**CURRENT PROBLEMS:**

triple\_df\_agg functions do not work when dataframes aren't equivalent (eg when certain years don't have data for the names we are searching for.)

**TODO:**

perhaps ditch the 3 year aggregator function, and use Flourish to create a race animation histogram (https://www.youtube.com/watch?v=dH\_Ap\_UagYE) that goes year-by-year)

**Code Process for Tax Name Discovery:**

It's simple. I iterate through my sets of dataframes, using a dictionary to count the total frequency of each virus name. If the virus name does not exist in the dictionary, it is added to said dictionary as a key, with its frequency added as its value. If the key (virus name) has been seen before, we simply add its frequency to the value already existing (all of the prior frequencies.) I do this through both the sub and pub networks, forming one dictionary. You will find the results in this folder. Additionally, if desired, I can split the dictionary into sub and pub networks for comparison.

**Code Process for Plotting:**

Our inputs: 52 .csv files, split into sub\_virus and pub\_virus networks of 26 files each, each file representing a year over a time period of 1992-2018, each containing a column of virus names on which papers have been submitted/published, and a column storing the frequency of virus name occurence.

We first import these .csv files using pandas into two lists - sub\_virus\_list and pub\_virus\_list, lists that contain 26 pandas dataframes each - where each .csv file of one year (for example 1992) is represented as a dataframe - visually and structurally equivalent to the .csv files.

Now that we have our lists of dataframes, we run sub/pub\_top\_name\_binder functions through the sub and pub lists. The binder functions take a dataframe and a list of desired virus names (as of 9/29/20, this list is manually entered - as soon as the name discovery algorithm is working, we will use this to determine names.) The function then finds the frequency values corresponding to each name, and returns a dataframe consisting of only the desired virus names and their values.

Then, the new\_sub/pub\_lister functions apply the above binder functions to each element of our list. Now, all our dataframes only contain the names and values we want.

Next, we pass these new lists to sub/pub\_triple\_df\_agg functions - these are aggregator functions, that take three dataframes and average the frequency of each virus name across all three years, returning a single dataframe representative of a 3 dataframe, 3 year section.

Now, we apply the aggregator function to the entirety of each list, returning 9 dataframes each for sub and pub - and plot the data into a trellis plot - 9 subplots, sharing the same axes, closely positioned for visual data comparison. As of 9/29/20, this doesn't work - see the "problems" section at the top of this file.