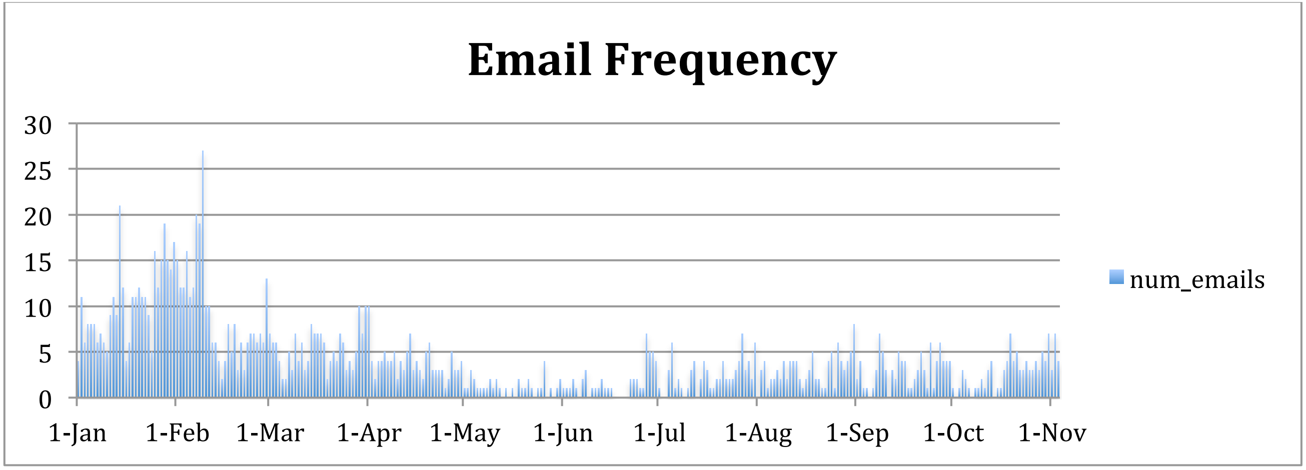
NLP Lab 3
Justin Postigo and Logan Williams

Preprocessing steps:

- We got rid of html tags and unicode characters in the email bodies.
- We got rid of http links in the email bodies.
- We used python's mailbox module for parsing the mbox files.

First finding: Frequency of Emails

Methods: We used NLTK's FreqDist to count the number of emails each candidate sent per day. We also parsed the date to just get the day and month. We take this data and put them into a CSV file and did a bar chart on the data. This gives us the amount of emails received on each date across all mailboxes.



Second finding: Authorship Attribution

For our second finding, we used a Naive Bayes classifier to figure out authorship of each email. For our features, we used a 5-gram. Within the 5-gram, the middle word must contain a dollar amount e.g. $10, $20, etc. The remaining four words must contain a word that denotes that the person is asking for money e.g. 'give', 'donate', etc. Using these 5 words as features, and the author as the label we train a Naive Bayes classifier on one-third of our data and test it on the remaining data. On average of about 5 trials, Naive Bayes gets about 82% accuracy for correctly guessing the author of the email.

Third finding: Sentiment Analysis/Positivity Scores

For our third finding, we use Vader Sentiment Analysis on each email from each candidate. We find the "positive" values from the sentiment analysis from each email. These values represent how positive the email actually is. To observe our findings, we put the data into a CSV file and did a line graph on the data. For each candidate, most of them stay fairly constant or got slightly lower positive scores overtime. There are some candidates such as Bush and Paul where their positive values went up overtime. This might be due to the low amount of emails from these candidates.