

Text Analytics Using JMP®

Melvin Alexander, Social Security Administration

ABSTRACT

JMP® version 11 introduced the Free Text Command in the Analyze > Consumer Research > Categorical Platform under the “Multiple” tab. This utility restricted users to just produce word frequency counts and create indicator columns of the words that appeared in free-text comment columns. For more extensive text mining, users must use other JMP® Scripting Language (JSL) scripts, functions, and tools. This presentation will review different ways how JMP® can parse and convert qualitative text data into quantified measures.

Text mining techniques covered in this presentation include forming Term-Document-Matrices (TDMs); applying singular value decomposition (SVD) to identify the underlying dimensions that account for most of the information found in documents and text, and clustering word groups to convey similar topics or themes. Attendees should be able to use the methods for further reporting and modelling.

INTRODUCTION

This presentation will review the ways JMP® can be used to perform the techniques of text mining. The basis for this paper came from an E-poster Josh Klick and I presented at the Discovery Summit 2014 conference, see Alexander and Klick (2014). The E-poster showed how JMP and R integration transformed, unstructured, free-text comments from Respondents to Mid-Atlantic JMP Users Group (MAJUG) meeting feedback surveys. Many visitors to our poster wanted to know how the fundamental text mining tasks (available in SAS® Text Miner, SAS/IML®, or R) could be done using JMP alone. With the JMP tools, users will learn how to apply the methods presented to mine their own textual data.

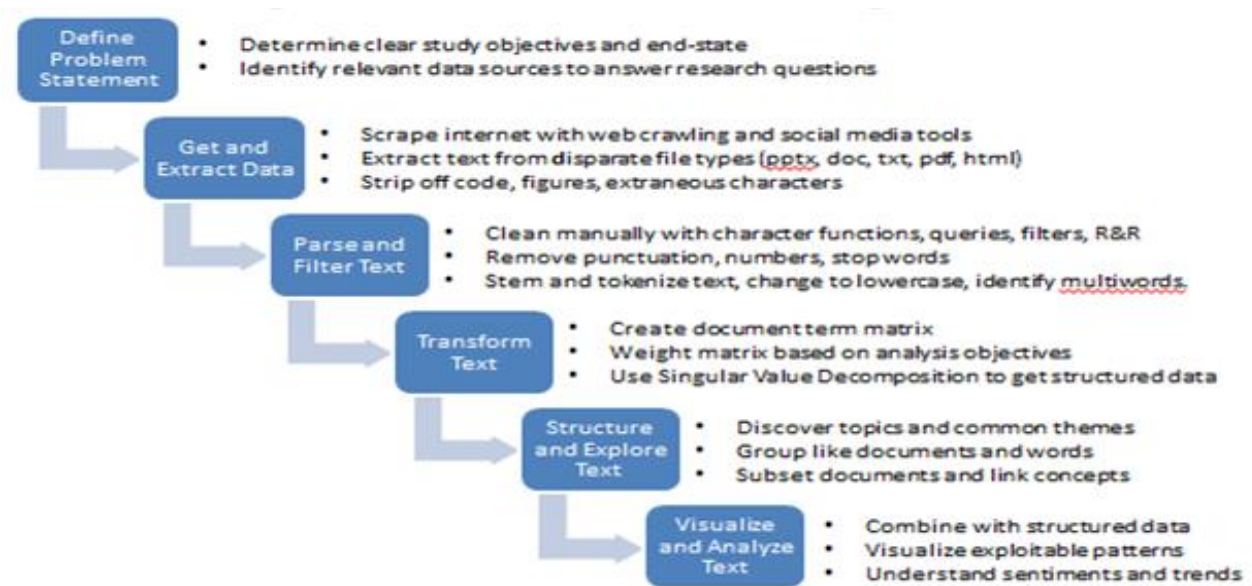
Text Analytics combines the disciplines of linguistics, statistics, and machine learning to model and analyze text data that guides business intelligence, Exploratory Data Analysis (EDA), research, and investigation. Text Analytics uses text mining techniques to transform unstructured, qualitative, source text into quantitative measures used for reporting and modeling. See McNeill (2014).

Text mining seeks to find predominant themes (topics) from documents (corpus) where singular value decomposition (SVD) is used to help extract and interpret the key topics from terms included in the text. Text mining methods increase statistical learning that takes advantage of the additional information found in text. Text parsing removes any terms that have little or no informative value (stop words); and filters, cleans, prepares, and keeps only those terms that are most informative for further analysis. See Karl and Rushing (2013), and Rushing and Wisnowski (2015).

Text mining tools in this presentation help reveal the “User’s Voice”, gain insights, and identify ways to improve the services given to user group members that were invisible in the structured data categories.

I will apply the text mining techniques on free-text comments from MAJUG meeting participants to help improve meeting planning that meets MAJUG member’s needs. Figure 1, from Rushing and Wisnowski (2015), depicts the process flow of text mining steps. The top left oval defines the study objectives (e.g., understand the “Voice of the User” – VOU – from feedback comments in order to deliver improved content). The second oval is where input text is collected from user-feedback surveys. The parsing and filtering oval breaks down text into more structured data (i.e., retaining meaningful terms). The transformation oval groups these terms and converts them into quantifiable form. The bottom ovals cluster documents and terms into groups that convey similar content which serves as input into models that provide reliable information for predicting outcomes and increase the user-group experience.

Figure 1: Text Mining Flow



By way of background, MAJUG meetings are held three or four times a year. Notices are posted on the MAJUG web site (<http://www.majug.com/>), see Figure 2. MAJUG also has a presence on the JMP User Community site (<https://community.jmp.com/community/regional-jmp-user-groups>).

Figure 2: MAJUG Web-site

MAJUG
MID-ATLANTIC JMP USERS GROUP

A COMMUNITY OF JMP USERS

WELCOME
The Mid-Atlantic JMP Users Group is comprised of users of JMP Software, the statistical discovery software from SAS.

Our membership includes individuals from Washington DC to Baltimore. We meet once every 3 months at a different location in order to better serve our geographically diverse membership.

Mid-Atlantic JMP Users Group (MAJUG) Spring Meeting
Location: Bureau of Labor Statistics
Thursday, April 2nd from 9-12
AGENDA*:

- 9:00 Welcome
- 9:15 JMP® 12 Preview (JMP Speaker via WebEx, Webcast, or in-person)
- 10:15 Break
- 10:30 Text Analytics Using JMP® (Mel Alexander)
- 11:15 Applications of JMP® and R Integration, Discussion, Awards, and Feedback
- 12:00 Adjourn

* This schedule is subject to change without notice.

MAJUG meeting will be located at the Bureau of Labor Statistics:
Please reply to Joshua Klick at jlick.joshua@bls.gov by 12pm Friday March 27th indicating whether you plan to attend in person or via WEBEX.
(A reminder email with instructions to access the meeting online will be sent to those who register by March 30th).

Note that in person attendees will need to be registered on the BLS visitor list prior to April 2nd. Directions for those arriving outside the BLS: BLS is located in the Postal Square Building at 2 Massachusetts Ave, NE., directly across from Union Station on the red line. The visitors' entrance is on 1st St. NE, between Massachusetts Ave and G Street. Attendees will need a photo ID and should allow 10 minutes to check-in with the security desk before the start of the meeting.

Benefits of Membership

- Learn about new JMP products, capabilities and services as soon as they become available.
- See demos of new software features.
- Succeed with hands-on workshops.
- Improve your efficiency and productivity by learning new coding and analysis techniques.
- Find out how your peers are solving problems.
- Talk with JMP developers one-on-one and influence the future direction of JMP software.
- Enhance your presentation and interpersonal skills in a relaxed and user-friendly environment.
- Sharpen your managerial and leadership skills by serving as a user group officer.

► e-Mail our Membership Chair, Susan Dorsey at sdorsey@son.umaryland.edu, for more information.

RESOURCES

- JMP User Community
- JMP File Exchange
- Other JMP Users Groups

STAY CONNECTED WITH JMP

facebook

LinkedIn

twitter

STEERING COMMITTEE

Chairs:

- Melvin Alexander
melvin.alexander@verizon.net
- Joshua Klick
jlick.joshua@bls.gov

Members:

- Manny Uy
Manny.Uy@jhuapl.edu
- Susan Dorsey
sdorsey@son.umaryland.edu
- Jennifer Barb
barbj@mail.nih.gov
- Kelly McVearry
Kelly.McVearry@ngc.com
- Merve Gurlu
mgurulu1@jhu.edu

Copyright ©, 2013 Mid-Atlantic JMP Users Group
SAS and JMP are registered trademarks of SAS Institute, Inc. All rights reserved.

GET AND EXTRACT TEXT

At the conclusion of each MAJUG meeting, attendees are asked to complete an evaluation form. The five questions in section 2 of Figure 3 are usually open-ended, unstructured text that respondents write in.

Figure 3: MAJUG Meeting Evaluation Form

MAJUG Meeting Feedback

1. Please rate how much you agree with the following statements by selecting a response using the following scale: 1 = strongly disagree and 5 = strongly agree.

	Strongly Disagree	Disagree	Neither agree / disagree	Agree	Strongly Agree
The meeting topics and program were informative.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Meeting time was convenient.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Meeting Place and venue were convenient for me.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Decision-making was shared at this meeting.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
All meeting participants were actively involved.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Meeting time was used effectively.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I was satisfied with this meeting overall.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I enjoyed this meeting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Please provide additional feedback about the MAJUG meeting below:

What aspects of this meeting were particularly good?

What aspects of this meeting need improvement?

Do you have any suggestions or additional comments about this meeting?

What JMP meeting topics would be useful to cover for future meetings?

Would you be interested in presenting at a future MAJUG meeting? Topics may range from lessons learned to a business

Table 1 shows a sample data table of the collected feedback. Even though the sample is small, MAJUG leadership was able to use the responses as a baseline to understand their members' needs.

Table 1: Sample Data Table of Respondent's Feedback from MAJUG Meeting Evaluations

Respondent	Date	Meeting Informative	Convenient Time	Place and venue
1	09/12/2013	Strongly Agree	Strongly Agree	Strongly Agree
2	09/12/2013	Neutral	Agree	Agree
3	09/12/2013	Strongly Agree	Strongly Agree	Strongly Agree
4	09/12/2013	Strongly Agree	Strongly Agree	Strongly Agree
5	09/12/2013	Strongly Agree	Strongly Agree	Strongly Agree
6	01/28/2014	Strongly Disagree	Disagree	Agree
7	01/28/2014	Strongly Agree	Strongly Agree	Strongly Disagree
8	01/28/2014	Agree	Neutral	Strongly Agree
9	01/28/2014	Strongly Agree	Strongly Agree	Strongly Agree
10	01/28/2014	Agree	Agree	Strongly Agree
11	01/28/2014	Agree	Strongly Agree	Strongly Agree

TEXT PARSING AND FILTERING

Figure 4 lists the Comments of the “Suggestions” column made by participants that were written to the JMP Log (Output 1) with the JSL << **Get Values** message.

Figure 4: MAJUG Meeting Evaluation Comments about Suggested Improvements

	Suggestions
1	improvements on the MAJUG site. Adding previous presentations to the website, best sources to learn JMP (online or books), and maybe tips and tricks of using JMP.
2	rotating location and WebEx access is important
3	next meeting email list of who is planning to come. list of topics of interest and discuss
4	time savings. laundry list of topics. Data analytics. Best practices, how to best summarize. review issues, problems. email beforehand - I'm coming JMP presentations. Who are
5	users in MAJUG, Professions, share email contacts. Web value (increase usefulness) What papers/presentations have occurred at MAJUG,
6	Query members planning to attend what they want to get out of the meeting so
7	their concerns, questions, issues can be addressed and discussed,
8	'Please start at 10',
9	'Have coffee break with coffee, more communication between meetings, suggesting topics',
10	'MAJUG should have a fee (perhaps \$5) to buy refreshments so participants can get coffee without leaving meeting',
11	''

```
Data Table ("SESUG 2015 Text Analytics demo");
Suggestion = Column("Suggestions") << Get Values ;
```

Output 1: Suggestions Comments

```
{"Improvements on the MAJUG site. Adding previous presentations to the website, best sources to learn
JMP (online or books), and maybe tips and tricks of using JMP.", "Rotating location and WebEx access is
important", "next meeting email list of who is planning to come. list of topics of interest and discuss",
"Time savings. laundry list of topics. Data analytics. Best practices, how to best summarize. review
issues, problems. email beforehand - I'm coming JMP presentations. Who are users in MAJUG,
Professions, share email contacts. Web value (increase usefulness) What papers/presentations have
occurred at MAJUG", "Query members planning to attend what they want to get out of the meeting so
their concerns, questions, issues can be addressed and discussed", "", "", "Please start at 10",
"Have coffee break with coffee, more communication between meetings, suggesting topics",
"MAJUG should have a fee (perhaps $5) to buy refreshments so participants can get coffee without
leaving meeting", ""}
```

The Word Freqs data table in Figure 5 shows the Term Frequency Vector (TFV). It was created from the Analyze > Consumer Research > Categorical Platform, and used "Free Text" under the "Multiple" tab. Alternatively you can use JMP **Free Text** message command:

```
Current Data Table () << Categorical( Free Text( :Suggestions, << Save Word Table ),
Crosstab Transposed( 1), Legend( 0) );
```

Here I parsed and filtered the TFV for stop words (i.e., unnecessary words and punctuations that add little value in retrieving meaningful information between respondents and increases the dimensions and “noise” variation in the unstructured text). I used the **Select Where** clause to highlight stop words in the TFV. These stop words can be excluded or deleted, leaving a final TFV for further exploration and analysis.

The **Select Where** clause in Figure 5 scanned for stop words specified in a list. For example,

```
/*
*****
Here are the stopwords as a list taken from
http://jmlr.csail.mit.edu/papers/volume5/lewis04a/all-smart-stop-list/english.stop
*****
*/
```

```

stopwords = {"able", "about", "above", "abroad", ... , "yours", "yourself",
"yourselves", "you've", "zero"};
dtswc = Current Data Table () ; /* Current TFF frequency table */
/* Select all rows which have any Word in the stop word list, stopwords */
dtswc << Select Where( Contains( stopwords, :Word ) );

```

Figure 5 Select Where Clause using a stopwords list from the Term Frequency Vector (TFV)

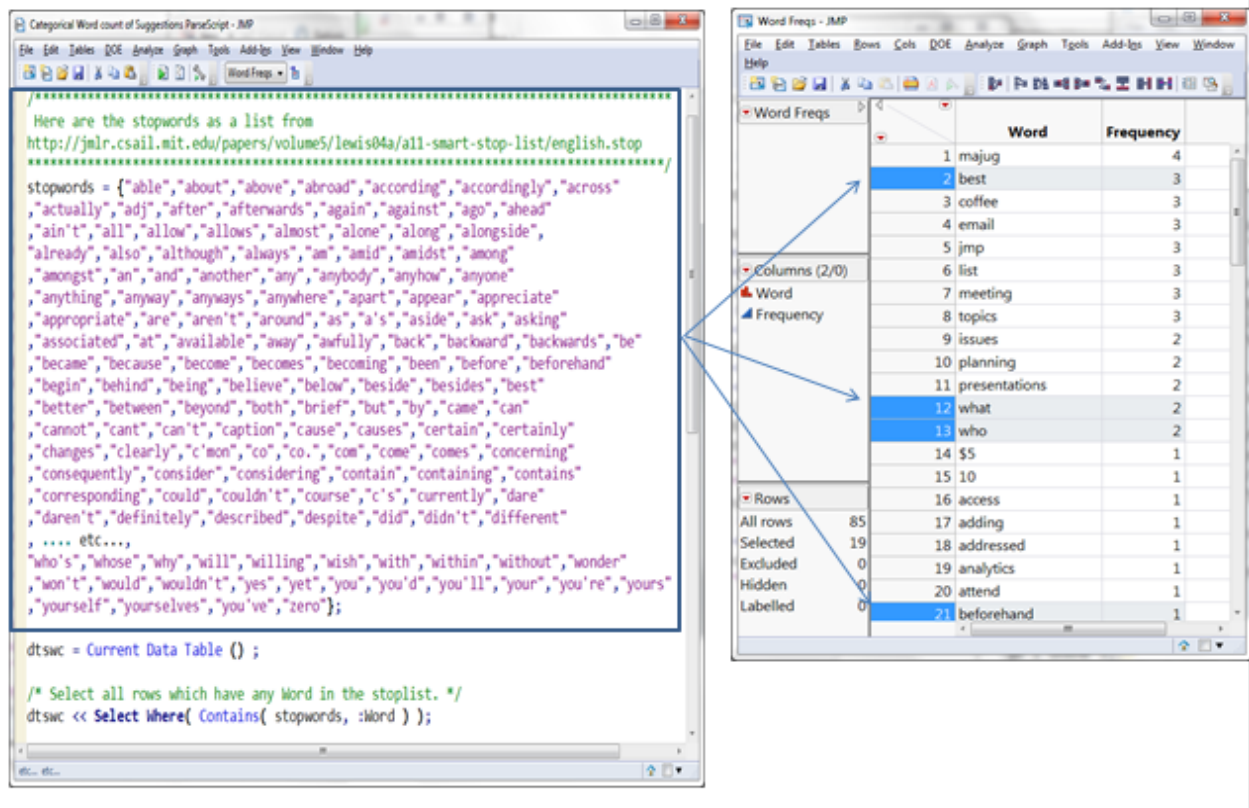


Table 2 showed the resulting TFV after 19 stop words were removed.

Table2: Results after Stop words were removed from the TFV

The screenshot shows the 'Word Freqs - JMP' table window. The table has two columns: 'Word' and 'Frequency'. It lists 21 words with their frequencies. The first 19 rows are selected, indicated by blue highlighting. The 'Rows' panel on the left of the table window shows 'All rows' as 66, 'Selected' as 0, 'Excluded' as 0, 'Hidden' as 0, and 'Labelled' as 0.

Word	Frequency
1 majug	4
2 coffee	3
3 email	3
4 jmp	3
5 list	3
6 meeting	3
7 topics	3
8 issues	2
9 planning	2
10 presentations	2
11 \$5	1
12 10	1
13 access	1
14 adding	1
15 addressed	1
16 analytics	1
17 attend	1
18 books	1
19 break	1
20 buy	1
21	1

TEXT TRANSFORMATION

Another data preparation step is to use the **RECODE** command. Table 3 below used the **RECODE** command for the Word column to change the values of “\$5” and “10” into “Charge-\$5-fee” and “Start-at-10” terms.

I used the remaining TFV terms without stemming. Stemming maps multiple words to a fixed-reduced root term, token, or stem. For example, words like *tap*, *tapping*, and *tapped* are similar. Stemming reduces them to the same term *tap*. I did not use stemming because of our small sample size. Stemming could be applied in future applications as more respondents (documents) increase the database size. In version 11, stemming is possible with the **RECODE** command, which was overhauled in JMP version 12, see Preiss (2015). Among the new features in JMP 12 is the “Group Similar Values” option. This feature lets users decide how to group terms in similar, unique categories based on character edit distances. Character edit distances are the minimum number of single-character edits needed to change a text-string into another. This is accomplished by grouping values that differ by certain percentages compared to the total number of characters of each value or by the number of characters between pairs of terms. Shift-click, control-click, and right-click options may be used to find related terms, put them into group categories, and save the recoded values to the original table or as new columns. Another new feature is the “Text to Column” option in the Utilities> **RECODE** command. This feature creates binary frequency weights that will be discussed further in the **TERM FREQUENCY WEIGHTING** section.

One could also create stem terms looping through the rows and change the values with the **For Each Row()** function. With the *tap* example, all words beginning with *tap* would be replaced with *tap* as its stem, regardless of any trailing letters that followed it. The JSL statement would be:

```
For Each Row(If(Starts With (:Word, "tap"), :Word = "tap")) ;
```

The **Match()** function provides another way to map multiple terms *tapped*, *tapping*, and *tap* to the stem *tap* using the following JSL statement:

```
Match( :Word, "tapped", "tapping", "tap", "tap", :Word) ;
```

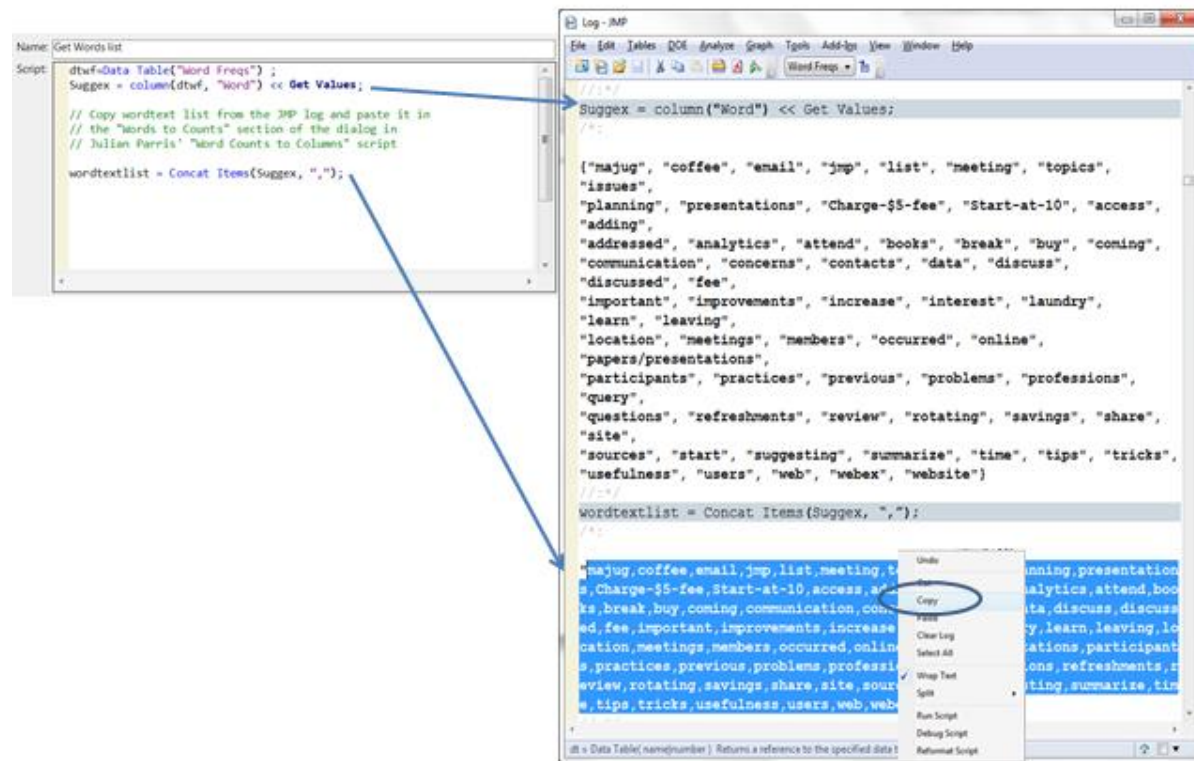
Note that using the **RECODE** command and the **For Each Row()** or **Match()** functions for stemming is a manual process which can become tedious and unwieldy with a large number of words in the TFV. Several stemming algorithms for automating the process (encoded in SAS, SQL, R, and other languages) exist. These algorithms are beyond the scope of this presentation. Many of them are available at Porter (2006).

Table 3: Recode Word Column to change values of “\$5” to “Charge-\$5-fee” and “10” to “Start-at-10”

Word	Frequency
1 majug	4
2 coffee	3
3 email	3
4 mp	3
5 list	3
6 meeting	3
7 topics	3
8 issues	2
9 planning	2
10 presentations	2
11 Charge-\$5-fee	1
12 Start-at-10	1
13 access	1
14 adding	1
15 addressed	1
16 analytics	1
17 attend	1
18 books	1
19 break	1
20 buy	1

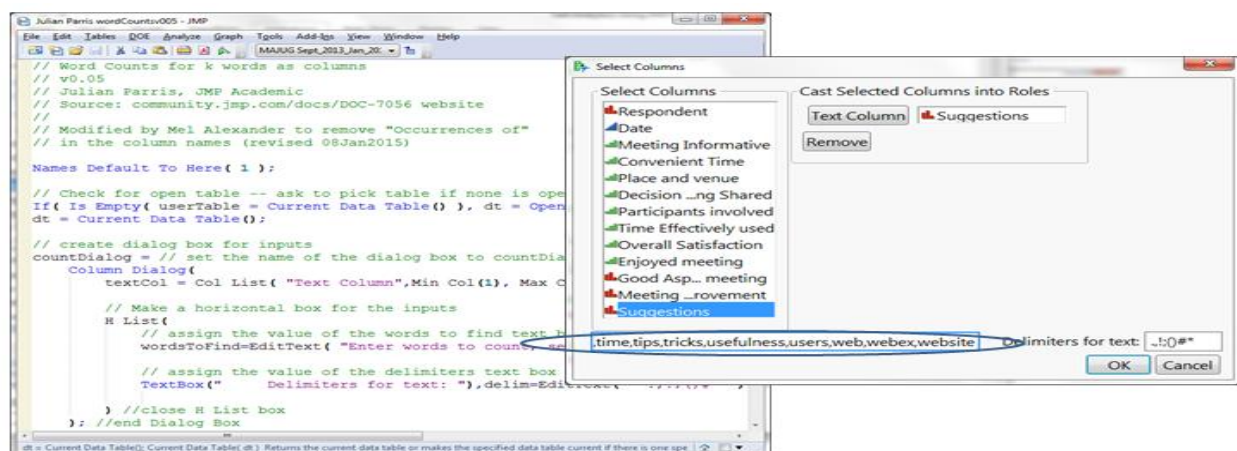
Next, I used the JSL **Concat Items** function to create a single string of comma-separated words from the TFW. The results were written to the JSL log, shown in Figure 6. I copied the text string words from the JSL log and returned to the original data table as the active table. This string would be copied and pasted into Julian Parris' "Word Counts for k words as columns" JSL script. See Parris (2014).

Figure 6: Creation of the single string of comma-separated words from the TFW.



After running the script dialog, I put the Suggestions column in the "Text Column" role. Next, I pasted the copied list of comma-separated words into the "words to count" section textbox of the script's dialog box in Figure 7. After clicking OK, the script added the words as columns to the active data table.

Figure 7: Julian Parris' "Word Counts to Columns" JSL script with pasted comma-separated terms.



I subset the word columns into a Document-Term-Matrix (DTM) data table called “Terms Matrix” in Table 3

Table 3: Terms Matrix Data Table

	majug	coffee	email	jmp	list	meeting	topics	issues	planning	presentations	Charge-\$5-fee	Start-at-10	access	ad
Word Count of Good Aspects of t	1	1	0	0	2	0	0	0	0	1	0	0	0	
Word Count of Aspects needing ir	2	0	0	0	0	0	0	0	0	0	0	0	1	
	3	0	0	1	0	2	1	1	0	1	0	0	0	
	4	2	0	2	1	1	0	1	1	0	1	0	0	
	5	0	0	0	0	0	1	0	1	1	0	0	0	
	6	0	0	0	0	0	0	0	0	0	0	0	0	
	7	0	0	0	0	0	0	0	0	0	0	0	0	
	8	0	0	0	0	0	0	0	0	0	0	0	0	
	9	0	2	0	0	0	0	1	0	0	0	0	0	
	10	1	1	0	0	0	1	0	0	0	0	0	0	
	11	0	0	0	0	0	0	0	0	0	0	0	0	

The “Terms Matrix” data table is converted into a DTM matrix with the **Get As Matrix** JSL message. Next, transpose the DTM into a Term-Document-Matrix (TDM) where rows represent terms, columns are the respondents (documents), and matrix entries represent the relative frequency of terms used by respondents. I prefer working with the transposed “long” TDM vs. the “wide” DTM.

Here, transposing was done to show ways of preparing the data as input for the next step, running the SVD function.

```
A = Data Table ( "Terms Matrix" );
DTM=A << Get As Matrix ;
/* B Transposes DTM to form Term-Document-Matrix (TDM) */
B = DTM` ;
```

Output 2: Selected Output of the Term Document Matrix (TDM) B

```
[10020000010,00000000210,
00120000000,20010000000,
00210000000,00101000010,
00110000100,
etc.
00010000000,10000000000,
10000000000,00010000000,
00010000000,00010000000,
01000000000,10000000000]
```


TERM FREQUENCY WEIGHTING

Term Frequency weighting is an intermediate step often used to assign weights to term counts in sparse TDMs so that the term's discriminatory power is enhanced. Terms that are more (or less) important than others are given higher (or lower) weight and sparse terms are assigned values of zeros. The different weighting schemes are the following :

Binary Frequency (Indicator) weights are useful when there is a lot of variance in the lengths of the documents (i.e., respondent's comments). The binary weighting scheme is what JMP uses with the "Save Indicators for Most Frequent Words" option from the **Free Text Word Counts** output report. In version 12, binary indicator columns can be created by clicking the red triangle for the Word column and selecting Utilities > Make Indicator Columns.

Raw Frequency (B) is most often used because some researchers have found that it improves interpretation results when it is important to distinguish between terms appearing rarely in documents vs. terms that appear several times. Although term-frequency weighting can make the interpretation of results more difficult, it can provide better predictive performance

Log transformations are used to shrink the weight of terms that appear in many documents while inflating the weight of terms that appear in few documents. Typically two basic log transformation weighting schemes are used: log base 2 or log base 10. Log base 2 (C) down-weights higher frequency terms, lower-frequency terms get higher weight, and terms with zeros remain zero. Log base 10 (D) dampens the presence of high counts in longer documents without sacrificing as much information as the binary weighting scheme.

There is no universal best weighting scheme, so the standard practice is to compare the different schemes before applying the SVD function.

```
//Raw Frequencies
B = DTM` ;
show(B) ;
```

B= [1 0 0 2 0 0 0 0 1 0,
0 0 0 0 0 0 0 0 2 1 0,
0 0 1 2 0 0 0 0 0 0 0,
2 0 0 1 0 0 0 0 0 0 0,
... etc. ...
1 0 0 0 0 0 0 0 0 0 0];

```
// log base2
C= J(Nrow(B),Ncol(B),0);
For( i = 1, i <= NRow(B), i++,
For( j = 1, j <= NCol(B), j++,
C[i,j] = log(B[i,j]+1,2);
) );
show(C) ;
```

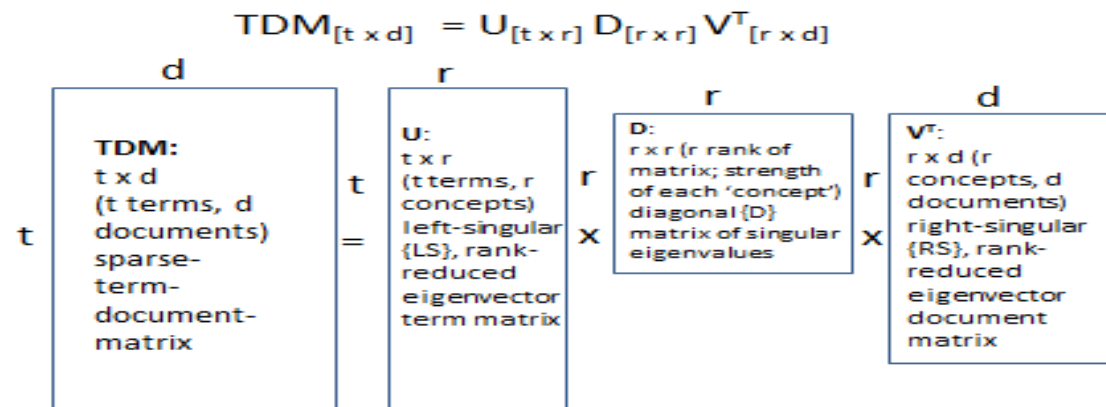
C =
[1 0 0 1.58496250072116 0 0 0 0 0 1 0,
0 0 0 0 0 0 0 1.58496250072116 1 0,
0 0 1 1.58496250072116 0 0 0 0 0 0 0,
1.58496250072116 0 0 1 0 0 0 0 0 0 0,
... etc. ...
1 0 0 0 0 0 0 0 0 0 0];

```
// log base10
D = J(Nrow(B),Ncol(B),0);
For( i = 1, i <= n, i++,
For( j = 1, j <= p, j++,
if(B[i, j]>0,
D[i,j] = 1 +log10(B[i,j]),
D[i,j] = 0 );
) );
show(D) ;
```

D =
[1 0 0 1.30102999566398 0 0 0 0 0 1 0,
0 0 0 0 0 0 0 1.30102999566398 1 0,
0 0 1 1.30102999566398 0 0 0 0 0 0 0,
1.30102999566398 0 0 1 0 0 0 0 0 0 0,
... etc. ...
1 0 0 0 0 0 0 0 0 0 0];

SINGULAR VALUE DECOMPOSITION (SVD) TEXT/DATA STRUCTURING

The decomposition formula for the SVD function is as follows:



$TDM = UDV^T$ where U denotes the TDM left-singular, rank-reduced eigenvector term matrix (**LS**); D (**D**) is the diagonal elements matrix consisting of the square root of descending, nonnegative **singular eigenvalues**; V^T , the Transpose of matrix V , is the right-singular, rank-reduced eigenvector documents matrix (**RS**) of the TDM. The matrix algebraic product of TDM and V^T forms the singular-value-decomposed term vector scores. Note that the **SVD** function applied on DTM reverses the order of the left-singular and right-singular matrices (i.e., $DTM = VDU^T$). The eigenvalues and eigenvectors (eigens) from the **SVD** function help explain the amount of descriptive information (variation) contained in words and documents.

More information about estimating the rank of rectangular matrices using the **SVD** function can be found in Albright(2004), Bogard (2012), and Wicklin (2015).

The term (**LS**) and documents (**RS**) eigenvectors assign weights to each topic or theme they represent. They serve as thresholds which determine the strength of association terms or documents have in “belonging” to specific topics. Terms with similar topic weights (eigenvalues) describe each topic and summarize the main ideas of the document collection (corpus of respondents).

```
/* SVD function on the TDM */
{LS,D,RS}= SVD(B); /* singular value decomposition of B = LS*D*RS` */
```

Output 3: Selected Output of Eigenvector from the SVD function on Matrix B

```
{[0.378096662351085 0.131824388966428 0.0211184899994071 0.10589219870066
0.180893870586793 - 0.249947997346168 0 0 - 0.0598591783533297 0.132522620357014 0,
0.0331182960691775 - 0.0292243730113285 0.337183161237851 0.583127882442963
0.125047226003007 0.154020292284754 0 0 0.16821759320533 - 0.185471956068679 0,
0.350648279538604 - 0.163307416493917 - 0.00534077688514531 - 0.0449366248671385 -
0.190147431306457 - 0.0225114783847062 0 0 0.142431314885967 - 0.68019673189248 0,
0.2492189633787 0.419512224556798 0.0192515782022323 - 0.0343837845368551 -
0.0279803345343858 0.0579557417687915 0 0
etc.
0 0 0 0 0 0 0 0 - 1 0, 0 0 0 0 0 0 0 1 0 0, 0 0 0 0 0 0 0 1 0 0 0, 0.0511449194105313 -
0.0617494272704414 0.375812713862112 0.725075996234368b0.0094742784022233
0.571411451830774 0 0 0 0 0, 0.0931094176656951 0.00702825750170741 0.404132013103739
0.379678618426814 0.32464030058507 -0.76053361389797 0 0 0 0 0, 0 0 0 0 0 0 0 0 1]]}
```

The **SVD** function reduces the dimensional size (variation) of the TDM, from a matrix having many columns into one with fewer columns (or reduced rank). SVD columns are linear combinations of the rows in the original TDM. The rank-reduced SVD preserves much of the structure (descriptive information) of the original TDM, with less “noise” (error) variation. The smaller dimensional size of the TDM matrix saves the amount of computational resources (time, memory storage, etc.) needed for data processing. Using the SVDs simplifies statistical modelling tasks because fewer variables and “noise” factors are involved.

STRUCTURE AND EXPLORE TEXT

Next, I computed the Principal Components (PCs) by multiplying U and D, see Hastie, Tibshirani, and Friedman (2009). Principal Components are special cases of SVDs. PCs project rows of the TDM into new sets of attributes (dimensions) so that they are orthogonal (i.e., have zero covariance) and are independent (uncorrelated) with each other. Most of the variance in the data are captured by the first few (usually two) attributes. The **As Table ()** function converted the Principal Components matrix into a data table using the JSL script below:

```
u = LS; /* independent eigenvectors of B*B` = LS*/
s = D; /* independent eigenvectors of B`*B = D */
v = RS; /* Singular values (sqrt(eigenvalues)) of B*B` or B`*B */

ID = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 ] ;

/* Compute Principal Components (PCs) */
PCs = u * diag(D);
/* Turn PCs into a Principal Components data table */
As Table (PCs) << Set Name ("PrinComps") ;
col = column(1) << Set Name("PC1"); col = column(2) << Set Name("PC2") ;
col = column(3) << Set Name ("PC3"); col = Column(4) << Set Name ("PC4") ;
col = Column(5) << Set Name ("PC5"); col = Column(6) << Set Name ("PC6") ;
col = Column(7) << Set Name ("PC7"); col = Column(8) << Set Name ("PC8") ;
col = Column(9) << Set Name ("PC9"); col = Column(10) << Set Name ("PC10") ;
col = Column(11) << Set Name ("PC11") ;

// Form Principal Component data table
Data Table( "PrinComps" ) << Join( With( Data Table( "Word Freqs" ) ),
SelectWith( :Word ), Select( :PC1, :PC2, :PC3, :PC4, :PC5, :PC6, :PC7, :PC8, :PC9,
:PC10, :PC11 ), SelectWith( :Overall Satisfaction ), By Row Number, Output Table(
"Principal Components" ) );

dtpc = Data Table ("Principal Components");
// Label the Word Column so that Word show up on Graph Builder plot
dtpc << Set Label Columns( :Word ) << Select All Rows ;
```

Table 4: Principal Components Data Table

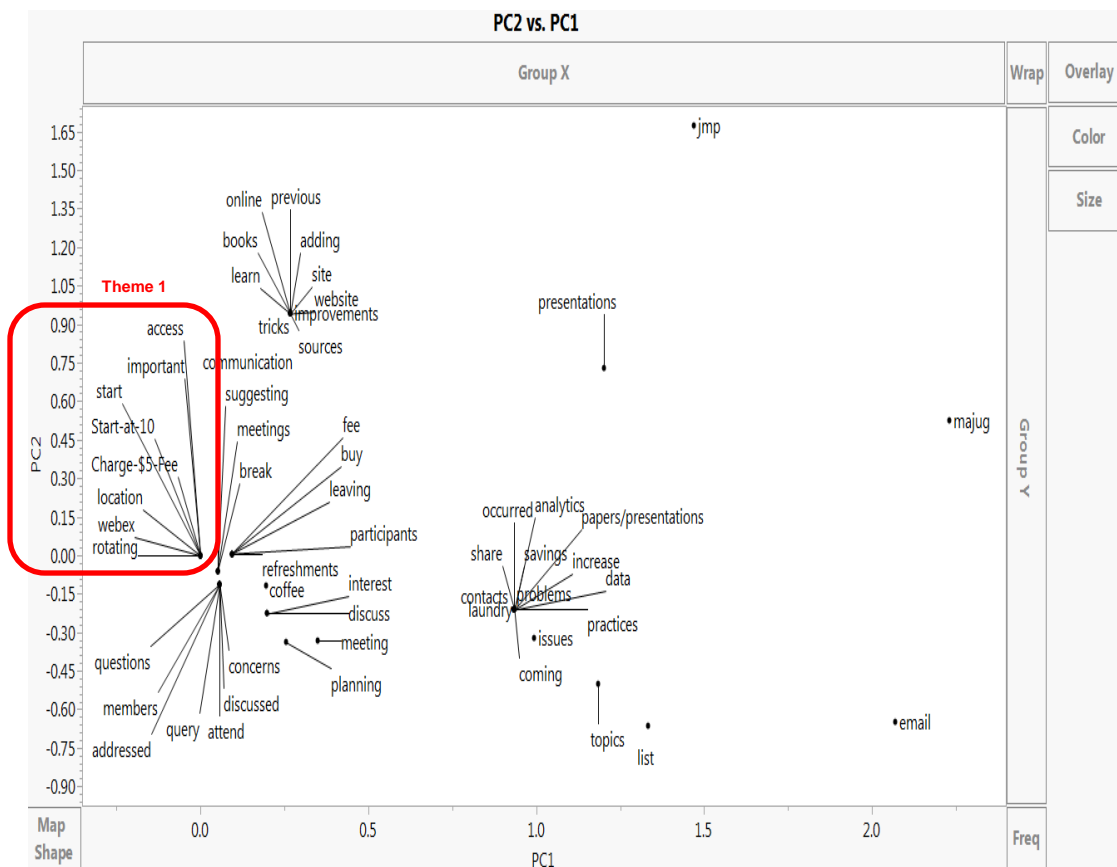
Word	PC1	PC2	PC3	PC4	P
1 majug	2.2307852599	0.5253719312	0.0723875174	0.332285237	0.4970
2 coffee	0.1953992565	-0.116470597	1.1557574408	1.8298306109	0.3435
3 email	2.0688387158	-0.650844153	-0.018306497	-0.141009227	-0.522
4 jnp	1.4704017394	1.6719208737	0.0659883331	-0.10789486	-0.076
5 list	1.3327274866	-0.66645478	0.6928643299	-0.295126551	-1.466
6 meeting	0.3494506726	-0.332410324	1.499350049	-0.321897746	-0.012
7 topics	1.1850003202	-0.500849072	0.6006653247	0.5796974037	-0.653
8 issues	0.9924524842	-0.323827955	0.3840472087	-0.54745904	0.6066
9 planning	0.256341255	-0.339438582	1.0952180359	-0.701576365	-0.337
10 presentations	1.2026925272	0.7300881825	-0.088585388	-0.051762747	0.031
11 Charge-\$5-fee	0	0	0	0	0
12 Start-at-10	0	0	0	0	0
13 access	0	0	0	0	0
14 adding	0.2677092122	0.9418326913	0.1545737206	-0.056132113	-0.108
15 addressed	0.0574691692	-0.112083446	0.6272063169	-0.551828406	0.4660
16 analytics	0.934983315	-0.211744509	-0.243159108	0.0043693659	0.1405
17 attend	0.0574691692	-0.112083446	0.6272063169	-0.551828406	0.4660
18 books	0.2677092122	0.9418326913	0.1545737206	-0.056132113	-0.108
19 break	0.0511449194	-0.061749427	0.3758127139	0.7250759962	0.0094
20 buy	0.0931094177	0.0070282575	0.4041320131	0.3796786184	0.3246
21 coming	0.934983315	-0.211744509	-0.243159108	0.0043693659	0.1405
22 communication	0.0511449194	-0.061749427	0.3758127139	0.7250759962	0.0094
23 concrete	0.0574691692	-0.112083446	0.6272063169	-0.551828406	0.4660

VISUALIZE AND ANALYZE TEXT

Figure 8 is a Graph Builder Bi-plot of the PCs scores that gave a data visualization of terms that appear close together by the same respondents (documents). Terms close together helped uncover similar themes or common topics concealed in the clustered text (a.k.a. latent semantic analysis). For instance words connected to the same vertex on the PC2 by PC1 Bi-plot showed common themes or words that were synonymous with each other.

The terms out of the vertex of Theme 1 (such as “Start-at-10”, “Charge-\$5-fee”, “rotating location”, webex access”) may suggest a factor and principal component described as “desired expectations that meeting attendees would like”.

Figure 8: Graph Builder Bi-plot of Principal Components.



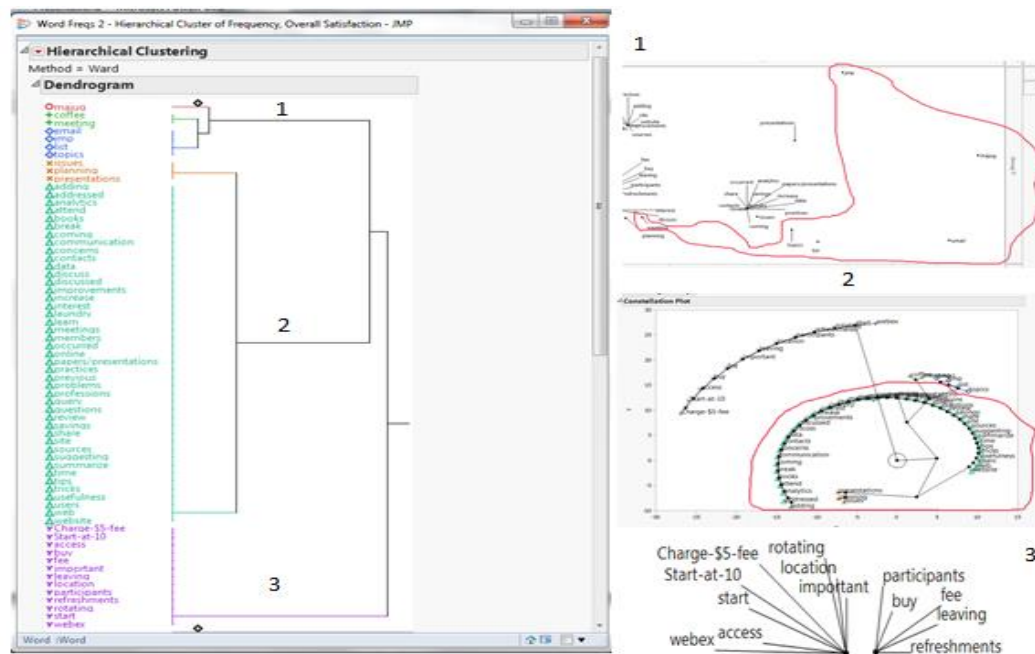
I further explored the association between terms and respondents via cluster analysis, principal components analysis, and regression modelling. Clustering is the unstructured technique that helps determine which documents are most similar; which groups of terms are similar to particular terms; and which clusters strongly relate to other variables than other clusters. Recursive partitioning (Partition) produces decision trees and Hierarchical Clustering returns dendrograms of overall satisfaction split by SVDs, PCs, or terms.

The Cluster analysis in Figure 9 indicated three distinct clusters from term frequencies by respondent's overall satisfaction, identified with 1, 2, and 3. Terms formed by these clusters and Constellation plot were matched with the Terms labeled on the principal components and SVD By-plots.

```
Data Table("Word Freqs") << Hierarchical Cluster( Y(:Frequency,:Overall Satisfaction
),Label( :Word ),Method( "Ward" ),Standardize Data( 1 ),Color Clusters( 1 ),
Mark Clusters( 1 ), Dendrogram Scale( "Distance Scale" ),Number of Clusters(5),
```

```
Constellation Plot( 1 ), SendToReport( Dispatch( {}, "Dendrogram", OutlineBox,
{SetHorizontal( 1 )} ) ) );
```

Figure 9: Hierarchical Clustering Dendrogram and Constellation Plot.



CONCLUSIONS

With the text mining tools I presented, we can gather the essence of text meanings from MAJUG-meeting respondents. The principal components and SVDs provided inputs used to estimate probability models that will lead to improved satisfaction and value to attendees of future meetings. That's a worthy goal users group leaders all want to achieve.

Cautions for users:

1. Using underscores, dashes, or other symbols in between words are necessary to form word-phrases or word combinations. Without them, Julian Parris' JSL script will treat them as spaces to be added as separate, unnecessary columns. These additional columns obscure the DTM or TDM.
2. TFV limits the numbers of terms used to create the DTM or TDM. This impacts the sparsity, dimensions, and size of the SVD.
3. The Principal Components computed with the **SVD()** function is different from the singular values from Contingency Analysis. Contingency Analysis computes singular values and eigens using different formulas than the eigens of the SVD() function. Note that JMP's Principal Components, Factor Analysis, PLS, and Correspondence Analysis from the Analyze >> Multivariate Methods platform uses normalization of term frequencies as weights. Each term is normalized so that the sum of each document vector is 1. Normalization is done by dividing the term counts in each document (each row of the DTM) by the total number of words in each document (the row sums of the DTM). This is useful when the documents have different lengths. This normalization is referred to as the Frobenius (or Hilbert-Schmidt, or Schur) norm. This norm measures the matrix root-mean-square gain or average effect of "noise" variation along the orthogonal directions in the vector space. Sall (2015) affirmed how JMP's singular value decomposition can efficiently perform the multivariate analysis of wide data consisting of covariance matrices in the order of squares of the number of columns.
4. Among the issues with SVD is the difficulty in interpretability with matrices having over 23,000 singular elements that have mixed signs. Non-negative Matrix Factorization (NMF) is an alternative approach that was developed to produce non-negative element matrices that correct for these SVD deficiencies. For more information and access to JMP JSL scripts and addins, see Fogel et al. (2013).

REFERENCES

1. Albright, R (2004), "Taming Text with the SVD", Cary, NC: SAS Institute, Inc., <ftp://ftp.dataflux.com/techsup/download/EMiner/TamingTextwiththeSVD.pdf> (accessed 03/06/2015).
2. Alexander, M and Klick, J (2014), "Text Mining Feedback Comments from JMP® Users Group Meeting Participants", <https://community.jmp.com/docs/DOC-6748> (accessed 02/13/2015).
3. Bogard, M (2012), "An Intuitive Approach to Text Mining with SAS IML", <http://econometricsense.blogspot.com/2012/05/intuitive-approach-to-text-mining-vis.html> (accessed 02/13/2015).
4. Hastie, T, Tibshirani, R, and Friedman, J (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. pp. 79-80, 535-536, New York, NY: Springer-Verlag.
5. Karl, A, and Rushing, H (2013) "Text Mining with JMP and R", http://www.jmp.com/about/events/summit2013/resources/Paper_Karl_Rushing.pdf (accessed 02/26/2015).
6. McNeill, F (2014) "The Text Frontier – SAS Blog", <http://blogs.sas.com/content/text-mining/> (accessed 02/26/2015).
7. Mroz, P (2014) "Word Cloud in Graph Builder?", <https://community.jmp.com/thread/58441> (accessed 03/24/2015).
8. Preiss, J (2015), "Coming in JMP 12: Overhauled Recode for easier data cleaning", <http://blogs.sas.com/content/jmp/2015/01/21/coming-in-jmp-12-overhauled-recode-command/> (accessed 01/21/2015).
9. Rushing, H and Wisnowski, J (2015), "Harness the Power of Text Mining: Analyse FDA Recalls and Inspection Observations", <https://community.jmp.com/docs/DOC-7204> (accessed 03/19/2015)
10. Parris, J (2014), "Word Counts to Columns", <https://community.jmp.com/docs/DOC-7056> (accessed 02/13/2015).
11. Porter, MF (2006), "The Porter Stemming Algorithm", <http://tartarus.org/martin/PorterStemmer/> (accessed 02/26/2015).
12. Wicklin, R (2015), "Compute the rank of a matrix in SAS", <http://blogs.sas.com/content/iml/2015/04/08/rank-of-matrix.html> (accessed 04/08/2015).
13. Sall, J (2015), "Wide data discriminant analysis," <http://blogs.sas.com/content/jmp/2015/05/11/wide-data-discriminant-analysis/> (accessed 05/11/2015).
14. Fogel, P, Hawkins, DM, Beecher, C, Luta, G, and Young, SS, (2013), *A Tale of Two Matrix Factorizations, Technical Report 85*, Research Triangle Park, NC: National Institute of Statistical Sciences.

ACKNOWLEDGMENTS

I thank Robin Moran, Gail Massari, Tom Donnelly, John Sall, and the JMP Division of SAS® for their contributions and support; and Lucia Ward-Alexander for her review and editorial assistance.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Melvin Alexander
Social Security Administration
6401 Security Blvd.; East High Rise Building (5-A-10)
Baltimore, MD 21235
Phone: (410) 966-2155
Fax: (410) 966-4337
E-mail: Melvin.Alexander@ssa.gov

JMP, SAS and all other SAS Institute, Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

DISCLAIMER

The views expressed in this presentation are the author's and do not represent the views of the Social Security Administration or SAS Institute, Inc.