Capstone Project 1: Displaying Large Organizational Networks: Data Wrangling Report

This project is designed to make an application that can be used by individuals who are not specialists in network analysis to view and explore a large organizational network on their own. It is built using data gathered from a survey or organizations, asking them with whom they interact as they make and influence water quality policy. This report first repeats the description of the data from the project overview and then speaks in more detail about the cleaning process. The codebook, which be found in the /data/ folder provides a complete set of details describing the data.

For this project, I used Python to repeat the data wrangling process that I previously completed using R. The purpose of doing this is to practice my data wrangling skills in Python, ensuring that I can complete this task using both tools.

Data

In 2014 and 2015, researcher with the Vermont Experimental Program to Stimulate Competitive Research (Vermont EPSCoR),[1] a branch of NSF EPSCoR, gathered two editions of an organizational network survey which gathered data on the networks that make and implement water quality policy in Vermont's portion of the Lake Champlain Basin (LCB). Table 1 lists the four organizational groups into which we sorted our respondents and reports the rate at organizations responded, by group. An online survey was developed and deployed, with respondents recruited through personal outreach. Each respondent was presented with a list of all the organizations in the survey and asked which organization's the respondent's organization shared information, provided technical assistance, collaborated or coordinated on projects, provided reports of their operations, and shared financial resources. We derived five different functional subnetworks, one from each of these types of interactions. Several characteristics, or node attributes, for each organization are also recorded, including a measure of the organization's budget and staff size, the organization's sector (such as public, private, or non-profit), the organization's geographic jurisdiction (Vermont, New York, Quebec, USA, Canada, etc.), and jurisdictional level (municipal, regional, state/province, federal, international). Each network and the attribute data are recorded on separate. Since these data were gathered with federal research dollars, a de-identified version is available for free distribution and will be posted to Github, along with a full codebook.[2]

[1] I personally led this effort, leading the drafting of the survey instrument, outreach to survey respondents, and data preparation, cleaning, and analysis.

[2] Data are found here: https://github.com/wmirecon/Water_Quality_Governance_Networks

Table 1: Survey Response Rates

Organizational Group	Number of Contacts		Completed Responses		Response Rate (%)		Observation Rate (%)[3]	
	2015	2014	2015	2014	2015	2014	2015	2014
Governmental Programs	53	56	30	26	56.6	46.4	81.6	71.75
Regional Actors and NGOs	51	50	24	26	47.1	52.0	72.5	73.47
Winooski Watershed	52	52	29	11	55.8	21.2	80.9	38.16
Missisquoi Watershed	34	40	12	12	35.3	30.0	58.8	51.54
Total	**190**	**198**	**95**	**75**	**50.0**	**37.9**	**75.1**	**60.26**

Survey Design as initial data wrangling

The survey used a customizable online survey tool, LimeSurvey, through a license provided by the University of Vermont (UVM). Several steps were taken to ensure that, when the raw survey returns were downloaded from LimeSurvey, they would be easily cleaned and prepared for analysis. The network interactions portions of the dataset provided the greatest challenge to this task. With an on our needs as analysts, the network links were designed to be as easy on respondents as possible. This meant providing a matrix of potential interaction partners, each with a list of interaction types. For each partner and each interaction type, the respondent could mark 1 for non-routine interactions, a 2 for routine interactions, or leave blank for no interaction. When downloaded, each respondent's answers are recorded on one line and each combination of a partner and interaction type receives a separate column. This data structure means that the network interactions are downloaded in a form that is similar to the kind of matrix that many network analysis platforms, including the *igraph* package that is used for SNA in both R and Python, making large-scale transformations of the dataset unnecessary.

Data cleaning problems presented by survey tool

However, networks are best analyzed with only one type of interaction.[4] This structure means that the matrices for all five interaction types are intermixed and must be separated. The large number of potential partners, over 200, multiplied by 5 network types, means that there are over 1,000 different columns of network, making hand cleaning impossible.

[3] Observation Rate records the percentage of non-directional network links that the survey was able to observe by obtaining a response from at least one of the two organizations involved in each link. See: Scheinert, S., Koliba, C., Hurley, S., Coleman, S., and Zia, A, 2015, The shape of watershed governance: Locating the boundaries of multiplex networks. *Complexity, Governance & Networks*, 2(1), 65-82. doi: 10.7654/15-CGN25.

[4] Analysis using multiple types is possible but much more complex. Often, the mixture of different types of links makes most standard network statistics meaningless, as they cannot be cleanly and clearly interpreted.

Additionally, meeting standards for the ethical conduct of research, as enforced by the UVM Institutional Review Board (IRB) requires that identifying information be scrubbed from the dataset. Most of this information is easily scrubbed by deleting columns from the raw downloaded data that contain names, titles, and contact information for individual respondents, leaving only their organizations. But the organizational names that are included in the downloaded column headers as well and respondent entries requires a more detailed approach, and is best down computationally.

Anonymization

Anonymization is done using a combination of translation keys. In R, this was done by joining data frames using keyed merging, specifically, using the *left_join()* function provided by R/*dplyr*. A better approach, which will be implemented in future cleaning efforts would use a kind of named-value vector referred to as a look up list. For cleaning in Python, I used Python's equivalent to a look up list, a dictionary. Dictionaries were applied first to the respondent's provided organizational names. This was done iteratively until all organizational acronyms were matched to a dictionary value and an associated anonymous identifier. The Codebook for the dataset describes how the anonymous identifiers are constructed.

Once the respondents' organizational affiliations were anonymized, data for the bimodal interactions, those which connect an organization to the geographic areas, policy domains, policy tools, action arenas, and accountability mechanisms that the organization uses or in which it is active, are removed and stored separately. There is no missing data in these interactions; the survey structure prevented it. Each option for a geographic area, domain, tool, action arena, and accountability mechanism presented respondents with a check box. If you used that item, you marked the checkbox. If you did not, then you left it blank. The data then downloaded this data as either a "Yes" or "No" for the status of the checkbox. Cleaning included changing "Yes" and "No" to 1 and 0, respectively.

After removing bimodal network data, the remaining organization to organization network links are stored in a *pandas* data frame. Instances where respondents indicated no interaction in this dataset are downloaded as missing values, but are in fact, an observation of the lack of a link, rather than a non-observation. For this reason, missing values are recoded from system missing to '0'.

Once this is down, the data frame is transposed, which makes the next two steps easier. First, the anonymization process is repeated. This did not require iteratively checking for matches, as these organization names were built into the survey system and downloaded using a standardized set of acronyms which I, as the research and survey designer standardized while designing the survey.

Network Matrix Separation

Once the organization names were anonymized, the separate matrix for each type of interaction (Information Sharing, Technical Assistance, Project Coordination & Collaboration, Reporting, and Financial Resource Sharing) had to be separated. All potential interaction partners were listed five times, one with each subnetwork, and always in the same order, allowing for a list to be made of subnetwork labels in that order, and then repeated until it was the proper length. This

list could then be entered as a column in the transposed data frame and used for filtering. Once filtered into five separate non-square matrices (NSQs), the subnetwork label column is removed and the data frames each re-transposed back to their original structure. I did this filtering this way as both R and Python filter move effectively on rows than on columns.

One complicated factor throughout this process is that duplicate organization names existed in both the column headers and the row headers. This is less problematic for the rows, but forces name changes in the columns. And, since the data is transposed multiple times, the duplicates in both the rows and columns would be changed, breaking data relationships that should not be broken; duplicate names represent multiple observations involving the same organization and so the names must remain consistent. This is dealt with by placing headers within the data frame rather than as row or column indices. Python struggled with this while R dealt with it naturally, though were able to function under this constraint.

Network Edgelist Construction

The final step involved a further transformation of the data. While the data did come as matrix, which network analysis platforms recognize, the actual structure of the matrix does not fit pre-set network platform expectations. A network 'mode' is a type of node, such as organizations or people. The bimodal section is referred to by that name, as it represents data in two different modes.[5] Except in unlikely coincidences, bimodal datasets that are coded as matrices will be rectangular matrices. Network analysis platforms understand this and are built to input rectangular matrices where the axes, that is the rows and columns represent different modes.

But unimodal data, like that in the organization to organization subnetworks, when coded as a matrix is coded as a square matrix, with the two different axes matching and the diagonal representing "self-loops," where links originate from and point to the same node. But, unless exactly one response is obtained from every potential interaction partner,[6] survey data will not provide a square matrix. Very few network platforms are capable of handling that variation.[7] Instead the data must be transformed, either into a square matrix or into another type of data structure, called an edgelist. Edgelists list node pairings where links exist between the nodes, forming a dyad. Each line of an edgelist represents a single dyad. My final cleaning step was to write a function that transformed the NSQs into edgelists.

[5] Really, there are several different modes, but the underlying structure is the same so that they can be stored together. Analysis requires separation, but the separation is not a necessary step during cleaning and preparation.

[6] This is generally the goal of a network survey, but it is *extremely* difficult to achieve, especially for a large network, like the one in this dataset.

[7] The only platform that can is *ORA, programmed by the CASOS Institute at Carnegie Mellon University. While it has a batch command mode, it is not programmable in the way that R and Python is, make it much more limited. Also, until recently, licenses were $500 each.