Project Report

**Team Members:** Matthew Berning, Mohammad Ronosentono, and Paula Gallardo

**Overview**:

Correlate 2018 congressional midterm election candidates to their financial status and their respective district’s economic indicators.

**Process:**

For this projects, we retrieved information from four different data sources, two relating to U.S. 2018 congressional candidates and two with economic data by congressional district from 2016 and 2017. We then cleaned and organized the data with Jupyter Notebook and subsequently the data was loaded into a series of tables in a MySQL relational database to complete the Extract, Transform, Load process.

Extract

Initially we decided to use the Census Bureau [API](https://www.census.gov/developers/) to retrieve the data of interest. However, the interface was extremely complex and the actual endpoints for the most recent set of data were not clear from the documentation available further it was not clear if we would have the ability to then subset the retrieved data by congressional district. It then became apparent that the hard work of identifying endpoints for information by congressional district was already complete; we found a custom tool built by the Census Bureau called ‘[My Congressional District](https://www.census.gov/mycd/)’. This tool allows for very simple querying of information from the [American Community Survey](https://www.census.gov/programs-surveys/acs/about.html)  using the Census Bureau API, the results of each search query can then be simply exported as CSV files (by individual congressional district or by state). This process was mildly time consuming, and required downloading 52 individual CSV files, one for each state/territory/district with congressional representation in the 115th congress.

Our other datasets were much more straightforward in extraction; we simply downloaded CSV files from a [Kaggel source](https://www.kaggle.com/eliasdabbas/midterm-elections-candidates-search-results-pages/version/1#midterm_election_candidates_2018_0921.csv) and the [Federal Election Commission](https://www.fec.gov/files/bulk-downloads/index.html) which had candidate information for the 2018 midterms, and the [Census Bureau’s County Business Patterns: 2016](https://www.census.gov/data/datasets/2016/econ/cbp/2016-cbp.html) dataset for employment and employment sector.

Transform

When manipulating the data in Jupyter Notebook, we imported the dependencies and libraries, such as pandas for dataframes, sqlalchemy to connect python to MySQL, and pymysql to load the data into MySQL. The data gathered from Kaggle ([2018 Midterm Elections Candidates](https://www.kaggle.com/eliasdabbas/midterm-elections-candidates-search-results-pages/version/1#midterm_election_candidates_2018_0921.csv)) was cleaned, sorted, and split into two pandas dataframes to better identify House and Senate candidates’ information individually. For this, we used pandas to read each CSV file, we built separate tables, and dropped columns that were not considered relevant for future analysis.

A connection string from the dataframes created in Jupyter Notebook and MySQL database localhost was created through “create\_engine”. Duplicates were dropped and an index was assigned for each database to be loaded into MySQL.

A similar process was used to manipulate candidates financial CSV file ([Candidates Financial data from FEC](https://www.fec.gov/files/bulk-downloads/index.html)). However, we did not segregate senate candidates from house candidates for clarity purpose as it can be manipulated in MySQL and assessed according to forthcoming research questions.

The transformation process for the data derived from the My Congressional District CSV’s was slightly more complex, and required creating a subset of the 52 downloaded state/territory CSV files for expediency. Five random states were chosen, their CSV files were read into pandas dataframes and then they were stripped of extraneous data, duplicate rows were dropped, and only the relevant (employment/unemployment) numbers were retained we also used a transposition to match the primary key (state and district) to the other data sets. Then the individual dataframes were concatenated and pushed to a table in MySQL.

Load

We created five tables in MySQL to load the 2018 midterm election candidates data with respect to their financial information and their districts’ economic data.

We built a database schema in MySQL that is called election\_2018\_db that has the following tables:

* house\_candidate
* senate\_candidate
* candidate\_finance
* congressional\_district\_mcd
* congressional\_district\_cbp

House\_candidate and senate\_candidate tables contain the personal information of the congressman and senator candidates respectively, such as name, address, political party, status, congressional district, etc. Both tables have fec\_candidate\_id as the primary keys.

Candidate\_finance table contains information about the financial status of each candidate. It has one-on-one relation with house\_candidate and senate\_candidate tables based on fec\_candidate\_id as the foreign key.

Congressional\_district\_mcd table has information on the population of the district, the number of people employed and unemployed as of the date collected, and the unemployment rate from the My Congressional District CSV files. The key relation to the other databases is the state\_dist column which is the same information in the other tables.

Congressional\_district\_cbp table has employment information per district that is taken from census.gov website based on 2016 County Business Pattern data. It links to house\_candidate, senate\_candidate, and candidate\_finance based on the state and district number.

**Summary:**

The project was finished and the databases were created. We are also in a position to extract from the CSV’s additional fine grained economic indicators using minute variations on this transformation process.

Appendix: Project Proposal

**Overview**:

Correlate 2018 congressional midterm election candidates to their financial status and economic indicators by district.

**Data Sources:**

* 2018 Midterm Elections Candidates from Kaggle: <https://www.kaggle.com/eliasdabbas/midterm-elections-candidates-search-results-pages/version/1#midterm_election_candidates_2018_0921.csv>
* Economic data by district 2017 (My Congressional District): <https://www.census.gov/mycd/?st=01&cd=07>
* Employment by District 2016 compiled (Complete Congressional District File): <https://www.census.gov/data/datasets/2016/econ/cbp/2016-cbp.html>
* Candidates Financial data from FEC: <https://www.fec.gov/files/bulk-downloads/index.html>
  + 2018: [candidate\_summary\_2018.csv](https://cg-519a459a-0ea3-42c2-b7bc-fa1143481f74.s3-us-gov-west-1.amazonaws.com/bulk-downloads/2018/candidate_summary_2018.csv)

**Economic Indicators by district:**

* Employment
* Financial data for each candidate
* Unemployment rate by district

**Extract**

* Federal Election Commission (CSV): candidate\_summary\_2018.csv
* Kaggle (CSV): Midterm\_election\_candidates\_2018\_0921.csv
* [Complete Congressional District File](https://www2.census.gov/programs-surveys/cbp/datasets/2016/cbp16cd.zip?)
* Use API calls on the census.gov website “<https://www.census.gov/mycd/>” to build dataset of economic indicators i.e. ‘unemployment rate’ by congressional district

**Transform**

* In Python, using pandas, join the two datasets (from Kaggle and the FEC) using FEC candidate ID as primary key.
* Based on congressional district number, join economic indicators (from API calls to census.gov) with respective candidates

**Load**

* Build a single database in MySQL i.e. ‘election\_db’,  create tables for each candidate and each congressional district?
* MongoDB (api to mongoDB)

**Process**

* From the midterm elections candidates database 2018 (Kaggle), use “fec\_candate\_id”, “name”, “party”, “status” (all)
* From the candidates 2018, join tables through Cand\_id. Financial data from “Total receipt” to “Coverage End date”