Group 62 - Progress Report

Neill Killgore, Rajiv Kamal, Eissa Qader, Kha Tran, Bryce (Bo) Meyering

4/24/2022

Introduction

Under the STOCK (Stop Trading on Congressional Knowledge) act of 2012, senior US government officials have to report any stock, bond, or commodities transaction within 45 days of the transaction, provided that the total transaction amount is greater than $1000. Recently, news of insider trading and violations of the STOCK Act by congressional members has made many headlines in US national news [1]. This type of behavior is troubling due to the timely and, many times, highly sensitive information they are exposed to on a regular basis. Though the purpose of the STOCK act is to prevent our representatives from acting on this information and gaining a distinct advantage in the market, the penalties for not complying are minimal and ineffective (a fee of $200 USD for each transaction reported late). While proving the intent behind a congressional member's trades is a difficult task since we do not know what they know, we thought it would be worthwhile to analyze representatives' trading behavior and transaction timing before major moves in either the market or individual securities they hold.

While there are some methods for detecting insider trading that have been proposed in recent articles[2][3][4], most rely on released litigation records for transactions that could form the basis of training data for the models they employed. While many congressional members have been implicated in recent allegations of insider trading, there are many legal obstacles to proving the 'materiality' of any information they receive.[5]. Thus, no members have yet been charged with any infractions of the law regarding trading securities based on non-public information. To make matters worse, there are several other obstacles to identifying suspicious trading activity in Congress. First, congressional members only have to report their data in discrete bins of the dollar amount of the purchase or sale, not in the number of shares they traded. So the starting data is inherently imprecise. In addition, persons engaged in insider trades are more likely to trade in options rather than stocks due to the potential payouts involved[6], congressmen are not required to report whether the transactions were options or stocks.

Main Objectives

For this project then, we have identified three main objectives that will form the basis of our analyses.

[1] see (https://www.theatlantic.com/politics/archive/2022/01/congress-stock-trading-ban/621402/) for more reading

[2] Identification of Insider Trading Using Extreme Gradient Boosting and Multi-Objective Optimization. Shangkun Deng et al. Information 2019, 10, 367; doi:10.3390/info10120367

[3] A Deep Learning Based Illegal Insider-Trading Detection and Prediction Technique in Stock Market. Sheikh Rabiul Islam. ArXiv, 2018, abs/1807.00939

[4] How to detect illegal corporate insider trading? A data mining approach for detecting suspicious insider transactions. M. Fevsi Esen et al. Intelligent Systems in Accounting, Finance, and Management. Vol 26, Issue 2, 2019

[5] How Senators May Have Avoided Insider Trading Charges. Robert Anello. Forbes https://www.forbes.com/sites/insider/2020/05/26/how-senators-may-have-avoided-insider-trading-charges/?sh=504a1bfb27ba

[6] Early Detection of Insider Trading in Option Markets. Steve Donoho. KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data miningAugust 2004 Pages 420–429https://doi.org/10.1145/1014052.1014100

- Thus, our first and main research goal is to leverage publicly available daily stock price data and Congressional stock transaction disclosures to identify 'suspicious' trades within Congress. We plan to investigate any correlations between congressional members' trades and price movement in the market following the trades that congressional members make.
- Our Second research goal for the project is to identify the best and worst traders within each respective House of Congress or political party by computing their individual returns and portfolio *alphas* compared to broader market indices.
- Our final goal is to identify any distinct clusters of congressional members based on similar trading patterns, positions held, cumulative returns, trading dates, etc. We will try to find the variables that are most associated with each cluster to see if any congressional members that have suspicious trades cluster together or are distinct from those members with 'typical' portfolios. In general, we will try to identify the data characteristics of outliers in the clustering analysis.

Problem Statement

To state our problem more succinctly:

Recent allegations of congressional insider trading have revealed the inherent difficulties of proving the materiality of any non-public information. Current methods for insider trading detection rely on labeled data and/or litigation records, records that do not exist for members of Congress who trade securities. We propose to use public data to calculate portfolio metrics for each congressional member and identify outlier trading activity by clustering in order to flag trades for either suspicious timing or amounts.

Dataset Description

We have identified three separate datasets that we plan to leverage in addressing our research questions.

- The first dataset is a live database containing all of the disclosed stock purchases/sales of all US Congress members. We have accessed this database through a Python API at Quiver Quant for a small fee. Pursuant to the Code of Federal Regulations, government officials are only required to report the value of a sale or purchase within 11 distinct bins and what investment they sold/purchased, but not the quantity or exact price. This leaves some ambiguity in the numerical side of the data that we will have to address by making some broad, less-than-ideal assumptions. Our dataset currently has coverage from 2016 until the present for all congressional members who disclosed any transactions.
- In order to supplement the above dataset, we have also pulled daily stock data for all of the unique ticker symbols found in our congressional trading data. This data was pulled from Yahoo Finance using the R package `tidyquant` starting from January 1, 2016, until the present date. We plan to focus most on the `adjusted_close` variable in order to calculate returns.
- Finally, we used curl script GET-requests to obtain metadata for each congressional member using the ProPublica API. This included a variety of information such as Party, gender, congressional committees, etc. We mainly used this data to quickly obtain all of the Party data for each of the 167 unique members in our trading dataset.

The first two datasets are in tidy formats with self-explanatory variable names. They are listed below.

Congressional Trading Data

- **Report Date:** Date that the transaction was disclosed under the STOCK act.
- **Transaction Date:** Date of the transaction
- **Ticker:** Ticker symbol of the transaction
- **Representative:** Name of the Congress member
- **Transaction:** Type of transaction, ‘Sale’ or ‘Purchase’
- **Amount:** Lower bound of the transaction range
- **House:** Congressional body that the representative is a member of, ‘Representatives’ or ‘Senate’
- **Range:** The range of the transaction value

Here is a glimpse at a few random rows of the congressional trading dataset (only select columns for brevity’s sake):

TransactionDate	Ticker	Representative	Transaction	Range
2020-05-13 00:00:00	ATVI	Gilbert Cisneros	Sale	$1,001-$15,000
2020-04-14 00:00:00	ANTM	David Perdue	Sale	$15,001 - $50,000
2020-01-22 00:00:00	NXPI	Lois Frankel	Purchase	$1,001-$15,000
2021-05-28 00:00:00	QCOM	Earl Blumenauer	Sale	$1,001-$15,000

Historical Market Data

- **Symbol:** Stock ticker symbol
- **Date:** Date of the observation
- **Open:** Opening price
- **High:** High price
- **Low:** Low price
- **Close:** Closing price
- **Volume:** The number of shares traded
- **Adjusted:** Closing price after accounting for major corporate actions

symbol	date	open	high	low	close	volume	adjusted
AAPL	2019-08-08	50.0500	50.8825	49.8475	50.8575	108038000	49.71067
AAPL	2021-06-11	126.5300	127.4400	126.1000	127.3500	53522400	126.81260
AAPL	2019-07-12	50.6125	51.0000	50.5500	50.8250	70380800	49.67890
AAPL	2019-02-13	42.8475	43.1200	42.4800	42.5450	89960800	41.42608

The data from ProPublica was a nested JSON file. We used Python to grab the data from the curl output and export it to .csv format to upload in R.

Congressional Member Data

- **Party** (This is the main variable we wanted, though many others were present in the database)

Approach and Methodology

Our approach to analyzing and modeling the data has several parts, which are discussed in the sections below. In general, we followed the following schema: Data Cleaning -> EDA -> Clustering -> Calculating Member Returns -> Identifying Suspicious Trades

Data Cleaning and Merging

There were inconsistencies in spelling that needed to be amended when joining the trading data to the Congressional member data. Around a third of the members either had some misspelling in their names or several versions of their names were recorded in the datasets. These were corrected through regular expression extraction and or exact string matching. There were also a few issues, such as a reporting date and transaction date that got switched or transactions that were associated with two different reporting dates. We also created a few more variables in the dataset, such as `report_lag` (`report_date-transaction_date`), `overdue`, which is the number of days past the 45-day limit that the transaction was reported, and a binary variable, `DNR` (Did not report) for any transaction that was not reported.

Exploratory Data Analysis

Once we cleaned the data, we first performed some exploratory data analysis to get a general idea of the structure of the data. We created interactive data visualizations using the `plotly` library to look at summaries for the most commonly traded stocks, the stocks with the highest transaction amounts, the congressional members with the highest and lowest reporting lags, and the relationships between all of these variables. We then plotted out the total number of transactions per day and the total transaction amount per day (by transaction type, `sale` or `purchase`) as a function of time. Finally, we looked at many of these key variables on a Party basis to see if there were any large differences between Democrats and Republicans. The results from these will be discussed in the results section below.

Member Clustering

Using the congressional trades data, we then created several sets of sparse matrices to use in cluster analysis. We decided to create matrices in two basic categories: All unique ticker symbols and transaction dates. Scaling methods were performed on five of the six matrices we created. Generally, if the matrix only contained binary data, the matrix was left unscaled so as to preserve sparsity. However, if the matrix consisted of continuous data for the non-zero elements, then only those non-zero elements in each column were centered and scaled to $\sigma = 1$ so that the mean of each column was a near-zero value.

Focusing on the first category of matrices using ticker symbols, we created three sparse matrices to cluster on. The first matrix, `A1`, was a basic matrix with the portfolio history for each stock for each member. If a member had held any security regardless of the amount or the duration, then we recorded a `1` for that stock and a `0` otherwise. The second matrix, `A2`, was calculated using the number of transactions each member made in a given security, both purchases, and sales, as a proportion of the total number of transactions they made. The columns of `A2` were scaled as described above. Matrix `A3` was created using the sum total of transaction amounts for each security as a proportion of the sum total of all transactions that member made, followed by non-zero element scaling. Each of these mxn matrices was 167x818 (number of members X number of unique tickers represented in the data).

For the second category we created, we pulled the transaction date history for each member and calculated three matrices `A4`, `A5`, and `A6` for the total transaction amount per day, transaction count proportion per day, and transaction amount proportion per day for each member, respectively. The matrices were filtered to remove all zero columns, i.e., any dates that had no transactions. The columns of all these matrices were scaled as described to retain sparsity.

We then worked on reducing data dimensionality by several different PCA algorithms for high-dimensional sparse data. Specifically, we used a few of the methods in the `sparsepca` package but found them to be computationally slow when solving for more than 20 components while not returning scores any different from other methods. In comparison, we were able to use the `sparsesvd` algorithm from the like-named package to quickly compute the truncated sparse SVD components. We used this to compute the first 20 principal components for each of the sparse matrices to use for clustering. We then implemented a spherical k-means algorithm from the `skmeans` package and also the DBSCAN clustering algorithm from the `dbscan` package. For the k-means clustering, we examined scree plots to choose an appropriate number of clusters by finding a natural elbow point in the graph.

After looking at the structure of the data, we then hypothesized that there would be one main cluster of *normal* members who traded very little or in a similar fashion to each other and N outlier points that were trading differently than other people. DBSCAN is well suited to this since it is a density-based approach that returns cluster membership for dense clusters and membership in a general noise group, i.e., the outlier points. We first calculated the k-nearest neighbors distance matrices for each matrix setting k equal to n of the matrix dimensions. After ordering the points, we identified an appropriate cutoff value for the ϵ parameter of the algorithm. After identifying `eps`, we ran the algorithm setting the minimum number of points to $n+1$, 21 since we computed 20 principal components.

We also tried using an extension of DBSCAN, the OPTICS method, for setting `eps` for the DBSCAN algorithm on the same PCA data, using the reachability plots to select a proper `eps` threshold and then extracting the clustering information with `extractDBSCAN`.

Finally, we changed the sparse matrices so that the columns were scaled but not zero centered to ensure that all values were greater than or equal to zero. We then constructed dissimilarity matrices based on non-linear distances metrics such as Bray-Curtis or Kulczyinski and performed ISOMAP, which is a non-linear method that calculates the distance between points.

In addition to these methods used above, we computed the cosine similarity between all members and also from each member within a given cluster against members of other clusters as a quality check to ensure that the cluster membership did indeed have more similarity to each other than out-of-cluster members. This approach yielded meaningful results when used with the k-means algorithm since the number of clusters generated was higher than DBSCAN, and the clusters were formed based on Euclidean distances. However, when applied to the DBSCAN results, both the average cosine similarity of members within a cluster and the average cosine similarity of cluster members to noise members was generally low since the main group contained much higher variability than the k-means clusters. In addition to this, we expected to see that the noise points had low similarity to each other and members of the dense cluster, which proved to be the case.

Member Returns

Assumptions

Due to the incomplete nature of the data, we had to make some assumptions in order to calculate returns for each member.

- **No member ever has a negative balance of either stocks or cash in their accounts.** In order to accomplish this, we used trades and price data to calculate the ending balance of each member as if they had started with $0 and zero shares. For negative values (e.g., -500 shares AAPL), we made the assumption that the congress member started the trading period with enough of the asset to be able to sell it. In order to adjust the data, the inverse of the minimum balance of each asset, if negative, was added to each member's starting balance.
- **All executed trades occur at the lower bound of the specified range.** The data did not specify the exact number of shares or value of the trade. For consistency between all traded and members, the `Amount` column of the congressional trading data was used to calculate the number of shares purchased or sold.

Calculation Method

To optimize the matrix operations required, we transformed the prices table into a data structure with each column representing a stock ticker and each row representing a date. The cells contained the value from the adjusted_close column of the original prices table. We added a signed_amount column to the congressional trades dataframe to represent sell orders with a negative number and buy orders with a positive number.

For each member in the dataset, the following operations were performed:

1. Get a list of all stocks traded by the member and first/last trades dates from the congressional trades table.
2. Create new dataframes member_prices and member_orders containing only data for the relevant member, dates, and stock tickers. Add CASH column to member_prices to track cash balance.
3. Copy the member_prices dataframe to a new member_trades dataframe. Replace all values with 0.
4. For each trade in the member_orders dataframe:
   - Calculate the number of shares using the signed_amount column and insert it into the appropriate row (date) and column (ticker) of member_trades.
   - Insert the inverse of the cash value from 4a into the CASH column.
5. Create a holdings dataframe by calculating the cumulative sum of each column of the member_trades dataframe. The holdings dataframe tracks the daily amount of each asset that each member owns. (e.g. 5 shares NFLX, 2 shares AAPL, $1000 cash).
6. Create a holdings_dollars dataframe by multiplying the member_prices dataframe by the holdings dataframe to calculate the daily cash value of each asset in the portfolio.
7. Create a portfolio_values dataframe by summing across each row to calculate the daily total cash value of the portfolio.

Once the daily portfolio cash value was calculated for each member, the tidyquant and PerformanceAnalytics libraries were used to calculate daily returns and alphas.

Identifying Suspicious Trades

The article by M.Fevsi Esen *et al.* suggests capturing returns after a set window of time for each trade. We calculated the return of each trade and the S&P500 index after 30 days. From this data, we can calculate how each trade performed in relation to the overall market. We can then use a clustering algorithm as suggested by M.Fevsi Esen *et al.* to identify suspicious trades. DBSCAN is a useful clustering technique for this since it labels outliers as "noise".

Results

Exploratory Data Analysis

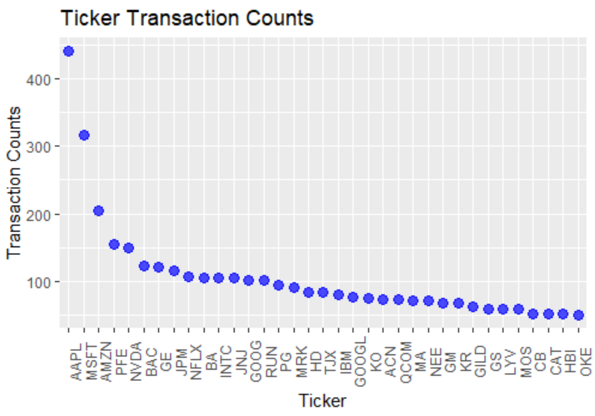

Figure 1: Transaction counts for the most commonly traded stocks

After cleaning the initial database of trades, we performed some general exploratory analysis. In regards to the transactions that members made, the vast majority occurred in just a handful of securities, such as Apple, Microsoft, and Amazon, with the data roughly following a Pareto distribution (see figure 1). There were over 441 transactions (including purchases and sales) in AAPL, followed by 317 in MSFT and 205 in AMZN. Contrast this with the mean transaction count for each stock of 12.5 transactions. We also noticed a similar trend for the lag in transaction reporting for each stock - the vast majority of transactions for each security were reported on time or shortly after the deadline (82.8%), but a few transactions were extremely late. In fact, only a few unique securities accounted for the highest average reporting lags for transactions (figure p3 supplementary material). We thought that there might be a relationship between the average reporting lag for a security and how frequently it was traded in our datasets, but while those securities with extremely high average reporting lags were on average traded less frequently, the trend was far from linear (figure 2). The relationship improved little after log transformation.

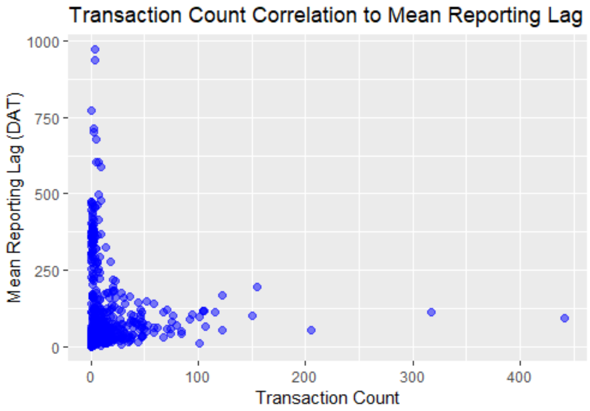

Figure 2: Correlation between transaction counts and average reporting lag for all ticker symbols.

From this, we can conclude that the majority of members are trading in a small number of popular stocks and the transaction counts for all stocks are far from being uniformly distributed. In addition to this, it seems as though the vast majority of transactions are reported on time, but those which are reported very late are less likely to be frequently traded stocks.

Moving on with this insight, then, we should look at the data from a congressional member basis. We might hypothesize several things. First, we might think that members who are making suspicious trades would be less likely to report transactions on time or report them at all. Let's first take a brief look at those members who did not report transactions in the table below. Only three members had transactions that were not reported. Diane Feinstein, Richard Burr, and Richard Blumenthal. In our data, none of these members have ever reported any of their transactions. While both Richards were moving around relatively smaller amounts of money, Dianne's undisclosed purchases and sales totaled over 2.5 million dollars.

Table 3: Transaction counts and sum amounts of members' DNR transactions

Representative	transaction_count	transaction_amount
Dianne Feinstein	12	2696012
Richard Blumenthal	5	565005
Richard Burr	28	294028

Let's also filter the data for those members who reported transactions after the due date, calculate the average reporting lag for their late transactions, and count the total number as well as find the sum of the late transaction amounts. We'll only take a look at the top ten members sorted by mean reporting lag in descending order.

Table 4: Top ten highest average reporting lags for delinquent transactions by member

Representative	transaction_count	transaction_amount	lag
Thomas Suozzi	203	2346203	668.1675
Cynthia Axne	71	71071	574.7746
David Perdue	200	613200	547.8700
Ted Cruz	1	15001	535.0000
Cheri Bustos	3	45003	501.6667
Daniel Meuser	28	56028	479.8214
Dave Joyce	2	16002	472.5000
Bill Cassidy	19	19019	463.7368
Tim Kaine	1	1001	455.0000
Tom Malinowski	120	1002120	454.6333

Both Thomas Suozzi (D-NY) and David Perdue (R-GA) are highly delinquent in a lot of transactions, 203 and 200 for Thomas and David, respectively. Tom Malinowski (D-NJ) was also highly delinquent in a large number of transactions (average of 455 DAT, 120 transactions).

On average, though, almost all of the transactions made are reported in a timely manner, as shown in figure 3. The dotted line represents the log of the 45 DAT reporting deadline. You can see that the vast amount of the data lies well underneath the cutoff. This is true regardless of the Party or the House of Congress that the member serves in. And though we see that some members who trade frequently have very high average reporting lags, this is not true across the board. In fact, the incidence of highly delinquent reporting decreases drastically as a member begins to trade more frequently though this is a non-linear relationship (figure p3, supplementary material)

Finally, we looked at the data over time to see if there were any abnormal spikes

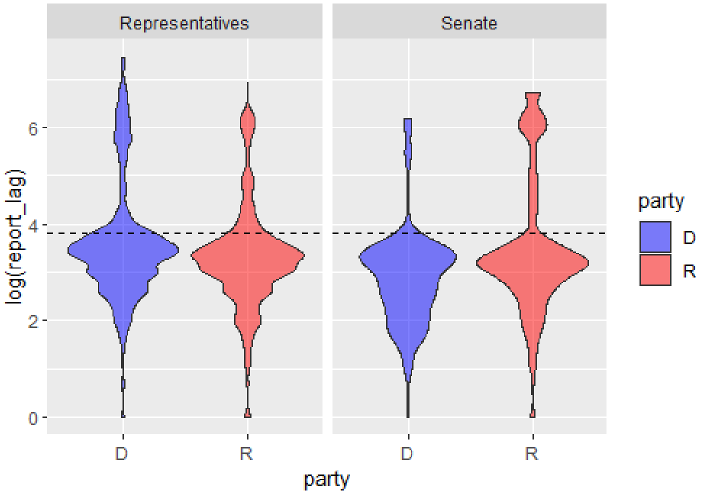

Figure 3: Reporting lag for all members by party and house of Congress. The dashed line represents the log of the 45 day deadline. Data was log scaled to better show the distribution.

in the number of transactions that occurred or the amount of money that was exchanged on a given day. The figure for the daily transactional amount, both purchases and sales, is shown in figure 4.

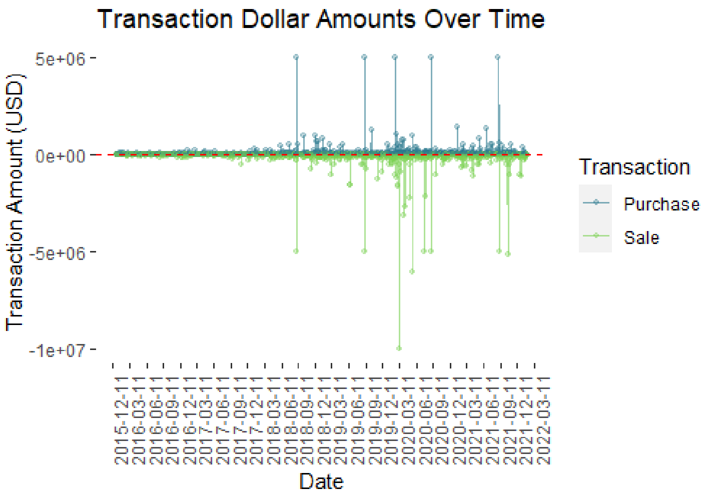

Figure 4: Total transactional amount of purchases and sales by date

We identified a series of 5 'spikes' in both purchases and sales that warranted a detailed look. These high dollar transactions belonged solely to Suzan Delbene (D-WA), who was trading in MSFT at amounts greater than $5 million, making purchases and sales within a few days of each other. While the market was not moving greatly during these periods, we still find it odd that she made these transactions.

These are particularly interesting since her husband, Kurt Delbene, was an Executive Vice President at Microsoft before stepping into his current role as Assistant Secretary of Information and Technology in the Department of Veteran's Affairs, and MSFT is the only ticker she has traded in beside one other sale of MN for $100k. This will have to be examined in greater detail later. We constructed many other plots in the EDA section that are available in the supplementary materials but could not be included in the final report for the sake of brevity.

Clustering

The sparse matrices we constructed from the trading data were decomposed by sparse SVD into their U, Σ, V components, and the first 20 components were used for clustering standardized for all the decompositions we made. Sparse SVD was both faster and more interpretable than some of the methods from the `sparsepca` package, such as the robust sparse PCA algorithm. In addition, since many of those algorithms rely on random projections, our results differed slightly every time we ran the functions. These methods also tended to be much slower when we were computing more than 20 components. When we plotted the first two components for each of the matrices, we observed a large cluster that contained the majority of members with some outlier points. With the components for matrices `A4`, `A5`, and `A6`, we observed an extremely dense center cluster with a few outliers.

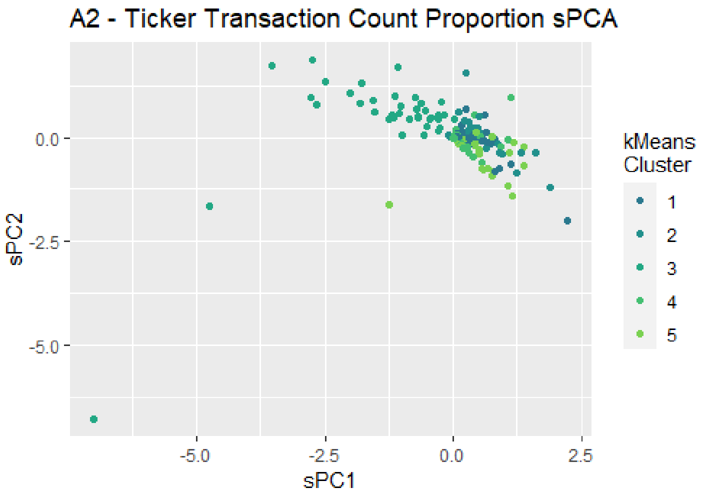

Figure 5: First two principal components of A2 with clusters from skmeans

The first clustering algorithm we tried was spherical k-means which scales and normalizes all of the data vectors and uses their cosine similarity as a metric to compute clusters. This is particularly helpful in situations with high dimensional data or when dealing with sparse matrices. We selected the `k` parameter from the elbow point on the scree plot, though in practice, the *within-cluster sum of squares* decreased rather gradually when we increased `k`, making selection difficult. When we used traditional k-means, it returned rather naive groupings of the data, which were much improved with the spherical k-means algorithm (See figure 5).

However, both methods generated clusters that were not useful in further analysis when trying to identify outliers, as these outliers might still have a high degree of cosine similarity to other data observations near the center cluster if they both pointed in the same direction. This prompted us to explore other methods.

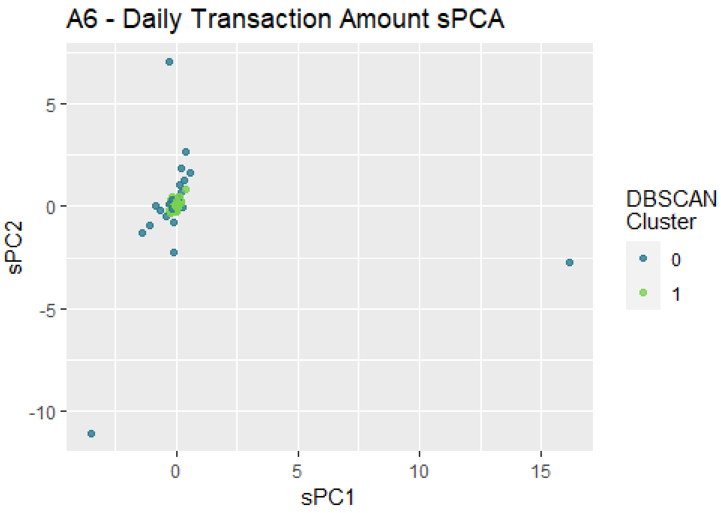

Figure 6: First two principal components of A6 with clusters from DBSCAN. '0' contains 'noise' points and '1' contains the dense cluster points.

We then used DBSCAN to generate clusters. We believe that this method yielded the most informative results for us since it creates clusters on a density basis instead of identifying an arbitrary set number of clusters. From the data, we know that most members trade very little and trade in a small number of common stocks; thus, we had an underlying assumption that the vast majority of members have similar trading portfolios and thus should cluster together (figure 6). Whereas k-means generates labels based on the centroids of the clusters, DBSCAN throws all of the non-clustered points into a 'Noise' category, which is perfect for identifying outlier traders. The DBSCAN algorithm takes a few parameters that need to be selected carefully, `eps` and `minPts`. One common way to select `eps` is to construct a k nearest neighbors distance matrix, order the points from smallest to largest, and select `eps` as the point at which the kNN distance drastically increases, indicating the points are spreading far away from the center of the cluster (figure 7). `minPts` was selected using the ndim(predictors) + 1.

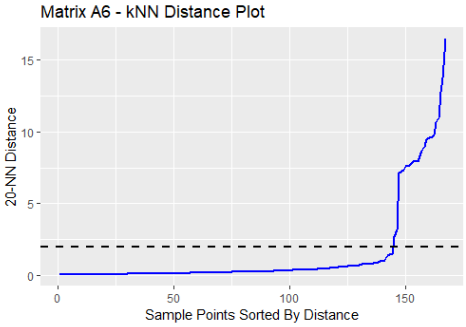

Figure 7: kNN distance plot to select optimal 'eps' threshold parameter for DBSCAN algorithm. K was set to the number of features in the data, 20 in this example

Principal components of the matrices constructed from member portfolio data `A1 - A3` yielded broad distributions with more noise points when clustered with DBSCAN; however, when we used the transactional time series data from each member to construct sparse matrices, DBSCAN was able to identify a single dense cluster of traders with just a handful of wild outliers. We used DBSCAN to calculate cluster membership on the first 20 principal components for each matrix, and then we cross-tabulated the members in the 'noise' category from each of the DBSCAN cluster results to see if there were any common outliers among the different ways of looking at the data.

In total, we flagged 33 unique outlier members across all of the datasets. The following list shows how many outliers were flagged for each matrix: `A1`- 13, `A2`- 8, `A3`- 7, `A4`- 12, `A5`- 17, `A6`- 23. Just over half of the outlier members were only flagged in only one dataset; however, 16 members were flagged in 2 or more. Four members, Gilbert Cisneros (D-CA), Greg Gianforte (R-MT), Josh Gottheimer (D-NJ), and Susie Lee (D-NV), showed up in the noise group in every matrix when we computed cluster membership with DBSCAN. David Perdue (R-GA) and Dean Phillips (D-MN) appeared as outlier members in 5 of 6 datasets.

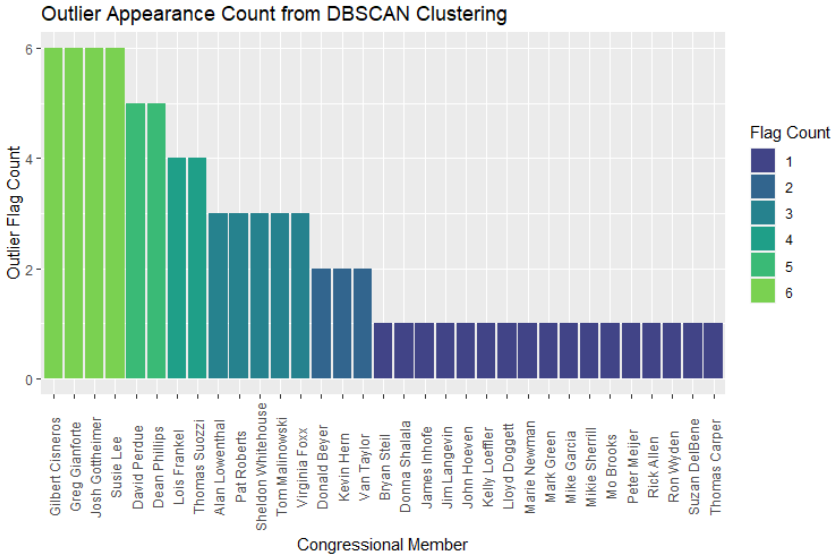

Figure 8: kmeans plot

A barchart showing the top outlier members and the total number of times that they were flagged as an outlier point is shown the figure 8 on the left. Euclidean dissimilarities were computed for each set of members and then ordered from most dissimilar to least dissimilar. We can see that most of the same members who show up as outliers in the DBSCAN clustering are also some of the most dissimilar data points when compared across datasets. Notice, though, that while some members such as Gilbert Cisneros are consistently dissimilar across all of the matrices, others such as David Perdue are only highly dissimilar in one set of matrices, namely

`A4-A6`, which looked at the timing of transactions and not in `A1-A3` which suggests he was trading in the same securities as most other people, but trading at different times than other people.

Table 5: Top ten dissimilar members. Values are the mean Euclidean dissimilarity scores for members.

member	A1	A2	A3	A4	A5	A6
David Perdue	8.829295	7.711450	6.759671	16.580064	16.966494	16.937476
Josh Gottheimer	12.083197	9.584328	8.554176	13.257268	14.573236	13.463439
Gilbert Cisneros	13.975176	11.161681	10.807688	10.151232	10.008839	10.568575
Greg Gianforte	10.834906	7.630704	7.778039	8.982076	8.227577	14.364808
Susie Lee	11.199986	8.131528	7.331251	9.654420	10.002906	10.452940
Dean Phillips	11.267278	8.823987	8.094791	7.674649	8.159097	8.255221
Lois Frankel	9.095190	6.139844	6.382141	9.474559	9.673832	9.980988
Thomas Suozzi	8.566922	6.464482	6.212569	8.845121	8.400582	10.489868
Sheldon Whitehouse	8.324590	6.923983	6.633463	8.821187	9.067000	8.961893
Alan Lowenthal	7.077798	5.118625	6.391517	8.825294	8.589364	9.229088

By creating sparse matrices that capture specific traits of each trading member, whether the securities they trade in, the dates they trade, or the amounts of the transactions they are making, and then reducing the dimensionality of these datasets with sparsesvd, we were able to use DBSCAN to identify outliers in each matrix. By themselves, these results don't point toward any wrongdoing but can be used as a starting point to identify congressional members whose trading patterns lie outside of the norm within Congress. We will use the outliers discovered in this section as a starting point to pin down members with any abnormal returns or abnormal trade timing in the next section.

Member and Trade Returns

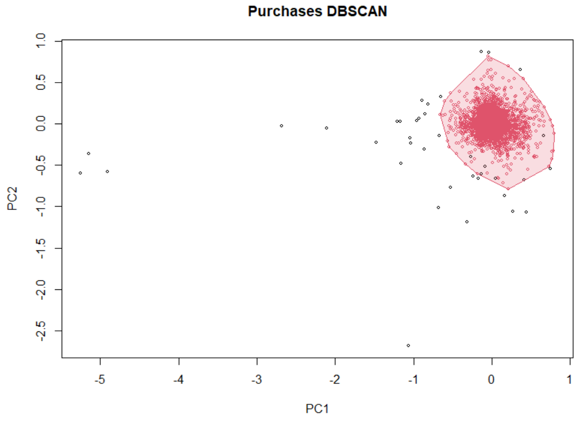

Figure 9: Hull Plot for DBSCAN clustering performed on the member purchase returns data

Portfolio returns and portfolio alphas for all members in the dataset seem to indicate that congresspeople do not substantially outperform the market. The largest alphas show the member outperforming the market by single digits over the course of the entire analysis period, and many members under-performed the market. Further investigation is needed to determine whether portfolio returns are a valid predictor of insider trading; however, initial results seem to indicate that they are not.

Thirty-day returns from individual trades are far more diverse, with some members, in a single trade beating the market by more than 100%. The mean and median of the thirty-day returns for all purchases are both near 3%, which is double the thirty-day market return and may indicate that a significant number of trades made by Congress are far more profitable than would otherwise be expected. In order to evaluate the trades based on the previous market and stock performance, we also calculated returns for both the stock and the market for the 30 days immediately preceding each trade.

Using all four of the 30-day calculated return measurements (30-day stock returns, 30-day market returns, previous 30-day stock returns, and previous 30-day market returns), we performed DBACAN clustering to identify outliers among both purchases and sales.

The plots of both the purchases and sales DBSCANS make it clear that most transactions fall into one large cluster, with only a few outliers. Out of all 5241 returns calculated from the purchase data, we identified a total of 43 outlier transactions that are suspicious. These transactions were made by 23 individuals in both houses of Congress and both parties. Out of the 4968 `Sale` transactions recorded in the data set, 102 transactions were flagged as suspicious outliers, made by only 17 different congressional members. Calculating the intersection between the purchase and sale outliers, we identified a list of just ten members listed below who made transactions that appeared as extreme outliers: `"Adam Kinzinger", "Brian Mast", "Tom Malinowski", "Thomas Suozzi", "Patrick Toomey", "Roger Marshall", "Josh Gottheimer", "Donald Beyer", "Gilbert Cisneros", "Greg Gianforte"`.

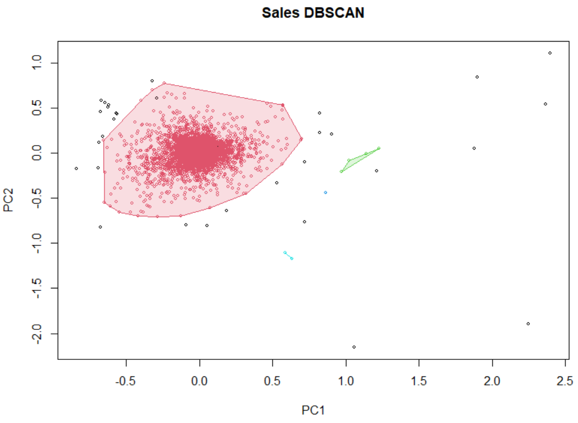

Figure 10: Hull Plot for DBSCAN clustering performed on the member sales returns data

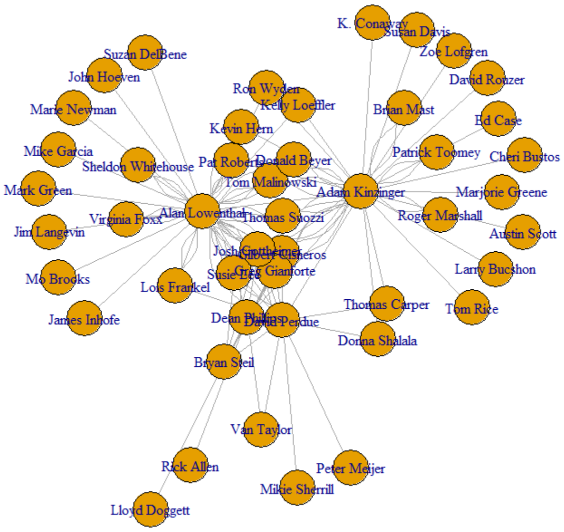

Figure 11: Network or members flagged as outliers in DBSCAN clustering analyses

There is also a deal of overlap between the DBSCAN clustering results from the returns data as well as the earlier DBSCAN results from the portfolio and transaction timing matrices we created. Out of the 33 members identified earlier clustering analysis, as shown in figure 8, 16 of these members also had transactions that were flagged as outliers. The members that were flagged in both sets of clustering analyses are as follows: `"Pat Roberts", "Tom Malinowski", "Susie Lee", "Ron Wyden", "Donna Shalala", "Thomas Suozzi", "Josh Gottheimer", "Thomas Carper", "Donald Beyer", "Gilbert Cisneros", "Greg Gianforte", "Kevin Hern", "Kelly Loeffler", "David Perdue", "Dean Phillips", "Alan Lowenthal"`

We constructed an undirected graph with the number of edges between nodes equal to the number of times connected nodes were flagged as outliers in the same dataset. In figure 11, you can see that there are a large number of connections to Adam Kinsinger, Alan Lowenthal, Gilbert Cisneros, Josh Gottheimer, and Greg Gianforte. David Perdue, Dean Phillips, and Tom Malinowski also have quite a few edges to other members.

The outliers identified in the DBSCAN clustering shown above also appear in the earlier analysis of returns that beat the market. Further, many of the members making these trades were also classified as outliers in the exploratory data analysis portion of this report.

We researched a few of these outliers - details are in the Interpretation and Discussion section.

Identifying the Best Traders

We consider the "best traders" to be those who made the highest percentage of trades that outperformed the 30-day market returns. For purchases, this means that the asset traded increased in price more than the market. For sales, outperforming the market means that the asset traded decreased in price more than the market.

Table 6: Top Ten Purchasers. Percent is the percentage of trades that beat the market

Representative	percent
Cynthia Axne	73.58491
Pat Roberts	66.45570
Donald Beyer	60.47904
Thomas Carper	60.31746
Kevin Hern	57.44681
Josh Gottheimer	57.31225
David McKinley	56.89655
Gary Palmer	55.71429
Ron Wyden	55.65217
Alan Lowenthal	54.76190

Table 7: Top Ten Sellers. Percent is the percentage of trades that beat the market

Representative	percent
Marie Newman	74.54545
Pat Fallon	56.25000
David Perdue	55.25813
Donna Shalala	55.14706
Lois Frankel	53.84615
Katherine Clark	53.57143
Tom Malinowski	50.68493
Sheldon Whitehouse	50.56180
Susie Lee	50.49020
Kelly Loeffler	50.41322

Many of the top traders are also identified in the exploratory data analysis, pointing to a potential consistent group of congressional members trading on information that allows them to outperform the market more than the average congressional trader, who only beats the market 49.5% of the time when selling and 51.9% of the time when purchasing.

Interpretation and Discussion

Highly Suspicious Trades

We wanted to take a look at some of the trades that were flagged as outliers in the DBSCAN analysis to see if they are indeed suspicious trades. Below are just a sample of the few that were flagged.

- Brian Mast purchased IPWR at the end of December 2020. In January 2021, less than 30 days later, IPWR published a whitepaper describing a new battery charging technique that could be used to improve charging efficiency in electric vehicles. Mast is a member of the transportation committee. The IPWR price tripled immediately following the announcement.
- Gilbert Cisneros and Josh Gottheimer purchased QDEL on March 10, 2020, and March 2, 2020, respectively. On March 17, 2020, QDEL announced that they received an FDA emergency use authorization for a new tool to help rapidly diagnose patients with COVID-19.
- Thomas Carper purchased NTLA on November 20, 2020. On December 5, 2020, NTLA announced a new treatment method for Leukemia.

The small sample above, combined with the clustering results and the analysis of trading patterns and returns, shows that some members of Congress are most likely trading on information that is not available to the general public. With a combination of clustering on portfolio data, transaction history, and returns generated from transactions, we were able to use DBSCAN to identify potential outliers. However, while we showed that this method works to some degree, there are several flaws that need to be addressed. First, we have an assumption that most people who trade are trading on public information, are trading at similar times, and are trading in similar securities in similar amounts. These could all be shown to be grossly oversimplified. At least for our data, we only observed one dense cluster whose members belonged to a common label. It could be that this would not be the case for all datasets that one might encounter, and a more advanced clustering algorithm should be used.

The inherent incompleteness of the data within our databases proved difficult, especially with the limited information regarding each trade. For instance, on the dates we identified where Suzan DelBene was making large transactions in Microsoft, we know from other sources that she was making transactions in excess of $25 million in most cases. However, the highest bin range for our data was “Greater than $5 million” which definitely reduced the degree of granularity in our analyses. In addition to this, we had no knowledge of the number of shares that were purchased, so all of this information had to be estimated to the best of our ability.

Generally speaking, identifying suspicious trades was very difficult and prone to many errors. Our greatest difficulties were in coming up with a reliable method to estimate the returns and finding an appropriate clustering method that was useful in identifying outlier points. Our initial hypothesis that most people are normal traders and would group out in one cluster proved to be true for the majority of the datasets we constructed. Additionally, by using DBSCAN in a ‘backward’ fashion to identify outlier points, we were able to effectively find those traders or trades that differed substantially from the rest. However, even with this information, one still has to conduct in-depth research into what happened with a given company in the weeks surrounding a given transaction. This could be automated with web scraping and NLP models but was far beyond the scope of this project. Improved identification of congressional members could be made with a larger labeled dataset of known inside trades from the SEC using supervised learning models.

Given the current minimal penalties regarding late transaction reporting and the fact that members of Congress are allowed to freely trade while serving on congressional committees, we believe that more action should be taken to restrict congressional members from benefiting from the information they receive. At a time of historic mistrust of politicians and lack of faith in the democratic processes we hold dear, it is imperative that we citizens and lawmakers take the necessary steps to ensure our markets are secure and free from undue outside influences.