Paper DM03

Analyzing Large Social Networks with MP Connect, SAS/IntrNet®, and %DS2CONST

Shane Hornibrook, Charlotte, NC

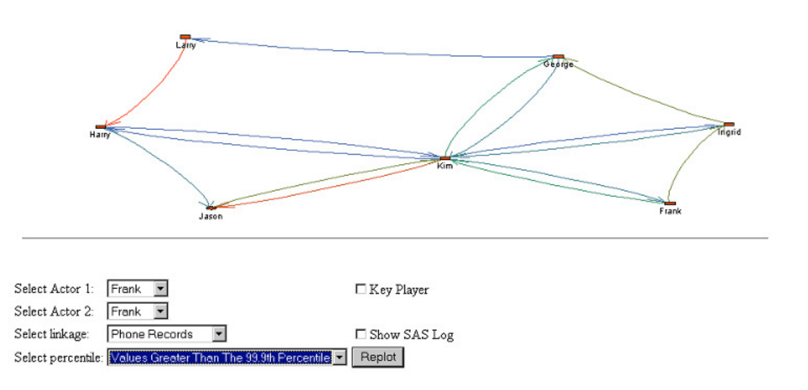
ABSTRACT

There are few analyses that rival the requirements of an interactive, web-delivered Social Network Analysis (also known as Link Analysis). Essentially the question posed to the data is “Who is connected to whom and how strongly are they connected?” Linking disparate multi-million-row databases and interactively reporting interesting links is a task that was, until recently, daunting. Historically, analysts have lacked the computational resources and analytical methods necessary to concisely report on the linkages between “key players” in networks of interest, preventing whole-network approaches. SAS/IntrNet® reports integrating the Constellation Applet and utilizing the parallel-processing capabilities of MP Connect provide an ideal method for returning relevant, timely Social Network Analyses from massive databases. Investigators can subsequently select “interesting” network components and individuals for further visualization and investigation.

INTRODUCTION

Social Network Analysis is a graphical and mathematical science that illuminates the linkages between entities such as persons or businesses. Business and government agencies contain vast quantities of information that must be filtered and summarized in order to understand the structure within the data. This structure is often best understood by using spider web like graphs variously referred to as “Link Charts”, “Organizational Network Analyses”, and “Social Network Analyses.” These graphs contain *nodes*, representing an entity such as a person or business. Joining the nodes are lines called *links* or *edges*. These edges represent connections between the nodes.

Example of SAS/IntrNet webpage integrating the Constellation Applet



Select Actor 1: Frank
Select Actor 2: Frank
Select linkage: Phone Records
Select percentile: Values Greater Than The 99.9th Percentile
☐ Key Player
☐ Show SAS Log

Often, an investigator may be interested in the connections between one or more entities. From this initial analysis, the investigator may wish to discover more of the network that connects at the boundaries of the starting network. Rather than require an investigator to make multiple requests, with the attendant multiple waiting periods, an alternative is the use of online analytical tools such as SAS/IntrNet.

Intuitively, Social Network Analysis data sources are rarely small (row-wise) or simple; otherwise, there would be little need for a link chart, as the connections and structure of the data would be easy to summarize or browse in tabular format. With the volume and often multiple-source data that is utilized, an online reporting tool must use the most efficient back-end data extraction and processing code possible.

This paper demonstrates methodologies for highlighting the perhaps hidden structure of the linkages within your data.

THE PROBLEM

SEEING THE MOUSE, BUT MISSING THE ELEPHANT IN THE ROOM

Long have investigators known that finding linkages between “interesting” persons is the key to delineating (and breaking) networks. Such investigations have, in the past, taken what could be described as a “shotgun” approach, or better yet, a “shot in the dark”. An investigation may be triggered by a tip, an arrest, or mention of a connection between one entity and another on a report. Such methods are useful, and fit into the category of hypothesis testing. They are also easy to perform, as they ask a specific question: Is Party A connected to Party B? However, this type of reporting suffers from a shortfall: while you are concentrating on a set of specific links you may not notice a neighboring, massively interconnected network, the “breaking” of which would provide an even greater payoff for a comparative amount of investigation.

THE DATA EXTRACTION – MP CONNECT

The object of this paper is to present a framework whereby Social Network Analysis may be performed in "real-time" via web browser. This methodology removes the huge turn-around time on typical ad hoc Social Network Analysis requests and places a powerful online analytical tool in the hands of investigators.

In order to present Social Network Analyses in "real-time" to investigators, the data must be summarized. Often, database management systems (DBMSs) are the repository for financial, associative, and billing information that is of interest to investigators.

In large business and government, each department is likely to have its own "data silos". A lack of a centralized data warehouse may prevent or otherwise impinge on a cross-discipline analysis. In addition, the data fields that may be of great interest to a network analyst may not be carried from the source systems (e.g. mainframes) to a data warehouse. Data may often be spread across databases and operating systems without regard to the requirements of data analysis.

One of SAS's strengths lies in its ability to extract data regardless of format. In the context of widely disparate data sources, MP Connect, SAS's multiprocesser utilization tools can turn this multiple source "problem" into an advantage! Selecting from different data sources are independent tasks, and can be run asynchronously. If your data must be extracted from more than one table, or more than one warehouse, MP Connect is the solution to improve the elapsed time between the user selecting a set of "Players" to display, and getting results back to the SAS/IntrNet session and web browser. MP Connect provides methodologies for submitting multiple pieces of SAS Code to multiple remote servers.

The example to the right presents a typical framework for spawning multiple data extraction processes. In this code snippet, we are interested in transactional data held in a Teradata database, physician visits held in an IBM DB2 database, phone records held in SAS files, financial data held in flat files, and ownership data held in a separate IBM DB2 database.

Each independent task is submitted in its own remote submission block, delineated by "rsubmit" and "endrsubmit." Each of these rsubmit blocks spawn separate processes. The controlling SAS job does not wait for one rsubmit block to return results before submitting the next rsubmit block. It is only at the waitfor statement that the SAS session waits for the submitted processes before continuing.

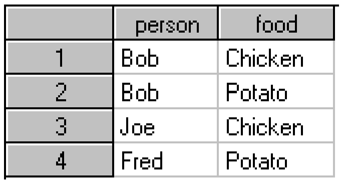
```
/* First SQL task: get shopping data
   from Teradata  */
rsubmit teradata wait=no;
proc sql;
   connect to teradata(&teraconnparams.);
   create table shopped as
   select  ...
   from connection to teradata (
   );
   disconnect from teradata;
quit;
endrsubmit;

/* Second SQL task: get physician visits
   from IBM DB2 */
rsubmit db2 wait=no;
   proc sql;
       connect to db2(&db2connparams.);
       create table visits as
       select ...
       from connection to db2 (
       ...
       );
       disconnect from db2;
   run;
endrsubmit;

/* Third SQL task: get phone records
   from SAS data sets */
rsubmit main wait=no;

libname phone 'path-to-sas-library';
proc sql;
   create table called_from as
   select   ...  ;
quit;
endrsubmit;

/* Fourth SQL task: get banking records
   from flat files*/
rsubmit bank wait=no;
data banking;
   infile ...;
   ...;
run;

endrsubmit;

/* Fifth SQL task: get business ownership
   records from separate db2 process */
rsubmit owner wait=no;
proc sql;
   connect to db2(&db2connparams2.);
   create table owners as
   select * from connection to db2 (
   ...);
quit;
endrsubmit;

/* Wait for all rsubmits to complete */
waitfor _all_ teradata db2 main bank
owner;
```

If each job executes in series, the wait time would be longer than the attention span of most web-based users. Using MP Connect we have SAS submit each job to separate servers, and wait for the results to return from all. This parallel processing ensures that the data gathering stage takes only the run-time of the single longest submission, not the sum of all queries.

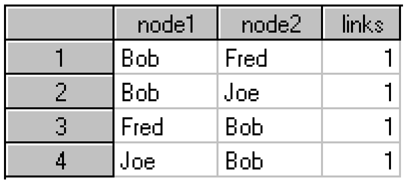
BUILDING NODE AND LINK DATASETS

When the basic data has been extracted, there are very few steps required to manipulate it into format required for the %DS2CONST macro. As with many other Social Network Analysis packages, the Constellation Applet requires **node** and **link** data sets. The **node** data set is a distinct listing of the unique nodes and their unique identifier, as well as optional descriptors for the properties of the nodes. The **link** data set structure is presented below. This link data structure is called a "dyad." A dyad is a data form that shows one association per line, with the connections between the source node and an associated node.

To the right is a typical record layout from a transactional database. In this example, one column is the identifier (person), and the other column is their favorite foods.



	person	food
1	Bob	Chicken
2	Bob	Potato
3	Joe	Chicken
4	Fred	Potato



	node1	node2	links
1	Bob	Fred	1
2	Bob	Joe	1
3	Fred	Bob	1
4	Joe	Bob	1

However, the %DS2CONST macro is expecting data in a standard dyad form, shown to the left. This format has one record per linkage, showing for example, that Bob and Fred are linked by one food in common, but there is no direct linkage between Joe and Fred. As this is an "undirected" dyad we can see that the linkages from Bob→Fred have the same counts as from Fred→Bob.

To transform the data from the standard form to a dyad, we need to join the data to itself. This may seem like a daunting task when dealing with multi-million record databases. However, modern relational databases, with properly built indexes can build dyads and return these results quickly. Some simple SQL (right) can be used, such as this undirected summary of the count of different foods in common.

```
/** dyadic link data set **/
proc sql;
create table dyad as
select a.person as node1,
       b.person as node2,
       count(distinct a.food)
              as links
from fav_foods as a,
     fav_foods as b
where a.food=b.food
  and a.person^=b.person
  and additional qualifiers
group by node1, node2;
quit;
```

Indexes *must* be present in order to return results quickly. Examine the tables and columns that will be your join variables, and ensure that the indexes required exist. Also, it is probably obvious, but should be stated nonetheless, that values common to many entities should not be used as linkage variables. For example, linking persons by gender would be an extremely poor choice as approximately half of your data would be joining to itself - a data set containing 100 males would create a link table of 9,900 links! When joining a large database to itself, having a limited number of possible records returned is very important to success.

```
/** Ninety Fifth percentile
    as a threshold for the
    strongest links  **/
proc univariate noprint
   data=links;
   var link_cnt; /** Link Counts */
   output out= ninety_fifth
          p95=p95;
quit;

/** selecting the 95th into a macro
    variable and where-clausing runs
    in 1/5 the time of a merge **/
proc sql noprint;
   select p95 into :p95
   from ninety_fifth;
quit;

data links;
   set links(where=(link_cnt>&p95.));
run;
```

THRESHOLDING

An important consideration when dealing with entire-network analysis is the elimination of spurious connections. In the above SQL, the *additional qualifiers* statement could include qualifiers to eliminate the possibility of joining based on "dummy" values or null values. Also, one should consider thresholding to put a damper on the "noise" and only highlight the most significant connections. Some arbitrary, yet often useful, thresholding include rankings such as the "Top 25" connections. In relational databases such as DB2 such data can be restricted by SQL statements such as "fetch first 25 rows only" with "order by" clauses. In SAS we can order by the link of interest and use an "obs=" qualifier. Such thresholding need not be as arbitrary as a "Top N", but can be based on percentiles, peer-groups, geographic profiling, and expert knowledge built into the system.

Some network centrality metrics such as degree centrality can be easily calculated in SQL statements, and can be used in thresholding. For more complex network analysis measures and determination of "normal" network connectivity data profiling should be performed and results stored in a profile database. This network profiling quantifies the levels of connections and is a tool for determining what is considered "normal". The payoff comes when an analysis is to be performed more than once: a profiling data set can be used to determine the level and type of connections that could be expected in the data. For example, if the number of linkages of a certain type is a suspicious linkage, but there are many low-level contacts that are not interesting to investigators, simple thresholding using the profile data set can suffice.

SIX DEGREES OF KEVIN BACON: NETWORK COMPONENTS AND SNA ALGORITHMS

Methods for determining commonly applied Network Analysis algorithms, such as determining flow through a network, include SAS/OR's PROC NETFLOW, and PROC IML matrix-based algorithms. When such techniques are not available there are other DATA step and SQL-based techniques for determining connections within a network. However, this paper cannot begin to delve into the mathematical techniques behind Social Network Analysis. For general theory and methodologies please refer to the book *Social Network Analysis* by Wasserman and Faust.

For discovering a distinct network component without having to retrieve all data from massive databases, one can use a simple SQL with an iterative sub-select. When an analysis is interested in only one entity and their immediate linkages, and perhaps *their* immediate linkages, a "snowball" technique can be used. Essentially it can be described as a "sub-select ad nauseum". For example, we could be interested in selecting the people who received a call from Bob, and subsequently made a phone call, and who those people called, and so on and so on …

In the context of the party game "Six degrees of Kevin Bacon", the SQL would look somewhat like the code to the right.

In practice, when the SQL is written as sub-selects rather than separate SQL statements, timely returns from large relational databases are very possible. In runs on large data sets (>30 million nodes, in 150 million records) the snowballing technique for finding connections and 'connections to connections' return results within a timeframe expected by web users.

Typical Social Network Analyses have not utilized large databases, mainly for the reason that these databases are not accessible to researchers. Thus, many tools are not suited to the memory and time requirements of loading 30-million node data set into memory and building network metrics. Generating SAS code for an iteratively sub-selecting macro (not shown) is straightforward, and results are returned quickly. If a particular path is desired, such as from "Kevin Bacon" to another actor, a macro exit is taken when the "target" is reached. After this minimum data has been extracted from the RDBMS, further metrics can be calculated to determine the sub-networks statistics.

```
/** Simplified example of iterative
    selections from the same table
    to build a data set that grows
    outward from one actor. **/
proc sql;
/* first level of selection pulls
   all movies with Kevin Bacon **/
   create table kevins_movies as
   select distinct movie_name
   from big_table
   where actor_name="Kevin Bacon" ;

/* all the actors Kevin Bacon
   sharing a movie with Kevin Bacon */
   create table second_level as
   select a.actor_name,
          a.movie_name,
          a.production_date
   from big_table as a,
        kevins_movies as b
   where a.movie_name=b.movie_name;

/* all the actors that
   were subsequently in a
   movie with these actors */
   create table third_level as
   select a.actor_name,
          a.movie_name,
          a.production_date
   from big_table as a,
        second_level as b
   where a.movie_name=b.movie_name
     and a.production_date>=
         b.production_date;
   /** and so on ...**/
   quit;
```

If you are interested in seeing only the local network for a target you can use a similar 'snowball' technique, iterating outward your target, exiting the macro when a set number of iterations are completed.

If you are interested in seeing the full extent of the network component you can iteratively call the macro until one SQL returns the same number of results as the previous SQL statement -- indicating that the extent of the network has been discovered.

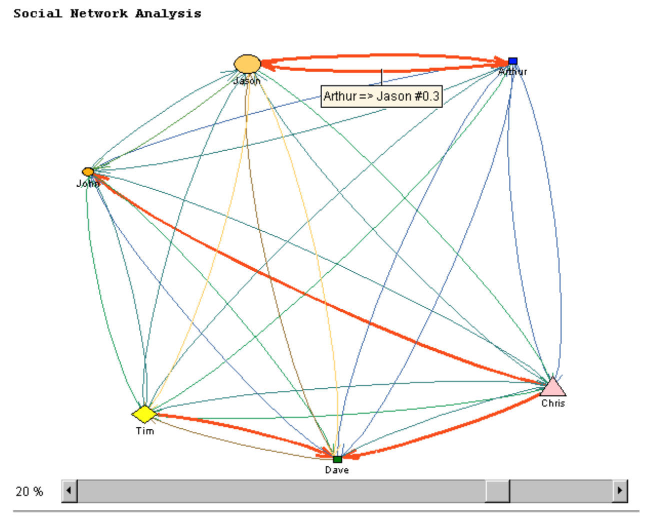
MERGING THE FINAL DATA SET

The results of the multiple extractions from disparate data sources are returned to SAS at the end of the MP Connect code. Once these individual link data sets are built and returned from the remote sessions, the final data set can be merged using a straightforward data step merge. In the final merge different link types should be weighted to arrive at a final score for the linkage between any two entities. For example, a linkage indicating that the entities "shop at the same store" would probably carry less weight than "own the same business."

THE VISUALS: %DS2CONST MACRO VARIABLES AND THEIR EFFECTS

The %DS2CONST macro and Constellation Applet provide a wide array of user-definable properties for the selection and display of network data. The display properties are for *nodes* and *links*. The main graph types are *Arc, Hierarchical,* and *Associative.* For advanced users, X and Y coordinates for nodes can be specified: this is useful for applying novel network layout algorithms.

Associative graphs display Nodes and Links based on their weighted values; Node size and link width can represent the relative size of the node and link values.



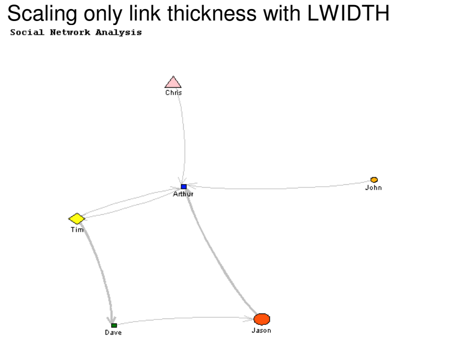
For this author's purpose, the most useful type of graph drawn by the Constellation Applet is the "*Associative*" graph. In this graph the nodes and edges can be scaled, with edge thickness representing strength of association and differing node sizes representing quantitative properties of the node, e.g. overall number of billings, or criminal convictions.

NODES

The Constellation Applet allows specification of variables that define each node shape, node background color, and node outline color. If the user wishes to display a quantitative variable as a node size, automatic scaling can be set, or user-defined formats can be applied. Qualitative properties, such as whether a node represents a business, an owner, a manager, or an employee can be represented by node shape. Node coloring is conducive to representing qualitative properties such as criminal activity. In addition, properties such as font, font size, and font style can be specified.

Static node shapes can be specified using the macro variable NSHAPE. If you wish to vary node shape you can specify the variable that contains the shape name using the NODESHAP macro variable. The shapes available are circles, diamond, squares, or triangles. Although there are a limited number of shapes, when combined with the unlimited color scheme, these shapes should be suitable for representing a variety of qualitative properties, such as "business", "owner", "manager", "fraudulent", and "not-fraudulent".

Scaling only link thickness with LWIDTH

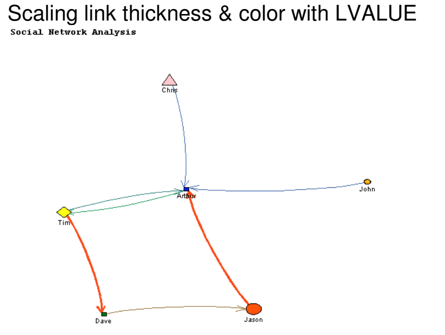


LINKS OR EDGES

The associative graph is useful for showing directional flow. In a directed graph, an arrow is drawn from one node to the other if there is a linkage. For example, if node A telephoned node B, this would be represented by an arrow from A→B. If node B also telephoned node A an *additional* arrow is drawn from B→A. These directional graphs allow investigators to see the properties of the connections; whether they are unidirectional or bi-directional. In small networks this can visually indicate the more 'important' entities.

The edges can be shown as different thickness according to edge weights. This option is only available when "Associative" graphs are chosen as the display method.

Scaling link thickness & color with LVALUE



Examples of usage of the associative graph are many. In the context of fraud analysis, these graphs can show the flow of money from one entity to another. In money laundering schemes, we can show the pathways as funds move from one account to another. By linking up these pathways we can determine the routes that funds have traveled and who is directing funds to whom, even by indirect means.

SAS's %DS2CONST macro provides variables for showing the thickness of these lines related to the quantitative properties. To allow SAS to automatically scale the line thickness AND color according to the same link weights, the macro variable LVALUE should equal the variable name containing the link weights. If you wish to scale the link width and link colors separately you can specify one variable for link width (LWIDTH) and another for link color (LCOLOR). Colors can be specified in hex format (e.g. FF0000), or by color name (RED).

In cases where a more sophisticated scaling methodology would be advantageous you can specify your own formats for application to your link data. To specify that formatting should be applied to the link values, use LCOLFMT =*your_format_name*. This has the obvious advantage of not having to modify or add to any data set.

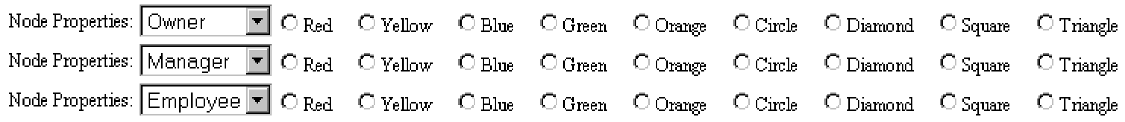
```
/** link formatting **/
proc format;
value   linkfmt low-100   = 1
                101-250   = 2
                251-high = 3;
quit;
```

Using PROC FORMAT, you can create a customized color gradient format according to the link strength. However, since linkage strength can easily be represented by thickness, the color property is best left to a categorical variable: an example from fraud analysis could be “Blue” for previously investigated and “okayed” linkages, “Gray” for linkages under ongoing investigation, and “Red” for linkages that have not been investigated.

Navigation within the graph applet is accomplished by a combination of keyboard and mouse shortcuts. The navigation possible includes panning, zooming, selection of nodes, and hyper-linking. In addition, the graph applet provides useful interactivity, such as moving a node. Often, in complex network drawings it may be difficult to determine the origin of one or more edges, or perhaps nodes are placed too closely. To move a node, merely select it with a click, then hold down Shift & Control and use your mouse to re-position.

INTEGRATING SAS/INTRNET QUERIES AND REPORTS

Using a SAS/IntrNet based query page, the appearance of the diagram can be modified. Essentially we want to connect each user-modifiable display variable (color, size) to a respective form variable in a SAS/IntrNet generated query page. Ideally we would determine the distinct values of the categorical variables in our data and present these variables as options. So, for example, if the category included “owner”, “manager”, and “employee” we could provide these values to a SAS/IntrNet based form. The user can then select the shapes and colors that each value will be given. This is accomplished by giving the user options based in HTML drop down menus and radio buttons. The values of the drop downs and radio buttons are submitted to the SAS job as macro variables.

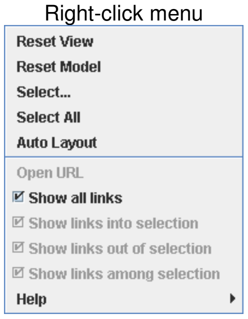

Node Properties: Owner ○ Red ○ Yellow ○ Blue ○ Green ○ Orange ○ Circle ○ Diamond ○ Square ○ Triangle
Node Properties: Manager ○ Red ○ Yellow ○ Blue ○ Green ○ Orange ○ Circle ○ Diamond ○ Square ○ Triangle
Node Properties: Employee ○ Red ○ Yellow ○ Blue ○ Green ○ Orange ○ Circle ○ Diamond ○ Square ○ Triangle

The values that can be set by the %DS2CONST macro span three pages. The most important macro variables for display are the link colors, link thickness, node colors, node size, and node shape. Simple SAS/IntrNet query pages can generate HTML form variables for distinct values of categorical variables from the node data set. These variables are used to specify color and shape formats for the values of these nodes. This provides additional functionality for investigators that are interested in specifying the shapes and colors that should be associated with each type of node.

SELECTING INTERESTING NETWORK COMPONENTS AND INDIVIDUALS

The Constellation Applet provides a right click menu with functionality that includes resetting the view, selecting links, or selecting only some of the edge types (into or out of the selected link).

Right-click menu


Reset View
Reset Model
Select...
Select All
Auto Layout
Open URL
☑ Show all links
☑ Show links into selection
☑ Show links out of selection
☑ Show links among selection
Help

The %DS2CONST macro provides a methodology for creating URL links from each node. By building a variable containing a URL, and using the NURL=*variableName* macro call, the developer can provide further functionality. Additionally, link IDs can be selected and passed to user-defined JavaScripts. By integrating the %DS2CONST macro and Constellation Applet with other reports, the analysts are able to easily bring the observations they make while browsing the output of the Constellation Applet into other reports and further drill-down into detail data. This URL linkage is a bridge to other reports, providing functionality that is lacking in many other Social Network Analysis applications.

The actions that occur when an investigator clicks a link are almost unlimited, but the following are suggestions:

- Additional reports are displayed for this node: Node ID used as a qualifier or ‘where clause’ for other reports
- Node gets added to database of ‘interesting nodes’ for further investigation and other queries
- Detailed data for node is shown
    - Billing history, Business ownership
    - Case file (Photos, Fingerprints)

CONCLUSION

Integrating SAS/IntrNet and the %DS2CONST macro with back-end processing using MP Connect provides a "real-time" methodology for web-based Social Network Analysis utilizing very large databases.

ACKNOWLEDGMENTS

Thanks to Denise Figliozzi and David Steves, the SESUG 2006 Data Presentation section chairs.

Thanks to Marje Fecht of Prowerk Consulting (http://www.Prowerk.com) for her assistance.

Thanks to Patricia Serrito, the SESUG 2007 Data Mining section chair.

DO YOU EVER GET THAT FEELING OF DÉJÀ VU?

This paper was originally written for SESUG 2006. Presentation of this paper was unfortunately delayed until SESUG 2007. Thanks to the SESUG committees for accepting my paper for SESUG 2006 and allowing me to present at SESUG 2007. Original paper: http://www8.sas.com/scholars/Proceedings/2006/DataPresentation/DP04_06.PDF

RECOMMENDED READING

The SAS Institute, *"Macro Arguments for the DS2CONST, DS2TREE, DS2CSF, and META2HTM Macros"*
http://support.sas.com/91doc/getDoc/graphref.hlp/a002606467.htm

The SAS Institute, *"Sample 1168: Constellation applet example with DS2CONST macro"*
http://support.sas.com/ctx/samples/index.jsp?sid=1168&tab=code

The SAS Institute, *"How to Use This Constellation"*
http://support.sas.com/rnd/datavisualization/webgraphs/v9_1/en/constchartapplet/mainmenu.htm

Wasserman and Faust, 1994, "Social Network Analysis: Methods and Applications" Cambridge University Press.

Scott, 2000, "Social Network Analysis: A Handbook" Sage Publications Inc.

Sparrow, MK, 2000, "License to Steal: How Fraud Bleeds America's Health Care System" Westview Press.
Commentary: Contains a description of a Social Network Analysis performed in Florida in 1993, related to Medicare Fraud detection (Page 243). In addition, a book review has been published in the FBI Law Enforcement Bulletin - January 2002 (http://www.fbi.gov/publications/leb/2002/jan02leb.pdf).

CONTACT INFORMATION

Shane Hornibrook
Charlotte, NC
Email: sesug_paper @ ShaneHornibrook.com
Web: http://www.ShaneHornibrook.com/
The most up-to-date version of this paper and associated SAS Code are available at
http://www.ShaneHornibrook.com/sesug2007/