FATCAT -- *for thick data*

A weakness with much of the published network research is that it is "thin" -- there is a lack of context, of complexity. Everett Rogers discussed this problem in the keynote address at the 1986 conference of the International Network for Social Network Analysis. While analytic methods are becoming more sophisticated and powerful, they are unable to bring the social context in which the relationships binding people to one another take place.

Data that is rich in this contextual detail is sometimes called "thick" data. The problem is that, while programs like SOCK, SSA-II, NEGOPY, etc., may be powerful or may have large capacities, they do not have the ability to mix different kinds of data into a single analysis. To compare data that describes the qualities of the relationships between people with data that describes characteristics of the people themselves typically requires the use of two or more packages. SPSS may be combined with NEGOPY; BMDP may be combined with CONCOR; SAS may be combined with SOCK, UCINET, or STRUCTURE.

These combinations are not generally "nice". They require the user to run the data through one program and then edit the results and run them through the second program. The editing necessary to make the results fit is usually not easy. Even simple kinds of analysis may require the use of "tricky" techniques or multiple runs to achieve the desired results. Thick data is not only difficult to analyze; it is also difficult to collect. The (un?)lucky analyst who gets ahold of really thick data is likely to experience a "thick headache" unless some of the thickness has been removed from the data in one way or another.

This manual describes a program designed to make thick network data easy to analyze without "squeezing" it first. The package is the result of the growing level of frustration I experienced as I attempted to work with a series of increasingly thick network datasets. At first I used SPSS and MIDAS, together with NEGOPY. Later I wrote one-shot programs to merge files of demographic data with files of network link data. Once this was done, the "normal" statistical packages, like SPSS, could do a lot more for me. But the merging process was pretty nasty, the merged files were *huge*, and I got tired of writing one-shot programs that seemed to become more and more complex over time.

By then I had come to the conclusion that the types of analysis I had been doing on a one-shot basis were not limited to unique situations. I had seen enough of a pattern in the steps that comprised the analysis that I could start work on a general-purpose package that would mind all the details and give me time to think about what I was doing, rather than having to focus my attention on the complexities of the process. The result (several revisions later) is FATCAT, a pro-

gram designed to deal with thick (multivariate multiplex contextual) network data.

The analysis FATCAT performs can be described as "categorical," in that it divides the members of the network into categories and then describes the contact between and among categories. The program uses two files of data simultaneously: a file of "index" variables that describe characteristics of the individuals in the network, such as age, sex, occupation, etc.; and a file of "link" variables that describe characteristics of links, such as frequency of interaction, purpose or content of the contact, level of satisfaction with the outcome, etc.

Variables for FATCAT

The program uses several types of variables. Two types describe the *individuals in the network*:

- ***ID numbers*** Each individual in the network must have an identification number. Ideally, these numbers begin at 1 and run up to N, where N is the number of individuals in the network, but this isn't necessary, as long as there are no ID numbers lower than 1 or higher than 2000.
- ***Index variables*** Individuals may be described by a set of variables such as sex, age, income, position, etc. The program uses these variables to sort individuals into categories, where each category is a subset of the entire network. Index variables must be discrete or categorical. That is, they must be whole numbers, like 1,2,3.., rather than fractional values, like 3.14179 or 1.1417. Also, because the program cannot handle more than twenty categories at once, index variables can only have values between 1 and 20. Common demographic variables are typically used as index variables.

Three types of variables describe the connections between *individuals*:

- ***General magnitude link variables*** These are variables that, taken individually, describe the overall magnitude of the interaction between a pair of individuals. Examples of magnitude variables include frequency of interaction, amount of time in contact, and number of lines of e-mail exchanged. These variables could take any value greater than zero, such as 2, 43, 1.29, 5.25, etc.
- ***Categorical link variables*** These are similar to index variables, only they describe the links between individuals, rather than the individuals themselves. Examples include year of the transaction (1=1970, 2=1975, 3=1980), which party initiates the transaction (1=YOU, 2=BOTH 3=THE OTHER PERSON), which communication channel/ medium was used for the transaction (1=FACE-FACE, 2=PHONE, 3=E-MAIL), etc. These variables are discrete and categorical, as an index variables. Any one of these variables, taken by itself, describes one aspect of the link. These variables may take whole-number values from 1 to 20.

- ***Proportional link variables*** The value of each variable in the set indicates what proportion of the interaction fits into the category associated with the variable. For example, a set of proportional link variables might describe how much of the interaction is done in order to get information, to get advice, and to get action, or how often the purpose of the interaction is for each of these reasons. These variables take values that are fractions or percent-ages, or values that may be converted to fractions or percentages by multiplying by a constant.[1]

There are two important difference between proportional and categorical link variables:

- First, categorical link variables must take whole-number values between 0 and 20. Each *value* is associated with a category of links ("1" means this link goes in the first row or column in the matrix). Where a categorical link variable tells *what kind* of link it is, a proportional link variable tells *how much* of the link is of the type associated with the variable. Proportional link variables can thus take fractional values, such as 3.1419 or 1.141. Here, each *variable* is associated with a category, and thus with a row or column.
- Second, it only takes *one* categorical variable to describe a link. For example, the variable "initiation" tells who usually initiates the contact. However , to describe the content or purpose of the interaction that forms a link, a *set* of proportional variables is required, usually along with a general magnitude variable. For example, the general magnitude variable might tell how much interaction there is, while one proportional variable would tell what portion of the interaction is to get advice, another would tell what portion is to give information, and another would tell what portion is for emotional support.

(Note that neither categorical nor proportional link variables give any indication about the overall **strength** of the tie between the individuals connected by the link. Instead, they both tell something about the **quality** of the connection. Ways of combining general magnitude variables with categorical and proportional variables are discussed in later sections of the manual.)

---

[1] i.e. The data might be coded so that: 0=NEVER, 1=SELDOM, 2=OCCASIONALLY, 3=USUALLY, AND 4=ALWAYS. These values could be converted to proportions by multiplying by 0.25. Thus NEVER-->0, SELDOM-->.25, OCCASIONALLY-->.50, USUALLY-->.75, and ALWAYS-->1.0. The program asks you to supply the multiplicative constant at the appropriate time. For the situations in which it is necessary to add a constant before multiplying, the program also asks you to supply an additive constant.

What does FATCAT do?

Say we have a network with 100 members. We have several pieces of information about each of those people -- sex, age, education, income, etc., as shown below for the first three people.

	id #	sex	age	education	income	1st lang.	#children
Sam	1	2	4	2	3	1	3
Linda	2	1	3	1	1	2	1
Susan	3	1	2	3	2	1	2

Say we also know who talks with whom and how much contact each person has with each of the others. The binary adjacency matrix shown below (a simplified version of this kind of data) will be used to show what the program does.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	.	1		1				1	1			1		1				1	1	
2	1	.	1		1					1	1		1						1	1
3		1	.				1			1	1			1	1					
4	1			.		1		1	1			1			1			1		1
5		1			.		1				1		1	1					1	
6				1		.	1	1			1	1						1		1
7			1		1	1	.			1			1						1	1
8	1			1		1		.	1						1	1			1	
9	1			1				1	.			1		1		1				
10		1	1	1			1			.	1				1					
11		1	1		1	1				1	.		1	1					1	
12	1			1		1			1			.				1			1	
13		1			1		1				1		.	1						1	1
14	1		1		1				1		1		1	.	1		1			
15			1	1				1		1				1	.		1	1	1	
16						1			1	1		1				.			1	
17	1			1		1				1				1	1		.			1
18	1	1					1				1	1			1			.	1	1
19					1		1	1					1		1	1		1	.	1
20		1		1		1							1				1	1	1	.

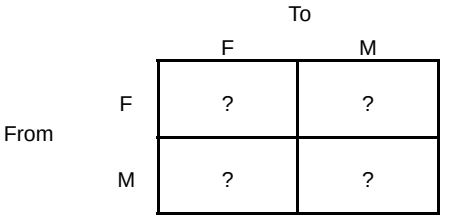
In this matrix there is one row and one column for each person in the network. A '1' in row *i*, column *j* indicates that there is a connection from person *i* to person *j*. For example, the matrix shows that person #1 has connections with #2, #5, #9, #11, #14, #17. A segment from an actual data file for the information in the matrix will look something like this:

```
                                    .
                                    .
                                    .
13  11   4  3  1  1  5  0  9  1  1  2
13  14   5  1  2  3  7  4  1  2  0  3
14  10   7  3  2  2  0  7  3  2  1  4
15  17   6  4  1  3  2  4  1  0  0  0
                                    .
                                    .
                                    .
```

The first two numbers on each line are the ID numbers of the person reporting the link and the person the link is with. The rest of the numbers are various link variables, indicating, perhaps, the frequency of interaction, the perceived importance of the contact, the length of the relationship between the two individuals, the amount of interaction that deals with work-related issues, the amount that deals with sports, the amount that deals with politics, etc.

The goal of FATCAT is to uncover the relation between the data that describes the individuals and the data that describes the connections between the individuals, and to assist you in identifying significant patterns in the link matrix. Let's see how it does this for the simplest type of analysis it does -- Type 1.

In Type 1 analysis, the program sorts individuals into categories on the basis of an index variable and constructs a new matrix, where there is one row and one column for each value of the index variable. Because each row and each column of the new matrix represents an entire category of people, these are sometimes called "categorical who-to-whom matrices".



If we choose "sex" for the index variable, the new matrix will look something like the one shown above. Of course the question marks will be replaced by numbers. What is important about this new matrix is that it has a much smaller number of rows and columns than the original data matrix. Furthermore, the rows and columns carry some meaning, since they are associated with values of the index variable.

After it asks you the names of your data files and what type of analysis you want to do, the program asks you to select an index variable to use for sorting people into categories. It reads the

index file, and sets up an "index" that has one entry per person. Each person's entry tells which category the person belongs to, based

Code: 1 = f; 2 = m

ID#	1	2	3	4	5	6	7	8	9	10	11	12. . .
Index value	1	2	2	1	2	1	2	1	1	2	2	1. . .

on the value of the index variable for that person. The matrices below show the index values (in *italics*) next to the ID numbers of the first twelve rows and columns of the matrix on page 5.

	1 1	2 2	3 2	4 1	5 2	6 1	7 2	8 1	9 1	10 2	11 2	12 1
1 1	.	1		1				1	1			1
2 2	1	.	1		1					1	1	
3 2		1	.				1			1	1	
4 1	1			.		1		1	1			1
5 2		1			.		1				1	
6 1				1		.	1	1			1	1
7 2			1		1	1	.			1		
8 1	1			1		1		.	1			
9 1	1			1				1	.			1
10 2		1	1	1			1			.	1	
11 2		1	1		1	1				1	.	
12 1	1			1		1			1			.

Raw adjacency matrix

	1 1	4 1	6 1	8 1	9 1	12 1	2 2	3 2	5 2	7 2	10 2	11 2
1 1	.	1		1	1	1						
4 1	1	.	1	1	1	1					1	
6 1		1	.	1		1				1		1
8 1	1	1	1	.	1							
9 1	1	1		1	.	1						
12 1	1	1	1		1	.	1					
2 2						1	.	1	1		1	1
3 2							1	.		1	1	1
5 2							1		.	1		1
7 2			1					1	1	.	1	
10 2		1					1	1		1	.	1
11 2			1				1	1	1		1	.

Ordered adjacency matrix

The program replaces ID numbers in the link matrix with the index values. In the matrix on the right, the order of the rows and columns has been changed so that all the "*1*" rows and columns come first, and then the "*2*" rows and columns. (The pattern of connections has not been changed; you can check this by comparing with the first matrix on the previous page.)

Now the program makes the categorical matrix, in which there is one row and one column for each value of the index variable. To do this, all the "*1*" rows are collapsed into one "categorical" row. The "*2*" rows are also collapsed, as are the "*1*" and "*2*" columns. Here is the result:

		To F	To M
From	F	24	4
	M	4	22

If you count the number of 1's in each of the four quadrants of the ordered matrix on page 7, you will see where the numbers in the categorical table come from. The data for this example are symmetric. Whenever there is a 1 in row *i*, column *j*, there is also one in row *j*, column *i*. This is why the numbers in the lower left and upper right cells of the categorical matrix are the same. In most social situations, the data are not symmetric, and one of these two numbers will be lar ger than the other, revealing a type of status differentiation in the social group.

If the index variable had more than two values, there would be more categories of people --

perhaps "*1*", "*2*", "*3*", and "*4*". The resulting categorical matrix would therefore have more rows and columns. The program can handle index variables with up to 20 dif ferent values, although tables with more than 9 rows and columns are difficult to interpret and not very nice to look at.

If we wish to consider the *amount of interaction* men have with women, rather than simply the *number of contacts*, the process is slightly different. Instead of a matrix of 1's and 0's, we begin with a matrix that has numbers that tell how much contact there is between each pair of individuals. This is where things get a bit more complicated.

- There can be different kinds of link data. In some cases links are *binary* -- they are either present or absent. All we would know about a pair of individuals is whether or not there is some contact between them. The large matrix at the beginning of this section contains binary link data.

- In other cases, the links are described by *scalar* numbers -- ordinary numbers, such as 1, 2, 3, 4, etc. Larger numbers mean more contact or "stronger" links. When there is no contact, the "strength" will still be zero, though. An example of a typical scalar strength variable is "frequency of contact", coded as the number of conversations in a typical month. If we have this kind of scalar link data, we tell the program that link strength is the frequency variable. Instead of adding 1 when it decides where a link goes, it adds the value of the frequency variable. The numbers in the table will therefore tell how many contacts there are in a typical month from men to men, men to women, etc.

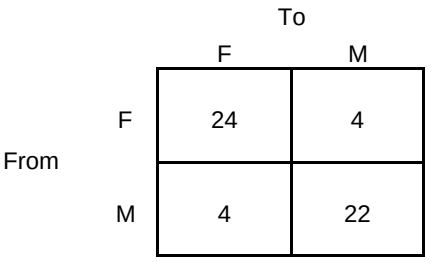
- In other cases, the network data is "more" than scalar. Rather than there being just one indicator of the amount of contact, there may be several variables. Each one may describe a different type of contact or a dif ferent aspect of the relation between the individuals connected by the link. This kind of data is called *multiplex*. To use multiplex data for the type of analysis described here, it is necessary to combine the appropriate link variables in such a way that you get a single indicator of the amount of contact between the individuals. You may do this by adding several indicators together, or by multiplying one indicator by another, or by some more complex type of operation. You may wish to use only the formal communication, or only the contact that involves financial information. The program is quite flexible in this regard, and it has the capability to deal with very complex situations.

Nine Types of Analysis

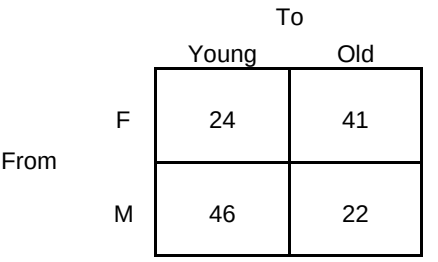
The program does nine types of analysis:

1. Index$_i$ by Index$_i$ --

Analysis Type 1, the one described above, uses a single index variable to sort people into categories for rows and columns. Type 2 uses one index variable for rows and another one for columns. For example, we could use "Sex" for rows and "Age" for columns. This would tell us whether men and women differ in the extent to which they interact with people of different age groups. Other than the fact that it uses different variables for rows and columns, Type 2 analysis is the same as Type 1.



Analysis Type 1



Analysis Type 2

Here the members of the network are sorted into categories on the basis of an index variable. The program describes the contacts members of each category send and receive, in terms of mean link strength, mean number of links per person, etc. Then it describes the amount of contact between and within categories. It does this by constructing "categorical who-to-whom matrices", in which there is one row and one column for each value of the index variable. Rows correspond to the senders of links; columns to receivers.

Analysis Types 1 and 2 can handle multiplex data, but they do this by either selecting one type of interaction or by combining several types into a single variable that indicates the strength of the connection.

This type of analysis allows questions like these to be addressed: "Do men dif fer from women in the amount of interaction they have with men and with women?" and "Do men

report more contact with women than women report with men?"

2. Index$_i$ by Index$_j$ --

This is the same as the first type, except two dif ferent index variables are used -- one for rows ("from") and the other for columns ("to"). There is one row for each value of the first index variable and one column for each value of the second. This allows questions like these to be asked: "How much contact with social workers, psychiatrists, and legal aid staff do female delinquents have in comparison to male delinquents?" and "Do people of different ages have different levels of contact with men and women?"

3. Index by Proportional Link --

Analysis Types 3 and 4 both use one index variable and a set of proportional link variables. For Type 3, each row corresponds to a value of the index variable. Each column corresponds to a single proportional link variable. For example, we could use "sex" for rows and different content areas for columns to compare men to women in terms of what they talk about (personal problems, politics, sports, etc.). Type 4 uses the index variable for columns and the proportional link variables for rows.

		Content			
		Personal	Politics	Sports	Money
From	F	32	17	12	19
	M	17	23	35	17

Analysis Type 3

Here an **index** variable is used to sort individuals who *send* links into categories. There is one row for each *value* of the index variable, and thus for each type of person (this is the same as analysis Types 1 and 2). There is one column for each of the **proportional link** *variables* you are using in this run. Thus each column describes a particular type of contact.

Analysis Types 3 and 4 were designed specifically to handle multiplex network data. When the data describe multiplex relationships, the connection between a pair of individuals may involve several types of interaction. Thus, the connection between a pair of individuals may contribute to several columns in the categorical matrix. This is in contrast to Types 1 and 2,

where the entire connection between a pair of individuals all goes into a single column.

Going across a row in the categorical matrix, you see how much of the contact of the members of the row category falls into each of the types described by the link variables. Going down a column, you see how much of one type of contact comes from each category of person. Type 3 allows questions like these to be addressed: "How much of the professional contact by communication researchers is made to get information, to get advice, to get action, and to discuss problems, in comparison with contacts made by sociologists, psychologists, and anthropologists?" or "To what extent do people who have dif ferent levels of education use electronic media to obtain different kinds of information?"

4. Proportional Link by Index --

This is similar to Type 3 above, except there is one row for each **proportional link** *variable* ("from"), and one column for each *value* of the **index** variable -- for each category of individuals who *receive* links. Going across a row, you see how much of one type of contact is directed to members of each category of person. Going down a column, you see how much of each type of contact is directed to one category of person.

Type 4 allows questions like these to be addressed: "For what purpose do respondents make contact with friends, relatives, and casual acquaintances?" and "How do the relationships people have with physicians, ministers, psychiatrists, and physiotherapists dif fer in terms of the extent to which they satisfy each of six categories of interpersonal need?"

Content	To: F	To: M
Personal	32	26
Politics	24	18
Sports	11	19
Money	14	22

Analysis Type 4

5. Index by Categorical Link --

Analysis Types 5 and 6 are similar to Type 2, except they replace one of the index variables with a categorical link variable. There might be a categorical link variable based on the answer to the question: "Who usually initiates the interaction -- you or the other person?" Call this variable "initiation".

		Initiator: Self	Initiator: Other
From	F	32	14
	M	17	35

Analysis Type 5

Here, the columns in the categorical who-to-whom matrix are determined by who initiates the interaction, while the rows correspond to the sex of the person the link is from. As each line of link data is read, the program looks for the sex of the person to determine which row the link belongs to, and then at the initiation variable to determine which column. Other than using a categorical link variable instead of a second index variable, analysis Type 5 is the same as Type 2. Links are counted or accumulated in the categorical matrix in exactly the same way.

Here an **index** variable is used to sort individuals who *send* links into categories. There is one row for each *value* of the **index** variable, and thus for each type of person. There is one column for each *value* of the **categorical link** variable. Thus each column describes a particular type of contact.

Because each link between a pair of individuals has only one value on the categorical link variable, the entire connection between a pair of individuals must fall into a single column, as it does for Types 1 and 2, and in contrast to Type 3, where the interaction between a pair of individuals may be spread across a number of columns. Analysis Types 5 and 6 handle multiplex relationships the same way Types 1 and 2 do.

Going across a row, you see how much of the contact of the members of the row category falls into each category of the link variable. Going down a column, you see how much of one type of contact comes from each category of person. Type 5 allows questions like these to be addressed: "How much of the social interaction of people in dif ferent age groups is directed towards people perceived to be more wealthy, equally wealthy, or less wealthy than the respondent?" or "How much of the e-mail sent by persons occupying different levels in the authority structure of an organization is sent in the morning, early afternoon, late afternoon, evening, or late at night?"

6. Categorical Link by Index --

This is similar to Type 5 above, except there is one row for each *value* of the **categorical link variable** ("from"), and one column for each value of the **index** variable -- for each category of individuals who receive links.

Initiator	To F	To M
Self	34	26
Other	24	42

Analysis Type 6

Going across a row, you see how much of one category of contact is directed to members of each category of person. Going down a column, you see how much of each category of contact is directed to one category of person. Type 6 allows questions like these to be addressed: "What communication channel do respondents use to make contact with superiors, peers, and subordinates?" or "How do the conversations people have with grandparents, parents, siblings, and children differ in terms of the basic rhetorical structure (competitive, cooperative, supportive, diplomatic, or argumentative)?"

7. Categorical Link by Proportional Link --

Analysis Type 7 is similar to Types 3 and 4, except it replaces the index variable with a categorical link variable. In Type 7, each row in the table corresponds to a value of the categorical link variable, while each column is associated with a proportional link variable. The value of the categorical variable for each link in the data file is used to assign the link to a category, which becomes a row in the table. The proportional variables determine how much of the link belongs to each column.

With this type of analysis we could see whether different communication channels (written, telephone, e-mail, face-face) are used for communication about different content areas.

This is similar to Type 3 above, except that there is one row for each *value* of the **categorical link variable** ("from"). Each link between a pair of people will thus be assigned to one row in the table. There is one column for each of the **proportional link variables**, so each link may contribute to some or all columns in the table.

Going across a row in the table, you see how much of the interaction of the category associated with the row is accounted for by the proportional link variables. If the categorical variable is "date" and the proportional variables are for content, each row would tell you how much of the interaction at one time is allocated to the different content areas. Going down a column, you would see whether or not the proportion allocated to a single content area was changing over time.

Type 7 allows questions like these to be addressed: "How do different communication channels compare, in terms of the extent to which they are used for different purposes?" or "How does the mix of information contents change from year to year over a five-year period?"

8. Index by Index Crosstabulation --

Type 8 does a standard crosstabulation on **index variables**. The tables produced by a Type 8 analysis are interpreted dif ferently from those produced by all other types. For **index variables**, Type 8 is like T ype 2, except that it is the individuals themselves that are accumulated in the cells of the table, rather than interaction between individuals. As in Type 2, there will be one row for each value of one index variable and one column for each value of the other index variable. The number in row *i*,column *j* is the number of individuals who have the value *i* on the first Index variable and *j* on the second one. This analysis is done without reference to the data that describes the links between individuals.

With Type 8, questions like these can be addressed: "Are the proportions of men in different age group similar to the proportions of women?" or "Is there a relationship between participation in continuing education programs and level of satisfaction with job placement?"

9. Categorical Link by Categorical Link Crosstabulation --

Type 9 is like Type 8, except it uses categorical link variables. In this case each interaction is classified according to the values it has on two categorical link variables, the first determining the row and the second determining the column. No reference is made to the Index data. Unlike Type 8, however, an equation can be used to weight each link according to the value(s) it has on other link variables.

Type 9 allows questions like these to be addressed: "What is the relation between the duration and the perceived satisfaction with interactions?" or "Is the location in which the interaction takes place related to the the the level of formality of the discussion?"

The nine types of analysis and exactly what they do are described in more detail in a later section.

Data Files for FATCAT

FATCAT uses two files of data simultaneously. One describes the links between individuals; the other describes the individuals themselves.

Both data files for FATCAT begin with *variable lists* which tell the program the names of your variables and where to find the variables. This feature was added to the program to make it easier to use and to reduce errors by eliminating the need for awkward format specifications.

The format of the data

The **link data** has the following format:

ID# ID# W1 W2 W3 ... Wn

The first "ID#" is the identification number of the respondent or the individual the link is "from" -- the individual who describes the link.

The second "ID#" is the identification number of the other person involved in the link -- the person the link is "to."

The "W1 W2 W3 ... Wn" are variables describing characteristics of the link. Some of these may be categorical variables; some may be scaled as ratio-level indicators of magnitude; others may be scaled as proportions. In some cases, if the scaling is not completely appropriate, the capability of the program to perform arithmetic transformations on the variables may be used to remedy the problem. An observation may extend over more than one line of data if necessary.

The **index data** has the following format:

ID# V1 V2 V3 ... Vn

The "ID#" is the identification number of the respondent this line of data describes.

The "V1 V2 V3 ... Vn" are variables describing characteristics of the person whose ID number begins this line of data. These must all be categorical variables. An observation may extend over more than one line of data if necessary.
Both data files must be in **fixed format**. This means that if a variable occupies columns 3 and 4

on the first line of data, it must be in those columns on all lines of data. It is not necessary to separate variables with spaces, but in some cases doing so may improve the readability of the files and make data entry and error correction easier to accomplish.

Variable lists

The purpose of the *variable list* is to tell the program the *names* of variables and the *columns* each variable occupies on each line of data. Both lists begin with ID number variables and are terminated with the word "End". The variable list for both files may also have *value labels*, and the one for the **link file** may have *variable type codes*.

The variable list has one entry for each variable being described. The basic format of an entry is a variable name followed by a column range. The name is what the program will call the variable that occupies the columns in the specified range. Take, for example, the entry:

```
Date        (6-7)
```

This indicates that the variable "Date" is found in columns 6 and 7 on the line of data.

The first letter of the variable's name must be in the first column on the line in the variable list part of the file. The column numbers must be enclosed within ordinary parentheses, with a hyphen (minus sign) separating the starting and ending columns of the field occupied by the value.

If you want to read the variables Age, Sex, and Height, your list might look like this:

```
Age         (6-6)
Sex         (8-8)
Height      (9-10)
End
```

The column numbers are enclosed in parentheses and appear after the variable name. For one-column variables, like "Age" and "Sex" in the example above, the column number is repeated twice, with a hyphen in between. Variables in the list must come in the order in which they appear in the data. (i.e. Sex, column 8, cannot appear in the list before Age, column 6.)

- The variable lists for both files begin with ID numbers. In the **index file** there is one ID number variable, "**ID**". For the **link file** there are two -- one for the respondent, and one for the other individual involved in the link. The first one is always "**ID1**" and the second is "**ID2**". The ID number variables must always be the first ones on the list; they must appear before any other variables you want the program to read.

- Following the ID numbers will be the other variables you wish to include in the analysis. (If there are some variables you don't expect to ever use, you shouldn't include them in the list.)

- Put the variable names in the same order in the list as the variables appear in the file (i.e. a variable in columns 7-8 should appear in the list before a variable in columns 9-10).

- If you wish, you may add a comment after the closing brackets around the column numbers. Anything after the closing bracket[2] will be ignored by the program.

- The variable list must be ended with the word "End" on a line by itself.

A sample variable list for a link file is shown below. This list says that there are eight link variables altogether (counting the two ID numbers for each link). The first one after the ID numbers is the frequency of interaction (a general magnitude variable).

```
ID1           (1-3)
ID2           (4-7)
Frequency     (9-9)      Q#8  1=1/mo 2=2-3/mo 3=1/wk 4=2-3/wk 5=1/day
initation     (10-10)    Q#9  1=self 2=half-half 3=other
school        (12-12)    discuss school 0=none 1=little 2=some 3=much
personal      (13-13)    discuss personal issues          "
politics      (14-14)    discuss politics                 "
em suppt      (16-16)    emotional support                "
END
```

The next variable tells who initiates the contact (a categorical link variable). The next four (all proportional variables) tell how much of the time in contact with this person is for each of four content areas -- school-related problems, personal problems, political issues, and emotional support.

The data follows immediately after the "End". There are no end-of-file codes necessary, and you do not have to provide a format specification. The program makes its own format specifications from the information in this list. If there are variables you know you won't be using in any analysis

[2] except *variable type codes*, explained below

with FATCAT, you should simply omit the description of that variable from the list. (Note that in the example above, the contents of columns 11 and 15 were not described. If there is data in these columns, it will be ignored by the program because it wasn't described on the variable list.)

The variable list in the **index file** differs from the one in the **link file** in two important ways:

- First, the **index file** has only *one* ID number per record, since the information on a line of data refers to only one individual. The **link file** has *two* ID numbers per record, because it takes two individuals to have a link (it takes two to tango...)

INDEX FILE		LINK FILE		
ID	(1-4)	**ID1**	(1-3)	
Age	(6-6)	**ID2**	(5-7)	
Sex	(7-7)	Frequency	(9-9)	[m]
Position	(8-9)	Initiation	(10-10)	[c]
Division	(10-11)	Work	(12-12)	[p]
Rank	(12-12)	Personal	(13-13)	[p]
	:		:	
	:		:	

- Second, you can have *variable type codes* in the **link file** (but *not* in the **index file**). A variable type code tells the program what kind of link variable it is dealing with. Remember there are three kinds of link variables: **C**ategorical, general **M**agnitude, and **P**roportional. The variable type codes are the letters "C", "M", and "P"; they are enclosed in square brackets, "[" and "]"; and they appear immediately after the column numbers that tell where the variables are located.

  The variable type codes tell the program what kind of variable it is dealing with. It uses this information to know which variables should be included in the menus it creates to help you select the variables you want to use in your analysis. If you have a categorical link variable, for example **INITIATION**, you would put the variable type code **[C]** after the column numbers for this variable. The program would recognize **YEAR** as a categorical variable, and it would put this variable only on the menus where a categorical variable would be a valid choice. The variable list for the link file above shows an example of variable type codes. An important consideration to make when deciding whether or not to use variable type codes is this: If you use a type code for a variable, that variable will only be available for analysis when a variable of that type is appropriate. Variables with no type codes are available for all analyses, even though they may not be appropriate for some. If you want the additional flexibility , or if a variable can be legitimately used in more than one way, don't use a type code.

Value labels

For all **index** variables and **categorical link** variables, you can include *value labels* in the variable list. To do this, put a line containing a slash (/) in the first column immediately after the line with the variable's name and column numbers. In the following lines you put the values and the labels you want to associate with the values. To indicate the end of the value label list, put another line with a slash in the first column.

If you don't use value labels for a variable, the program will use either the values themselves or labels that look like "CAT 1", "CAT 2", etc., as labels for rows and columns in the results, Here is an example:

```
ID          (1-4)
Years       (6-6)
/
1 0-5
2 6-10
3 more
/
Tenured     (7-7)
/
1 No
2 Yes
/
Position    (8-9)
/
1 Editor
2 Production Coordinator
3 Manager
4 Division Head
/
Division    (10-11)
    :
    :
```

Pseudo-variables

There are two *pseudo-variables* that look a bit like variable names. The first is "**END**" which signifies the end of the variable list and the beginning of the data. The Second is "*****", which indicates the end of a record of data.

If your index file has two lines of data per person, you will need a way of telling the program where the new line begins. You do this with a "*****", as shown in the example below.

```
  :
  :
Class (76-76)
/
1 Manager
2 Clerical
/
*
```