    - Reset the index (.reset_index()) so that the state abbreviations, which was treated as the index column, ‘pop out’ into their own manipulable column

    - Combine State Abbreviation and District Number into new column Congressional District
    - Export dataframe into .csv file (e.g. age2018.csv)

Combining the data into one dataframe/csv file:

- Read each of the four .csv files (e.g. age2018.csv) into jupyter notebook combined_2018_cleanup.ipynb
- To combine data into one table, start by creating a new DataFrame (combined_2018_df) with the columns all tables share ("State", "State Abbreviation", "District Number", "Congressional District", and "Voting-Age Population") and also the cols unique to the age_2018 Dataframe from age_2018_df
- Join in unique cols from sex and poverty DataFrame (sap_2018_df) into combined_2018_df
- Did the same thing for education and race DataFrames
- Export resulting combined_2018_df as csv (combined2018.csv)

Unemployment Data

Description:

- We decided to compare the 2016 and 2018 Unemployment data available from the Bureau of Labor Statistics
- The data provided from the Bureau of Labor Statistics were 2 CSV files. One file for the 2016 Election and the other for the 2018 Primary Election.

Clean Up Process:

- To begin cleaning up the data, we used Jupyter Notebook to upload both CSV files
- We began by reviewing the files and create 2 data frames. One for 2016 dataset and the other for 2018 dataset
- Once we uploaded both datasets, we began renaming the column names of both data frames to begin centralizing the dataframe we want
- We joined both data frames using the join function in pandas by the "*State*" column
- We realize that we would have a problem merging the data with Postgres, so we had to rename our columns again so the data can be read in Postgres
- After we cleaned up our data, we saved the file as the "*finalunemploymentcombine.csv*"
- Using the clean unemployment data, we can use this data with our other datasets to gather insightful information regarding the eligible voters of the prior elections

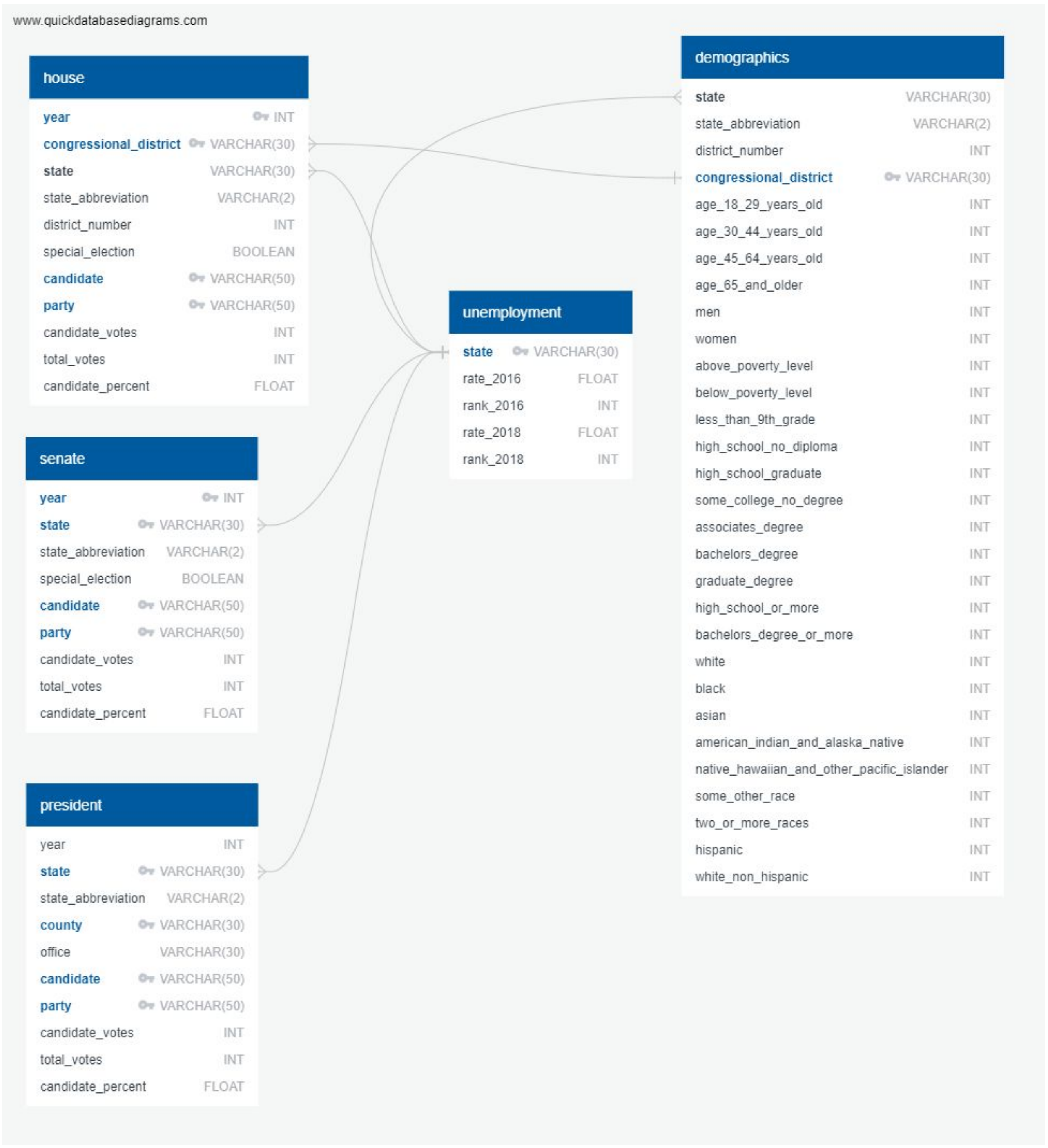
Database Creation and Data Load

Given the structured nature of our data, we chose to use a SQL database composed of five tables created from the cleaned dataframes we created in our transformation process:

- unemployment
- house
- senate
- president
- demographics

To create the database, we wrote a SQL script with the table schema, and ran the script in postgres. We then used python (in a jupyter notebook) to load the cleaned dataframes into the tables we created. An ERD diagram illustrating the various tables is available on the next page.

ERD Diagram:



house

year	🔑	INT
congressional_district	🔑	VARCHAR(30)
state		VARCHAR(30)
state_abbreviation		VARCHAR(2)
district_number		INT
special_election		BOOLEAN
candidate	🔑	VARCHAR(50)
party	🔑	VARCHAR(50)
candidate_votes		INT
total_votes		INT
candidate_percent		FLOAT

senate

year	🔑	INT
state	🔑	VARCHAR(30)
state_abbreviation		VARCHAR(2)
special_election		BOOLEAN
candidate	🔑	VARCHAR(50)
party	🔑	VARCHAR(50)
candidate_votes		INT
total_votes		INT
candidate_percent		FLOAT

president

year		INT
state	🔑	VARCHAR(30)
state_abbreviation		VARCHAR(2)
county	🔑	VARCHAR(30)
office		VARCHAR(30)
candidate	🔑	VARCHAR(50)
party	🔑	VARCHAR(50)
candidate_votes		INT
total_votes		INT
candidate_percent		FLOAT

unemployment

state	🔑	VARCHAR(30)
rate_2016		FLOAT
rank_2016		INT
rate_2018		FLOAT
rank_2018		INT

demographics

state		VARCHAR(30)
state_abbreviation		VARCHAR(2)
district_number		INT
congressional_district	🔑	VARCHAR(30)
age_18_29_years_old		INT
age_30_44_years_old		INT
age_45_64_years_old		INT
age_65_and_older		INT
men		INT
women		INT
above_poverty_level		INT
below_poverty_level		INT
less_than_9th_grade		INT
high_school_no_diploma		INT
high_school_graduate		INT
some_college_no_degree		INT
associates_degree		INT
bachelors_degree		INT
graduate_degree		INT
high_school_or_more		INT
bachelors_degree_or_more		INT
white		INT
black		INT
asian		INT
american_indian_and_alaska_native		INT
native_hawaiian_and_other_pacific_islander		INT
some_other_race		INT
two_or_more_races		INT
hispanic		INT
white_non_hispanic		INT