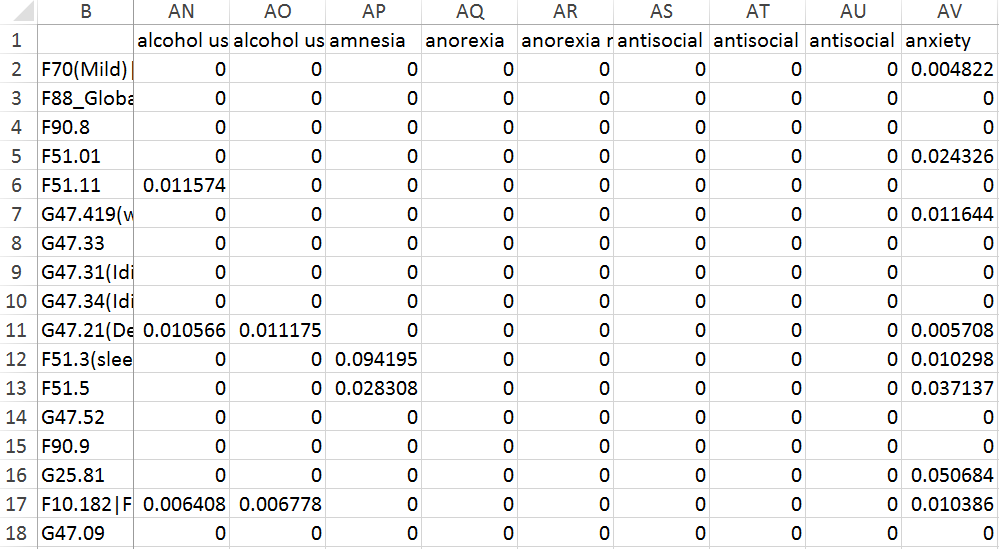
Hi Kelly and Maggie, welcome back to the lab! How’s your trips? I am right now in Atlanta and this document will tell you what we have done since last week and what we will do in this week.

Firstly, there is a bad news for Kelly. It seems that Jordan did not like our cosine similarity matrices and adjacency matrices and he said that because the adjacency matrix just contained too much information to let us compare efficiently. Maybe I misunderstood what he meant, but I think Maggie could explain to you more clearly.

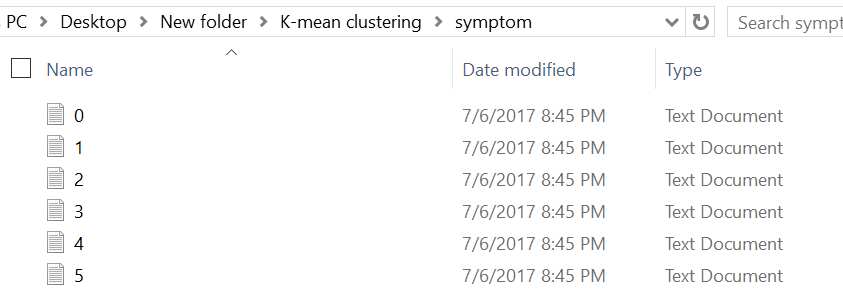
**Generating Tf-idf Matrix**

Ok, Jordan’s new idea is create the tf-idf matrix with all codes for diagnosis on the first column and all word chunks on the first row. We wrote the program only captured 500 features (word chunks) and we created two versions of file, one is all size of word chunks (n-gram size from 1 to 6) and another version with n-gram size from 3-6.



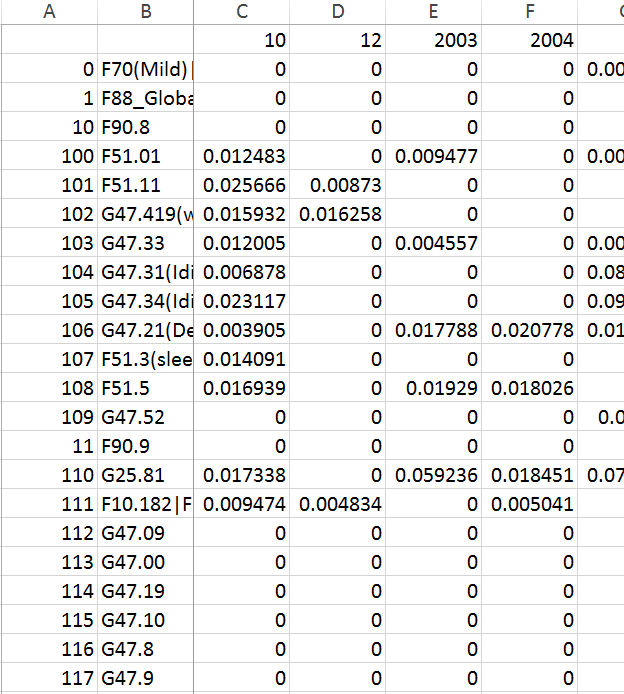
NOTE: If you want to create another matrix with larger number of features captured and other size of n-grams. Here are something to notice:

1. Because we split the all DSM-5 text and I name the diagnosis from 0 to 222. (223 diagnoses in total)



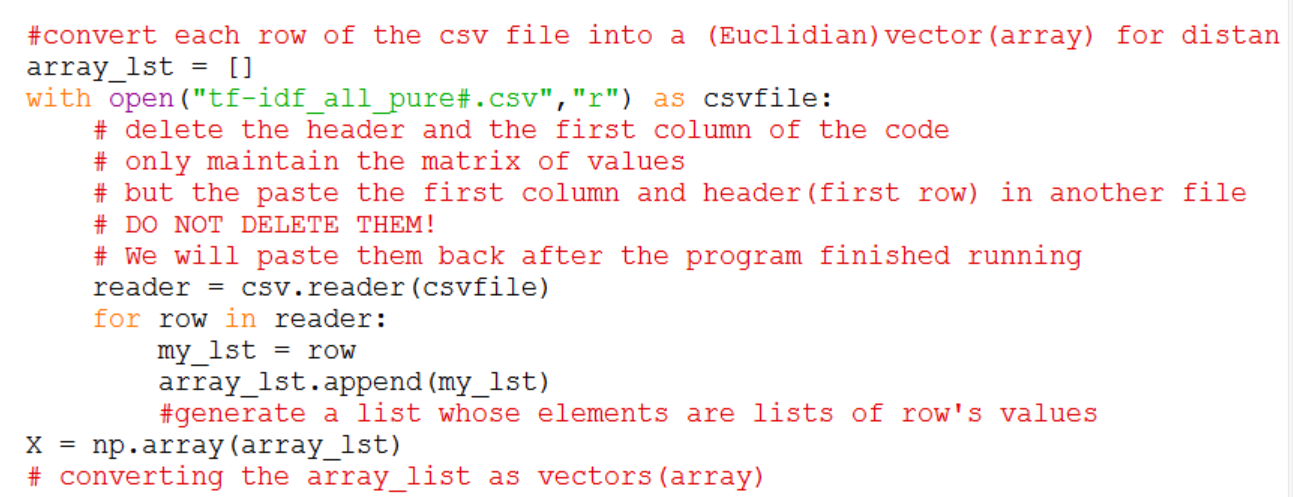
But I did not expect that on the first column of the matrix generated, the order of the diagnoses was not listed numerically (from 0 to 222) but instead, appeared in a strange order. I matched the rows with the corresponding indices (from 0 to 222) and F codes. If you want to use the same code to generate similar matrix, you can just copy and paste the very first two columns and they should be the same.

2. The first row would not be generated directly. (Sorry I was a little bit running out of time to think of a good way to print out the nice format) So what I did was to convert all 500 features (500 captured common word chunks) into a data frame, to export the data frame as the csv file and then copy and paste the feature names to the first row of the matrix we want.

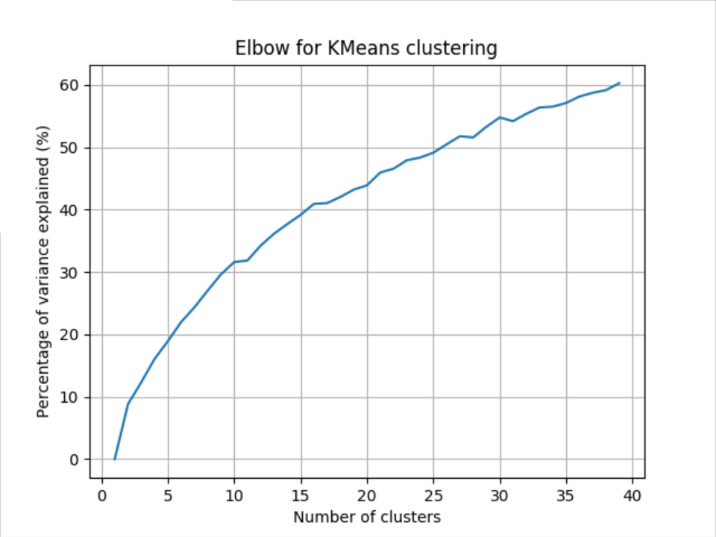
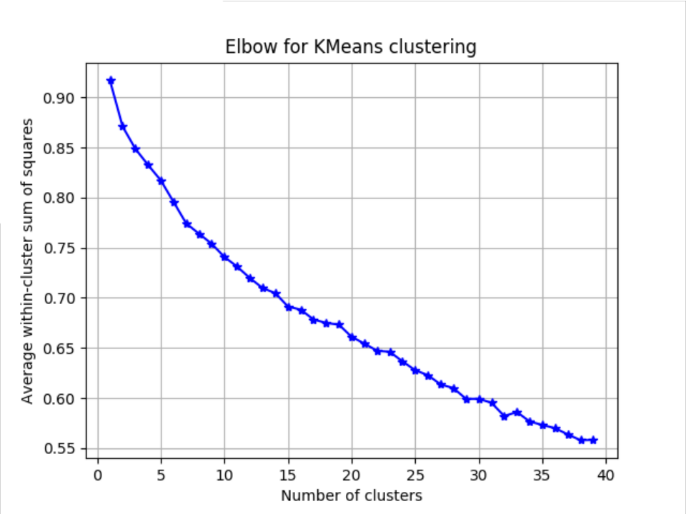


**K-mean Clustering**

And then we did K-mean clustering directly on tf-idf matrix values. Firstly, K-mean clustering could only cluster elements based on the Euclidian distance, so the first step we need to do was to convert each row of the matrix we generated to be a vector (array). To make things easier, I just delete the first two columns and the first row (cut and paste them in a new file because we need to paste them back later), and read the csv file row by row as lists and convert the lists into arrays.



And then in order to cluster all of the diagnoses, we need to figure out the optimal number of the clusters we want. So I used the elbow method, trying to find the best number of the clusters. (<https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set>)



We really don't want cluster number more than 40, so I used the range of cluster from 1 to 40. The figure to the left is the elbow plot, the elbow point is where the average sum of square distance within clusters does not vary so much. To the right is the figure Jordan wanted us to created too; it is the percent of variance been explained by the number of clusters. However, we found that both graph cannot effectively show where the elbow point is.

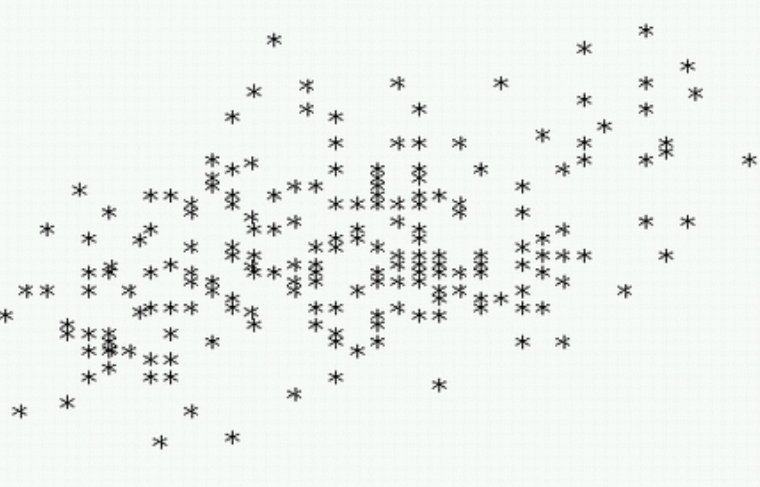
**Future Thoughts**

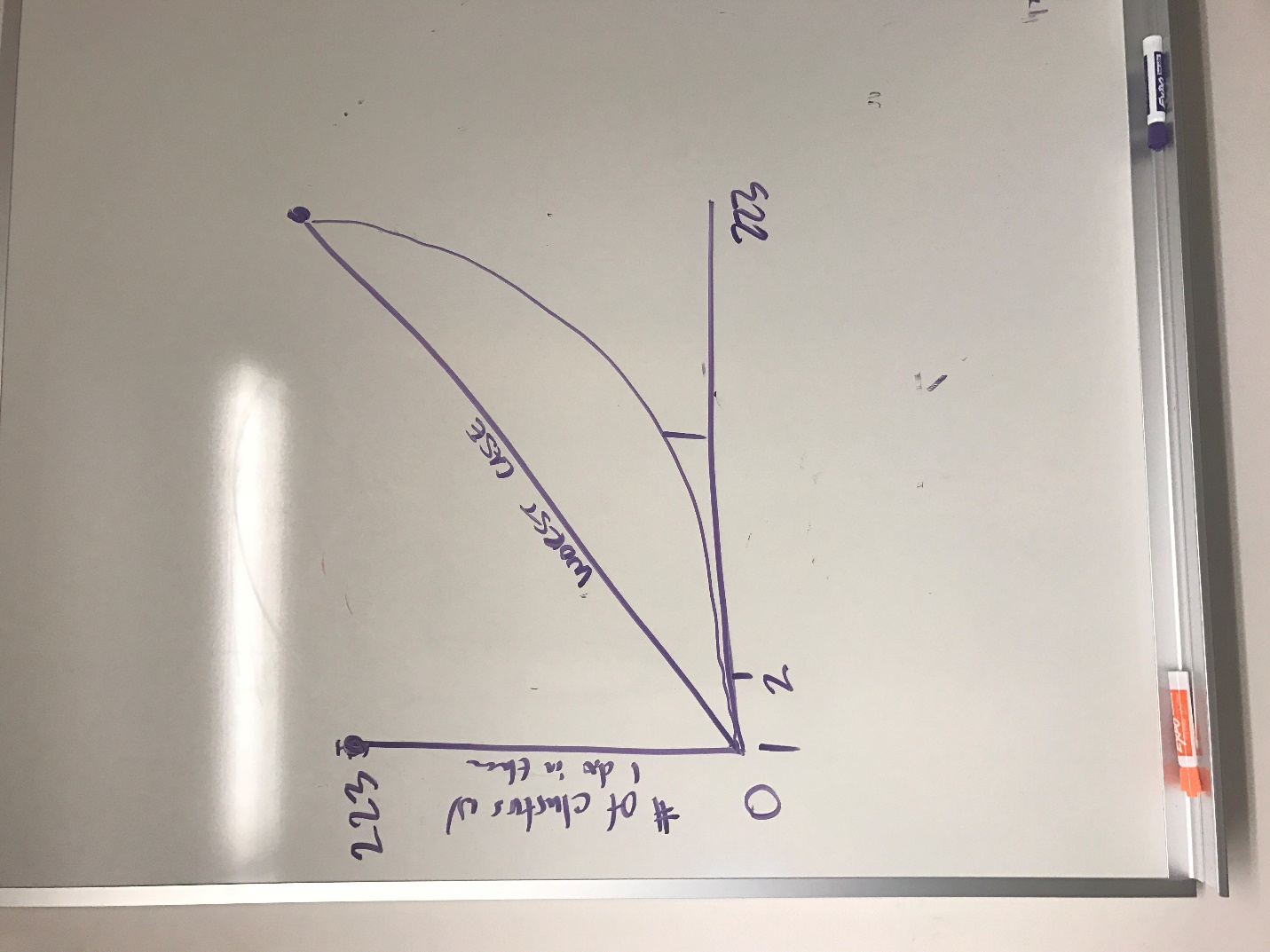
Here are some drawbacks with the K-mean clustering we have done so far. And I will try to explain what Jordan had told me about the future steps we should consider.

**Drawback 1: Cannot determine the elbow point and potential risks of using elbow method**

The first question I had mentioned before was even we made the elbow plot, we still cannot determine where the elbow point is (what is the optimal number of cluster we should use)

And a potential risk was even though we see the decrease in the average sum of squares of the distance within clusters, we are not sure whether the decrease means something.

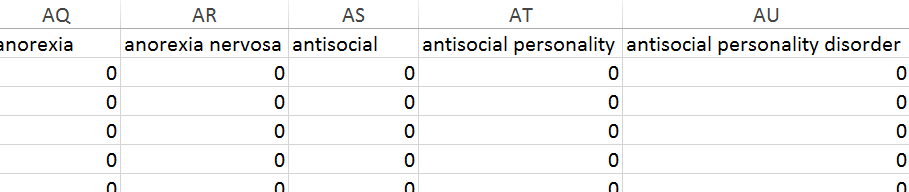


In the figure above, we have two ways of clustering: 2 clusters, 4 clusters. Let’s suppose the average sum of square of distance of 4 clusters is smaller than that of the 2 clusters. However, such decrease if we divide one large cluster into sub-clusters does not mean anything because they can be grouped together and that is good enough.

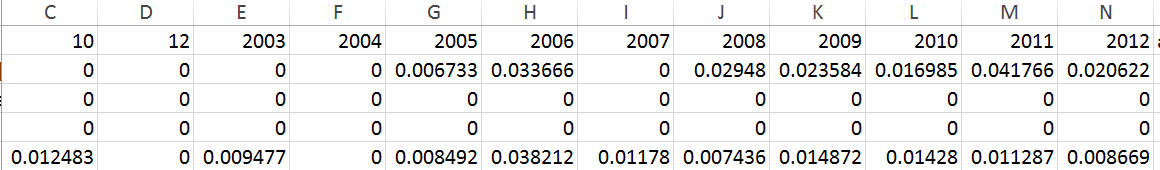
So Jordan thinks a better way to solve these two questions is making a new version of graph: X axis is still number of clusters, and Y axis is number of clusters with only one diagnosis in that cluster and we possibly would see a curved line, which initially increases slowly and after a certain point, increases dramatically.

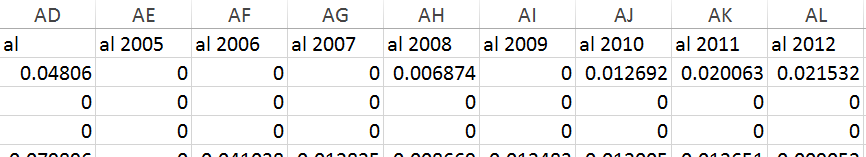
**Drawback 2: Repetitive word chunks captured**

Another problem is that in the mTf-idf matrix we created right now, the first row captured many extremely similar word chunks. For example:

****

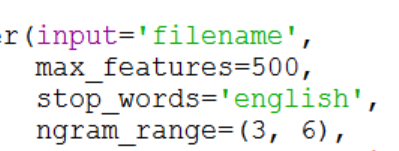
Also many word chunks that are references and numbers:





Possible ways to solve these problems include:

1. Manually delete those similar or meaningless word chunks. Because the word chunks are ordered alphabetically, so these cases must be adjacent to each other. It would not take so much time to finish deleting these word chunks. And we always want to keep the most specific word chunks. For example, antisocial personality disorder in first figure of this page, we want to keep “antisocial personality” and delete the other two because disorder appears so many times in the whole text and it is definitely not a key word. Antisocial personality is more specific than antisocial, so we want to keep antisocial personality. As you go through the word chunks, you can decide by your own and maintain the one that makes the most sense to you.

2. Add stop words. Right now we get rid of some unimportant words based on English language but we can also build such stop word list based on the DSM-5 language. Thus, words like disorder, diagnosis, personality, etc would not be included when we calculated the tf-idf values.

Based on what Jordan told me, he suggested us to manually delete the word chunks firstly unless we have more efficient programming method to get rid of those similar word chunks. I think it is better to ask him again on Monday.

***Thank you so much for continuing our work! I definitely have great time working with you during this summer and I would be the side of my computer as much as I can and definitely throw questions to me!***

***Hope to see you next semester, highly possible in stats class and economics class. Enjoy rest of your summer break! ☺***

**Best regards,**

**Lucy**