Data Preperation

In this chapter we discuss how to import data into software and then look at issues such as cleaning the data, dealing with missing values and transforming and manipulating the data to get it into a form suitable for analysis. The most time consuming error in this area, is collecting the data and not knowing what format the results need to be in to easily import them into programs such as UCINET. A little early work pays huge dividends later and any potential user of network methods would do well to give full consideration to these issues before they collect any data.

Whether the data is collected electronically, by observation, from historical or other records or via a questionnaire then the norm is for the researcher to have captured this information in a format that they have some control over. The data is then held in a database, on a spreadsheet, in a word processing file or simply in a pile of completed questionnaires. It is rarely, if ever, in matrix form or as a graph. Programs such as UCINET recognise this and support a variety of formats that aid data entry. The medium that is holding the data is of less importance, it is only of importance if the format has to be manipulated in some way so that it conforms to protocols required for data entry. Clearly having the data in a proper database such as Access provides the user with the greatest flexibility and for anyone familiar with an appropriate tool this would be the recommended way forward. Spreadsheets offer slightly less flexibility but still have enough functionality to allow for the user to configure the data in a way that is suitable for importing. Even having the data in a word processing file is usually sufficient. If the data ends up on completed questionnaires then there is no flexibility and so it is important to avoid the need to manipulate the data before importing into a software package.

One issue to consider is whether any labels, that is the actors names, should be directly used or whether the data is entered using a code or number. If labels are used throughout then consideration has to be given over consistency and this is a common source of errors. The same actor may have slightly different formats for their names, even down to the use of upper and lower case letters, or simply a mistype or misspelling can create a new fictitious node in the network. Another problem that can occur particularly in larger studies is two individuals have the same name.

3.1 Linked Lists

Most network data is sparse. That is the number of links that actually exist is much smaller than the number that could exist. Therefore the simplest way of representing the data is to list the adjacencies that exist and to proceed with the assumption that unlisted connections are not there. These are known as link lists. Programs take these lists and use them to ‘construct’ the corresponding adjacency matrix. There are a

number of different forms the list can take, we consider the most commonly used procedure. The easiest way to understand this is to consider a simple example. The following is a six node undirected network

1 3 4 5
5 6
4 6 5
3 6

The first number in each row gives the starting node of and edge the numbers that follow in the same row are a list of end nodes. Hence the first row 1 3 4 5 states that actor 1 is connected to actors 3, 4 and 5. The second row states that actor 5 is connected to actor 6 and so on. Note that there is no order amongst the rows nor within the rows. Actor 2 is not listed and hence is an isolate. The network corresponding to this linked list is given in Figure 1.

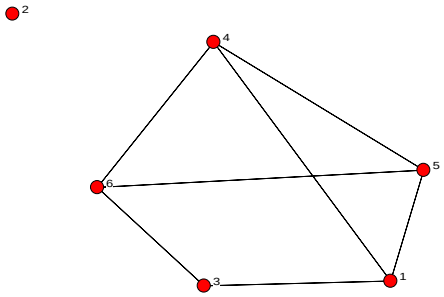

Figure 1

The network analysis programs Pajek and UCINET use a special data entry language developed by Stephen Borgatti called DL (Data Language). This has a series of keywords that appear before the link list data which describe the type of data and gives vital information that allows the software to construct the adjacency matrix. As an example to import the data described above we would construct a text file using a simple editor such as notepad (or the text editor in UCINET) which would look like

```
dl n=6
format=nodelist1
symmetric=yes
data:
1 3 4 5
5 6
4 6 5
3 6
```

The dl tells the software to expect a dl file, n=6 gives the number of actors, format=nodelist1 tells the program to expect a linked list of the form described here, symmetric=yes states that this is symmetric data so that every link is reciprocated and data: indicated that the data will follow.

The method so far described explains how to get the data into the software but has made no mention of the labels or names associated with the nodes. Typically this can be brought in at the same time as the network data either as part of the data or alongside the data. Alternatively it can be imported in a separate operation. If we used the same format as above then a linked list would look like (note this is a different network)

```
sanders    skvoretz s.smith
skvoretz   sanders s.smith "t smith"
s.smith    sanders skvoretz
"t smith"    skvoretz
```

This is a network with four actors sanders, skvoretz, s.smith and t smith. Here we can see that if this was a larger network there is potential for confusion over the two smiths. It is also necessary to enclose the name t smith in quotation marks so that we don't create two actors namely t and smith. It is advisable to place quotation marks around any label (or file name) that contains spaces.

One problem with the format described above is that it is not possible to input any values and so it is only suitable for binary data. A simple extension can overcome this problem. Instead of listing an actor followed by all their adjacencies we list all the edges followed by the value of the edge. The following is an example of such a format:

```
1 2 4
1 3 2
2 1 1
2 3 1
2 4 4
3 1 1
```

Here actor 1 is connect to actor 2 and the link has a value of 4. Note this data is not symmetric since further down the list we see that actor 2 is connected to actor 1 by an edge of value 1.

Each software package will support a variety of link lists and an early decision as to which is the most suitable is advisable so that it can be incorporated into the data collection design. This is particularly true if the data collection results in completed paper questionnaires.

7.2 Other Data Types

The only other general type of data will be an adjacency matrix. This can be read directly as a format by most packages. If the adjacency matrix is stored in a spreadsheet then some packages such as UCINET have the ability to directly import Excel files, or have an internal spreadsheet in which the matrix can be pasted. Most packages support the importing of network data from one to the other. So for example file types such as vna (see visualization chapter), Negopy, Pajek and UCINET can be imported and exported from one program to the other.

Many network techniques use attribute data with the network data. We are interested for example if gender plays an important explanatory role on the structure of say a friendship network. We therefore need to be able to import attribute data alongside the network data. The nature of the attribute data is important in determining which techniques are applicable. We are interested in whether the attribute can take on a continuous value such as age or height, or whether we just have discrete values which correspond to certain categories for example gender or membership of an organization. Most software packages handle attribute data as a column vector of numbers, usually with the rows in the same order as the rows of the corresponding adjacency matrix. These are then imported in the same was as a raw adjacency matrix. Often there is more than one attribute, some programs (Eg Pajek) insist that a different vector is imported for each attribute (Pajek also uses different file extensions for continuous and categorical data), other such as UCINET permit all attributes to be part of a rectangular attribute matrix. A simple attribute matrix for four attributes on five actors looks like

	Age	Income	Gender	Nationality
Roberta	28	126000	1	1
Steve	34	85000	2	1
Martin	44	96000	2	3
Vanessa	41	62000	1	3
Maggie	19	185000	1	2

The age column gives the age in years; the income is the gross annual income in US dollars; Gender is 1 for female and 2 for male; Nationality 1 is American, 3 is British and 2 is Swiss. The first two columns are continuous and the last two are categorical. Hence UCINET would allow the whole matrix to be imported but Pajek would require four separate vectors, two of each type.

It should be noted that attribute data can be converted to network type data. For example we could create a network based upon sharing an attribute. Hence in the above we can create a network in which two actors have a tie if they share the same nationality. We can do the same for the continuous data creating at tie between two actors if the absolute difference in their annual income is less than 10000 dollars.

7.3 Cleaning the Data

Once the data is imported into the software it is advisable to examine it in some detail. There are usually a number of problems and if these are detected early it will save a lot of time in repeating an analysis because a problem with the data has revealed itself later on. Usually if there has been a serious error this is apparent immediately but always check the imported data against the original to verify that the process has been completed properly. The most common errors are repeated nodes. This occurs when an actor has been entered twice (correctly on both occasions) or more commonly when there are slight differences in the labels. To find where this has occurred it is necessary to search for rows that are the same. One way to do this is to run a structural equivalence routine (see chapter 7?). Another common problem is that there are some missing actors. This is either because of non-response or they have simply been missed out. The former is a serious issue which we shall return to.

It is always worth thinking if the data should have some special features and seeing if they are present. The most common of these is whether the ties are reciprocated. Many relations are inherently symmetric and if this is the case then it is worth examining the data to see if it exhibits this property. Usually it does not! When this happens the best approach is to return to the respondent for clarification although this is often not possible. In which case the user needs to decide what to do. In most cases the normal practice is to assume that one of the actors failed to report the tie and that it does exist. Suppose we were dealing with friendship type data and asked both actors who they socialized with outside work. If actor x mentions actor y but y did not mention x the most likely explanation is that y simply forgot to mention x. We can use the same approach for verifying non-symmetric data provided we had constructed the study to collect both directions. For example we can ask to whom do you go to for advice? and from whom do you receive advice? We now compare the two matrices the transpose of one should equal the other. If there is disagreement then we need again to think what action to take. This is not as straight forward since it is less likely that this would be forgotten, particularly by the one receiving advice. It may be that the advice as not relevant and so the receiver decided it did not constitute advice. Or it may have been unsolicited and so the receiver did not think of it as advice. In general we should expect that both parties would agree and so if one of the links were missing we not accept that advice had occurred. Each case needs to be considered on its own merits with the analyst taking account of the type of relation, the method of collection and if known the reliability of the informants.

A major difficulty with network data is missing data which has occurred because of a non response. Unlike normal statistical sampling this is a serious problem and all efforts should be made to collect the data. This includes asking other parties if the respondent is not forthcoming. We can use similar techniques as above to deal with missing data. Clearly if it is symmetric data and only one actors is missing then we can use all the others to infer the results. If more than one actor is missing we can again deduce everything except the relationship between the missing actors. For non-symmetric data if we had asked both parties for the data then we can use this in the same way as completing the advice example above. Only in this case we would take the advice receiver as a clear indication of advice giving. Unfortunately most network methods will not tolerate missing data and the actor will need to be removed from the analysis. In this case some effort will be required to establish how crucial this is. One approach is to establish how important the actor is to the study, either by interviewing actors associated with the missing respondent or looking at relevant attribute data. As an alternative one can see how robust any analysis is by including missing data and trying to deduce or even guess missing ties to establish if the results are robust.

7.4 Data Management

Part of the process of data cleaning was taking data that should be symmetric that was not, and forcing it to be symmetric.The process of converting unsymmetric data to symmetric data is known as symmetrizing. This is sometimes necessary with inherently unsymmetric data when we wish to use a routine that only works on symmetric data. Clearly if we do this then great care needs to taken in interpreting the output since we have changed the relation we are investigating. In essence we are comparing the (i,j) entry with the (j,i) entry. We can make these agree by a variety of different methods, we could take the minimum of the two or the maximum. These are

the two options for binary data. If we think one report of the pair is sufficient then we take the maximum, if we feel we need both for it to be valid we take the minimum. The minimum method clearly reduces the nuber of edges whereas the maximum method increases the number. We can see this clearly in the following example.

										1
	1	2	3	4	5	6	7	8	9	0
	-	-	-	-	-	-	-	-	-	-
1	0	1	0	0	1	0	1	0	1	0
2	1	0	1	1	1	0	1	1	1	0
3	0	1	0	1	1	1	1	0	0	1
4	1	1	0	0	1	0	1	0	0	0
5	1	1	1	1	0	0	1	1	1	1
6	0	0	1	0	0	0	1	0	1	0
7	0	1	0	1	1	0	0	0	0	0
8	1	1	0	1	1	0	1	0	1	0
9	0	1	0	0	1	0	1	0	0	0
10	1	1	1	0	1	0	1	0	0	0

Original unsymmetric data

										1
	1	2	3	4	5	6	7	8	9	0
	-	-	-	-	-	-	-	-	-	-
1	0	1	0	1	1	0	1	1	1	1
2	1	0	1	1	1	0	1	1	1	1
3	0	1	0	1	1	1	1	0	0	1
4	1	1	1	0	1	0	1	1	0	0
5	1	1	1	1	0	0	1	1	1	1
6	0	0	1	0	0	0	1	0	1	0
7	1	1	1	1	1	1	0	1	1	1
8	1	1	0	1	1	0	1	0	1	0
9	1	1	0	0	1	1	1	1	0	0
10	1	1	1	0	1	0	1	0	0	0

Symmetrized data using maximum method

										1
	1	2	3	4	5	6	7	8	9	0
	-	-	-	-	-	-	-	-	-	-
1	0	1	0	0	1	0	0	0	0	0
2	1	0	1	1	1	0	1	1	1	0
3	0	1	0	0	1	1	0	0	0	1
4	0	1	0	0	1	0	1	0	0	0
5	1	1	1	1	0	0	1	1	1	1
6	0	0	1	0	0	0	0	0	0	0
7	0	1	0	1	1	0	0	0	0	0
8	0	1	0	0	1	0	0	0	0	0
9	0	1	0	0	1	0	0	0	0	0
10	0	0	1	0	1	0	0	0	0	0

Symmetrized data using minimum method

Clearly if the data is valued we can also use the minium and maxmium method but we could also use the average. There are a number of other alternatives but these are only used on rare occasions and unless there is a very good reason it is best to limit the choice to these three alternatives.

Another common requirement is to convert valued data to binary data. This process is known as dichotomizing the data. In this case we take a valued adjacency matrix and convert all cells greater (or less) than a certain number to one and set all the remaining cells to zero. Again the reason for doing this is to use methods that are only applicable to binary data. It is not advisable to simply select a single dichotomization but to repeat this for a number of different levels and use the same routine. The results from doing this retain the richness of the data and often reveal insights non the structure that were not easy to deduce from techniques designed to deal with he valued matrix directly. The effect of dichotomization is shown below in which the original valued matrix is dichotomized to form a binary matrix in which values 2 or greater are replaced by a 1 and all other values are set to zero.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	2	1	0	2	0	0	2	2	2	2	1	1	1	0
2	2	0	3	3	1	4	2	0	2	1	1	2	0	2	0
3	1	3	0	6	1	2	2	1	2	0	2	2	1	1	0
4	0	3	6	0	2	2	1	0	0	0	4	3	1	0	0
5	2	1	1	2	0	1	1	2	1	1	2	1	1	0	0
6	0	4	2	2	1	0	1	2	2	0	2	0	1	0	0
7	0	2	2	1	1	1	0	1	1	0	1	0	2	1	0
8	2	0	1	0	2	2	1	0	2	1	2	0	2	0	0
9	2	2	2	0	1	2	1	2	0	3	3	0	1	1	0
10	2	1	0	0	1	0	0	1	3	0	3	1	0	1	0
11	2	1	2	4	2	2	1	2	3	3	0	0	1	0	0
12	1	2	2	3	1	0	0	0	0	1	0	0	0	0	0
13	1	0	1	1	1	1	2	2	1	0	1	0	0	1	0
14	1	2	1	0	0	0	1	0	1	1	0	0	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Original valued data

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	0	0	1	0	0	1	1	1	1	0	0	0	0
2	1	0	1	1	0	1	1	0	1	0	0	1	0	1	0
3	0	1	0	1	0	1	1	0	1	0	1	1	0	0	0
4	0	1	1	0	1	1	0	0	0	0	1	1	0	0	0
5	1	0	0	1	0	0	0	1	0	0	1	0	0	0	0
6	0	1	1	1	0	0	0	1	1	0	1	0	0	0	0
7	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0
8	1	0	0	0	1	1	0	0	1	0	1	0	1	0	0
9	1	1	1	0	0	1	0	1	0	1	1	0	0	0	0
10	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0
11	1	0	1	1	1	1	0	1	1	1	0	0	0	0	0
12	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Dichotomized data

One of the features of network analysis is the fact that we examine multiple relations on the same actor set. It is sometimes necessary to combine these. Either because they are not completely independent, or possibly because the combined relation better captures what we are after. For example we may want an overall socializing matrix constructed from three separate relations such as who do you attend sports events with, who do you go to the theatre with and who do you go out to dinner with. We can combine matrices by summing them, averaging the entries or taking the union. Alternatively we may wish to extract a single matrix and analyze this on its own, or perform the same analysis on all the matrices separately. In dealing with multiple relations we need to keep in mind that not all matrices are the same basic type of data. For example some matrices may represent negative sentiment and others positive. Also if the data is valued and we are combining them we need to make sure that the scales are consistent. One matrix may have entries that vary form 1 to 10 whilst another has values that go from 10 to 1000. This is fine if we expect the second matrix to carry significantly more weight but if this is not the case we may need to scale the entries or normalize the two matrices so that they have the same overall mean.

Finally it may happen that we do not want analyze the whole network. We may wish to delete a node or nodes from the network. This may be because they have certain attributes that are not relevant or possibly they are isolated from the rest of the network and so should be removed from the study. Or we may wish to combine nodes to form one node that is connected to the same nodes as the individuals were. One reason for combining nodes may have been that the data collected was at too finer level and we need to take courser grained analysis. Combining nodes in the same departments would be an example of moving up form the individual level to the department level.

We have mentioned just a few of the more common data management issues. Most software packages provide tools that help with the practical side of undertaking these.

If the data management has fundamental changed the nature of the relation then this must be taken into account when interpreting the output obtained from any analysis. Care needs to be taken to ensure that any of these processes do not undermine the research question that the analysis is trying to answer.

7.5 Valued Data

There are some special considerations when dealing with valued data. Probably the most basic for data that measures strength of a relation is whether the data is distance or similarity data. Similarity data is when higher values imply a closer connection or stronger relationship. In distance data the reverse is true and lower values imply a stronger connection. A simple example of distance data would be a matrix in which each actor was asked to rank the other actors by saying who would be their first choice, second choice etc to spend time with. An example of a similarity matrix would be one that recorded how much time people spent time together. It is fairly simple to convert similarity data to distance data and vice versa. If this is necessary it is usually built into any procedure. However all software programs require the user to declare whether the data is similarity or distance. The data matrix associated with symmetric similarity or distance data is called a proximity matrix.

Not all valued network data measures the strength of a relationship. Sometimes the values correspond to categorical data. For example we may have data that records a 1 if the two actors went to a sports game together, a 2 if they went to the theatre and a 3 if they went out to dinner. In this case the numbers merely signify the type of activity and give no indication of the strength of the relationship.