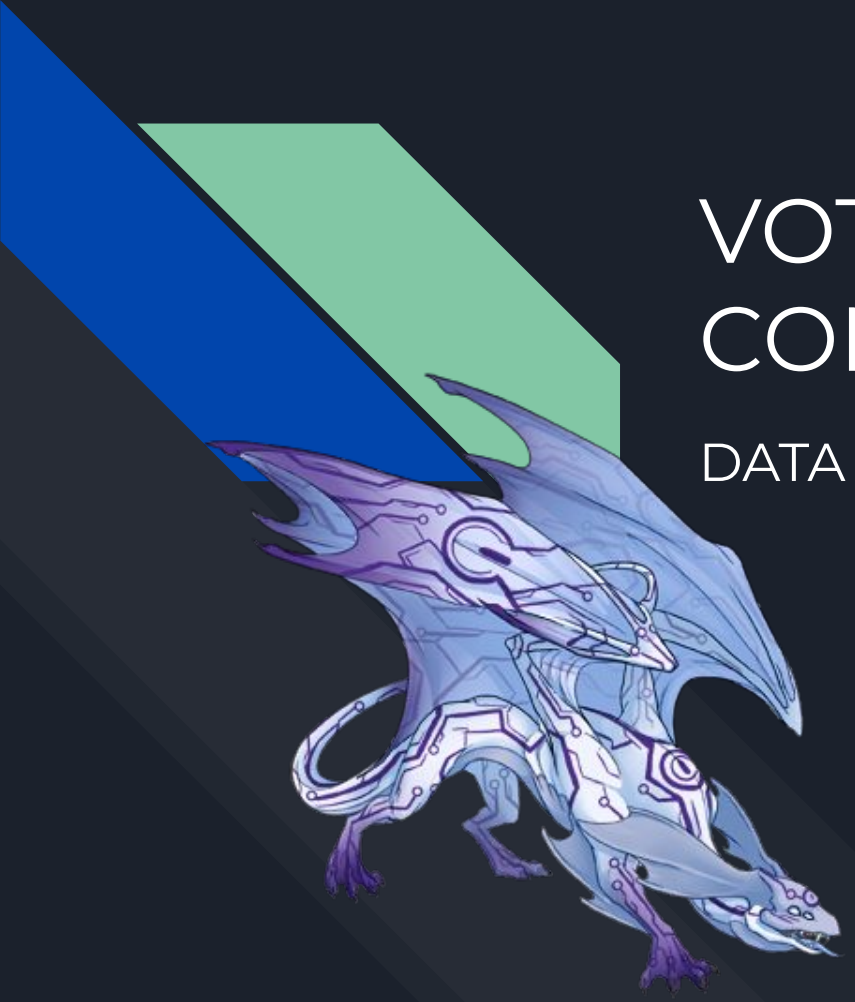

VOTERS AND VOTING CONDITIONS

DATA EXTRACT, TRANSFORM, LOAD

TEAM DATABASE DRAGONS
Rona Bourdeau
Jane Hur
Peter Smith

OBJECTIVE

Gather data on state demographics and voting conditions by state from 2016 General Election to see if there's a correlation between quality of voting condition and state population's race, economic status, and gender.

Extract using API, scrape and tables. Transform by cleaning, filtering and aggregating data. Load into SQL database.

TOOLS AND LIBRARIES

- Python
- Sqlalchemy
- Jupyter
- Pandas
- Requests
- Json
- Beautiful Soup
- Census API

DATASET 1: CENSUS API

Demographics data by state - Economic status, gender and race

Source: https://www.census.gov/data/developers.html

Data:

- State
- Household income
- Per capita income
- Poverty count
- Total employable population, over 16
- Total in labor force
- Civilian labor force unemployed
- Poverty rate
- Unemployment rate
- Total male
- Total female
- % Non-Citizen
- % Citizen

- White alone
- Black or African American alone
- American Indian and Alaska Native alone
- Asian alone
- Native Hawaiian and Pacific Islander alone
- Some other race alone
- Two or more races
- Two races including some other race
- Two races excluding some other race and three or more races
- Male Voting population - 18 & up
- Female Voting population - 18 & up

DATASET 1: CENSUS API

Extract

- Download and install Census wrapper for python (**census 0.8.14).**
- Use wrapper to query Census API and convert JSON data to dataframe
    - Used 5-year estimates (acs5.get) to gather 5 year estimates for population data. Selected year 2016 for query for each state.
    - Identified query codes for particular variables to query (e.g. "B01003_001E": "Population"

Transform

- Used Pandas to convert queried data into a data frame
- Renamed column headers from query code to definition of query code for ease of understanding.
- Performed calculations to determine each states poverty rate, unemployment rate, % citizenship and % non-citizenship as well as tabulated total male and female population for those of voting age. This data was not aggregated already.
- Remove columns that were only used for calculations.

Extract

- Use SQLAlchemy create_engine to create connection to PostgreSQL database
- Use Pandas to load dataframe into database
- Confirm data has been added by querying table

DATASET 2: CENSUS TABLES

Voter data by state - Citizenship, registered voter count, reported voter count

Source: https://www.census.gov/data/tables/time-series/demo/voting-and-registration/p20-580.html

Data:

- State
- Population
- Citizen population
- Total registered voters
- Percent registered voters
- Total reported voters
- Percent reported voters

DATASET 2: CENSUS TABLES

Extract

- Download excel spreadsheet from US Census Voting and Registration for Election of November 2016, then save it as csv file

Transform

- Used Pandas to load the csv onto a dataframe
- Set the state names as index.
- Dropped the totals row from the dataframe
- Chose only the pertinent columns to include in the dataframe
- When converting a csv to a dataframe, all the data is returned as string, even numeric data.
    - The numeric data includes a comma to separate thousands.
    - Use lambda function to replace the comma with a space
    - Converted numeric data from a string to float using pd.to_numeric
- Performed calculations to determine Percent Registered and Percent Voted

Load

- Use SQLAlchemy create_engine to create connection to PostgreSQL database
- Use Pandas to load dataframe into database
- Confirm data has been added by querying table

DATASET 3: ELECTION PERFORMANCE INDEX

Data on voting conditions from 2016 General Election
Source: https://elections.mit.edu

Data:

- State
- Data completeness
- Disability or illness related voting problems
- Mail ballots rejected
- Mail ballots unreturned
- Military and overseas ballots rejected
- Military and overseas ballots unreturned
- Online registration available
- Post election audit required
- Provisional ballots cast
- Provisional ballots rejected
- Registration or absentee ballot problems
- Registrations rejected
- Residual vote rate
- Turnout
- Voter registration rate
- Voting information lookup tools available
- Voting wait time

DATASET 3: ELECTION PERFORMANCE INDEX

Extract – Web Data Scrape

- Use Chromedriver and Splinter Browser to open Chrome window
- Loop through each state to:
  - Visit https://elections.mit.edu/#state-' + state
  - Use Beautiful Soup to parse into html page
  - Retrieve row section
  - Iterate through each row section to retrieve state name, indicator name and indicator value
  - Save all dictionary keys and values into list

Transform – Clean, Normalize

- Use Pandas to convert list of dictionaries into dataframe
- Use Pandas to pivot dataframe to use indicator names as column headers to normalize data

Load –PostgreSQL

- Use SQLAlchemy create_engine to create connection to PostgreSQL database
- Use Pandas to load dataframe into database
- Confirm data has been added by querying table

THANK YOU