MSBA 5303
PROGRAMMING FOR ANALYTICS
FALL 2019

POLIDATA

Final Report

Team Name: ABRACA-DATA

Cynthia Kagambirwa

Emilie Semo

Latonya McHale

Omotola Adeoye

Due December 4, 2019

Table of Contents

1. Executive Summary

Today, data is vital to an organization's success; it is not only having the data but understanding it that is key. Having knowledge of customers and their DNA (demographic, psychographic, and geographic behavior) gives insight to help organizations become customer centric and meet the needs of their target market group. It goes without saying, a business organization cannot be sustained without the financial support of the donors.

The aim of this group project is to provide a fundraising organization an in-depth analysis of donor's demographics. It is the objective of a successful fundraising company to find individuals who are willing to donate to campaigns and causes. Therefore, understanding the donor's characteristics and the correlation that these factors may have with donations will help the organization hone its campaign marketing practices.

The datasets were retrieved from various sources; two datasets were obtained from PoliData[1], and the two others from open sources (FBI.gov and simplemaps.com). We successfully obtained four files but only three were used for our analysis.

- AristotleFieldDefinitions: consists of 8 sheets and acts as a data dictionary but is unfortunately not defining the variables of our main dataset, that is why this dataset was not retained for our analysis.
- 91750_Match_Original_Data: consists of census information of U.S. registered voters.
- Crime.csv: consists of crime report in the U.S by state, 2017-2018.
- US Zipcodes: consists of 5-digit U.S. zip codes by state along with other geospatial variables.

[1] Per the COO's request, for privacy and legal purposes the name of the company was changes.

After collection of the datasets, the next steps taken were cleaning, constructing, integrating, merging, and formatting the data. The columns that were not purposeful to our analysis were removed. Also, the data consisted of a large number of missing values, which was drastically reduced once the data was cleaned.

Upon completion of transforming the data, we ran *t*-tests, Fisher/chi-square tests, and logistic regression. The first *t*-test was conducted to determine if there is a significant difference between voters who donate more than once and those who donate once with respect to mean age. The second *t*-test was conducted to determine if there is a significant difference between voters who donate more than $900 and those who donate less than $900 with respect to mean age. In both cases, we found that there is a significant difference (p-value<.05). We performed a third *t*-test to determine if there is a significant difference between male and female with respect to mean total donations amount. Although total donations amount mean for male is higher than the one for female, there was no significant difference found (p-value>.05). After running logistic regressions, we found two explanatory variables (voters age and median education years) that can explain the variability in number of donations and total donations amount.

It is recommended that PoliData obtains, if possible, a more complete dataset and provide information as to how the missing values should be handled. It is also suggested that PoliData incorporate additional information such as the date of each donation, to help us bring more insight to the analysis.

2. Introduction and Business Problem

PoliData is a public affairs company that provides a wide range of services such as strategy management, creative communications, data management, research, but their main focus in all of these is fundraising. They help national associations and Fortune 500 corporations succeed in today's competitive political environment by providing market research and communication solutions to raise funds for federal and state political action committees (PACs). They purchase and manage data for their clients, provide fundraising and a variety of other services to help boost the total gifts amount.

PoliData has access to a mass group of donors and would like to personalize their approach for each audience to increase the potential donation amount and target the most profitable donors. In fact, as we now have multiple generations in the workforce, Baby Boomers (1946-1964), Generation X (1964-1981), and Millenials (1981-1996) with Generation Z right behind (1995-Present), PoliData needs to be flexible in the way they market. Each one of these groups will need a different marketing strategy.

When marketing to different generations, it is important to understand their preferred means of communication. Gen Zs prefer to communicate online, through apps and social media (Facebook and Twitter), more often than in person. Millennials also favor online communication through text messaging and social media; they also tend to avoid face-to-face interaction and phone calls. Gen Xs will let calls go to voicemail and never check it but they are relatively quick answering text messages and emails. Boomers prefer in person interactions, but they are pretty flexible and adaptive and will answer phone calls, text messages, check emails, and use social media (The Ultimate List Of Charitable Giving Statistics For 2018, 2018).

The problem is the effort it takes to filter the data in a meaningful way to identify the ideal donors. Our solution to this issue will not only provide an analysis of the data, but will involve cleaning and transforming the data, and eliminating any unneeded variables from the data. This will improve our results and reliability of the data. This is important because it assists them in understanding their data and the best way to utilize their resources and streamline efforts. Our goal with this project is to identify who are the most profitable and dependable donors through the data analysis, so that PoliData can streamline their communication to those donors in hopes of increasing amounts and frequencies of donations.

3. Data Understanding and Data Preparation

3.1 Data Understanding

Our group obtained two tables of data from the Chief Operating Officer of PoliData and utilized it to conduct an analysis on the factors that could potentially affect the number of donations and donations amount. The first Excel file is named “AristotleFieldDefinitions” and is essentially a data dictionary. It is composed of 8 different sheets: Contributor_Consumer Specific, Vote History Fields, Turn out Fields, Occupation Code Appendix, Ethnicity-Language-Religion, Mortgage Lenders, Email Match Criteria and CBSA (A Core Based Statistical Area is a U.S. geographic area that include both Metropolitan areas (populations of 50,000 or more) and Micropolitan areas (populations 10,000 - 49,999)).

The second Excel file we obtained, “91750_Match_Original_Data.csv”, is where the actual data we will be analyzing is located. It consists of census information of U.S. registered voters collected through state and county level registered voter files, current U.S. census data, election return data, and third-party syndicated datasets like Experian. (Rhiza, 2016). After some internet research, we found out that PoliData probably bought this dataset from L2 Political, who is “the nation’s leading independent voter data and technology firm, processing voter data around the clock for all 50 states and DC and making that data available for analysis in the industry’s leading technology platform, L2 VoterMapping™.” (L2 Political, n.d.)

Upon review of these two tables, it was determined that we only needed “91750_Match_Original_Data.csv”, to conduct our analysis. It is composed of 45,744 rows and 367 columns and has 84.2% of missing values. A great number of the columns will be unusable as 278 columns have more than 70% of missing values, which will need to be dropped. Moreover, our two most important columns, which are the total donations amount and the number of donations, have 96.5% of missing values. We didn’t know if the missing values meant that the voter didn’t donate or if there were just no information on that voter, so we decided to delete the rows with missing values for these two particular columns. Lastly, we will need to drop all the columns containing confidential information such as residence address, phone number, voter ID, family ID, etc.

We also wanted to incorporate violent crimes statistics to our analysis and see if it would have an impact on the frequency and amount of donations. We found on the FBI.gov (Crime in the United States, Table 5, 2018) a report of the Crime in the United States by region, geographic division, and state for 2017-2018. We had to make substantial changes to that table (removing columns and rows to only keep the total per state and realigning headers) directly on Excel to be able to analyze it in Python. It is composed of 52 rows and 11 columns. There are no missing values. It has 1 object variable (state) and the rest of the variables are integers.

Since the crime dataset was by state and we didn't have a column called "state" in our original dataset, we needed to find one that would match each zip code to its respective state. After some research, we found a dataset that had zip code information per state along with other variables (US Zip Codes Database, 2019). It is composed of 33,099 rows and 16 columns. The only variable that has missing values is parent_ztca. It has 3 boolean variables, 4 floats variables, 3 integer variables and 6 object variables.

After cleaning and merging the different datasets, we drastically reduced the number of missing values. Our final dataset is composed of 1360 rows and 90 columns for only 4.1% of missing values.

3.2 Data Preparation

An analysis on each table was performed using Python. The .csv files were read into python using the following code:

- politic = pd.read_csv('91750_Match_Original_Data.csv', index_col='crna_ID')
- zip_to_state = pd.read_csv('US_Zip_to_State.01.csv')

- fbi = pd.read_csv('Table5_FBI_Gov_2018.csv')
- pd.set_option('display.max_column', 112)

Upon review of the tables, it was found that the 91750_Match_Original_Data.csv table consisted of a very large number of missing values. We looked at the number of columns that have an average of at least 70% of missing values and we looked at the percentage of missing values for our two most important columns, which are the total donations amount and the number of donations. The following codes were used to make this determination:

- zip_to_state.info()
- fbi.info()
- politic.shape
- politic.isna().mean().mean()
- (politic.isna().mean()>=0.7).sum()
- politic['FECDonors_TotalDonationsAmount'].isna().mean()
- politic['FECDonors_NumberOfDonations'].isna().mean()

We then proceeded to remove each column that was judged to be irrelevant to our analysis. We also removed the columns that seemed to be confidential and sensitive. Moreover, we removed all the rows that have missing values for both columns total donations amount and number of donations. The following codes were performed:

- politic.head()
- politic.drop(columns=politic.filter(like='InHome'), inplace=True)
- politic.drop(columns=politic.filter(like='In_Household'), inplace=True)
- politic.drop(columns=politic.filter(like='Primary_'), inplace=True)
- politic.drop(columns=politic.filter(like='PRI_BLT_'), inplace=True)
- politic.drop(columns=politic.filter(like='General_'), inplace=True)

- politic.dropna(subset=['FECDonors_TotalDonationsAmount'], inplace=True)
- politic.dropna(subset=['FECDonors_NumberOfDonations'], inplace=True)
- politic.dropna(axis=1,how='all', thresh=0.3*len(politic), inplace=True)
- cols=['SEQUENCE','LALVOTERID','Voters_StateVoterID','Voters_CountyVoterID','VoterTelephones_TelConfidenceCode','VoterTelephones_TelCellFlag','Residence_Addresses_ZipPlus4','Residence_Families_FamilyID','Mailing_Families_FamilyID','CommercialData_ISPSA','CommercialData_HomePurchaseDate','CommercialData_LandValue','School_District','Residence_Addresses_HouseNumber','Residence_Addresses_StreetName','Residence_Addresses_Designator','CommercialData_StateIncomeDecile','CommercialData_MosaicZ4','CommercialData_LikelyUnion','Voters_CalculatedRegDate','Voters_OfficialRegDate','County_Commissioner_District','Designated_Market_Area_(DMA)','Precinct','CommercialData_EstimatedIncome', ' CommercialData_DwellingUnitSize']
- politic.drop(columns=cols, inplace=True)
- politic.shape

Once the unnecessary columns were removed from the dataset, we looked at the columns we had left to see if we could rename some of them in a less confusing and shorter way. We realized that the columns starting with the string “FEC_Donors”, “'CommercialData”, “'CommercialDataLL” or ending with “Description” were not adding any meaning, therefore we decided to remove these prefixes and suffixes. The following codes were performed:

- politic.head()
- politic.rename(columns={col: col.split('FECDonors_')[-1] for col in politic.columns}, inplace=True)
- politic.rename(columns={col: col.split('CommercialData_')[-1] for col in politic.columns}, inplace=True)

- politic.rename(columns={col: col.split('CommercialDataLL_')[-1] for col in politic.columns}, inplace=True)
- politic.rename(columns={col: col.split('_Description')[0] for col in politic.columns}, inplace=True)

We also wanted to check the consistency of our data. We looked if there were any major differences between the Residence and Mailing columns for HHParties, HHCount, and HHGender. The following codes were performed:

- mixed_HHParties=np.where(politic['Mailing_HHParties']==politic['Residence_HHParties'], 0, 1)
- mixed_HHParties.sum()
- mixed_HHCount=np.where(politic['Mailing_Families_HHCount']==politic['Residence_Families_HHCount'],0, 1)
- mixed_HHCount.sum()
- mixed_HHGender=np.where(politic['Residence_HHGender']==politic['Mailing_HHGender'],0,1)
- mixed_HHGender.sum()

Since there were no significant differences between the Residence and the Mailing columns values, we decided to drop one of them, Mailing, to just keep the Residence column. We also renamed those columns to get rid of the prefix. The following codes were performed:

- cols = ['Mailing_HHParties', 'Mailing_Families_HHCount', 'Mailing_HHGender']
- politic.drop(columns=cols, inplace=True)
- politic.rename(columns={col:col.split('Residence_Addresses_')[-1] for col in politic.columns}, inplace=True)

- politic.rename(columns={col:col.split('Residence_')[-1] for col in politic.columns}, inplace=True)

The politic dataframe was now ready to be merged with the dataframe containing the state name. The ‘Zip’ from the original dataframe was used to match the ‘zip’ from the zip_to_state’ dataframe. We renamed the population variable as the next dataset we will be merging also contains a population variable. The following codes were performed:

- result = politic.merge(zip_to_state, left_on=Zip', right_on='zip')
- result.rename(columns={'population':'population_zip'}, inplace=True)

We repeated the same process with the fbi dataframe. The “state_name” of the previous merge (result) dataframe was matched to the “State” of the fbi dataframe and the population variable was relabeled appropriately. The following codes were performed:

- politic02 = result.merge(fbi, left_on='state_name', right_on='State')
- politic02.rename(columns={'population':'population_state'}, inplace=True)

As we finished merging the different dataframe, we decided to continue removing the unnecessary columns. The following codes were performed:

- cols = ['zcta', 'parent_zcta', 'county_fips', 'county_name', 'all_county_weights', 'imprecise', 'military', 'timezone']
- politic02.drop(columns=cols, inplace=True)
- politic02.shape

We then calculated the total crime per state by adding the different types of crime present in the dataset. The following code was performed:

- politic02['total_crime']=politic02['violent_crime']+politic02['murder_nonnegligent _manslaughter']+politic02['rape']+politic02['robbery']+politic02['aggravated_assau lt']+politic02['propert_crime']+politic02['burglary']+politic02['larceny_theft']+politic02['motor_vehicle_theft']

- politic02['total_crime']

In preparation for the statistical analysis, we looked at the unique values for the following variables: gender, political party, education, occupation, dwelling type, household gender, and marital status, and created an integer for each value. In order to reduce the number of integers per column, we grouped some of the strings into one meaningful common group. Moreover, some of these columns had missing values, therefore we filled them with an appropriate label. As it was lengthy lines of codes, please find a complete list of the codes in the appendix. Below is a partial list of the codes performed for this step:

- def parties (x):

```
if x =='Republican':
  […]
```

- politic02['parties_dummy'] = politic02['HHParties'].apply(parties)
- politic02['gender_dummy']=pd.get_dummies(politic02['Voters_Gender'],drop_first=True)
- politic02['Education'].fillna('Not Specified', inplace=True)
- def education (x):

```
if 'Bach' in x:
  return 1
    […]
```

- politic02['education_dummy'] = politic02['Education'].apply(education)
- politic02['Occupation'] = politic02['Occupation'].fillna(' Unknown ')
- def occupation (x):

```
if 'Financial' in x:
```

return 1

[…]

- politic02['occupation_dummy'] = politic02['Occupation'].apply(occupation)
- politic02['DwellingType'] = politic02['DwellingType'].fillna('Unknown')
- def dwelling (x):

if 'Multi' in x:

return 1

[…]

- politic02['dwelling_dummy'] = politic02['DwellingType'].apply(dwelling)
- def hhgender (x):

if 'Female' in x:

return 1

[…]

- politic02['HHgender_dummy'] = politic02['HHGender'].apply(hhgender)
- def marital (x):

if 'Married' in x:

return 1

[…]

- politic02['marital_dummy'] = politic02['MaritalStatus'].apply(marital)

We proceeded with the same method for the state variable. We didn’t want to have 50 different integers for each state, so we grouped them by region: south, west, midwest, and northeast (The Regions of the United States, 2019). The following codes were performed:

- west = ['Alaska', 'Arizona', 'California', 'Colorado', 'Hawaii', 'Idaho', 'Montana', 'Nevada', 'New Mexico', 'Oregon', 'Utah', 'Washington', 'Wyoming']

- midwest = ['Illinois', 'Indiana', 'Iowa', 'Kansas', 'Michigan', 'Missouri', 'Minnesota', 'Nebraska', 'North Dakota', 'Ohio', 'South Dakota', 'Wisconsin']
- south= ['Alabama', 'Arkansas', 'Delaware', 'Florida', 'Georgia', 'Kentucky', 'Louisiana', 'Maryland', 'Mississippi', 'Oklahoma', 'North Carolina', 'South Carolina', 'Tennessee', 'Texas', 'Virginia', 'West Virginia']
- northeast = ['Connecticut', 'Maine', 'New Hampshire', 'Massachusetts', 'New Jersey', 'New York', 'Pennsylvania', 'Rhode Island', 'Vermont']
- def region (x):

```
 if x in west:
  return 1
elif x in midwest:
  return 2
elif x in south:
 return 3
else:
  return 4
```

- politic02['region_dummy'] = politic02['State'].apply(region)

The last step before we were ready for statistical analysis was to fill the non-missing values for the voters age, estimated income amount, median education years, and estimated home value variables. We decided that we would fill them with the average and round it to a whole number to keep the data consistent. The following codes were performed:

- mean=politic02['Voters_Age'].mean()
- politic02['Voters_Age'] = politic02['Voters_Age'].fillna(mean).round(0)
- politic02['Voters_Age'].isnull().sum()
- mean02=politic02['EstimatedIncomeAmount'].mean()

- politic02['EstimatedIncomeAmount']=politic02['EstimatedIncomeAmount'].fillna(mean02).round(0)
- politic02['EstimatedIncomeAmount'].isnull().sum()
- mean03 = politic02['MedianEducationYears'].mean()
- politic02['MedianEducationYears']=politic02['MedianEducationYears'].fillna(mean03).round(0)
- politic02['MedianEducationYears'].isnull().sum()
- mean04=politic02['EstHomeValue'].mean()
- politic02['EstHomeValue'] = politic02['EstHomeValue'].fillna(mean04).round(0)
- politic02['EstHomeValue'].isnull().sum()

4. Methodology

Once we finished cleaning and merging the different datasets, we were ready to start working on the statistical analysis. In order to get the best results, we used both Python and SAS to perform *t*-test, Fisher/chi-square test, and logistic regression. We decided to create two groups for our dependent variables: number of donations greater than one (0=No, 1=Yes) and total donations amount greater than 900 (0=No, 1=Yes), which roughly represents the average of total donations amount in our dataset. We used the cut method to create these dichotomous variables.

When we first started on this project, we had the intuition that the voters age would affect both our dependent variables. In fact, it seems logical to think that older people tend to donate more money than their younger counterparts as they are more established in life. Therefore, we performed a *t*-test to see if there would be a significant difference between people who donate once and people who donate more than once with respect to mean age. We repeated the same test with our second dependent variable: total donations amount greater than 900. We also performed a *t*-test to investigate the relationship between the total donations amount and voters' gender; no significant difference was found.

The goal of our analysis is to find meaningful information that could explain the variation in number of donations as well as in the total donations amount. Therefore, we performed a logistic regression with the two dichotomous dependent variables we created. As Python doesn't have the stepwise selection method, we had to complement our analysis by using SAS. We first ran the logistic regression with Python and looked at the *p*-values to have a rough idea of which variables would be best for our model. After looking at our preliminary findings from Python, the following explanatory variables were selected: Voters_Age, occupation_dummy, education_dummy, EstimatedIncomeAmount, and Median Education

Years. These variables had *p*-values less than 0.05, and we included education as we wanted to investigate its relationship. Since occupation_dummy and education_dummy variables are composed of different categories (bachelor/graduate/highschool/other for education, and financial/management/manufacturing/medical/skilled/civil/education/other for occupation), we created dichotomous dummy variables for each of them.

We finished our analysis by creating visualization with Python. We used Tableau for map visualization.

5. Results

5.1 *t*-Test: Total Donations Amount v. Voters Gender

Title: Relationship between the mean of total donations amount for male versus female.

Summary Statistics:

Gender	N	Mean	Standard Deviation	Range	P-value
Female	660	984.8	2182.7	25-45640	.8627
Male	700	1002.8	1587.7	20-19500	

Conclusion: There is no significant difference between male and female with respect to mean total donations amount (p-value=0.8627).

5.2 *t*-Test: Voters Age v. Number of Donations>1

Title: Relationship between the mean age of voters who donate once versus voters who donate more than once.

Summary Statistics:

Number of Donations>1	N	Mean	Standard Deviation	Range	P-value
No	718	52.3	10.0	27-83	<.0001
Yes	642	54.6	9.5	30-81	

Conclusion: There is a significant difference between voters who donate once and voters who donate more than once with respect to mean age (p-value<0.0001).

5.3 *t*-Test: Voters Age v. Total Donations Amount>900

Title: Relationship between the mean age of voters who donate less than $900 versus voters who donate more than $900.

Summary Statistics:

Total Donations Amount>900	N	Mean	Standard Deviation	Range	P-value
No	977	52.6	10.0	27-83	<.0001
Yes	383	54.4	9.1	32-80	

Conclusion: There is a significant difference between voters who donate less than $900 and voters who donate more than $900 with respect to mean age (p-value<0.0001).

5.4 Logistic Regression: Number of Donations>1

Summary Statistics

	N (%)	Mean	Std Dev	Range
Number of Donations>1=Yes	642 (47.2)			
Voters Age	1360 (100)	53.4	9.9	27-83
Estimated Income Amount	1360 (100)	117962.4	59293.6	11000-250000
Median Education Years	1360 (100)	13.3	1.2	11-17
Estimated Home Value	1360 (100)	391693.3	292389.6	12500-4322395
Bachelor=Yes	728 (53.5)			
Master=Yes	320 (23.5)			
High School=Yes	26 (1.9)			
Other=Yes	286 (21.0)			
Financial=Yes	19 (1.4)			
Management=Yes	19 (1.4)			
Manufacturing=Yes	4 (0.3)			
Medical=Yes	925 (68.0)			
Skilled=Yes	6 (0.4)			
Civil=Yes	1 (0.1)			
Education=Yes	1 (0.1)			
Other_Unknown=Yes	398 (29.3)			

Pearson's and Phi Correlation Coefficient (*p*-value)

	Voters Age	**Median Education Years**	**Bachelor**	**Medical**
Voters Age	1.0000	-0.0357 (0.1883)	-0.0216 (0.4252)	0.2173 (<.0001)
Median Education Years		1.0000	0.0204 (0.4529)	0.0328 (0.2267)
Bachelor			1.0000	0.4326 (<.0001)
Medical				1.0000

p<0.0001 collinearity (significant correlation between predictor variables).

Best Model for predicting Number of Donations>1

Coefficients from Python:

Log odds (Number of Donations>1=Yes) = -2.4697 + 0.0239*Voters Age + 0.1054* Median Educations Years

Coefficients from SAS:

Log odds (Number of Donations>1=Yes) = -2.8184 + 0.0247*Voters Age + 0.1040*Median Education Years

Odds Ratio (95% CI) with interpretation

	Odds Ratio		**95% Confidence Interval**
	SAS	**Python**	
Voters Age	1.025	1.024	1.014-1.036
Median Education Years	1.110	1.111	1.017-1.210

Voters Age: For each 1-year increase in voters age, the likelihood of the number of donations being greater than 1 is 1.025 times higher, while controlling for median education years. Voters Age is a good predictor to estimate the donation frequency as the 95% confidence interval does not include 1.

Median Education Years: For each 1 unit increase in median education years, the likelihood of the number of donations being greater than 1 is 1.10 times higher, while controlling for voters age. Median education years is a good predictor to estimate the donation frequency as the 95% confidence interval does not include 1.

Fit of the Model

Percent Concordance (the larger, the better): 57.6%

Area under the curve (value close to 1 indicates a better fit of the model): 0.580

Hosmer and Lemeshow: The model is a good fit of the data (p=0.4684)

Probability of Number of Donations>1 for 60 years old and 14 years median education (using SAS best model)

Log Odds (Number of Donations>1=Yes) = -2.8184+0.0247*60+0.1040*14 = 0.1196

Odds (Number of Donations>1=Yes) = $e^{.1196}$= 1.1270

Probability = 1.1270/(1+1.1270) = **52.99%**

There is a 52.99% probability that a 60 years old voter with a 14 years median education will donate more than once.

5.5 Logistic Regression: Total Donations Amount>900

Summary Statistics

	N (%)	Mean	Std Dev	Range
Total Donations Amount>900=Yes	383 (28.2)			
Voters Age	1360 (100)	53.4	9.9	27-83
Estimated Income Amount	1360 (100)	117962.4	59293.6	11000-250000
Median Education Years	1360 (100)	13.3	1.2	11-17
Estimated Home Value	1360 (100)	391693.3	292389.6	12500-4322395

Pearson's and Phi Correlation Coefficient (p-value)

No variables were excluded due to collinearity (p<0.0001 collinearity (significant correlation between predictor variables)) but Estimated Income Amount was excluded from the final model as the 95% confidence interval included 1.

Best Model for predicting Total Donations Amount>900

Coefficients from Python:

Log odds (Total Donations Amount>900=Yes) = -4.5758 + 0.0294*Voters Age

Coefficients from SAS:

Log odds (Total Donations Amount>900=Yes) = -2.5136 +0.0292*Voters Age

Odds Ratio (95% CI) with interpretation

	Odds Ratio		95% Confidence Interval
	SAS	Python	
Voters Age	1.030	1.030	1.017-1.042

Voters Age: For each 1-year increase in voters age, the likelihood of total donations amount being greater than $900 is 1.030 times higher. Voters Age is a good predictor to estimate the total donations amount as the 95% confidence interval does not include 1.

Fit of the Model

Percent Concordance (the larger, the better): 56.3%

Area under the curve (value close to 1 indicates a better fit of the model): 0.578

Hosmer and Lemeshow: The model is a good fit of the data (p=0.6457)

Probability of Total Donations Amount>900 for a 60 years old voter (using SAS best model)

Log Odds (Total Donations Amount>900=Yes) = -2.5136 +0.0292*60 = -.7616

Odds (Total Donations Amount>900=Yes) = $e^{-.7616}$ = 0.4669

Probability = 0.4669/(1+0.4669) = **31.83%**

There is a 31.83% probability that 60 years old voter will donate more than $900 total.

5.6 Visualization

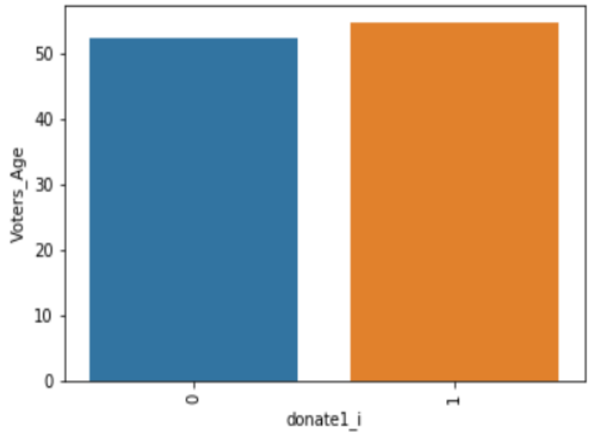

Figure 1: Voters Age for Number of Donations>1

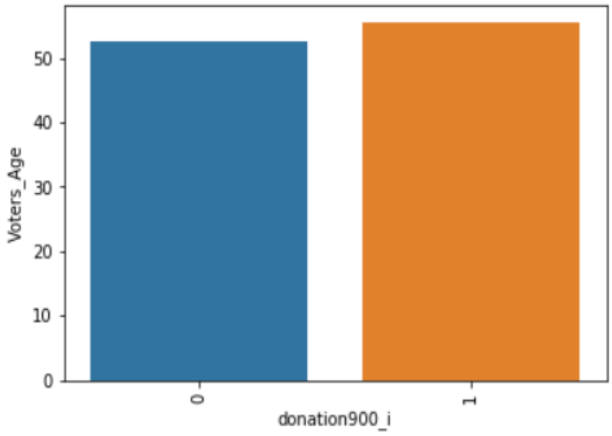

Figure 2: Voters Age for Total Donations Amount>900

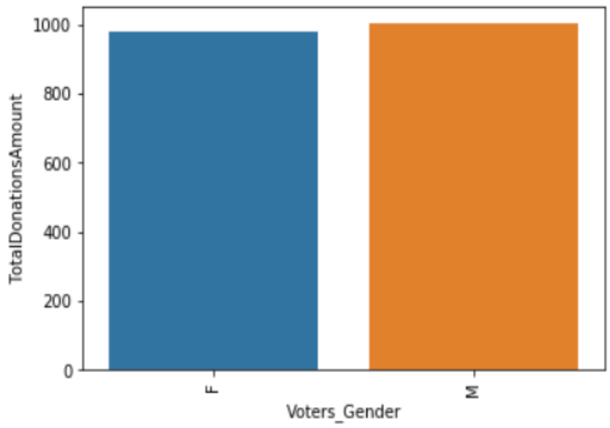

Figure 3: Total Donations Amount for Voters Gender

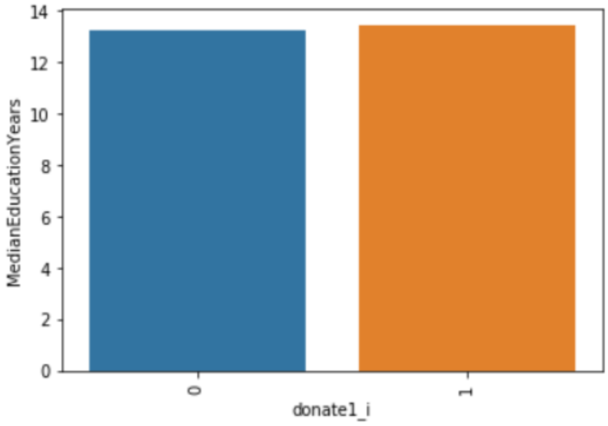

Figure 4 : Median Education Years for Number of Donations>1

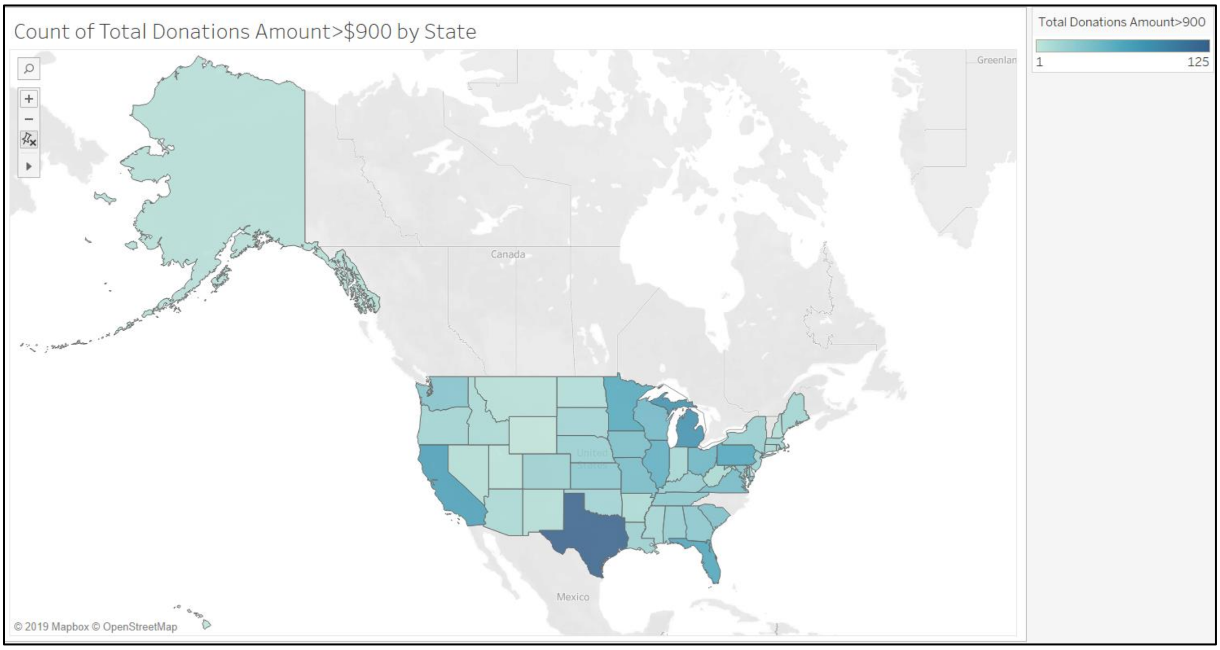

Figure 5: Count of Total Donations Amount>900 by State

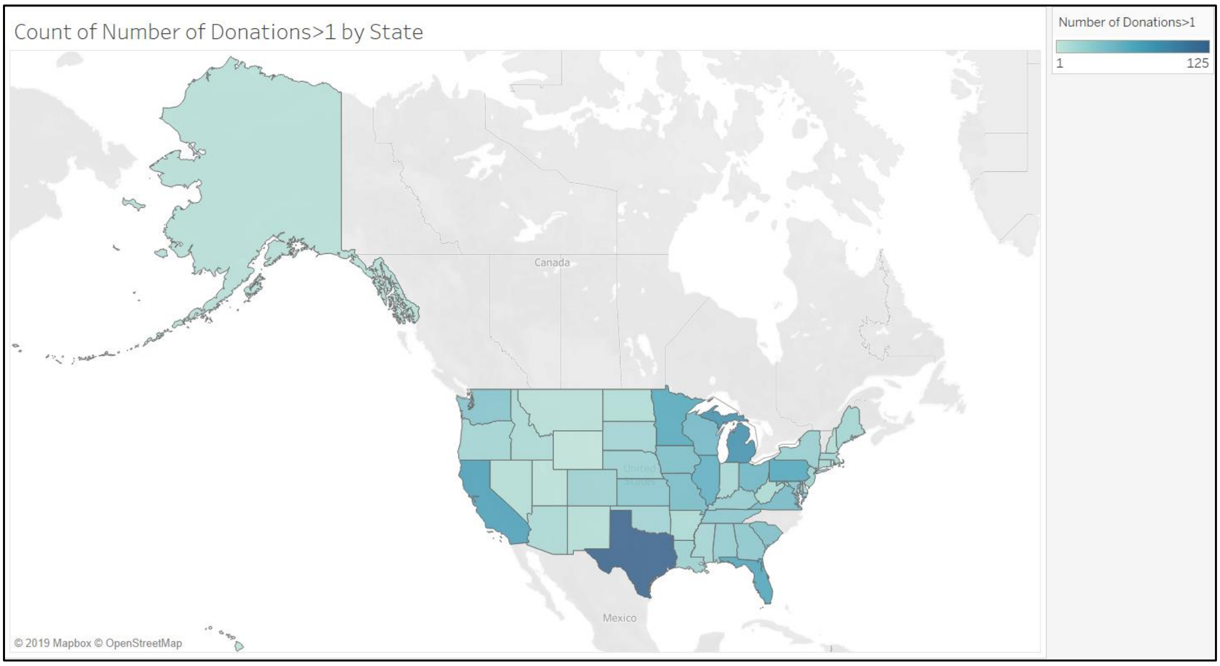

Figure 6: Count of Number of Donations>1 by State

6. Conclusion

Working with census information and political data can be challenging. As a team, we conducted a thorough analysis to look for characteristics such as gender, age, dwelling type, education, occupation, region, along with other variables that could influence the total donations amount and the number of donations. We hoped to find multiple explanatory variables. Unfortunately, we only found two relevant variables. After conducting *t*-tests, we found that there was a relationship between the mean age of the voters and the number of donations as well as the mean age of the voters and the total donations amount. We also found out, through logistic regression, that the variability in total donations amount greater than $900 could be explained by the voters age, and that the variability in number of donations greater than one could be explained by the voters age and the median education years. Due to missing values, our analysis doesn’t reflect the true story. Find below the limitations of the study and the recommendations we have for future analysis on this dataset.

6.1 Limitations

The dataset we obtained had a very large number of missing values. When working with a limited dataset, it is difficult to comfortably and confidently conduct an analysis to answer crucial questions. It would have been helpful to know what these missing values represent: does a cell with a missing value mean that there was nothing collected for that voter, or does it mean that the voter omitted that information when completing the form, or does it mean something else? For example, do the missing values in the total donation amount column indicate no donation, which in this case a 0 should have been entered, or does it mean that they don’t have the information?

We also think that providing additional information, such as the exact date of each donation, would have been beneficial to our analysis. If we had this data in hand, we could have correlated it to major events such as natural disasters, new bills being passed or presidential elections. We could have also looked for any patterns in the number of donations; do people tend to give once a year, every quarter, every 4 years, etc.? In addition, we suggest the company to add data on how the donations come through, i.e., social media, direct mail, text messaging, crowdfunding, fundraising events, etc., as well as which generation uses which of these means. This information would help them understand how to market effectively.

We decided to integrate the FBI crime dataset for the purpose of analyzing the data to determine if there was a correlation with donations and crime rates. Although we were able to obtain the data, it was only broken down by state. We could not find it by zip code or any other data that would allow us to narrow down the area. This was a too broad of a view and did not work with our data.

6.2 Recommendations

After reviewing the limitations, our team recommends PoliData to obtain, if possible, a more complete dataset and provide information as to how the missing values should be handled: should we get rid of it, replace it with an appropriate label, replace it with the mean, median, minimum, or maximum, or else? We also suggest PoliData to incorporate additional information such as the date of each donation and the means of donations to help us bring more insight to the analysis.

7. Appendix

7.1 FBI table variables

state: U.S. territory in which the crime was committed.

population: Amount of people living in that state (provided by the U.S. Census Bureau as of July 01, 2018).

violent_crime: crime where force was used on the victim.

murder_nonnegligent_manslaughter: offense resulting in death from one human being by another that was intentional and deliberate.

rape: offense involving the penetration of one human body part by another without consent of the victim.

robbery: offense involving stolen property through force, intimidation, or threats.

aggravated_assault: offense involving the attack of a person causing harm with the use of a dangerous/deadly weapon.

property_crime: offense involving burglary, larceny-theft, motor vehicle theft, and arson.

burglary: offense involving the unlawful entry of a structure to commit a felony or theft.

larceny_theft: offense involving the unlawful taking, carrying, leading, or riding of property from one person by another.

motor_vehicle_theft: offense involving the theft or attempted theft of a motor vehicle.

7.2 Zip to State table variables

zip: 5-digit zip code assigned by the U.S. Postal Service

lat: latitude of the zip code

lng: longitude of the zip code

city: official USPS city name

state_id: official USPS state abbreviation

state_name: state's name

zcta: TRUE if the zip code is a Zip Code Tabulation area

parent_zcta: ZCTA that contains this zip code

population: estimate of the zip code's population

density: estimated population per square kilometer

county_fips: zip's primary county in the FIPS format

county_name: name of the county_fips

all_county_weights: A JSON dictionary listing all counties and weights associated with the zip code

7.3 Python Codes

Dataset reading:

```
import pandas as pd
import numpy as np
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
politic = pd.read_csv('91750_Match_Original_Data.csv', index_col='crna_ID')
pd.set_option('display.max_column', 112)
zip_to_state = pd.read_csv('US_Zip_to_State.01.csv')
fbi = pd.read_csv('Table5_FBI_Gov_2018.csv')
```

Reviewed the different dataframes:

```
politic.shape
fbi.info()
zip_to_state.info()
politic.isna().mean().mean()
politic['FECDonors_TotalDonationsAmount'].isna().mean()
politic['FECDonors_NumberOfDonations'].isna().mean()
(politic.isna().mean()>=0.7).sum()
politic.head()
```

Removed irrelevant columns and columns with 70% of missing values:

```
politic.drop(columns=politic.filter(like='InHome'), inplace=True)
politic.drop(columns=politic.filter(like='In_Household'), inplace=True)
politic.drop(columns=politic.filter(like='Primary_'), inplace=True)
politic.drop(columns=politic.filter(like='PRI_BLT_'), inplace=True)
```

```
politic.drop(columns=politic.filter(like='General_'), inplace=True)
politic.dropna(axis=1,how='all', thresh=0.3*len(politic), inplace=True)
cols=['SEQUENCE','LALVOTERID','Voters_StateVoterID','Voters_CountyVoterID','VoterTelephones_TelConfidenceCode','VoterTelephones_TelCellFlag','Residence_Addresses_ZipPlus4',Residence_Families_FamilyID','Mailing_Families_FamilyID','CommercialData_ISPSA','ComercialData_HomePurchaseDate','CommercialData_LandValue','School_District','Residence_Addresses_HouseNumber','Residence_Addresses_StreetName','Residence_Addresses_Designator','CommercialData_StateIncomeDecile','CommercialData_MosaicZ4','CommercialData_LikelyUnion','Voters_CalculatedRegDate','Voters_OfficialRegDate','County_Commissioner_District','Designated_Market_Area_(DMA)','Precinct','CommercialData_EstimatedIncome','ComercialData_DwellingUnitSize']
politic.drop(columns=cols, inplace=True)
```

Removed rows that have a missing value for the variables Total Donations Amount and Number of Donations:

```
politic.dropna(subset=['FECDonors_TotalDonationsAmount'], inplace=True)
politic.dropna(subset=['FECDonors_NumberOfDonations'], inplace=True)
```

Renamed columns by deleting the prefix and suffix:

```
politic.rename(columns={col: col.split('FECDonors_')[-1] for col in politic.columns}, inplace=True)
politic.rename(columns={col: col.split('CommercialData_')[-1] for col in politic.columns}, inplace=True)
politic.rename(columns={col: col.split('CommercialDataLL_')[-1] for col in politic.columns}, inplace=True)
politic.rename(columns={col: col.split('_Description')[0] for col in politic.columns}, inplace=True)
```

Checked the consistency of the data:

```
mixed_HHParties=np.where(politic['Mailing_HHParties']==politic['Residence_HHParties'], 0,1)
mixed_HHParties.sum()
mixed_HHCount=np.where(politic['Mailing_Families_HHCount']==politic['Residence_Families_HHCount'],0, 1)
mixed_HHCount.sum()
mixed_HHGender=np.where(politic['Residence_HHGender']==politic['Mailing_HHGender'], 0,1)
mixed_HHGender.sum()
```

Dropped and renamed more columns:

```
cols=['Mailing_HHParties','Mailing_Families_HHCount','Mailing_HHGender']
politic.drop(columns=cols, inplace=True)
politic.rename(columns={col:col.split('Residence_Addresses_')[-1] for col in politic.columns}, inplace=True)
politic.rename(columns={col:col.split('Residence_')[-1] for col in politic.columns}, inplace=True)
```

Merged the FBI dataset and renamed the population column:

```
result = politic.merge(zip_to_state, left_on=Zip', right_on='zip')
```

```
result.rename(columns={'population':'population_zip'}, inplace=True)
```

Merged the State dataset and renamed the population column:

```
politic02 = result.merge(fbi, left_on='state_name', right_on='State')
politic02.rename(columns={'population':'population_state'}, inplace=True)
```

Removed irrelevant columns from the previous merge:

```
cols = ['zcta', 'parent_zcta', 'county_fips', 'county_name', 'all_county_weights', 'imprecise',
'military', 'timezone']
politic02.drop(columns=cols, inplace=True)
```

Created a new column, ‘total_crime’ to store the calculation of total crime per state:

```
politic02['total_crime']=politic02['violent_crime']+politic02['murder_nonnegligent_manslaug
hter']+politic02['rape']+politic02['robbery']+politic02['aggravated_assault']+politic02['propert
y_crime']+politic02['burglary']+politic02['larceny_theft']+politic02['motor_vehicle_theft']
```

Checked for NaN, created a function to assign a unique integer to each party and created a new column to apply the function:

```
politic02['Parties'].isna().sum()
def parties (x):
    if x =='Republican':
        return 1
    elif 'Democratic' == x:
        return 2
    elif 'Non-Partisan' == x:
        return 3
    elif 'Registered Independent' == x:
        return 4
    elif 'American Independent' == x:
        return 5
    elif 'Green' == x:
        return 6
    else:
        return 7
politic02['parties_dummy'] = politic02['HHParties'].apply(parties)
```

Created dummy variables for the voters gender column:

```
politic02['gender_dummy']=pd.get_dummies(politic02['Voters_Gender'],drop_first=True)
```

Checked for NaN in the Education column and filled them with an appropriate label:

```
politic02['Education'].isna().sum()
politic02['Education'].fillna('Not Specified', inplace=True)
politic02['Education'].value_counts()
```

Created a function to assign a unique integer to each education type, and created a new column to apply the function:

```
politic02['Education'].value_counts()
def education (x):
    if 'Bach' in x:
        return 1
```

```
    elif 'College' in x:
        return 1
    elif 'Grad' in x:
        return 2
    elif 'HS Diploma - Likely' == x:
        return 3
    elif 'HS Diploma - Extremely Likely' == x:
        return 3
    else:
        return 4
politic02['education_dummy'] = politic02['Education'].apply(education)
```

Checked for NaN in the Occupation column and filled them with an appropriate label:

```
politic02['Occupation'].isna().sum()
sorted(politic02['Occupation'].unique())
politic02['Occupation'] = politic02['Occupation'].fillna(' Unknown ')
```

Created a function to assign a unique integer to each occupation type, and created a new column to apply the function:

```
def occupation (x):
    if 'Financial' in x:
        return 1
    elif 'Management' in x:
        return 2
    elif 'Manufacturing' in x:
        return 3
    elif 'Medical' in x:
        return 4
    elif 'Skilled' in x:
        return 5
    elif 'Civil' in x:
        return 6
    elif 'Education' in x:
        return 7
    else:
        return 8
politic02['occupation_dummy'] = politic02['Occupation'].apply(occupation)
```

Checked for NaN in the Dwelling Type column and filled them with an appropriate label:

```
politic02['DwellingType'].isna().sum()
sorted(politic02['DwellingType'].unique())
politic02['DwellingType'] = politic02['DwellingType'].fillna('Unknown')
```

Created a function to assign a unique integer to each dwelling type, and created a new column to apply the function:

```
def dwelling (x):
    if 'Multi' in x:
        return 1
    elif 'Single' in x:
```

```
        return 2
    else:
        return 3
politic02['dwelling_dummy'] = politic02['DwellingType'].apply(dwelling)
```

Checked for NaN, created a function to assign a unique integer to each household gender, and created a new column to apply the function:

```
politic02['HHGender'].isna().sum()
sorted(politic02['HHGender'].unique())
def hhgender (x):
    if 'Female' in x:
        return 1
    elif 'Male' in x:
        return 2
    else:
        return 3
politic02['HHgender_dummy'] = politic02['HHGender'].apply(hhgender)
```

Checked for NaN, created a function to assign a unique integer to each marital status, and created a new column to apply the function:

```
politic02['MaritalStatus'].isna().sum()
sorted(politic02['MaritalStatus'].unique())
def marital (x):
    if 'Married' in x:
        return 1
    elif 'Single' in x:
        return 2
    else:
        return 3
politic02['marital_dummy'] = politic02['MaritalStatus'].apply(marital)
```

Grouped each state name by region:

```
west = ['Alaska', 'Arizona', 'California', 'Colorado', 'Hawaii', 'Idaho', 'Montana', 'Nevada', 'New Mexico', 'Oregon', 'Utah', 'Washington', 'Wyoming']
midwest = ['Illinois', 'Indiana', 'Iowa', 'Kansas', 'Michigan', 'Missouri', 'Minnesota', 'Nebraska', 'North Dakota', 'Ohio', 'South Dakota', 'Wisconsin']
south= ['Alabama', 'Arkansas', 'Delaware', 'Florida', 'Georgia', 'Kentucky', 'Louisiana', 'Maryland', 'Mississippi', 'Oklahoma', 'North Carolina', 'South Carolina', 'Tennessee', 'Texas', 'Virginia', 'West Virginia']
northeast = ['Connecticut', 'Maine', 'New Hampshire', 'Massachusetts', 'New Jersey', 'New York', 'Pennsylvania', 'Rhode Island', 'Vermont']
```

Created a function to assign a unique integer to each region, and created a new column to apply the function:

```
def region (x):
 if x in west:
  return 1
elif x in midwest:
  return 2
elif x in south:
```

```
 return 3
else:
  return 4
politic02['region_dummy'] = politic02['State'].apply(region)
```

Filled the missing values of the Voters Age with the mean of the column:

```
mean=politic02['Voters_Age'].mean()
politic02['Voters_Age'] = politic02['Voters_Age'].fillna(mean).round(0)
politic02['Voters_Age'].isnull().sum()
```

Filled the missing values of the Estimated Income Amount with the mean of the column:

```
mean02=politic02['EstimatedIncomeAmount'].mean()
politic02['EstimatedIncomeAmount']=politic02['EstimatedIncomeAmount'].fillna(mean02).ro
und(0)
politic02['EstimatedIncomeAmount'].isnull().sum()
```

Filled the missing values of the Median Education Years with the mean of the column:

```
mean03 = politic02['MedianEducationYears'].mean()
politic02['MedianEducationYears']=politic02['MedianEducationYears'].fillna(mean03).round(
0)
politic02['MedianEducationYears'].isnull().sum()
```

Filled the missing values of the Estimated Home Value with the mean of the column:

```
mean04=politic02['EstHomeValue'].mean()
politic02['EstHomeValue'] = politic02['EstHomeValue'].fillna(mean04).round(0)
politic02['EstHomeValue'].isnull().sum()
```

Created a function to set 1 for bachelor degree and 0 for all others, and created a new column to apply the function:

```
def bach (x):
    if x is 1:
        return 1
    else:
        return 0
politic02['bachelor'] = politic02['education_dummy'].apply(bach)
```

Created a function to set 1 for master degree and 0 for all others, and created a new column to apply the function:

```
def grad (x):
    if x is 2:
        return 1
    else:
        return 0
politic02['master'] = politic02['education_dummy'].apply(grad)
```

Created a function to set 1 for high school degree and 0 for all others, and created a new column to apply the function:

```
def hs (x):
    if x is 3:
        return 1
```

```
    else:
        return 0
politic02['highschool'] = politic02['education_dummy'].apply(hs)
```

Created a function to set 1 for other and 0 for all others, and created a new column to apply the function:

```
def other (x):
    if x is 4:
        return 1
    else:
        return 0
politic02['other'] = politic02['education_dummy'].apply(other)
```

Created a function to set 1 for financial industry and 0 for all others, and created a new column to apply the function:

```
def fin (x):
    if x is 1:
        return 1
    else:
        return 0
politic02['financial'] = politic02['occupation_dummy'].apply(grad)
```

Created a function to set 1 for management industry and 0 for all others, and created a new column to apply the function:

```
def mgmt (x):
    if x is 2:
        return 1
    else:
        return 0
politic02['management'] = politic02['occupation_dummy'].apply(mgmt)
```

Created a function to set 1 for manufacturing industry and 0 for all others, and created a new column to apply the function:

```
def mfg (x):
    if x is 3:
        return 1
    else:
        return 0
politic02['manufacturing'] = politic02['occupation_dummy'].apply(mfg)
```

Created a function to set 1 for medical industry and 0 for all others, and created a new column to apply the function:

```
def med (x):
    if x is 4:
        return 1
    else:
        return 0
politic02['medical'] = politic02['occupation_dummy'].apply(med)
```

Created a function to set 1 for skilled industry and 0 for all others, and created a new column to apply the function:

```
def skilled (x):
    if x is 5:
        return 1
    else:
        return 0
politic02['skilled'] = politic02['occupation_dummy'].apply(skilled)
```

Created a function to set 1 for civil industry and 0 for all others, and created a new column to apply the function:

```
def civil (x):
    if x is 6:
        return 1
    else:
        return 0
politic02['civil'] = politic02['occupation_dummy'].apply(civil)
```

Created a function to set 1 for education industry and 0 for all others, and created a new column to apply the function:

```
def edu (x):
    if x is 7:
        return 1
    else:
        return 0
politic02['education'] = politic02['occupation_dummy'].apply(edu)
```

Created a function to set 1 for other industry and 0 for all others, and created a new column to apply the function:

```
def other (x):
    if x is 8:
        return 1
    else:
        return 0
politic02['other_unknown'] = politic02['occupation_dummy'].apply(other)
```

Created a correlation coefficient matrix in preparation for logistic regression:

```
logistic=politic02[['Voters_Age','gender_dummy','parties_dummy','education_dummy','occupation_dummy','dwelling_dummy','HHgender_dummy','marital_dummy','region_dummy','Families_HHCount','EstimatedIncomeAmount','CensusBlockGroup','EstHomeValue','MedianEducationYears']]
logistic.corr()
```

Split the Number of Donations variable into a binary variable:

```
politic02['donate1']=pd.cut(politic02['NumberOfDonations'],[0,1,politic02['NumberOfDonations'].max()], labels=[0,1])
politic02['donate1_i']=politic02['donate1'].astype(int)
```

Logistic regression for Number of Donations>1 with odds ratio

```
model=smf.logit('donate1_i ~ Voters_Age + gender_dummy + parties_dummy + education_dummy + occupation_dummy + dwelling_dummy+HHgender_dummy + marital_dummy + region_dummy + Families_HHCount + EstimatedIncomeAmount + CensusBlockGroup + EstHomeValue + MedianEducationYears', data=politic02)
results=model.fit()
results.summary()
```

Split the Total Donations Amount variable into a binary variable:

```
politic02['donation900']=pd.cut(politic02['TotalDonationsAmount'],[0,900,politic02['TotalDonationsAmount'].max()], labels=[0,1])
politic02['donation900_i']=politic02['donation900'].astype(int)
```

Logistic regression for Total Donations Amount>900 with odds ratio

```
model=smf.logit('donation900_i ~ Voters_Age + gender_dummy + parties_dummy + education_dummy + occupation_dummy +dwelling_dummy + HHgender_dummy + marital_dummy + region_dummy + Families_HHCount + EstimatedIncomeAmount + CensusBlockGroup +EstHomeValue + MedianEducationYears', data=politic02)
results=model.fit()
results.summary()
```

Created a new dataframe called sas with the variables needed for the statistical analysis on SAS:

```
sas=politic02[['Voters_Age','gender_dummy','bachelor','master','highschool','other','financial','management','manufacturing','medical','skilled','civil','education','other_unknown','MedianEducationYears','EstimatedIncomeAmount','TotalDonationsAmount','EstHomeValue','donate1_i','donation900_i']]
sas.to_csv('sas.csv')
```

Created a new dataframe called cleaned data in preparation for Tableau visualization:

```
politic02.to_csv('cleaned data.csv')
```

Created visualization Figure 1:

```
ax = sns.barplot(x='donate1_i', y='Voters_Age', data=politic02, estimator=np.mean, ci=None)
ax.tick_params(axis='x', rotation=90);
```

Created visualization Figure 2:

```
Ax = sns.barplot(x='donation900_i', y='Voters_Age', data=politic02, estimator=np.mean, ci=None)
ax.tick_params(axis='x', rotation=90);
```

Created visualization Figure 3:

```
Ax = sns.barplot(x='Voters_Gender' ,y='TotalDonationsAmount', data=politic02, estimator=np.mean, ci=None)
ax.tick_params(axis='x', rotation=90);
```

Created visualization Figure 4:

```
ax = sns.barplot(x='donate1_i', y='MedianEducationYears', data=politic02, estimator=np.mean, ci=None)
ax.tick_params(axis='x', rotation=90);
```

7.4 Python Output

Logit Regression Results

Dep. Variable:	donate1_i	**No. Observations:**	1336
Model:	Logit	**Df Residuals:**	1321
Method:	MLE	**Df Model:**	14
Date:	Tue, 03 Dec 2019	**Pseudo R-squ.:**	0.01726
Time:	13:28:30	**Log-Likelihood:**	-907.71
converged:	True	**LL-Null:**	-923.65
Covariance Type:	nonrobust	**LLR p-value:**	0.004172

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-2.4692	0.875	-2.823	0.005	-4.183	-0.755
Voters_Age	0.0238	0.006	3.966	0.000	0.012	0.036
gender_dummy	0.0380	0.120	0.318	0.751	-0.196	0.272
parties_dummy	-0.0142	0.061	-0.232	0.816	-0.134	0.105
education_dummy	0.0820	0.069	1.195	0.232	-0.052	0.216
occupation_dummy	-0.0847	0.041	-2.061	0.039	-0.165	-0.004
dwelling_dummy	0.0071	0.132	0.054	0.957	-0.251	0.265
HHgender_dummy	0.0083	0.101	0.082	0.935	-0.189	0.205
marital_dummy	-0.0555	0.102	-0.546	0.585	-0.255	0.144
region_dummy	0.0212	0.063	0.338	0.735	-0.102	0.144
Families_HHCount	-0.0279	0.086	-0.323	0.746	-0.197	0.141
EstimatedIncomeAmount	8.782e-07	1.14e-06	0.772	0.440	-1.35e-06	3.11e-06
CensusBlockGroup	0.0204	0.051	0.401	0.689	-0.079	0.120
EstHomeValue	-2.34e-07	2.23e-07	-1.049	0.294	-6.71e-07	2.03e-07
MedianEducationYears	0.1019	0.050	2.020	0.043	0.003	0.201

```
Intercept                0.084649
Voters_Age               1.024078
gender_dummy             1.038717
parties_dummy            0.985925
education_dummy          1.085449
occupation_dummy         0.918816
dwelling_dummy           1.007143
HHgender_dummy           1.008292
marital_dummy            0.946029
region_dummy             1.021422
Families_HHCount         0.972524
EstimatedIncomeAmount    1.000001
CensusBlockGroup         1.020618
EstHomeValue             1.000000
MedianEducationYears     1.107258
dtype: float64
```

Logit Regression Results

Dep. Variable:	donation900_i	**No. Observations:**	1336
Model:	Logit	**Df Residuals:**	1321
Method:	MLE	**Df Model:**	14
Date:	Tue, 03 Dec 2019	**Pseudo R-squ.:**	0.02833
Time:	09:57:46	**Log-Likelihood:**	-766.90
converged:	True	**LL-Null:**	-789.26
Covariance Type:	nonrobust	**LLR p-value:**	4.531e-05

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-4.5758	0.979	-4.673	0.000	-6.495	-2.657
Voters_Age	0.0294	0.007	4.316	0.000	0.016	0.043
gender_dummy	0.1272	0.134	0.950	0.342	-0.135	0.390
parties_dummy	0.0475	0.067	0.705	0.481	-0.085	0.180
education_dummy	0.1214	0.077	1.572	0.116	-0.030	0.273
occupation_dummy	-0.0694	0.047	-1.478	0.139	-0.161	0.023
dwelling_dummy	-0.0670	0.149	-0.451	0.652	-0.358	0.224
HHgender_dummy	0.1357	0.113	1.196	0.232	-0.087	0.358
marital_dummy	0.0413	0.111	0.373	0.709	-0.176	0.258
region_dummy	0.1114	0.070	1.591	0.112	-0.026	0.249
Families_HHCount	-0.0059	0.095	-0.062	0.950	-0.191	0.180
EstimatedIncomeAmount	3.069e-06	1.25e-06	2.455	0.014	6.19e-07	5.52e-06
CensusBlockGroup	0.0529	0.057	0.932	0.352	-0.058	0.164
EstHomeValue	-1.01e-07	2.41e-07	-0.419	0.675	-5.74e-07	3.72e-07
MedianEducationYears	0.0770	0.056	1.380	0.168	-0.032	0.186

```
Intercept                0.010298
Voters_Age               1.029840
gender_dummy             1.135615
parties_dummy            1.048691
education_dummy          1.129034
occupation_dummy         0.932945
dwelling_dummy           0.935186
HHgender_dummy           1.145337
marital_dummy            1.042151
region_dummy             1.117883
Families_HHCount         0.994112
EstimatedIncomeAmount    1.000003
CensusBlockGroup         1.054318
EstHomeValue             1.000000
MedianEducationYears     1.080030
dtype: float64
```

7.5 SAS Codes

```
PROC IMPORT OUT= WORK.all
            DATAFILE= "C:\Users\sebmi\Desktop\sas.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
proc contents;

proc freq; tables bachelor master highschool other financial management
manufacturing medical skilled civil education other_unknown;
run;

proc sort; by gender_dummy;
proc means; by gender_dummy; var TotalDonationsAmount;
proc ttest; class gender_dummy; var TotalDonationsAmount;
run;

proc sort; by donate1_i;
proc means; by donate1_i; var voters_age;
proc ttest; class donate1_i; var voters_age;
run;

proc sort; by Donation900_i;
proc means; by Donation900_i; var voters_age;
proc ttest; class Donation900_i; var voters_age;
run;


/*no relation*/
proc freq; table donate1_i*gender_dummy/fisher chisq;
run;

/*no relation*/
proc freq; table donate1_i*parties_dummy/fisher chisq;
run;

/*RELATION*/
proc freq; table donate1_i*education_dummy/fisher chisq;
run;

/*RELATION*/
proc freq; table donate1_i*occupation_dummy/fisher chisq;
run;

/*no relation*/
proc freq; table donate1_i*dwelling_dummy/fisher chisq;
run;

/*no relation*/
proc freq; table donate1_i*HHgender_dummy/fisher chisq;
run;

/*no relation*/
proc freq; table donate1_i*marital_dummy/fisher chisq;
run;

/*no relation*/
proc freq; table donate1_i*region_dummy/fisher chisq;
run;
```

```
proc means; var Voters_Age EstimatedIncomeAmount MedianEducationYears
EstHomeValue;
run;


/*logistic 1*/
proc corr;var Voters_Age MedianEducationYears bachelor medical;
proc logistic descending; model donate1_i = Voters_Age
EstimatedIncomeAmount EstHomeValue MedianEducationYears bachelor master
highschool other financial management manufacturing medical skilled civil
education other_unknown/selection=stepwise;
proc logistic descending; model donate1_i = Voters_Age MedianEducationYears
bachelor medical /selection=score;
proc logistic descending; model donate1_i = Voters_Age
MedianEducationYears/lackfit ctable;
run;

/*logistic 2*/
proc corr; var Voters_Age EstimatedIncomeAmount MedianEducationYears;
proc logistic descending; model Donation900_i = Voters_Age
EstimatedIncomeAmount MedianEducationYears EstHomeValue bachelor master
highschool other financial management manufacturing medical skilled civil
education other_unknown/selection=stepwise;
proc logistic descending; model Donation900_i = Voters_Age/selection=score;
proc logistic descending; model Donation900_i = Voters_Age/lackfit ctable;
run;
```

7.6 SAS Output

The SAS System

The FREQ Procedure

bachelor	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	632	46.47	632	46.47
1	728	53.53	1360	100.00

master	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1040	76.47	1040	76.47
1	320	23.53	1360	100.00

highschool	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1334	98.09	1334	98.09
1	26	1.91	1360	100.00

other	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1074	78.97	1074	78.97
1	286	21.03	1360	100.00

financial	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1341	98.60	1341	98.60
1	19	1.40	1360	100.00

management	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1341	98.60	1341	98.60
1	19	1.40	1360	100.00

manufacturing	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1356	99.71	1356	99.71
1	4	0.29	1360	100.00

medical	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	435	31.99	435	31.99
1	925	68.01	1360	100.00

skilled	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1354	99.56	1354	99.56
1	6	0.44	1360	100.00

civil	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1359	99.93	1359	99.93
1	1	0.07	1360	100.00

education	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1359	99.93	1359	99.93
1	1	0.07	1360	100.00

other_unknown	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	962	70.74	962	70.74
1	398	29.26	1360	100.00

The SAS System

The MEANS Procedure

gender_dummy=0

Analysis Variable : TotalDonationsAmount				
N	Mean	Std Dev	Minimum	Maximum
660	984.8181818	2182.66	25.0000000	45640.00

gender_dummy=1

Analysis Variable : TotalDonationsAmount				
N	Mean	Std Dev	Minimum	Maximum
700	1002.81	1587.72	20.0000000	19500.00

The SAS System

The TTEST Procedure

Variable: TotalDonationsAmount

gender_dummy	N	Mean	Std Dev	Std Err	Minimum	Maximum
0	660	984.8	2182.7	84.9601	25.0000	45640.0
1	700	1002.8	1587.7	60.0100	20.0000	19500.0
Diff (1-2)		-17.9875	1899.8	103.1		

gender_dummy	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
0		984.8	818.0	1151.6	2182.7	2070.9	2307.2
1		1002.8	885.0	1120.6	1587.7	1508.7	1675.6
Diff (1-2)	Pooled	-17.9875	-220.2	184.2	1899.8	1831.0	1974.1
Diff (1-2)	Satterthwaite	-17.9875	-222.1	186.1			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	1358	-0.17	0.8615
Satterthwaite	Unequal	1199.2	-0.17	0.8627

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	659	699	1.89	<.0001

The SAS System

The MEANS Procedure

donate1_i=0

Analysis Variable : Voters_Age				
N	Mean	Std Dev	Minimum	Maximum
718	52.3300836	10.0472225	27.0000000	83.0000000

donate1_i=1

Analysis Variable : Voters_Age				
N	Mean	Std Dev	Minimum	Maximum
642	54.6417445	9.5145509	30.0000000	81.0000000

The SAS System

The TTEST Procedure

Variable: Voters_Age

donate1_i	N	Mean	Std Dev	Std Err	Minimum	Maximum
0	718	52.3301	10.0472	0.3750	27.0000	83.0000
1	642	54.6417	9.5146	0.3755	30.0000	81.0000
Diff (1-2)		-2.3117	9.7994	0.5323		

donate1_i	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
0		52.3301	51.5939	53.0662	10.0472	9.5531	10.5957
1		54.6417	53.9044	55.3791	9.5146	9.0210	10.0656
Diff (1-2)	Pooled	-2.3117	-3.3558	-1.2675	9.7994	9.4443	10.1824
Diff (1-2)	Satterthwaite	-2.3117	-3.3527	-1.2707			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	1358	-4.34	<.0001
Satterthwaite	Unequal	1353.5	-4.36	<.0001

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	717	641	1.12	0.1575

The SAS System

The MEANS Procedure

donation900_i=0

Analysis Variable : Voters_Age				
N	Mean	Std Dev	Minimum	Maximum
977	52.6438076	10.0429429	27.0000000	83.0000000

donation900_i=1

Analysis Variable : Voters_Age				
N	Mean	Std Dev	Minimum	Maximum
383	55.4046997	9.1067364	32.0000000	80.0000000

The SAS System

The TTEST Procedure

Variable: Voters_Age

donation900_i	N	Mean	Std Dev	Std Err	Minimum	Maximum
0	977	52.6438	10.0429	0.3213	27.0000	83.0000
1	383	55.4047	9.1067	0.4653	32.0000	80.0000
Diff (1-2)		-2.7609	9.7886	0.5901		

donation900_i	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
0		52.6438	52.0133	53.2743	10.0429	9.6165	10.5092
1		55.4047	54.4898	56.3196	9.1067	8.5042	9.8018
Diff (1-2)	Pooled	-2.7609	-3.9186	-1.6032	9.7886	9.4340	10.1712
Diff (1-2)	Satterthwaite	-2.7609	-3.8710	-1.6508			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	1358	-4.68	<.0001
Satterthwaite	Unequal	765.02	-4.88	<.0001

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	976	382	1.22	0.0247

The SAS System

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of donate1_i by gender_dummy		
	gender_dummy		
donate1_i	0	1	Total
0	351 25.81 48.89 53.18	367 26.99 51.11 52.43	718 52.79
1	309 22.72 48.13 46.82	333 24.49 51.87 47.57	642 47.21
Total	660 48.53	700 51.47	1360 100.00

Statistics for Table of donate1_i by gender_dummy

Statistic	DF	Value	Prob
Chi-Square	1	0.0773	0.7809
Likelihood Ratio Chi-Square	1	0.0773	0.7809
Continuity Adj. Chi-Square	1	0.0501	0.8229
Mantel-Haenszel Chi-Square	1	0.0773	0.7810
Phi Coefficient		0.0075	
Contingency Coefficient		0.0075	
Cramer's V		0.0075	

Fisher's Exact Test	
Cell (1,1) Frequency (F)	351
Left-sided Pr <= F	0.6302
Right-sided Pr >= F	0.4115
Table Probability (P)	0.0417
Two-sided Pr <= P	0.7861

Sample Size = 1360

The SAS System

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
Voters_Age	1360	53.4213235	9.8635869	27.0000000	83.0000000
EstimatedIncomeAmount	1360	117962.44	59293.61	11000.00	250000.00
MedianEducationYears	1360	13.3419118	1.2371521	11.0000000	17.0000000
EstHomeValue	1360	391693.29	292389.64	12500.00	4322395.00

The SAS System

The CORR Procedure

4 Variables:	Voters_Age MedianEducationYears bachelor medical

Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
Voters_Age	1360	53.42132	9.86359	72653	27.00000	83.00000
MedianEducationYears	1360	13.34191	1.23715	18145	11.00000	17.00000
bachelor	1360	0.53529	0.49894	728.00000	0	1.00000
medical	1360	0.68015	0.46659	925.00000	0	1.00000

Pearson Correlation Coefficients, N = 1360 Prob > |r| under H0: Rho=0				
	Voters_Age	MedianEducationYears	bachelor	medical
Voters_Age	1.00000	-0.03569 0.1883	-0.02164 0.4252	0.21733 <.0001
MedianEducationYears	-0.03569 0.1883	1.00000	0.02037 0.4529	0.03280 0.2267
bachelor	-0.02164 0.4252	0.02037 0.4529	1.00000	0.43257 <.0001
medical	0.21733 <.0001	0.03280 0.2267	0.43257 <.0001	1.00000

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Voters_Age	1.025	1.014	1.036
MedianEducationYears	1.110	1.017	1.210

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	57.6	Somers' D	0.160
Percent Discordant	41.7	Gamma	0.161
Percent Tied	0.7	Tau-a	0.080
Pairs	460956	c	0.580

Partition for the Hosmer and Lemeshow Test					
		donate1_i = 1		donate1_i = 0	
Group	Total	Observed	Expected	Observed	Expected
1	136	49	48.07	87	87.93
2	134	49	53.00	85	81.00
3	136	63	57.80	73	78.20
4	130	54	58.18	76	71.82
5	140	57	65.31	83	74.69
6	137	68	66.36	69	70.64
7	139	79	69.66	60	69.34
8	139	75	72.34	64	66.66
9	138	77	75.01	61	62.99
10	131	71	76.27	60	54.73

Hosmer and Lemeshow Goodness-of-Fit Test		
Chi-Square	DF	Pr > ChiSq
7.6495	8	0.4684

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Voters_Age	1.030	1.017	1.042

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	56.3	Somers' D	0.155
Percent Discordant	40.7	Gamma	0.160
Percent Tied	3.0	Tau-a	0.063
Pairs	374191	c	0.578

Partition for the Hosmer and Lemeshow Test					
		donation900_i = 1		donation900_i = 0	
Group	Total	Observed	Expected	Observed	Expected
1	150	29	28.32	121	121.68
2	140	25	30.40	115	109.60
3	126	28	30.13	98	95.87
4	117	32	30.40	85	86.60
5	157	52	43.30	105	113.70
6	138	39	40.50	99	97.50
7	120	42	37.07	78	82.93
8	109	35	35.08	74	73.92
9	148	49	49.82	99	98.18
10	155	52	58.02	103	96.98

Hosmer and Lemeshow Goodness-of-Fit Test		
Chi-Square	DF	Pr > ChiSq
6.0141	8	0.6457

8. References

(n.d.). From L2 Political: https://l2political.com/about/history/

(2016, March 22). From Rhiza: https://jenniferbc.online/Content/L2_datacatalog_methodology.htm

Crime in the United States, Table 5. (2018). From FBI: https://ucr.fbi.gov/crime-in-the-u.s/2018/crime-in-the-u.s.-2018/tables/table-4

The Regions of the United States. (2019, July 2020). From Wordl Atlas: https://www.worldatlas.com/articles/the-regions-of-the-united-states.html

The Ultimate List Of Charitable Giving Statistics For 2018. (2018). From NonProfitSource: https://nonprofitssource.com/online-giving-statistics/

US Zip Codes Database. (2019, July 11). From Simple Maps: https://simplemaps.com/data/us-zips