Visualizing Two Social Networks Across Time with SAS[®]:

Collaborators on a Research Grant vs. Those Posting on SAS-L

Larry Hoyle, Institute for Policy and Social Research, University of Kansas

ABSTRACT

This paper explores the visualization of two different social networks across time. One consists of people at 23,827 email addresses posting 267,209 messages to 82,279 threads on the email list SAS-L from 1996 through August of 2008. The other, much smaller but more complex, social network includes faculty, staff and students working on the Kansas NSF EPSCoR grant. Capabilities of the %DS2CONST macro and the SAS/GRAPH Annotate facility are compared. The use of Enterprise Guide for managing the process is described.

INTRODUCTION

Previous papers have discussed using SAS to visualize social networks (Hornibrook) and the computation of network statistics (Johnston et al). This paper focuses on the visualization of two particular social networks across time. The processes used to gather and clean the data, to manage the intermediate datasets, and to produce the graphics are all described.

One of the networks consists of the people posting to the Listserv list SAS-L. Data were gathered from the archive website (http://listserv.uga.edu/archives/sas-l.html) as a collection of HTML files – one for each week since 1996. Each of these files listed a set of threads (subject lines from posted emails) and the emails posted within each thread for that week. People were identified by the email address of the posting. Content of the emails was not collected for this project. People were considered to be connected if they posted to the same thread within the same year. One of the challenges posed by this social network is its size - people at 23,827 email addresses posting 267,209 messages to 82,279 threads.
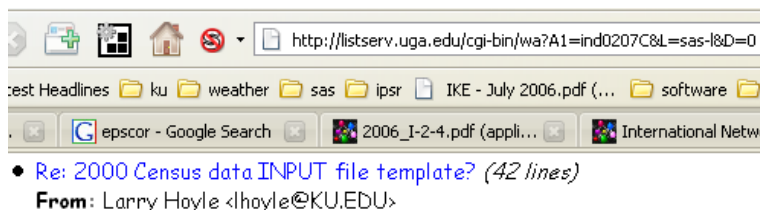
The other network consists of faculty, staff and students working on the Kansas NSF EPSCoR (KNE) phase V grant at several universities. Data were read from a spreadsheet with a number of separate worksheets. People were considered to be connected by different measures – co-authorship on a publication or other scientific product, being listed as co-investigators on a grant proposal, or the combination of the two measures. Some individuals also needed to be displayed distinctively as principal participants. Of interest was change in patterns of collaboration among research groups and institutions across time.

With both the SAS-L and EPSCoR data it was desirable to visualize changes in linkages across time. Multidimensional scaling, when possible, using the aggregate distances among the observations, ignoring time, produced a fixed spatial arrangement for plotting observations at each time period. A variation of this technique was used to generate coordinates for the 23,827 node dataset.

SAS-L

The SAS-L Archive website is organized into a collection of index pages that can be read by SAS using the URL access method. Each of these index pages (listed in http://listserv.uga.edu/archives/sas-l.html) was downloaded into a separate text (HTML) file as shown in the macro below.


```
%macro getWeek(week=0808d);
data _null_;
  d=datetime();
  call sleep(3,1);
  put "reading index for week &week" d datetime.;
  run;
filename weekurl url "&prepart.&week.&postpart";
filename weekfile "D:\projects\sugs\SGF2009\SAS-Lnetwork\sas-l_WEEKS\sasl&week..html";
data _null_;
  infile weekurl lrecl=500;
  file weekfile;
  input ;
  put _infile_;
run;
%mend getWeek;
```

The thread name and full email address of each poster, as seen in red in the box above, could be readily parsed from these data using SAS functions *htmldecode* and *prxparse*. A name and email address could then be separated from the full email address. Names can appear in these addresses in many forms: Last, First; First Middle Last; and so on. For the Multidimensional scaling it was convenient to standardize on a form that could be used as a SAS variable name. A Data Step procedure put the names into a standard form: LAST_FIRST_MIDDLE as shown in the table below.

|	StandardName	name	email	year	weekdate	posting	thread
4723	HOYLE_LARRY	Larry Hoyle	lhoyle@KU.EDU	2002	15JUL02	8	2000 Census data INPUT file template?

These data can be summarized to produce a table of counts by year or across all years which, in turn, can be used by the %DS2CONST macro to draw the nodes in a constellation chart.

	StandardName	PostsPerPerson
1	CASSELL_DAVID	8618

This table can also be joined with itself on thread=thread (see the Hornibrook papers) to produce counts of each dyad. These counts can be used by %DS2CONST to plot the links between nodes. In this case, an undirected graph, the count from-to is equal to the count to-from. Keeping the redundant records proved useful in computing triples so they were kept.

Project Designer   DYADCOUNTS (read-only)

|	StandardNameFrom	StandardNameTo	nPosts
1	DUNN_TOBY	SCHREIER_HOWARD	961
2	SCHREIER_HOWARD	DUNN_TOBY	961

These two summary files contain the information needed to plot a constellation chart, but 23,827 nodes is beyond the capacity of the constellation applet. Subsetting the data to include only those who have posted more than 100 times to the list since 1996 yields 291 people, which is within the capability of the applet. Three different views are shown below, displaying successively more links, links representing fewer common posts to the list. It is apparent that there is a core group with many posts to the same threads in common.
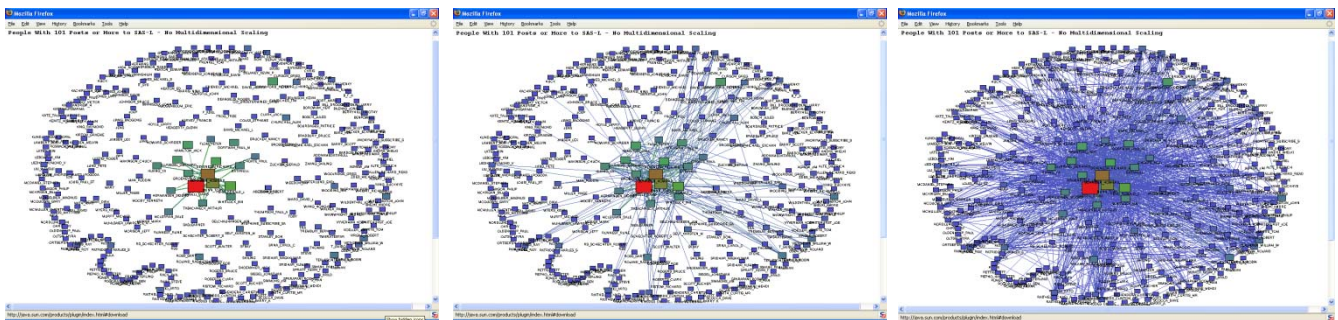
This core group will be familiar to those who read the SAS-L list, as many of them have been honored as "Most Valuable SAS-Lers" by members of the list. This makes sense in that those who are responsive to others on the list would be expected to have more postings in common than might less responsive participants.

The automatic layout performed by the constellation applet facility is convenient, but would not allow comparison across multiple invocations. Fortunately, the applet allows the user to specify a layout for the nodes. Computing a layout is relatively easy using PROC MDS.


```
proc mds
    data=SGF2009.TOPPOSTERSSIMILARITY
    out=SGF2009.TopPosters2D
    similar
    dimension = 2
    level=ordinal;
```

ct Designer   TOPPOSTERS2D (read-only)
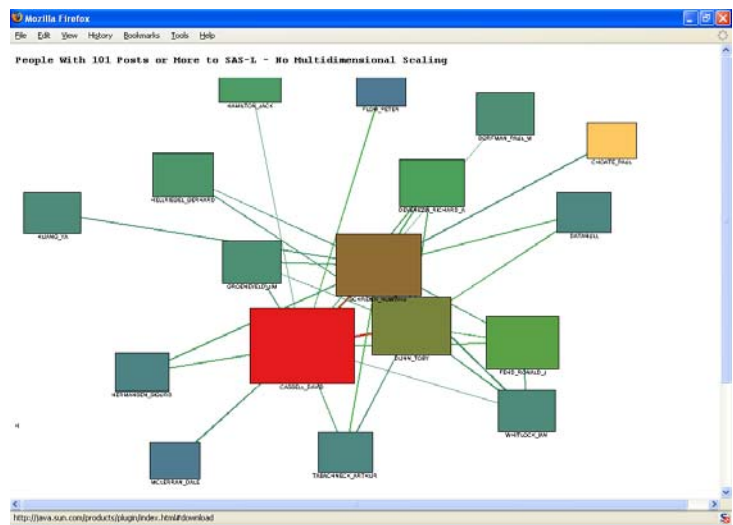
DIMENS	_MATRIX_	_TYPE_	_LABEL_	_NAME_	Dim1	Dim2
2	.	CONFIG	CASSELL_DAVID	CASSELL_DAVID	-0.193027413	0.1471488151
2	.	CONFIG	CHAHAL_RAHUL	CHAHAL_RAHUL	0.2577039875	-1.503109868
2	.	CONFIG	CHAKRAVARTHY	CHAKRAVARTHY	-0.272382977	0.5138918626

Note that this PROC requires input of an association matrix. The DATA Step code shown below computes that matrix. An SQL query first puts the list of names into a macro variable which can be used as a list of variable names and parsed with the scan function to find individual variable names. This is why the names were initially put into a form that could be used as SAS variable name. The lower right corner of the matrix is shown beside the code. Running the MDS procedure on a 23,827 X 23,827 matrix was not practical, so the group of top posters was used. In charts generated through the annotate facility, which does not have a limitation on the number of nodes, the less frequent posters can be placed around an outer circle or placed in proximity to the few in the scaled group to which they link.

```
  proc sql noprint;
    select distinct StandardName into :TopNameList separated by ' '
    from sgf2009.topPosters
    order by StandardName;
  quit;

  data SGF2009.TopPostersSimilarity;
   length &TopNameList 8 StandardName $ 50 ;
   length NameList $ 10000;
   retain &TopNameList;
   keep StandardName &TopNameList;

   retain row 0 NameList;
   array SN{*} &TopNameList;
   set SGF2009.TopPosterDyadCounts;
   by StandardNameFrom  StandardNameTo;

   if row = 0 then do;
     NameList = symget("TopNameList");
   end;

   if first.StandardNameFrom then do;
    row=row+1;
    do ixn= 1 to dim(SN);
      SN{ixn} = 0;
    end;
   end;
          /*  use scan to find the     */
          /*  index of StandardNameTo. */
    do ixn= 1 to dim(SN);
      w = scan(NameList,ixn," ");
      if w = StandardNameTo then do;
        SN{ixn} = sum(0, nPosts);
      end;
    end;

   if last.StandardNameFrom then do;
     SN{row} = .;
     StandardName = StandardNameFrom;
     output SGF2009.TopPostersSimilarity;
   end;
run;
```
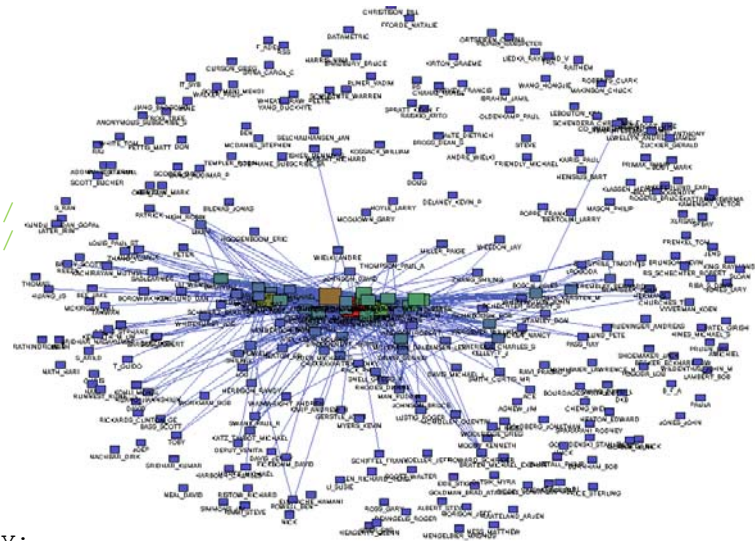
| ZHANG_SHILING	ZHANG_YU	ZUCKIER_GERALD	StandardName
.	1	0	ZHANG_SHILING
1	.	0	ZHANG_YU
0	0	.	ZUCKIER_GERAL

As with the automatic layout, the core group lies in the center of the 2 dimensional space. This same spatial arrangement can be used for multiple charts – as for instance, for separate charts for each year. Written to an animated GIF file with the GIFANIM driver, these produce an animation of the link structure across time where the nodes stay fixed and the only the links and node sizes change from frame to frame. Examples of this technique will be shown for the KNE data below.

The SAS-L group of top posters is highly connected as shown in the view below where all links are displayed. The ability to zoom and pan with the applet allows for exploration of the links in parts of the graph. Individual links may be identified. Inferences may also be drawn from the direction of the links. On the right below is a magnification of an area just above the core group. Note that most of the lines are nearly vertical – indicating links to the core group. Even within these 291 top posters many links are to a central group. This pattern will be seen again in a plot of all 23,827 people.
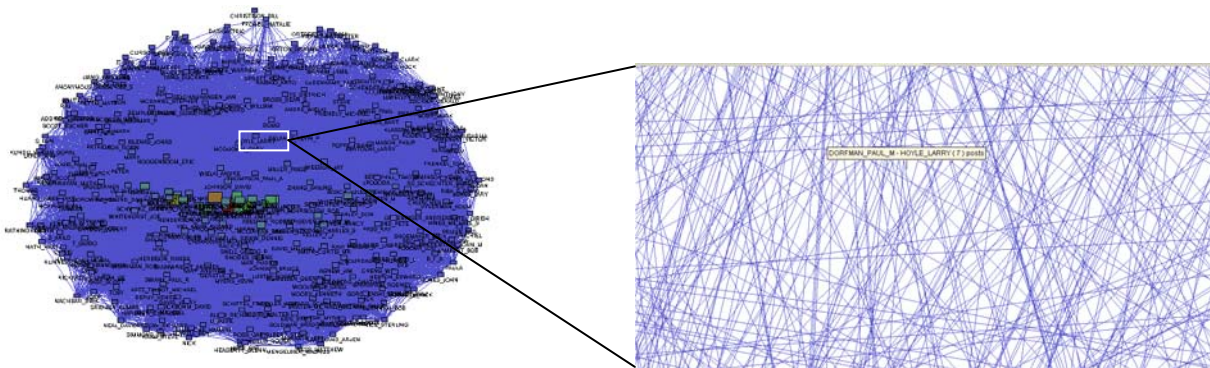
To display more than the number of nodes than the constellation applet can handle, it is possible to create annotate datasets and plot them with PROC GSLIDE. The figure below shows all people who have posted to SAS-L since 1996. Since it is impractical to do PROC MDS on 23,827 objects, those who have posted more than 100 times are shown in red with the same spatial arrangement as above, and the other 23,536 are arranged roughly alphabetically in a circular band in gray outside of that top posters group. Links to one of the 291 are shown in blue. Links that are between pairs of the 23,536 are shown in black. It can be seen that SAS-L is a "small world" in terms of common posts. Most threads have at least one posting from the 291. Code for generating the links is shown below.

All SAS-L People Posting to SAS-L since 1996, >5 common posts
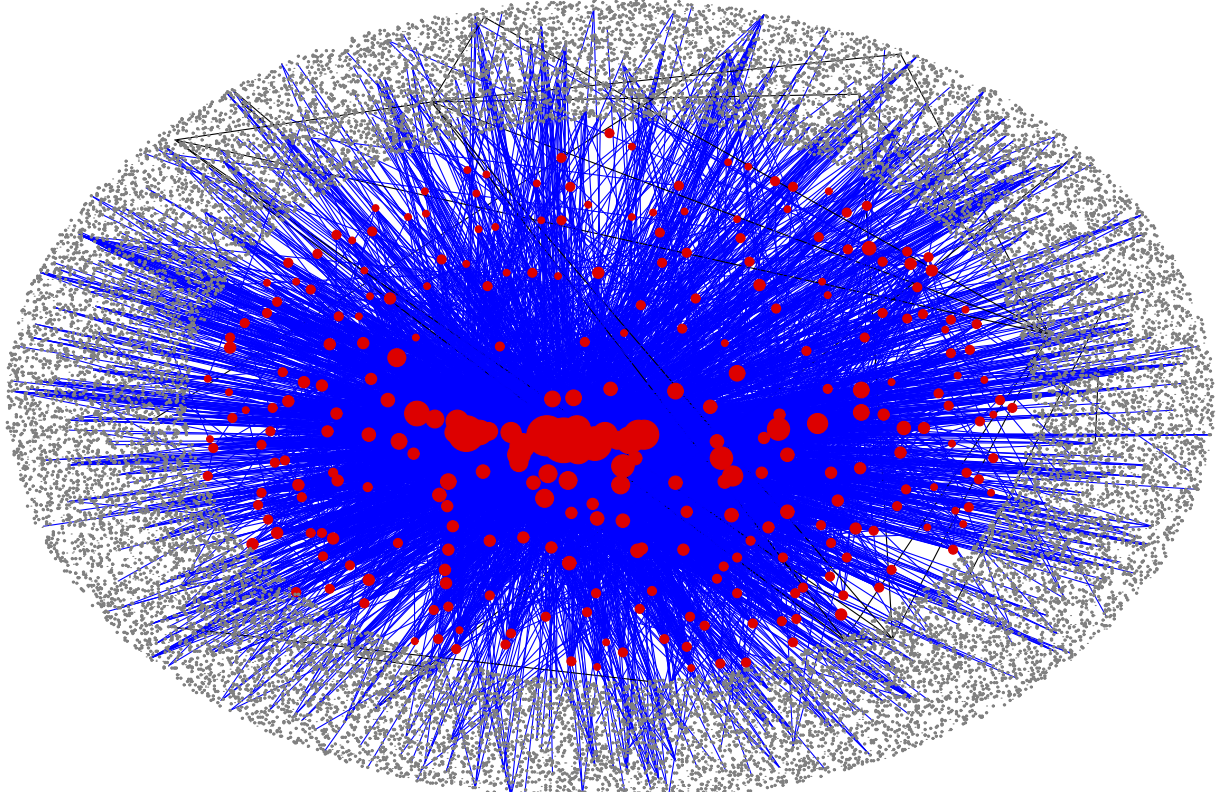
Those with more than 100 posts total are red, others are gray
Links with a top poster are blue, other links are black

```
data SGF2009.AllpersonLinkAnno;
   set  SGF2009.DYADSBYTHREADCOUNTTOTAL;
   where StandardNameFrom < StandardNameTo;
   retain order 0  color 'BLACK   ' line 1 size .05 XSYS  '3'  YSYS  '3'   HSYS  '3';
      FUNCTION = 'MOVE    ';                                       /*  MOVE   */
               x = 20 + 60*input(StandardNameFrom,topNameX.);
      if x=. then x=   5 + 90*input(StandardNameFrom,circleX.);
               y = 20 + 60*input(StandardNameFrom,topNameY.);
      if y=. then y=   5 + 90*input(StandardNameFrom,circleY.);
      when = 'B';         order=order+1;          output;

      FUNCTION = 'DRAW    ';                                       /*  DRAW   */
               x = 20 + 60*input(StandardNameTo,topNameX.);
      if x=. then x=   5 + 90*input(StandardNameTo,circleX.);
               y = 20 + 60*input(StandardNameTo,topNameY.);
      if y=. then y=   5 + 90*input(StandardNameTo,circleY.);
      if input(StandardNameTo,topNameX.) = . and
         input(StandardNameFrom,topNameX.) = . then do;
         color = 'BLACK   ';              size = .05;
      end;
      else do;
          color = 'BLUE    ';             size = .1;
      end;
      order=order+1;
      output;
    run;
```

4

The informats topNameX and topNameY place the 291 people as determined by the PROC MDS. The informats *circleX* and *circleY* above were generated for all standardized names. The code to generate those formats is shown below.

```
%let totalN=23823;

                      /*  create informats for the name to produce spatial coordinates  */
                      /*  generate x and y to place people in a circular band          */
                      /*   angular placement in the band is a function of  _n_          */
data SgF2009.CircleCntlout;
 set standardNames;

 length fmtname $ 40 label $ 256 start $ 50 end $ 50 type $ 1 HLO $ 11;
 keep   fmtname       label      start   end  type HLO;

 retain pi piOver2 ;
 retain type  'I' ;

 if _n_ =1 then do;
   pi = 4*atan(1);
   piOver2 = pi/2;

   fmtname = 'circleX';
   HLO = 'O';
   label = '.';
   start = '**OTHER**';
   end =   '**OTHER**';
   output;

   fmtname = 'circleY';
   output;
 end;

 HLO = ' ';

 alpha=2*_n_*pi/&totalN;
 r= 1 - (  .3*ranuni(1312121)  );
                             /* x=sin(alpha), y=cos(alpha) plots a  circle with radius 1  */
                             /*  where 0 <= alpha <= 2*PI                                  */
                             /* center the circle on .5 with radius .5                    */
                             /*  and place points randomly from .7 to 1 times the radius */
 circleX=r*sin(alpha)/2 + .5;
 circleY=r*cos(alpha)/2 + .5;

 fmtname = 'circleX';
  start = StandardName;
  end =   StandardName;
  label = put( circleX ,13.10);
  output;

 fmtname = 'circleY';
  start = StandardName;
  end =   StandardName;
  label = put( circleY ,13.10);
  output;
run;
```
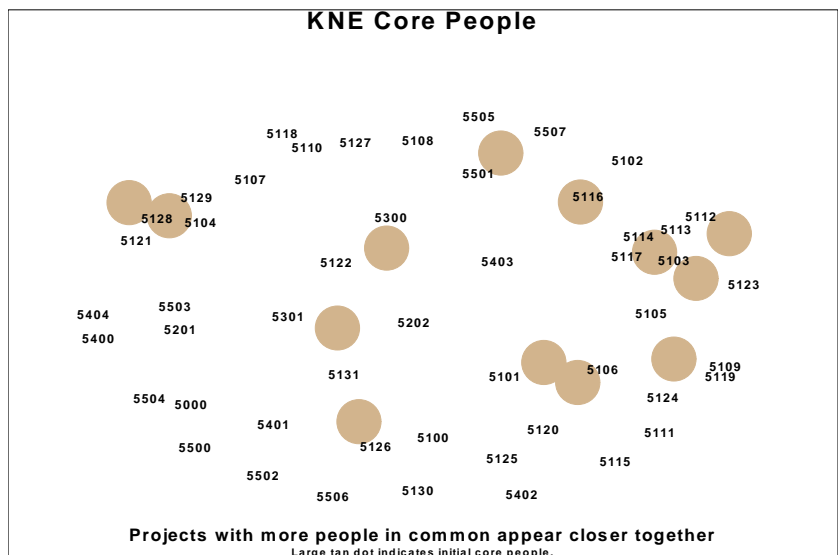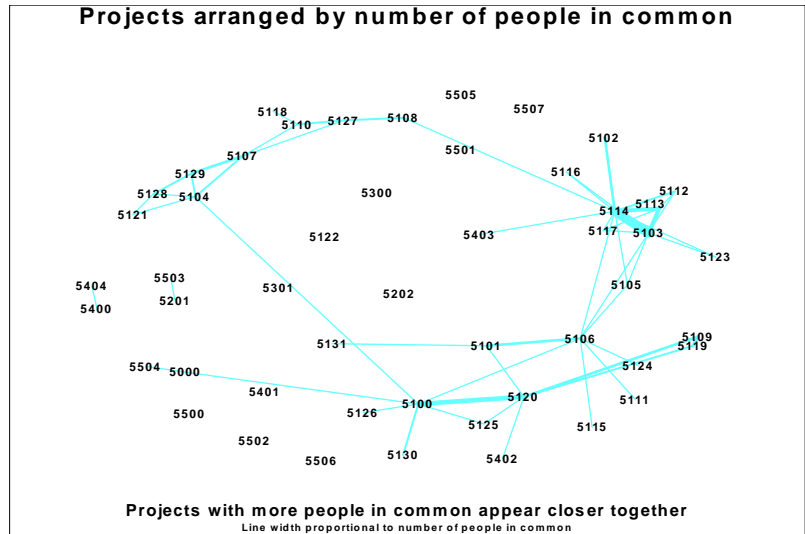
KANSAS NSF EPSCOR PHASE V (KNE)

Kansas NSF EPSCoR Phase V involved 247 people working on 50 projects. Data were gathered on proposals submitted and proposals awarded as well as scientific products (journal articles, conference presentations, etc.) submitted and scientific products accepted. For each pair of persons there were therefore multiple possible measures of association – working together on the same project, coauthoring a paper or a conference presentation, writing a grant proposal together. The project spans several years and involved several universities. Part of the evaluation of the project involved looking at changes in the pattern of collaboration across time.

In order to accommodate all of these factors we used the Annotate facility of SAS/GRAPH to generate static and animated GIF files as well as pdf files (and rtf files for this document). The

Projects arranged by number of people in common

Projects with more people in common appear closer together
Line width proportional to number of people in common

first step after importing the data from multiple worksheets in a spreadsheet was to clean the data, parse and standardize names. Manual editing was sometimes needed to resolve multiple forms of the name for an individual – as in nicknames vs. formal names. Determining university affiliation sometimes required searching universities' web sites. The next step was to compute a similarity matrix of projects and scale them into 2 dimensions. The number of people in common between a pair of projects was used as the basis of the similarity matrix.

It's a straightforward task to rescale the coordinates from the MDS to be percent of graphics area and generate an annotate file for the association of projects. Some projects that overlapped were manually adjusted as in the code below where the label for project 5113 is moved by a small percentage. Coordinates for each person were computed as the centroid of the coordinates of all of the projects with which the person was associated plus a small random amount to avoid overlaps.

```
create table ProjectAnno as
  select "LABEL" as FUNCTION,
         .75   as   size,
           put(project_number,4.) as text,
         case
           when project_number = 5113 then 5+90*( ProjectX-&minX)/(&maxX-&minX) -1
           else                            5+90*( ProjectX-&minX)/(&maxX-&minX)
         end     as x,
         case
         when project_number = 5113 then max(3,5+90*( ProjectY-&minY)/(&maxY-&minY) ) + 1
           else                          max(3,5+90*( ProjectY-&minY)/(&maxY-&minY) )
         end  as    Y,
         '5'   as   xsys,
         '5'   as   ysys,
         '8'   as   position,
         'cx000000' as  color,
         'projlabel' as object,
         monotonic() as order
  from KNEdb.ProjectsXY;
```

A small number of people involved in the overall KNE project were identified as "core people", to be highlighted in the graphics. A separate annotation file could draw then as larger filled circles.

KNE Core People

Projects with more people in common appear closer together
Large tan dot indicates initial core people.

6

Smaller filled circles were generated proportional in size to the association measure (e.g. number of proposals for that person) for each person for each year of the grant, shaded by institution (e.g. University of Kansas blue, Kansas State University purple) and lines drawn for links, with the width proportional to the number of associations between the pair. The annotation files were concatenated and then plotted with PROC GSLIDE, one plot per year both as individual files and successively into an animated GIF file.

These plots were generated as a separate series for proposals awarded, scientific product (publications accepted etc.), and the combination of the two measures. The scientific products series and the combined series are shown below.
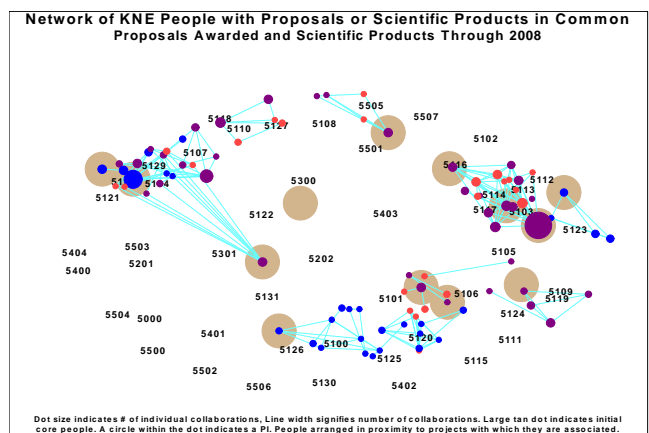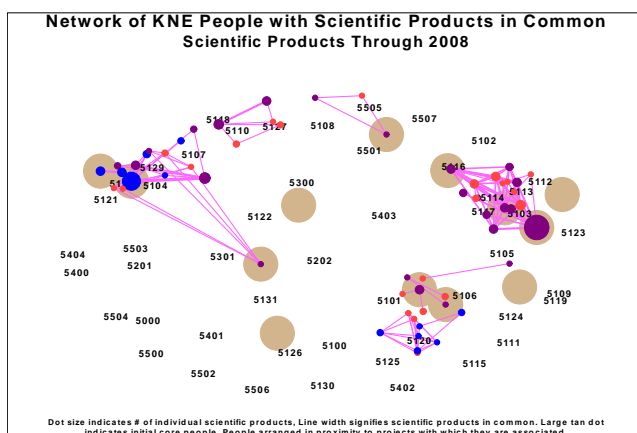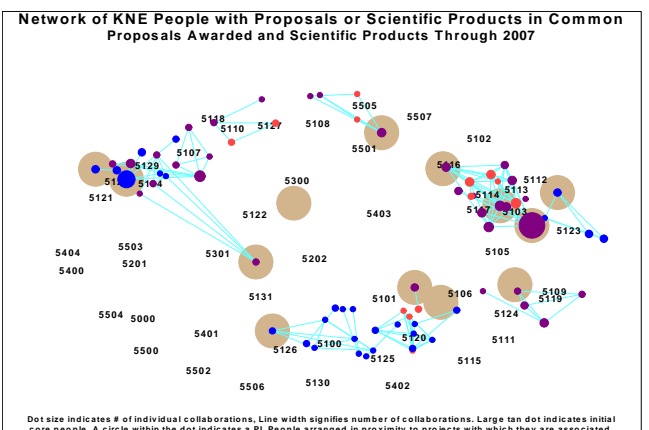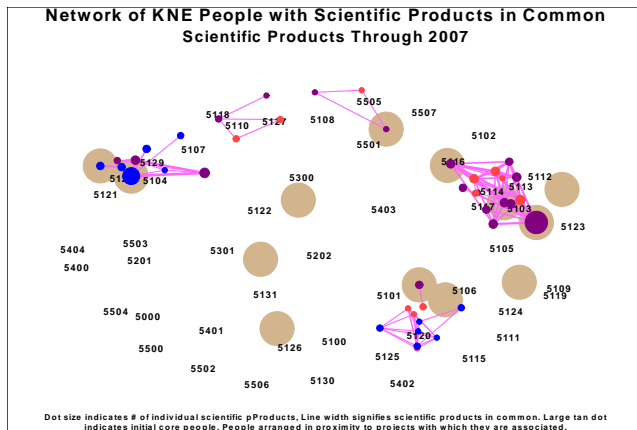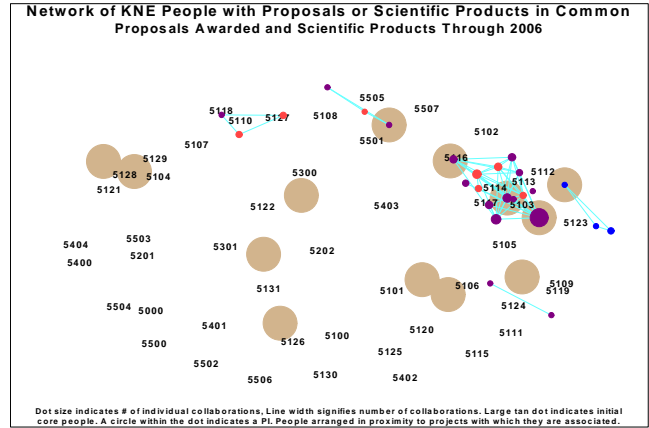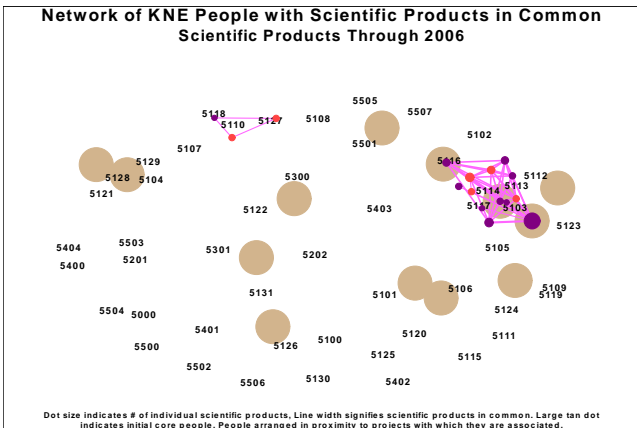
A plot of all the data for each series (e.g. submitted as well as accepted publications) and the combination of all of the data was also generated.

Network of KNE People with Proposals or Scientific Products in Common
All People involved in KNE, and all Submissions

Dot size indicates # of individual collaborations, Line width signifies number of collaborations. Large tan dot indicates initial core people. A circle within the dot indicates a PI. People arranged in proximity to projects with which they are associated.


Using ODS, the whole sequence of plots were written to an HTML file, and to pdf and rtf files. The latter two allowed printing at a much higher level of quality. The HTML file allowed a reference to the animated GIF files to be inserted using the text= option on the ODS command as in the code below.

```
ods HTML text='<H2>Animations: Proposals over
     time</H2><img alt="Proposal Animations"
  src="ProposalAnimation.gif" border="0" class="c">
```

A NOTE ABOUT ENTERPRISE GUIDE

These projects involved creating multiple files with somewhat complicated interrelationships. An example: the PROC MDS generated x.y coordinates for projects, which were then joined to the records for each individual which were then used to output a CNTLOUT dataset of informats for the standardized names. These informats were then used in the generation of annotate files for nodes and links. Tracking these dependencies was helpful when, for example, an overlooked nickname showed up in a plot and needed to be standardized manually.

Even though most of the nodes were code nodes, developing a process flow diagram in Enterprise Guide was helpful. The diagram at right shows some of the relationships among the code nodes and datasets for the KNE analysis.

8
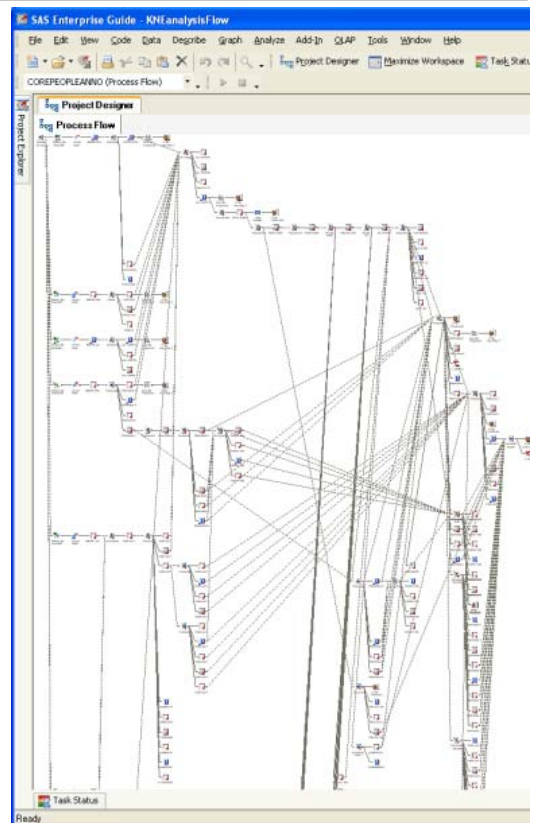
STATISTICS AND COMPUTING THE CLUSTERING COEFFICIENT - TRIPLES

Once there are tables of nodes and dyads it is a straightforward task to produce some of the standard social network statistics. As an example, distribution analysis shows that the number of others to which each person links on SAS-L is distributed exponentially (e.g. Kolmogorov-Smirnov D 0.4177 Pr > D <0.001).

One statistic that requires additional computation is the clustering coefficient. This is a measure that for all triples a, b, and c; if b links to a and b links to c, the clustering coefficient is the proportion of times that a links to c. There are two approaches to computing the clustering coefficient (see Newman). The code below computes what Newman called $C^{(1)}$, or what is referred "the fraction of transitive triples". (3 X number of triangles) / (number of connected triples). It also computes $C^{(2)}$, the mean of "(number of triangles connected to vertex i) / (number of triples centered on i)".
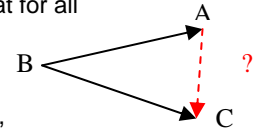
This code uses a hash to count the triangle closures (A to C in the figure at right). This approach was more than 10 times faster than an SQL based approach (4 minutes vs. 45 min) when computing this coefficient for SAS-L.

```
%macro ClusteringCoefs(
    DyadCounts=test,                    /*  dataset containing counts of dyads        */
    dyadFrom=StandardNameFrom,          /*  character variable - name of FROM node     */
    dyadTo=StandardNameTo,              /*  character variable - name of TO node       */
    dyadCount=PostCount,                /*  weight or count for each dyad              */
    fractionTriples=fractionTriples,    /*  dataset with C1 statistics                 */
    localTripleRatios=localTripleRatios, /* dataset with all local ratios              */
    meanLocals=meanLocals,              /*  dataset with the mean of the local ratios (C2) */
    ClusteringStats=ClusteringStats );  /*  dataset with C1 and C2                      */

proc sql;
 create view dyadCounts as
   select &dyadFrom  as From,
          &dyadTo    as To,
          &dyadCount as dyadCount length=3
   from &DyadCounts;
quit;

data &fractionTriples  (keep = nTriples nTriangles C1)
     &localTripleRatios(keep = localTriples localClosed localRatio );

 retain iteration 0  ixIDTo 0;
 retain nTriples 0  nTriangles 0;
 retain localClosed 0 localTriples 0;

 length localTriples 8 localClosed 8 localRatio 8 From $ 50 To $ 50  dyadCount 3;

 label C1 = 'C1 Clustering Coefficient - fraction of transitive triples';

 if iteration = 0 then do;
     dyadCount = 0;      localClosed = 0;      localTriples = 0;      localRatio = 0;
                                /* dyadhash will just be used to mark the presence of a dyad  */
                                /* load it from the view  with the variable names we want     */
  declare hash dyadHash(dataset: "dyadCounts", hashexp: 9);
  rc = dyadHash.defineKey('From', 'To');
  rc = dyadHash.defineData('dyadCount');
  rc = dyadHash.defineDone();
 end;
 iteration=iteration+1;

 set &DyadCounts end=last;
 by &dyadFrom &dyadTo;
                /*   each IDFrom will be considered as the center of a triple (ID2) */
                /*    the set of triples for that ID2 is determined by all          */
                /*    combinations of NameTos taken two at a time                   */
 array NameTos{25000} $ 50 _temporary_;

 if first.&dyadFrom then do;
   ixIDto = 0;
   localClosed = 0;
```

9

```
      localTriples = 0;
   end;
                                    /*  the dyads file has links to self - don't include them here  */
                                    /*   also ID1 in the triple will always be less than ID3         */
                                    /*    so we don't need the duplicates                            */
   if &dyadFrom ne &dyadTo then do;
      ixIDto = ixIDto + 1;
      NameTos{ixIDto} = &dyadTo;
   end;
                                       /*  here is where we find all the combinations          */

   if last.&dyadFrom and ixIDto > 1 then do;
      do ixID1 = 1 to ixIDto;
         From = NameTos{ixID1};
         do ixID3 = ixID1+1 to ixIDto;
            To = NameTos{ixID3};

            nTriples = nTriples + 1;                  /* this counts all triples */
            localTriples = localTriples + 1;          /*  this counts triples with a common middle   */
                              /* if the dyad from the start of the triple to the end of the triple */
                              /*   is in the hash (find returns 0) then the triangle is closed      */
            if not dyadHash.find() then do;
               nTriangles  = nTriangles + 1;
               localClosed = localClosed + 1;
            end;
         end;   /*  ixID3 = ixID1+1 to ixIDto */
      end;   /*  ixID1 = 1 to ixIDto */
   end;  /* last.StandardName and ixIDto > 1  */

   if last.&dyadFrom then do;
      if localTriples = 0 then localRatio = 0;
      else localRatio = localClosed / localTriples;
      output &localTripleRatios;
   end;

   if last then do;                             /* each triangle was found as 3 closed triples  */
    nTriangles = nTriangles / 3;                /* divide by 3 to get the unordered count        */
    C1 = 3 * nTriangles / nTriples  ;
    output &fractionTriples;
   end;
end;
run;
proc sql;
 create table &meanLocals as
  select mean(localRatio) as C2
                       label='C2 Clustering Coefficient - mean ratio of closed triples',
         count(*) as nNodes
  from &localTripleRatios;

create table &ClusteringStats as
 select &fractionTriples..* ,
        &meanLocals..*
 from &fractionTriples, &meanLocals;

%mend ClusteringCoefs;

%ClusteringCoefs( DyadCounts=SGF2009.DyadsbyThreadCountTotal,
                  dyadFrom=StandardNameFrom,
                  dyadTo=StandardNameTo,
                  dyadCount=nPosts,
                  fractionTriples=fractionTriples,
                  localTripleRatios=localTripleRatios,
                  meanLocals=meanLocals,
                  ClusteringStats=SGF2009.ClusteringStatistics );


   quit;
```

Number of Nodes	Number of Links	Number of Triples	Number of Triangles	$C^{(1)}$ Clustering Coefficient - fraction of transitive triples	$C^{(2)}$ Clustering Coefficient - mean ratio of closed triples
23,823	425,667	95,726,110	2,994,894	0.093858	0.453219

Some basic statistics for SAS-L are shown in the table above. The pattern of $C^{(1)}$ and $C^{(2)}$ is somewhat similar to that of biology co-authorship ($C^{(1)}$ =0.088 and $C^{(2)}$=0.60) reported by Newman.

There are other statistics that measure each node's relationship to the network. These can include simple statistics such as number of links to other nodes. The Hoyle SGF paper listed below contains an example of code to compute the Betweenness Centrality statistic for each node. This measure relates to the number of shortest paths that run through a given node.

CONCLUSION

The %DS2Const macro and the constellation applet provide an easy to use and flexible tool for interactive exploration of moderate sized networks. Larger networks or those requiring more complex or customized plots may be visualized with the annotate facility with not much additional effort. Non-interactive animations may be easily created with the annotate facility. The annotate facility along with ODS is a good choice for producing high quality plots for publication.

REFERENCES

- Hornibrook, Shane *&Degrees. of Separation: Social Network Analysis Using The SAS System* (NESUG2006) http://www.nesug.org/proceedings/nesug06/an/da21.pdf.
- Hornibrook, Shane *Analyzing Large Social Networks with MP Connect, SAS/IntrNet®, and %DS2CONST* (SESUG 2007) http://analytics.ncsu.edu/sesug/2007/DM03.pdf.
- Hoyle, Larry *Implementing Stack and Queue Data Structures with SAS® Hash Objects.* SAS Global Forum paper 084-2009, Washington D.C., 2009
- Johnston, M. Francis, Xiao Chen, Phillip Bonacich, Silvia Swigert *Modeling Indegree Centralization in NetSAS: A SAS Macro Enabling Exponential Random Graph Models* http://www.insna.org/PDF/Connections/v27/2006_I-2-4.pdf
- Newman, M. E. J. *The structure and function of complex networks.* SIAM Review 45, 167-256 (2003)

ACKNOWLEDGMENTS

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Sample code associated with this paper can be found at:

http://www.ipsr.ku.edu/ksdata/sashttp/SGF2009/

Contact the author at:
Larry Hoyle
Institute for Policy & Social Research, University of Kansas
1521 Lilac Lane, Suite 607 Blake Hall
Lawrence, KS, 66044-3177
785-864-9110
LarryHoyle@ku.edu
www.ipsr.ku.edu