SESUG Paper 113-2019

Comparison Word Clouds Using the %PROC_R macro and Base SAS® Interface

Melvin Alexander, Analytician

ABSTRACT

This presentation gives an example of calling and integrating R code from the Base SAS® environment. SAS makes it possible to run R code via SAS/IML®, SAS/IML Studio®, or SAS/Viya® as described by Gilsen (2018). Interfacing R with these other SAS modules requires separate and installations that incur additional costs.

R has a rich set of machine learning, text mining packages, and advanced graphic capabilities and complements SAS. I will demonstrate the R and Base SAS integration to construct comparison word clouds using a modified version of the %PROC_R macro of Wei (2012).

INTRODUCTION

R integration with Base SAS has been possible using special macros as described by Wei (2012) and Bettinger (2016), or using the Java object described by Hall (2015), Revolution Analytics (2015), and Wujek (2015). Hemken (2019) described an R SASmarkdown package that launches SAS and redirects SAS logs and list output to simple document files in R or RStudio. SASmarkdown calls out the Windows operating system to launch SAS processes.

I will demonstrate this capability using a modification of the %PROC_R macro by Wei (2012).

To integrate R and base SAS, follow these steps:

1. Download the PROC_R macro code from Wei (2012).
2. Save the file in a location (e.g., "P:\My SAS Files\Proc_R3_fgname.sas").
3. Open the code and update the path of R executable file in the code below.

```
* add the location path where R version 3.5.0 is installed ;
    %macro quit(rpath=%str(C:\Progra~1\R\R-3.5.0\bin\R.exe));
```

Here the rpath refers to location where the R executable is saved on the same machine where SAS is installed. Be sure to install the R library packages used for the analysis.

4. Open Base SAS and call the PROC_R3 macro, a modification of Xin Wei's PROC_R macro (2012) that executes R code in base SAS. The Proc_R3_fgname.sas program (with the %PROC_R3 macro) suppresses macro variables &fgname and &fgsw from being printed to the SAS log that avoids errors when running the program.
5. Run R inside SAS environment. See the SAS program below:

```
%include "P:\My SAS Files\Proc_R3_fgname.sas"  ;
%Proc_R3(SAS2R =, R2SAS= MAJUGTDM) ;
   cards4 ;
   /********/
   R code ;
   /*******/
```

```
dev.off()
;;;;
%quit;
```

The macro variables are described as follows:

SAS2R - specifies the names of SAS datasets converted into R dataframes or objects. The dataset can be single file names or multiple files names separated by spaces.
R2SAS - specifies the names of R dataframes or objects that convert into SAS datasets. These objects can be single file names or multiple files names separated by spaces.

Example

Word clouds are text mining visualizations that shows the most common words that appear from a collection of comments made by respondents (documents) from post-meeting surveys about their impressions (what they liked, disliked, or what could be improved). Word clouds are an example of basic text mining.

In text mining, documents are the string of words, phrases, paragraphs, or other comments made by individual respondents who answer voice of the customer surveys. The collection of all comments (documents) is a corpus (that is, the collection of documents containing all the words of the collection).

Quite often in surveys, much of the important information in the corpus is not used, and any free-response text can be difficult to analyze unless one knows what to look for.

Unlike analyses for structured data, text mining requires special software to analyze unstructured, natural language data. R has these special packages in the tm, SnowballC, and wordcloud libraries. The data used in this example were from Mid-Atlantic JMP® Users Group (MAJUG) text comments reported by Alexander (2015).

SuggEx in Output 1 shows a corpus consisting of 8 “documents,” one per respondent. The term-document matrix (corpus2b.tdm) contains the frequencies of all 79 terms that occurred in the corpus. Term frequency weighting (tf) is the raw count of the row term found in the document columns. Corpus2b.tdm has 538 zeros and 94 nonzero values. “Sparsity” is 85 percent, indicating much of the matrix contains zeroes. This is not surprising because not all terms show up in all comments

Output 1. Output from the R Output log after running PROC_R3

**********R OUTPUT**********

R_OUTPUT_LOG

```
> library(grDevices)
>
>
```

```
> # Satisfaction scores from MAJUG meeting Attendees
> # (1 = low satisfaction, 3 = neutral, 5 = high satisfaction)
>
> # Load suggestions from Alexander (2015) Text Analytics paper from MAJUG meeting comments
> SuggEx <- c("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")
>
> # load libraries
> library(wordcloud) # draw wordclouds
Loading required package: RColorBrewer
> library(tm) # text analysis functions
Loading required package: NLP
> library(SnowballC) # reduce words to common stems
> library(qcc) # prepare Pareto chart
Package 'qcc' version 2.7
Type 'citation("qcc")' for citing this R package in publications.
Warning message:
package 'qcc' was built under R version 3.5.1
> library(ggplot2) # create other plots

Attaching package: 'ggplot2'

The following object is masked from 'package:NLP':

annotate

>
> # load text data
```

```
> Suggestion <- matrix(SuggEx)
> TEXTFILE = Corpus(VectorSource(c(Suggestion)))
> inspect(TEXTFILE)
<>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 8

[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 users in MAJUG, Professions, share email contacts. Web value (increase usefulness) What papers/presentations have occurred at MAJUG
[5] Query members planning to attend what they want to get out of the meeting so their concerns, questions, issues can be addressed and discussed
[6] Please start at 10
[7] Have coffee break with coffee, more communication between meetings, suggesting topics
[8] MAJUG should have a fee (perhaps $5) to buy refreshments so participants can get coffee without leaving meeting
>
> # clean and process text data
> newstopwords <- c("and", "for", "the", "to", "of", "in",
+ "as", "is", "with", "an", "then", "by", "they", "than", "he", "she")
> skipWords <- function(x) removeWords(x, stopwords("english"))
> funcs <- list(tolower, removePunctuation, removeNumbers,
+ stripWhitespace, stemDocument, skipWords)
> corpus2.proc <- tm_map(TEXTFILE, FUN = tm_reduce, tmFuns = funcs)
> corpus2.proc <- tm_map(corpus2.proc, removeWords, newstopwords)
> corpus2.proc <- tm_map(corpus2.proc, removeWords, stopwords("english"))
>
> # create term-document matrix and label column names with scores
> corpus2b.tdm <- TermDocumentMatrix(corpus2.proc)
> corpus2b.tdm
<>
Non-/sparse entries: 94/538
Sparsity : 85%
Maximal term length: 13
Weighting : term frequency (tf)
> m <- as.matrix(corpus2b.tdm)
> colnames(m) <- c("1:Score=5", "2:Score=4", "3:Score=5",
+ "4:Score=5", "5:Score=5", "6:Score=4", "7:Score=5",
```

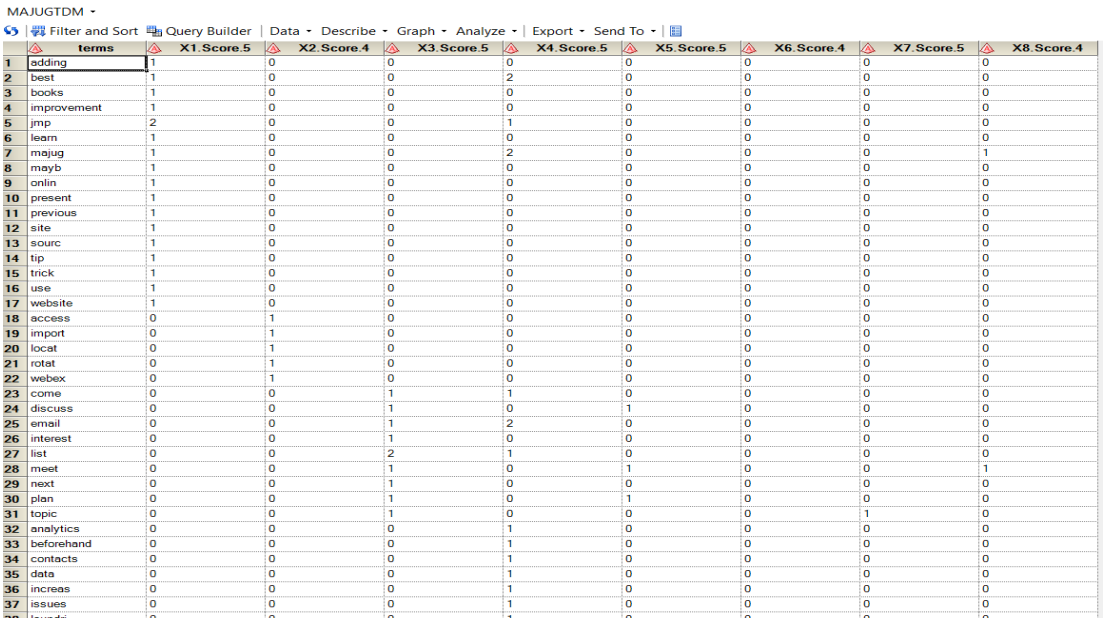
```
+ "8:Score=4")
> terms <- row.names(m)
>
> TDM3 <- cbind(terms,m)
> MAJUGTDM <- data.frame(TDM3)
>
> # produce Pareto chart with top 9 terms
> freq_terms <- rowSums(as.matrix(corpus2b.tdm))
> # freq_terms <- freq_terms[which(freq_terms>1)]
> x11()
> #  open the file MAJUGfreqtermsPareto to receive the graphics output
> png( file="P:/My SAS Files/MAJUGfreqtermsPareto.png", bg="white" )
> pareto.chart( rev(sort(freq_terms))[1:9],
+ main="MAJUG Meeting Comments Frequent Terms")

Pareto chart analysis for rev(sort(freq_terms))[1:9]
Frequency Cum.Freq. Percentage Cum.Percent.
majug 4 4 16 16
meet 3 7 12 28
list 3 10 12 40
email 3 13 12 52
jmp 3 16 12 64
best 3 19 12 76
coffe 2 21 8 84
get 2 23 8 92
can 2 25 8 100
>
> x11() # allow multiple screens to be diplayed with x11() function
> # produce word cloud
> sugwordcloud <- wordcloud(words=names(freq_terms),freq=freq_terms,
random.order=FALSE, colors=c("tomato","blue"))
> text(x=0.50, y=1.0, "Word Cloud of Most Used Terms Used by All MAJUG
Respondents")
> x11()
> # produce comparison cloud
> # open the file MAJUGComparisoncloud.png to receive graphics output
> png( file="P:/My SAS Files/MAJUGComparisoncloud.png", bg="white" )
> comparison.cloud(m,max.words=100,
+ colors = c("tomato","orangered2","cyan4","blue","maroon2",
+ "coral","chartreuse4","red"),
+ title.size=0.8,random.order=FALSE)
> text(x=0.5, y=1.0, "Comparison Cloud of Terms and Scores with Separate Colors for
Each Respondent")
>
```

```
> dev.off()
windows
2
> write.csv(MAJUGTDM,'MAJUGTDM.csv',row.names=F)
> q()
> proc.time()
user system elapsed
0.76 0.45 1.34
```

Using the text and graphical packages from R, I converted these comments into the Term-Document-Matrix (MAJUGTDM) after removing common stop words, such as “the”, “and”, punctuations, and reducing terms to their root, stem, or base form (e.g., replacing “location” with “locat”).

Table 1: Portion of the SAS Dataset MAJUGTDM



MAJUGTDM

Filter and Sort | Query Builder | Data | Describe | Graph | Analyze | Export | Send To

	terms	X1_Score_5	X2_Score_4	X3_Score_5	X4_Score_5	X5_Score_5	X6_Score_4	X7_Score_5	X8_Score_4
1	adding	1	0	0	0	0	0	0	0
2	best	1	0	0	2	0	0	0	0
3	books	1	0	0	0	0	0	0	0
4	improvement	1	0	0	0	0	0	0	0
5	jmp	2	0	0	1	0	0	0	0
6	learn	1	0	0	0	0	0	0	0
7	majug	1	0	0	2	0	0	0	1
8	mayb	1	0	0	0	0	0	0	0
9	onlin	1	0	0	0	0	0	0	0
10	present	1	0	0	0	0	0	0	0
11	previous	1	0	0	0	0	0	0	0
12	site	1	0	0	0	0	0	0	0
13	sourc	1	0	0	0	0	0	0	0
14	tip	1	0	0	0	0	0	0	0
15	trick	1	0	0	0	0	0	0	0
16	use	1	0	0	0	0	0	0	0
17	website	1	0	0	0	0	0	0	0
18	access	0	1	0	0	0	0	0	0
19	import	0	1	0	0	0	0	0	0
20	locat	0	1	0	0	0	0	0	0
21	rotat	0	1	0	0	0	0	0	0
22	webex	0	1	0	0	0	0	0	0
23	come	0	0	1	1	0	0	0	0
24	discuss	0	0	1	0	1	0	0	0
25	email	0	0	1	2	0	0	0	0
26	interest	0	0	1	0	0	0	0	0
27	list	0	0	2	1	0	0	0	0
28	meet	0	0	1	0	1	0	0	1
29	next	0	0	1	0	0	0	0	0
30	plan	0	0	1	0	1	0	0	0
31	topic	0	0	1	0	0	0	1	0
32	analytics	0	0	0	1	0	0	0	0
33	beforehand	0	0	0	1	0	0	0	0
34	contacts	0	0	0	1	0	0	0	0
35	data	0	0	0	1	0	0	0	0
36	increas	0	0	0	1	0	0	0	0
37	issues	0	0	0	1	0	0	0	0

Table 1 consisted of 79 terms as rows and 8 respondent as columns. This table was created specifying MAJUGTDM as the SAS dataset name for R2SAS macro variable.

Figure 1: Pareto Chart of the Most Frequently Used Term

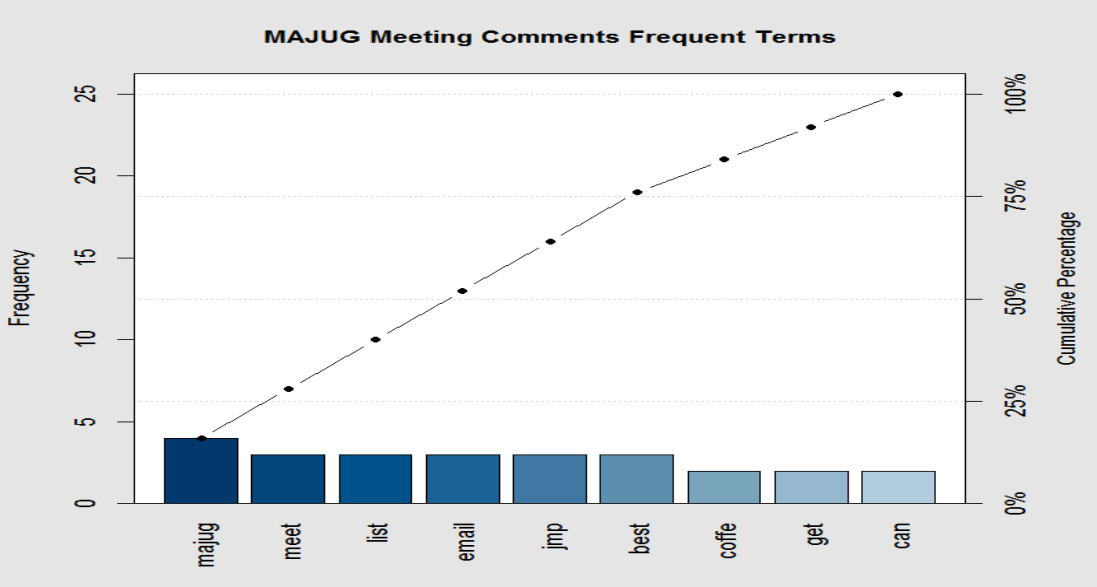
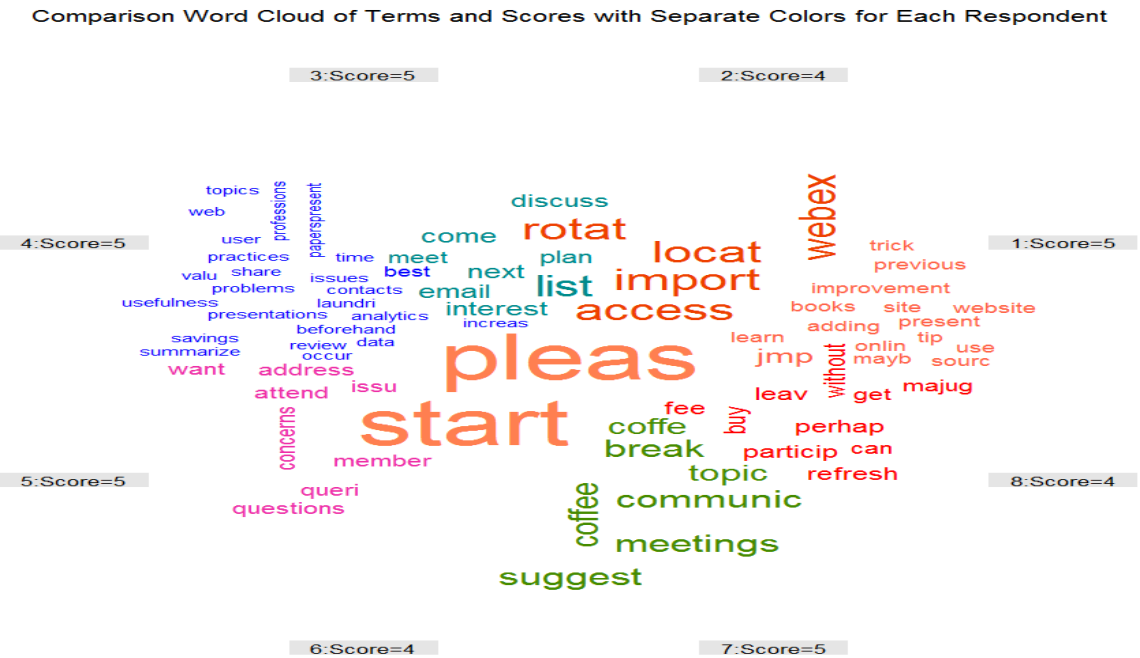


Figure 1 showed a Pareto chart of the most frequently occurring words in SuggEx corpus from Table 1.

Figure 2: Comparison Word Cloud showing relative frequency of Terms and Satisfaction Scores made by Each Respondent



The comparison cloud of Figure 2 shows distinguishing words that particular respondents used. Comparison clouds compare the relative frequency of terms used in two or more documents. They show the difference between how frequently words are used across documents. Comparison clouds contrast how individual respondents scored the particular meeting aspects that were recorded for these data.

The comparison cloud in Figure 2 shows that the most prominent terms “pleas” and “start” were from respondent 6 (stemmed from the comment “Please start at 10”). Larger (or smaller) sized terms used by respondents were more (less) prominent and had greater (less) weight or importance because of the principle of term frequency-inverse document frequency (or tf-idf). The tf-idf in text analytics increases in importance when terms appear multiple times in the same comments.

The second, most prominent terms from respondent 2 were “rotat”, “webex”, “import”, “access”, “locat” (stemmed from the “Rotating location and WebEx access is important” comment). This indicated that these terms were the most prominent factors for that respondent.

Respondents 1 and 4 had the least prominent (smaller) terms of the other respondents. This is because they had so many comments that made it difficult to determine what was of greatest importance to them.

CONCLUSION

Combining R with base SAS gives users enhanced analytics capabilities without the sole reliance of SAS/IML integration.

REFERENCES

Wei, X. (2012), "%PROC_R: A SAS Macro that Enables Native R Programming in the Base SAS Environment" *Journal of Statistical Software*, v.46, Code Snippet 2, https://www.jstatsoft.org/article/view/v046c02 .

Bettinger, R. (2016), "%SUBMIT R: A SAS(R) Macro to Interface SAS and R", *SESUG 2016 Conference Proceedings*, https://analytics.ncsu.edu/sesug/2016/AD-118_Final_PDF.pdf.

Gilsen, B. (2018), "Using the R interface SAS® to Call R Functions and Transfer Data", *SESUG 2018 Conference Proceedings*, https://www.lexjansen.com/sesug/2018/SESUG2018_Paper-119_Final_PDF.pdf.

Hall, P. (2015), “Tip: Open Source Integration Using the Base SAS Java Object”, https://communities.sas.com/t5/SAS-Communities-Library/Tip-Open-Source-Integration-Using-the-Base-SAS-Java-Object/ta-p/223697.

Hemken, D. (2019), “SAS Using R Markdown (Windows)”, https://www.ssc.wisc.edu/~hemken/SASworkshops/Markdown/SASmarkdown.html (accessed 7/25/2019).

Alexander, M., (2015), "Text Analytics Using JMP®", *SESUG 2015 Conference Proceedings*, https://analytics.ncsu.edu/sesug/2015/RV-31.pdf.

Revolution Analytics (2015), “Call R and Python from base SAS”, https://blog.revolutionanalytics.com/2015/05/call-r-and-python-from-base-sas.html .

Wujek, B (2015), “Open Source Integration Using the Base SAS Java Object”, https://github.com/sassoftware/enlighten-integration/tree/master/SAS_Base_OpenSrcIntegration

CONTACT INFORMATION

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

Melvin Alexander
811 Milford Mill Road
Baltimore, MD 21208-4633
Phone: (410) 458--7129
E-mail: Melvin.Alexander@verizon.net

ACKNOWLEDGMENTS

I thank Lucia Ward-Alexander for review and editorial assistance. Thanks to Jason Brinkley and Brian Varney for their help in this presentation.

APPENDIX

```
/*************************************************************************
 MAJUG Comparison Cloud using SAS PROC_R macro.sas

PROC_R3 is a modification of Xin Wei's PROC_R macro (2012) that executes R
code
in base SAS. Proc_R3 suppresses macro variables &fgname and &fgsw from being
printed to the SAS log that avoids errors when running the program.

MACRO Version:  1.0
**
SAS Version:    9.1.3/9.2/9.3
**
R Version:      3.5.0
**
Date:           Nov 24, 2011

Instruction: This SAS macro enables native R language to be embedded in and
executed along with a SAS program in the Base SAS environment under Windows
OS. This macro executes a user-defined R code in batch mode by calling the
```

```
unnamed pipe method within base SAS.

The R textual and graphical output can be routed to the SAS output window and
result viewer, respectively. Also, this macro automatically converts data
between
SAS datasets and R data frames such that the data and results from each
statistical
environment can be utilized by the other environment.

The objective of this macro is to leverage the strength of the R programming
language within the SAS environment in a systematic manner. Moreover,
this macro helps statistical programmers to learn a new statistical language
while staying in a familiar environment. The macro variables are described as
follows:

SAS2R - specifies the names of SAS datasets to be converted to R dataframe.
         Can be single file name or multiple files whose names are separated
         by space.
R2SAS  - specifies the names of R dataframes to be converted to SAS datasets.
         Can be single file name or multiple files whose names are
         separated by space.
**

rpath  - The full path and file names of R executable file for various R
version from
         2.11 to 3.5.0

 Authored:    Xin Wei, Ph.D.,
 Affiliation: Roche Pharmaceuticals, INC.
 Modified by: Mel Alexander


OUTPUT:
  SAS datasets retrieved from R:
  MAJUGTDM;

  Visualizations from R graphics:

  Pareto Chart of the most frequent used terms saved on
  "P:/My SAS Files/MAJUGfreqtermsPareto.png";

  Comparison Word cloud of terms used by documents on
  "P:/My SAS Files/MAJUGComparisoncloud.png".

R_OUTPUT_LOG shows the results after running R interactively from SAS
Enterprise
 Guide from the %Proc_R3 macro stored in the "P:\My SAS
Files\Proc_R3_fgname.sas"
 folder.

R_OUTPUT_LOG shows the results after running R interactively from SAS
Enterprise Guide
from the %Proc_R3_fgname macro stored in the "P:\My SAS
Files\Proc_R3_fgname.sas" folder.

References:
```

Wei, X. (2012), "%PROC_R: A SAS Macro that Enables Native R Programming in the Base SAS Environment" Journal of Statistical Sofware, v.46, Code Snippet 2, https://www.jstatsoft.org/article/view/v046c02.

Betinger, R. (2016), "%SUBMIT R: A SAS(R) Macro to Interface SAS and R", SESUG 2016 Conference Proceedings,https://analytics.ncsu.edu/sesug/2016/AD-118_Final_PDF.pdf.

Gilsen, B. (2018), "Using the R interface SAS® to Call R Functions and Transfer
Data",SESUG 2018 Conference Proceedings,
https://www.lexjansen.com/sesug/2018/SESUG2018_Paper-119_Final_PDF.pdf.

Alexander, M., (2015), "Text Analytics Using JMP®", SESUG 2015 Conference Proceedings, https://analytics.ncsu.edu/sesug/2015/RV-31.pdf.

Hall, P. (2015), "Tip: Open Source Integration Using the Base SAS Java Object",
https://communities.sas.com/t5/SAS-Communities-Library/Tip-Open-Source-Integration-Using-the-Base-SAS-Java-Object/ta-p/223697.

Revolution Analytics (2015), "Call R and Python from base SA",
https://blog.revolutionanalytics.com/2015/05/call-r-and-python-from-base-sas.html.

Wujek, B (2015), "Open Source Integration Using the Base SAS Java Object",
https://github.com/sassoftware/enlighten-integration/tree/master/SAS_Base_OpenSrcIntegration.

```
**************************************************************************/

/* Proc_R3_fgname.sas suppresses macro variables &fgname
  and &fgsw from being printed to the SAS log that avoid
  error when running the program.
*/
%include "P:\My SAS Files\Proc_R3_fgname.sas"  ;
%Proc_R3(SAS2R =, R2SAS = MAJUGTDM ) ;
cards4 ;

# Satisfaction scores from MAJUG meeting Attendees
# (1 = low satisfaction, 3 = neutral, 5 = high satisfaction)
# Load suggestions from Alexander (2015) Text Analytics paper from MAJUG meeting comments
SuggEx <- c("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")

# load libraries
library(wordcloud)   # draw wordclouds
library(tm)          # text analysis functions
library(SnowballC)   # reduce words to common stems
library(qcc)         # prepare Pareto chart
library(ggplot2)     # create other plots

# load text data
Suggestion <- matrix(SuggEx)
TEXTFILE = Corpus(VectorSource(c(Suggestion)))
inspect(TEXTFILE)

# clean and process text data
newstopwords <- c("and", "for", "the", "to", "of", "in",
"as", "is", "with", "an", "then", "by", "they", "than", "he", "she")
skipWords <- function(x) removeWords(x, stopwords("english"))
funcs <- list(tolower, removePunctuation, removeNumbers,
      stripWhitespace, stemDocument, skipWords)
corpus2.proc <- tm_map(TEXTFILE, FUN = tm_reduce, tmFuns = funcs)
corpus2.proc <- tm_map(corpus2.proc, removeWords, newstopwords)
corpus2.proc <- tm_map(corpus2.proc, removeWords, stopwords("english"))

# create term-document matrix and label column names with scores
corpus2b.tdm <- TermDocumentMatrix(corpus2.proc)
corpus2b.tdm
m <- as.matrix(corpus2b.tdm)
colnames(m) <- c("1:Score=5", "2:Score=4", "3:Score=5",
    "4:Score=5", "5:Score=5", "6:Score=4", "7:Score=5",
    "8:Score=4")
terms <- row.names(m)

TDM3 <- cbind(terms,m)
MAJUGTDM <- data.frame(TDM3)

# produce Pareto chart with top 9 terms
freq_terms <- rowSums(as.matrix(corpus2b.tdm))
# freq_terms <- freq_terms[which(freq_terms>1)]
x11()
# open the file MAJUGfreqtermsPareto.png to receive graphics output
png( file="P:/My SAS Files/MAJUGfreqtermsPareto.png", bg="white" )
pareto.chart( rev(sort(freq_terms))[1:9],
    main="MAJUG Meeting Comments Frequent Terms")

x11() # allow multiple screens to be displayed with x11() function
# produce word cloud
sugwordcloud <- wordcloud(words=names(freq_terms),freq=freq_terms,
random.order=FALSE, colors=c("tomato","blue"))
text(x=0.50, y=1.0, "Word Cloud of Most Used Terms Used by All MAJUG
Respondents")
x11()
# produce comparison cloud
# open the file MAJUGComparisoncloud.png to receive graphics output
```

```
png( file="P:/My SAS Files/MAJUGComparisoncloud.png", bg="white" )
comparison.cloud(m,max.words=100,
colors = c("tomato","orangered2","cyan4","blue","maroon2",
"coral","chartreuse4","red"),
title.size=0.8,random.order=FALSE)
text(x=0.5, y=1.0, "Comparison Cloud of Terms and Scores with Separate Colors
for Each Respondent")

dev.off()
;;;;
%quit;
```