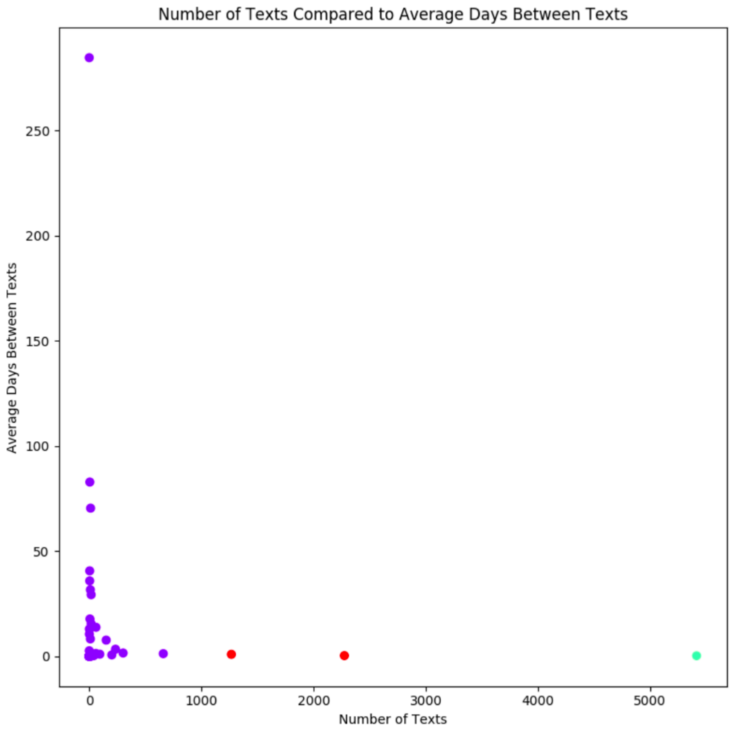
Dylan Martin

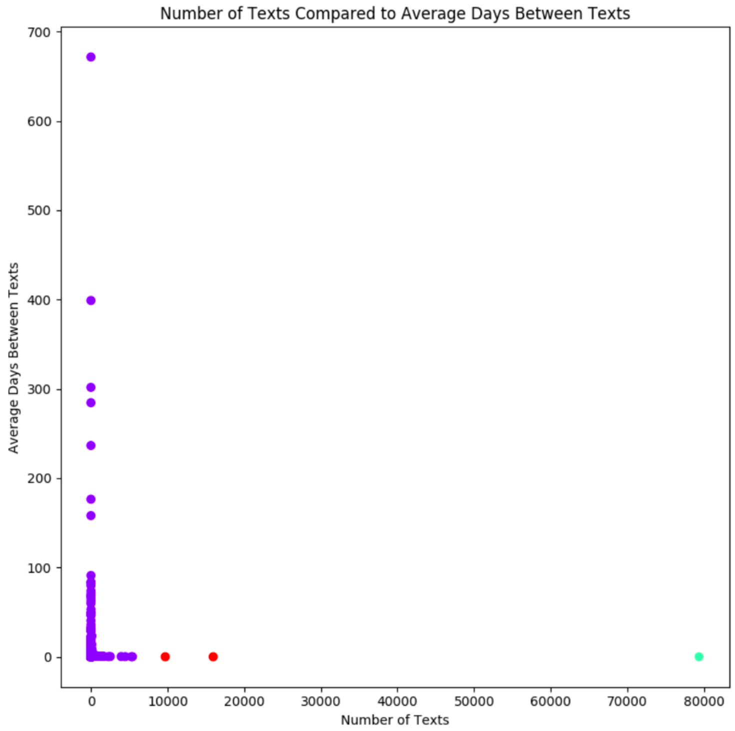
K-means Analysis on iMessage Data

The data set I chose to work on is my own iMessage database file. Since the data came in a database file I had to use sqlite to format the data into a useable data frame. Luckily the internet has examples on how to use sqlite for this task. The next step was to take the data frame gotten by using sqlite and perform more operations on it so that I had something that I could perform k-means analysis on. This involved making a coordinate data frame that included the number of texts every phone number sent for the x value and the average difference between every text that every phone number sent for the y value.

My hypothesis was that based on this coordinate scheme the k-means analysis would be able to group the phone numbers into people who are close friends or family i.e. people who have a high number of texts and a low average of days between texts, automated numbers used as conformation codes or family members who only text for specific occasions would have a small number of texts but a larger difference between days, and either spam, wrong numbers or one off texts have a low number of texts and a low average between texts (zero of they only sent one text). Overall I am happy with how the k-means analysis worked out since the graphs tend to show the distribution I described in the hypothesis. I used a k-means analysis with 3 clusters one for each of the groups I talked about earlier and as you can see in this graph where each color represents one of the three clusters there are groupings that follow the descriptions that I made fairly closely.



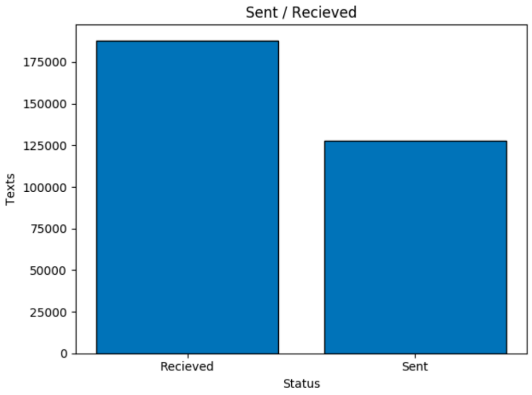
Data is taken from a random sample of 50 phone numbers

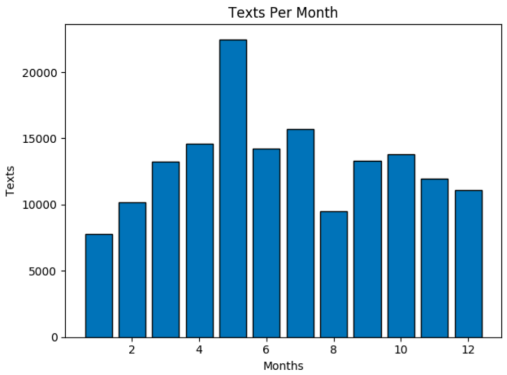


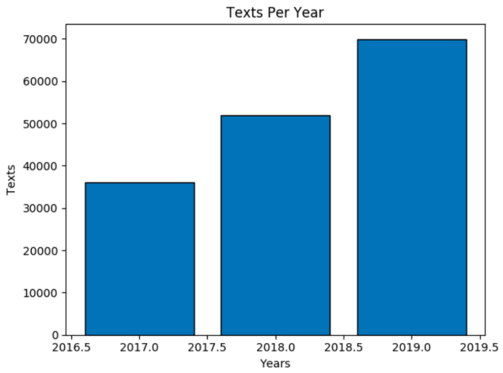
Data is taken from all of the phone numbers I have in my iMessage history.

The most difficult part of this process was finding a way to find the average difference between a list of days that I had received a text from a particular phone number. This was difficult mainly because of the sheer size of my data file. This meant that a good amount of the time I spent cleaning up the data was spent trying to optimize my for loops. I eventually got it down to about and average of around 30-40 seconds if there are no phone numbers with an extremely large amount of texts. In my data specifically I had one phone number that has around 80000 total text messages which is more than 4 times the amount of the second most text messages. This one node in particular could take about 20 seconds to run all by itself while most numbers finish in about 1 second.

The iMessage database also contains various other interesting information. I was working on this data set earlier in the semester before this project was assigned so I had already looked into the database a bit and came up with these graphs:







For future analysis of this data I think that a lot can be done especially with all of the information included in the database. It includes the text of every message that has been sent so perhaps a program could be made that learns to text like anyone who texts you or even yourself. Other interesting stuff would be predicting the time that a specific person will text you on any given day or a list of the top closest people to you. You could also use the data to block spam messages so if a person has been only sending messages with no response the program could ask the user if they want to block or silence the messages coming from the offending number.

Overview:

The program can be run with either my included chat database or one that you can get off of your computer assuming that you have a Mac and an iPhone with your iCloud synced to iMessage. To run the program the only necessary parameter to include is the path to your chat.db file. I highly recommend if you are going to use your own file that you make a copy of it and give the program the copy. It shouldn't do anything too damaging to the file but it did make my iMessage app a little messed up until I restarted my computer when I didn't make a copy of the file.

Steps for finding chat.db file:

1. With a new finder window open click on the settings drop down list.
2. From there go to Show View Options and check the Show Library Folder box.
3. Next navigate to /Users/yourname/Library/Messages/chat.db make a copy of this file and use that copy as the file for the program

After you've specified the correct chat.db file when you run the program it will ask if you would like to choose a sample size or run the program over the whole database. Running over the whole database can take a minute or two depending on how many texts you keep saved on your phone but eventually the program will output a scatter plot like the one I have included in on the other pages and a print-out of the data frame used by the k-means clustering algorithm along with the clusters each phone number was in. The program will then ask if you would like the view some extras and the extras are the three bar plots above. As a disclaimer the database file I have included does include my personal text messages so I would appreciate if you could use discretion when viewing data in the program. The only data frame that contains the raw text should be the df_messages data frame. The program used matplotlib to display the plots, scikitlearn.cluster for the k-means clustering, pandas for the data frame manipulation and sqlite3 to parse the database file.