Exploring Public Venmo Transactions:

What if I told you everyone was able to see every payment you made? The transactions we make on a daily basis can be a direct line to our personal behaviors, habits, and preferences. The value of these insights goes far beyond simply knowing how much we spend. The frequency at which we make payments, who we make payments to, and what we make payments for are only a few of the insights that if stitched together, could highlight trends in our spending and preferences that we did not even realize existed. This paper will expose just that.

Venmo is a digital wallet platform that facilitates electronic peer-to-peer payments. This means that it makes it easier to pay back Johnny for the beer last week (or Johnny can send a request money for the beer unfortunately) or pay your babysitter at the end of the night when you don't have cash. Venmo users have to create an account on the platform with a username and password, either through their own email or an existing Facebook account. This enables users to add other users as friends and create a similar social network as on other social media platforms. This network of friends is then used to create a news feed that displays the payments that your friends are making according to account settings. For each payment made or requested, users are required to include a message along with the transaction quantity. The message is meant to explain the purpose of the transaction but is frequently a field that only hints to the real nature of the payment through inside jokes, emojis, and other innuendos.

The purpose of this analysis was meant to make sense of what seems to be non-sense or random payments but is rather a treasure trove of data and value. The first part explores the data and the different components of a Venmo transaction. The second part is an attempt to understand each transaction's message in order to be able to cluster like-transactions to understand what types of payments user are making on Venmo.

The Public Venmo API:

All Venmo transactions are public by default.

Only once a user changes their account setting to hide their transactions from the public are they made private. These public transactions can be obtained via Venmo's public API here. There are use cases that mentioned that via other arguments ("since" and "until") within the API call, you were able to retrieve historical transactions. However, since this was discovered and bots were created to exploit this feature, Venmo has since limited its public API call to return only the last 50 most current transactions. While accessing historical transactions no longer seems to be possible, being able to gather current transactions still provides a wealth of insight into the types of payments that are made by Venmo users. This data was gathered over the span of several days around Halloween.

In addition, there are rate checks that limit the number of times you ping the Venmo API. For the purposes of this analysis, it was not critical that every transaction was captured in sequential order, thus the script that gather the data only pinged their servers every 30 seconds (I assume this is well below their rate limit because I was able to let the script run for multiple days in a row).

A Venmo Transaction:

Each Venmo transaction is structured the same way and contains the same type of data. The following fields is a selection of the available that will be used as part of this analysis:

Payment_id: A unique ID created for each Venmo transaction

Actor: Contains the following fields that pertain to the account that initiated the transaction
Actor / username: The username of the user
Actor / is_business: Signifies whether it is a business account or a personal account
Actor / name: The full name of the user
Actor / first_name: The first name of the user
Actor / last_name: The last name of the user
Actor / date_created: The date on which the user's account was created
Actor / id: The unique id of the user

Transaction: Contains the following fields that pertain to the account(s) that are the recipients of the transactions

**Depending on the type of the transaction, the recipient(s) can either be the recipient of a payment or a request. There can also be more than one recipient. **

Transaction / target / username: The username of the user
Transaction / is_business: Signifies whether it is a business account or a personal account
Transaction / name: The full name of the user
Transaction / first_name: The first name of the user
Transaction / last_name: The last name of the user
Transaction / date_created: The date on which the user's account was created
Transaction / id: The unique id of the user

Created_time: The time and date of the transaction

Message: The field that is supposed to contain a description of the transaction

Type: Signifies whether the transaction was a payment or request

----
Each of the above fields were captured for every transaction that the API call returned. Personal information such as first and last name were only used to attempt to predict the gender of the user. For the purposes of the analysis, the transactions were collected and saved in a CSV file so that the data gathering script did not have to be run at each time.

Processing Emojis:

An important characteristic of Venmo transactions are the emojis that are used to describe its purpose. Emojis are little symbols and pictures that users are able to make part of the text in their messages. This occurs for a multitude of reasons including a more fun way of saying something than text or a discrete way to describe an illicit or intimate transaction. Since emojis are regular text and can be used within the message field, it is critical to be able to parse out the emojis since often times they are the most telling features of a message.

Emojis are images that can be included as regular text because they have a text representation. The Unicode-text for each emoji can be found [here](). The text representation of emojis follow a consistent structure. Rather than trying to describe it, below is an example of the Unicode-text of similar emojis:

```
u':adult:': u'\U0001F9D1',
u':adult_dark_skin_tone:': u'\U0001F9D1\U0001F3FF',
u':adult_light_skin_tone:': u'\U0001F9D1\U0001F3FB',
u':adult_medium-dark_skin_tone:': u'\U0001F9D1\U0001F3FE',
u':adult_medium-light_skin_tone:': u'\U0001F9D1\U0001F3FC',
u':adult_medium_skin_tone:': u'\U0001F9D1\U0001F3FD',
```

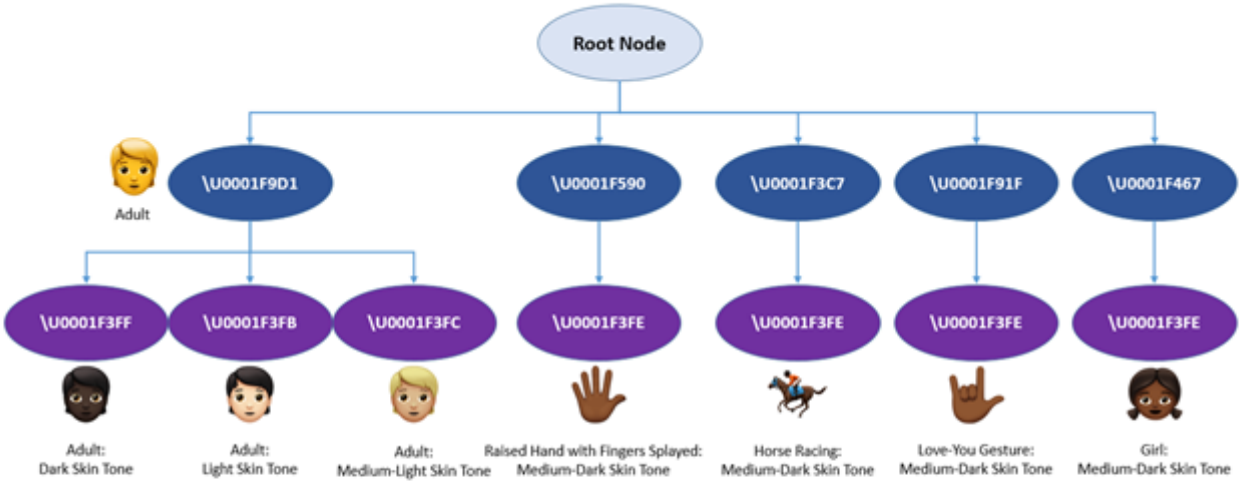
As seen in the structure above, emojis are composed of strings separated by the “\” symbol. For description purposes, we will call these strings “segments”. Similar emojis often contain a similar subset of segments and only differ on one segment. In addition, some segments can be considered “descriptors” rather than object. In the example below, all the emojis described are vastly different objects but have the same characteristic of “medium_dark_skin_tone” represented by the “U0001F3FE” Unicode text compared to the example above is all different types of adult emojis.

```
u':hand_with_fingers_splayed_medium-dark_skin_tone:': u'\U0001F590\U0001F3FE'
u':horse_racing_medium-dark_skin_tone:': u'\U0001F3C7\U0001F3FE'
u':love-you_gesture_medium-dark_skin_tone:': u'\U0001F91F\U0001F3FE'
u':girl_medium-dark_skin_tone:': u'\U0001F467\U0001F3FE'
```

Given the structure of the text representation of the emojis, a tree was constructed in order to organize the emojis. The emoji-tree can defined by the following characteristics:

- There is one root node for the entire tree
- Each node in the tree, but the root, corresponds to a segment
- The first segment of each emoji is a child of the root node
- Any subsequent segment in an emoji’s representation corresponds to another child for the specific segment
- The number of segments for each emoji corresponds to the number of levels in the tree plus one due to the root node

Visual representation of the emoji tree:



Parsing Emojis:

The tree structure coincides with the modularity of the emoji's structure and thus facilitates the parsing of emojis within text. The first step in the process was extracting the Unicode segments from the rest of the message. This was done using a simple pattern recognition script that identified the Unicode segments based on their syntax. Once the segments were extracted, the tree could be traversed segment by segment until the next segment was not a child of the current node. This represented the end of one emoji and the beginning of another. In this event, the current value of the node was deemed to be the emoji used and the process was repeated from the root node until there were no more segments.

Exploratory Data Analysis:

Gender Classification of Users:

Given that Venmo only provides the first and last name of the users, we wanted to predict the gender of the users to determine if there were any differences in how males engaged with the platform vs. females. In order to do so, we simply used a corpus of English names with corresponding gender to make to the first names of the sender. There were also some assumptions that were made in regards to the typical gender of a name which was reflected in a manual change for the specific names. A list of these names can be seen below. Obviously, this does not allow us to predict the gender for every name, but for the purposes of this analysis, it was sufficient.

Initial gender distribution of the origin of the transaction:

```
Male = 28.814917767 %
Female = 37.5691452397 %
Neutral = 20.6541579801 %
Misc. = 12.9617790132 %
```

Gender distribution after manual mapping of below names to genders:

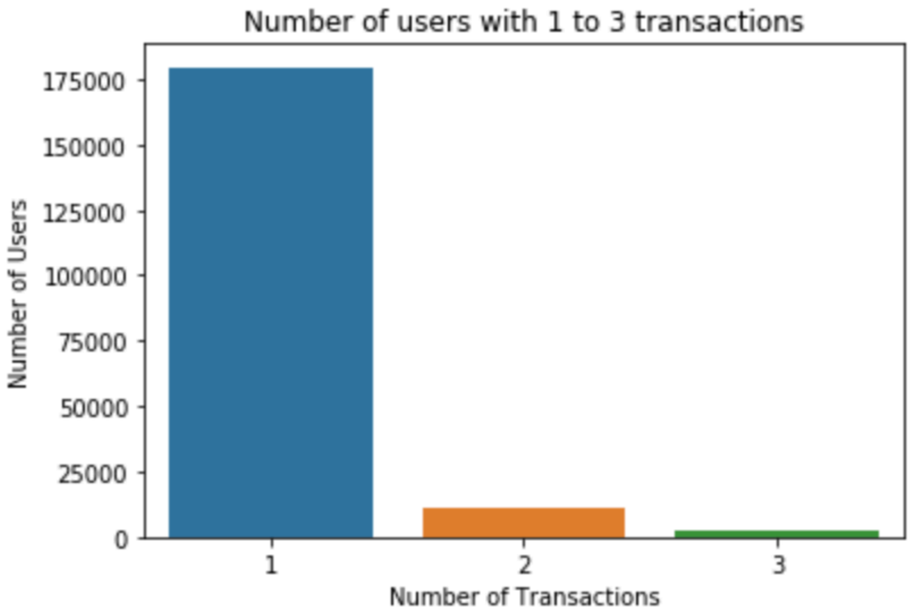
```
setGender("Male", ["Cameron", "Gabriel", "Ryan", "Lucas", "Chris",
                   "Jackson", "Tony", "Cody","Andy","Tyler", "Chase",
                   "Riley", "Danny", "Austin", "Michael", "Kyle",
                   "Sam", "Sean", "Dylan", "Drew", "Christian"])
setGender("Female", ["Mackenzie", "Jess", "Meredith", "Kim", "Madeline",
                     "Shelby", "Bailey", "Tori", "Jackie", "Dana",
                     "Melanie", "Ali", "Lauren", "Jessica", "Ashley",
                     "Michelle", "Brooke", "Kelsey", "Shannon", "Paige",
                     "Olivia", "Erin", "Kelly","Morgan", "Madison",
                     "Courtney", "Sydney"])
```

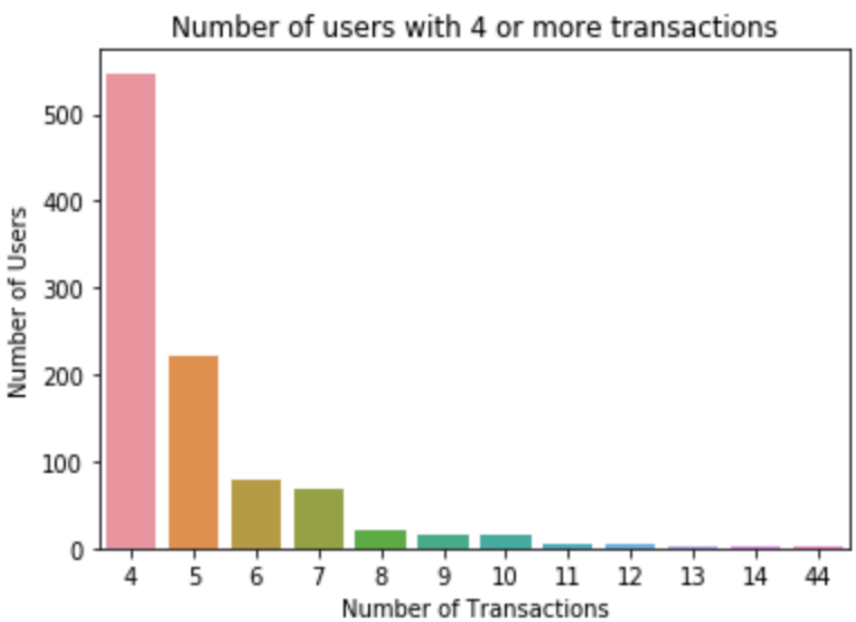
```
Male = 34.3108640259 %
Female = 43.4120917304 %
Neutral = 9.31526523048 %
Misc. = 12.9617790132 %
```

While it cannot be said for sure sure due to names that cannot be mapped to a gender or can be both male or female, the results above highlight that more transactions seem to originate from female users than male. This conclusion could be used to understand who engages more the Venmo platform the most and could help tailor the right service offering to the right demographics.

Furthermore, an analysis of the frequency of transactions for unique users can also highlight how rooted Venmo is into an individual user’s daily lives. Since the data was collected over the span of several days, it would have capture any users who have built Venmo transactions into their daily lives.

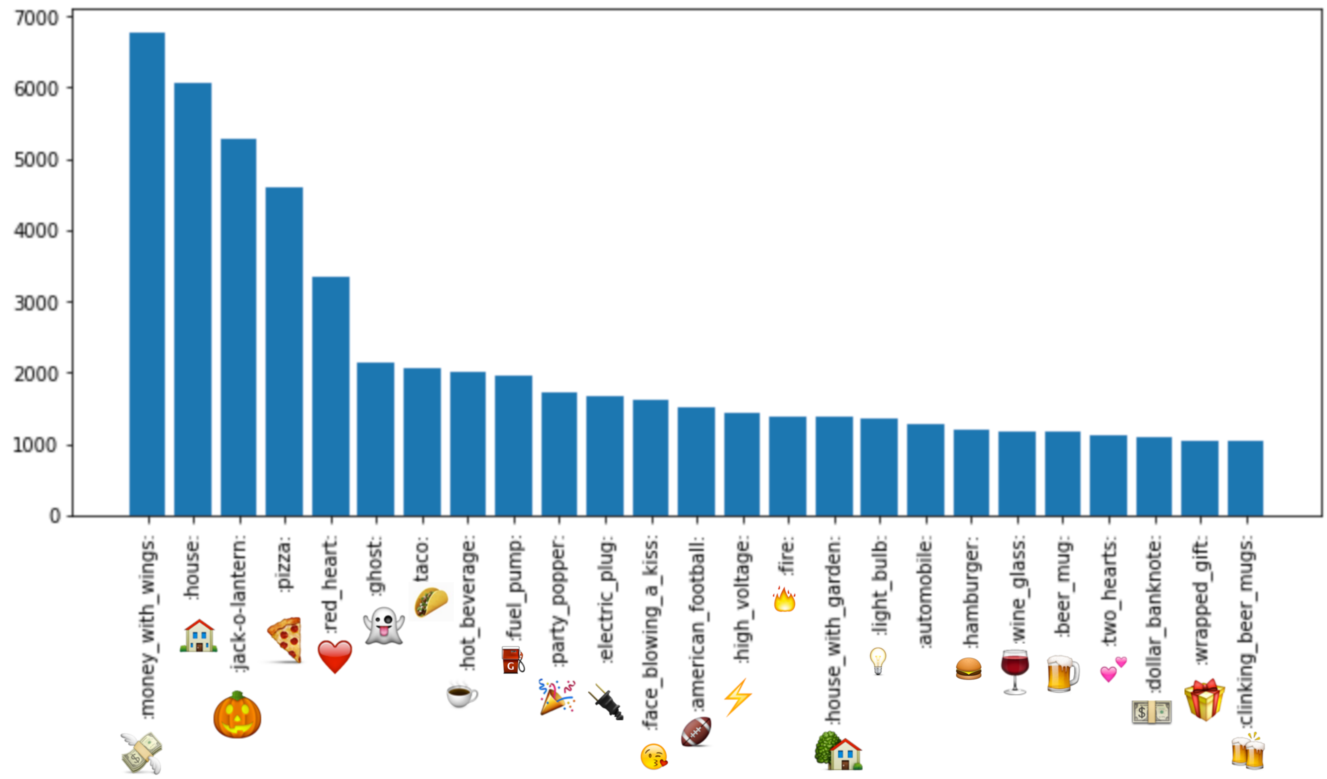
Transaction Frequency based on the concatenation of First and Last Name:





Emoji Analysis:

The analysis of the emojis used in the messages of Venmo transactions can shed additional light on the purpose or description of the transaction. The top 25 emojis used in Venmo transactions can be found below:



The tokenization of the messages that contain emojis enables a better interpretation of the meanings of emojis. The primary example is the relationship between the "Money with Wings" emoji and the "House" emoji to represent a payment of rent. Without the other emoji, it would have been difficult to interpret either emoji independently as a rent payment. This was facilitated through the clustering of the transactions by leveraging Word2Vec and K-means.

Transaction Clustering:

A deeper analysis of the transactions set was needed in order to be able to extract meaningful insights that speak to users' engagement with the Venmo platform. In order to achieve this, the transaction's corresponding message was deemed to be representative of the purpose of the transaction. While a logical assumption, in practice, this does not always hold true as some transaction messages are completely irrelevant or random. This seems to occur given the social aspect of Venmo that allows users to interact with their friends. We would expect these transactions to not accurately align to a cluster as seen below.

In order to cluster like-messages together, the messages needed to be quantified. This was done by creating vectors of the features of each of the messages using the Doc2Vec model. Doc2Vec builds upon the Word2Vec model that quantifies a document, or message, by building a vector based on the words in the body of text. A more detailed explanation of Doc2Vec can be found here.

Preprocessing:

In order to properly tokenize the messages, stop words had to be removed in order to ensure that only meaningful words were included in the vectors. The corpus of English stop words from NLTK was used as the initial set. However, additional slang and informal versions of the original words had to be added to the corpus.

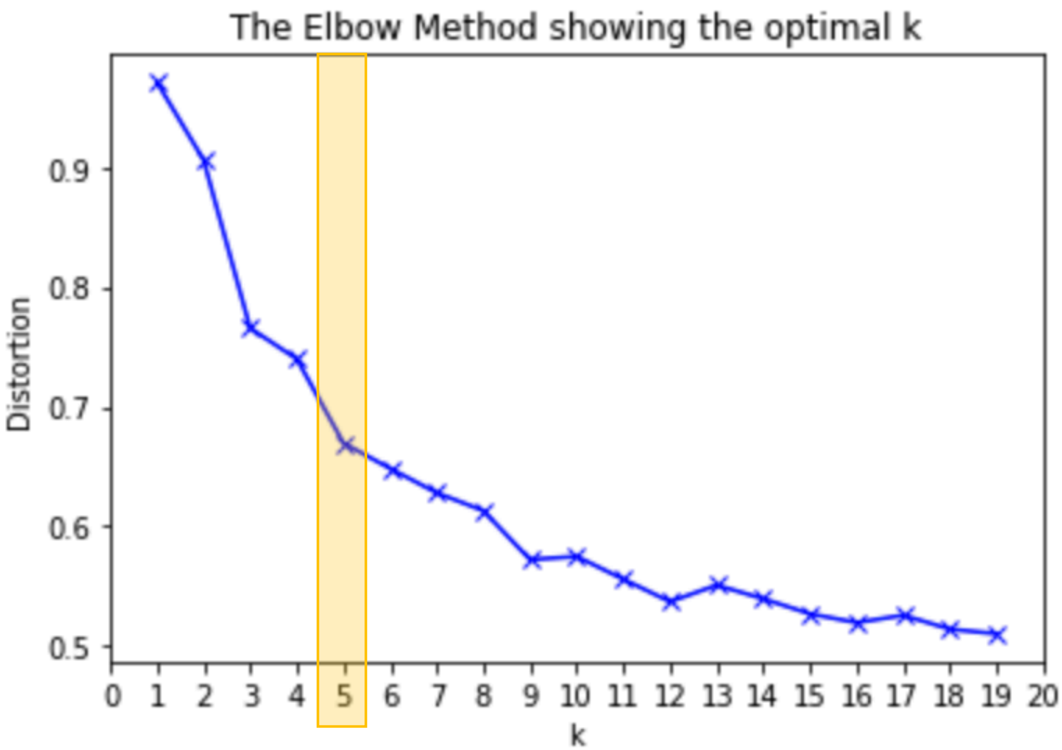
Additions to the corpus of stop words:

```
stop_words.append(["im", "ur", "u", "'s", "n", "...", "z", "n't"])
```

Once the tokens were created and lemmatized, the ones that represented the Unicode text for the emojis were replaced with English word-representation. Colons were used to distinguish an emoji representation from an English word that was written as regular text. For example, "pizza" is interpreted as the English word for the type of food while ":pizza:" represents the emoji for a pizza slice. This is useful when interpreting the results of the most similar "words" for a given entity.

K-means Model for Clustering:

The elbow method was used to determine the optimal number of clusters in our data. The goal of the elbow method is to find the lowers number of clusters that minimizes the distortion value of the model the most. To create the graph, the distortion value was plot against its corresponding number of clusters. As seen in the graph below, the optimal number of clusters for this data set is 5 as it is the smaller number of clusters that minimizes the distortion value the most.



Interpreting the Clusters:

Once we knew the optimal number of clusters in which to organize our transaction, a Latent Dirichlet Allocation (LDA) model was used in order to derive a “topic” or “structure” for each cluster. The word vectors that came out of the Word2Vec model were used as the input to the LDA model and the number of topics was determined by the optimal cluster number from the Elbow graph for K-means models. It is important to keep in mind that the Venmo transaction messages can almost be considered “random” bodies of text (both topically and structurally) when attempting to classify them into a limited number of topics. The results of the LDA model are below and we can see that some topics make more sense than others:

```
Topic 0 has a 7.3 % chance of being about ":house:"
Topic 1 has a 1.9 % chance of being about ":fuel_pump:"
Topic 2 has a 4.6 % chance of being about "food"
Topic 3 has a 8.8 % chance of being about ":jack-o-lantern:"
Topic 4 has a 9.1 % chance of being about ":pizza:"
```

Most-Similar Results:

Below are the top 5 most similar words to the entities that represent our categories that were described before. In addition, there are the top 10 similar words to the top 10 most used emojis in the transactions.

```
categories = [":house:", ":fuel_pump:", "food",":jack-o-lantern:",":pizza:",":red_heart:"]
```

```
Emoji - :pizza::
Top 10 Similar Words:
EMOJI: :spaghetti: --- 0.935360372066
Word: za --- 0.902567982674
EMOJI: :Italy: --- 0.872519850731
EMOJI: :night_with_stars: --- 0.819901227951
EMOJI: :turtle: --- 0.818804621696
EMOJI: :video_game: --- 0.810384273529
EMOJI: :fork_and_knife: --- 0.807278513908
Word: pasta --- 0.801966786385
EMOJI: :backhand_index_pointing_left: --- 0.801196396351
EMOJI: :drooling_face: --- 0.799467086792


Emoji - :red_heart::
Top 10 Similar Words:
EMOJI: :raising_hands: --- 0.849886000156
EMOJI: :smiling_face_with_smiling_eyes: --- 0.822620749474
EMOJI: :kiss_mark: --- 0.821949005127
EMOJI: :blue_heart: --- 0.820443153381
Word: thank --- 0.808254718781
EMOJI: :growing_heart: --- 0.806346178055
EMOJI: :hugging_face: --- 0.795048177242
EMOJI: :smiling_face: --- 0.792094409466
EMOJI: :folded_hands_medium-light_skin_tone: --- 0.788210391998
EMOJI: :smiling_face_with_halo: --- 0.781944274902


Emoji - :house::
Top 10 Similar Words:
EMOJI: :laptop_computer: --- 0.933083832264
EMOJI: :television: --- 0.930178940296
EMOJI: :mobile_phone: --- 0.93001973629
EMOJI: :houses: --- 0.921006083488
EMOJI: :telephone_receiver: --- 0.920080542564
EMOJI: :office_building: --- 0.918852090836
EMOJI: :mobile_phone_with_arrow: --- 0.914995312691
EMOJI: :desktop_computer: --- 0.906780958176
EMOJI: :heavy_dollar_sign: --- 0.906500101089
EMOJI: :satellite_antenna: --- 0.899487376213
```

```
Emoji - :fuel_pump::
Top 10 Similar Words:
EMOJI: :dashing_away: --- 0.912296652794
EMOJI: :heavy_dollar_sign: --- 0.849217772484
EMOJI: :sport_utility_vehicle: --- 0.841816604137
EMOJI: :P_button: --- 0.839294910431
EMOJI: :toilet: --- 0.825160980225
EMOJI: :office_building: --- 0.823118209839
EMOJI: :oncoming_automobile: --- 0.82271873951
EMOJI: :heavy_minus_sign: --- 0.821644663811
EMOJI: :automobile: --- 0.81920337677
EMOJI: :droplet: --- 0.815144538879


Emoji - :jack-o-lantern::
Top 10 Similar Words:
EMOJI: :ghost: --- 0.974654436111
EMOJI: :clown_face: --- 0.967918276787
EMOJI: :goblin: --- 0.965528190136
EMOJI: :man_zombie: --- 0.96545368433
EMOJI: :skull: --- 0.965453445911
EMOJI: :smiling_face_with_horns: --- 0.963592290878
EMOJI: :skull_and_crossbones: --- 0.963282942772
EMOJI: :kitchen_knife: --- 0.961363434792
EMOJI: :face_screaming_in_fear: --- 0.949325025082
EMOJI: :woman_zombie: --- 0.949137091637
```

Closing Remarks:

The topics of discussion in this analysis highlighted many underlying trends in how users engage with the Venmo platform. Within our dataset, we were able to derive insights into the composition of the user base, highlight trends in the frequency and types of transactions between users, and identify the top 50 emojis used in the transaction messages.

Furthermore, we were able to cluster like-transactions based on the set of features that characterize a transaction.

However, the inability to access historical transactions is this work's main limitation. As is the case for many studies, the inability to apply this analysis on a wider scope of data prevents us from understanding how the trends we identified changed over time. We are only able to comment on what kind of users engage with Venmo and how for a certain period of time. Understanding how these trends change over time could help explain how user preferences and habits evolve and help set the direction for the platform moving forward. Overall, the experience of this analysis showcased that valuable insights can be derived from user interaction data, but reinforced the importance of access to historical data to understand the evolution of user engagement.