Quality Tools used in Text Mining and Analytics (Tutorial- Matrix Diagram Analytics)

Melvin Alexander - Social Security Administration

ASQ Baltimore Section Meeting
October 10, 2017

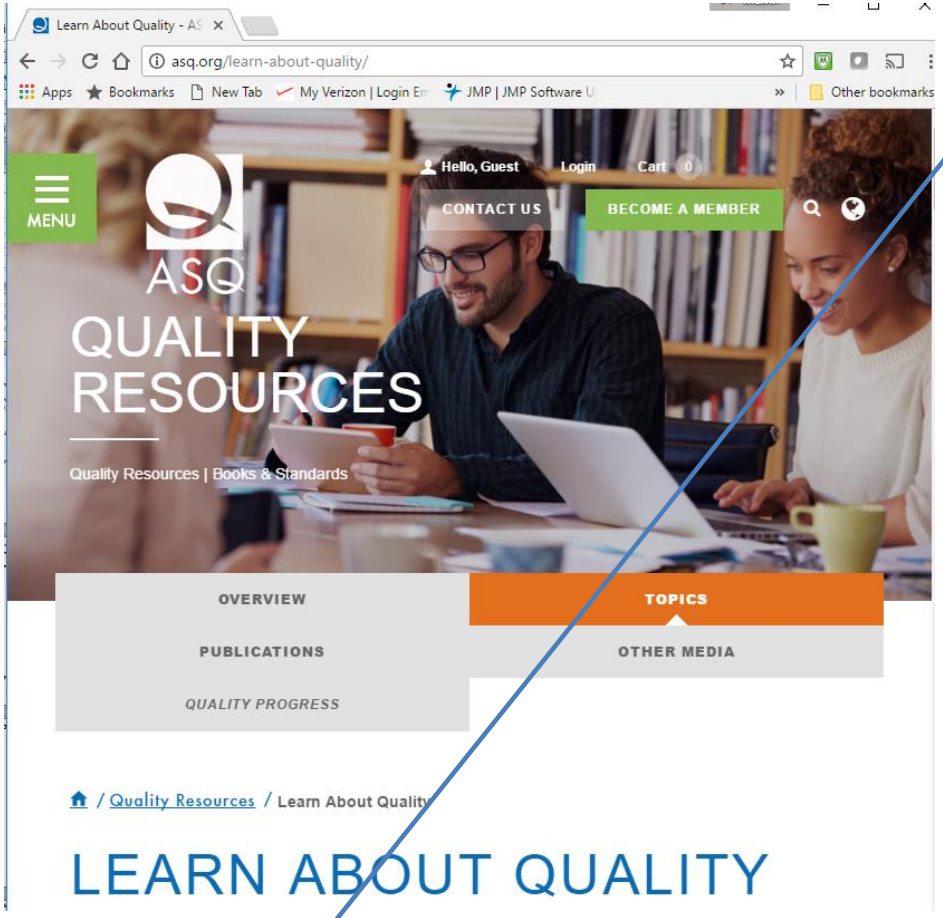
Disclaimer

- The views expressed in this presentation are those of the presenters and do not necessarily represent the views of the Social Security Administration(SSA).

Agenda

- Purpose: Review Basic Glossary of Text Mining concepts
- Apply Matrix Diagrams from the Quality Toolbox to unstructured, free-text analyses
- Summary and Conclusions
- Q & A

http://asq.org/learn-about-quality/



/ Quality Resources / Learn About Quality / New Management Planning Tools / Matrix Diagram

MATRIX DIAGRAM

Also called: matrix, matrix chart

The matrix diagram shows the relationship between two, three or four groups of information. It also can give information about the relationship, such as its strength, the roles played by various individuals or measurements.

Six differently shaped matrices are possible: L, T, Y, X, C and roof–shaped, depending on how many groups must be compared.

When to Use Each Matrix Diagram Shape

Table 1 summarizes when to use each type of matrix. Also click on the links below to see an example of each type. In the examples, matrix axes have been shaded to emphasize the letter that gives each matrix its name.

- An L–shaped matrix relates two groups of items to each other (or one group to itself).
- A T–shaped matrix relates three groups of items: groups B and C are each related to A. Groups B and C are not related to each other.
- A Y–shaped matrix relates three groups of items. Each group is related to the other two in a circular fashion.
- A C–shaped matrix relates three groups of items all together simultaneously, in 3-D.
- An X–shaped matrix relates four groups of items. Each group is related to two others in a circular fashion.
- A roof–shaped matrix relates one group of items to itself. It is usually used along with an L – or T–shaped matrix.

L-Shaped Matrix Diagram

This L-shaped matrix summarizes customers' requirements. The team placed numbers in the boxes to show numerical specifications and used check marks to show choice of packaging. The L-shaped matrix actually forms an upside-down L. This is the most basic and most common matrix format.

Customer Requirements

	Customer *D*	Customer *M*	Customer *R*	Customer *T*
Purity %	> 99.2	> 99.2	> 99.4	> 99.0
Trace metals (ppm)	< 5	—	< 10	< 25
Water (ppm)	< 10	< 5	< 10	—
Viscosity (cp)	20-35	20-30	10-50	15-35
Color	< 10	< 10	< 15	< 10
Drum		✔		
Truck	✔			✔
Railcar			✔	

Text Mining Basics

- Document is a string of words (e.g., phrases, comments made by respondents to a voice-of-the customer survey, paragraphs, etc.).
- A corpus is a collection of documents that contain all the words that appear in the collection.
- The term-document matrix (TDM) is a L-shaped matrix whose rows are the terms and columns are the documents. Each cell entry, $T_{(i, j)}$, represents the frequency (number of times) that the i-th term T was used in the j-th document.
- The TDM is used for information retrieval that produce a term frequency—inverse document frequency (tf-idf) measure. The tf-idf increases when term i appears frequently in document j, but decreases as the term appears in other documents.
- This tf-idf score is often used to rank documents in search results. It ranks high for documents that contain high frequency the terms in the search query, especially rare or unique words rather than common words (e.g. "the", “I”, “a”), often called stopwords.

Uses of Text Mining

- Predict probabilities of fraudulent insurance claims based on text in the claims.
- Produce lists of documents or responses which are most similar.
- Find sources of system failures associated with text in incident reports, error or repair logs.
- Identify root causes of error in FMEA studies
- Obtain fast, representative summaries of topics in a collection of documents.

Corpus - Collection of comments or documents found in Voice-of-the-Customer (VOTC) table made up of Text Comments made by 3 Survey Respondents (documents)

A corpus with 3 text documents (Comments about the Section Meeting Program)

The metadata consists of 2 tag-value pairs and a data frame
Available tags are:
create_date creator
Available variables in the data frame are:
MetaID

[[1]]
Enjoyed this venue. Food was good.

[[2]]
Environment good for meetings. Good sound, dual screens.

[[3]]
The location at the CCMT is great. The food was good except for what I thought was mashed potatoes

Lower-case Text Comments with Blank Term Document Matrix (TDM)

[Documents [Respondent]]
[[1]]
Enjoyed this venue.
Food was good.
[[2]]
Environment good for meetings. Good sound, dual screens.
[[3]]
The location at the CCMT is great. The food was good except for what I thought was mashed potatoes

Term\Document	[[1]]	[[2]]	[[3]]
enjoyed			
this			
venue.			
food			
was			
good.			
environment			
good			
for			
meetings			
sound,			
dual			
screens.			
the			
location			
at			
ccmt			
is			
great.			
except			
what			
I			
thought			
mashed			
potatoes			

Text Comments with Partially Filled Term Document Matrix (TDM)

Exercise: Fill in the blanks with the counts of the other terms

[Documents [Respondent]]
[[1]]
Enjoyed this venue.
Food was good.
[[2]]
Environment good for meetings. Good sound, dual screens.
[[3]]
The location at the CCMT is great. The food was good except for what I thought was mashed potatoes

Term\Document	[[1]]	[[2]]	[[3]]
enjoyed	1	0	0
this	1	0	0
venue.	1	0	0
food	1	0	1
was	1	0	2
good.	1	0	0
environment	0	1	0
good	0	2	1
for			
meetings			
sound,			
dual			
screens.			
the			
location			
at			
ccmt			
is			
great.			
except			
what			
I			
thought			
mashed			
potatoes			

Text Comments with Filled Term Document Matrix (TDM) Solution

[Documents [Respondent]]
[[1]]
Enjoyed this venue.
Food was good.
[[2]]
Environment good for meetings. Good sound, dual screens.
[[3]]
The location at the CCMT is great. The food was good except for what I thought was mashed potatoes

Term\Document	[[1]]	[[2]]	[[3]]
enjoyed	1	0	0
this	1	0	0
venue.	1	0	0
food	1	0	1
was	1	0	2
good.	1	0	0
environment	0	1	0
good	0	2	1
for	0	1	0
meetings	0	1	0
sound,	0	1	0
dual	0	1	0
screens.	0	1	0
the	0	0	3
location	0	0	1
at	0	0	1
ccmt	0	0	1
is	0	0	1
great.	0	0	1
except	0	0	1
what	0	0	1
I	0	0	1
thought	0	0	1
mashed	0	0	1
potatoes	0	0	1

Final Term Document Matrix (TDM) Some Stopwords removed (e.g., I, at, for, is)

```
> corpus2b.tdm <- TermDocumentMatrix(TEXTFILE)
> inspect(corpus2b.tdm)
A term-document matrix (22 terms, 3 documents)

Non-/sparse entries: 26/40
Sparsity           : 61%
Maximal term length: 11
Weighting          : term frequency (tf)

              Docs
Terms          1 2 3
  ccmt         0 0 1
  dual         0 1 0
  enjoyed      1 0 0
  environment  0 1 0
  except       0 0 1
  food         1 0 1
  for          0 1 1
  good         0 2 1
  good.        1 0 0
  great.       0 0 1
  location     0 0 1
  mashed       0 0 1
  meetings.    0 1 0
  potatoes     0 0 1
  screens.     0 1 0
  sound,       0 1 0
  the          0 0 3
  this         1 0 0
  thought      0 0 1
  venue.       1 0 0
  was          1 0 2
  what         0 0 1
> dim(corpus2b.tdm)
[1] 22  3
```

Total elements = 22 *3 = 66

# of Non-zero cells=26, zero cells =40; 26+40 = 66

Sparsity percentage = 100*(40/66)= 60.6 or 61%

Because most of the cells (61%) were zeros, the standard statistical methods such as chi-square or Fisher's Exact Tests cannot be used to analyze sparse data.

Why?
For chi-square, Critical Assumption that 5% of expected cells must be at least 5 is violated which invalidates the chi-square approach.

For Fisher's Exact Test, lots of tables with probabilities that have the same row and marginal totals are needed which require too many computational resources (as the size of the TDM increases). Fisher's Exact Test compares the observed table with all other tables having the same row-marginal totals.

Solution: Use statistical approaches and software designed to analyze sparse data, i.e., Singular Value Decomposition and free, open-source R software.

SVD Formula Definition

$$TDM_{[t \times d]} = U_{[t \times r]} D_{[r \times r]} V^{T}_{[r \times d]}$$

d

t

TDM:
t x d
(t terms, d documents)
sparse-term-document-matrix

=

r

t

U:
t x r
(t terms, r concepts)
left-singular {LS}, rank-reduced eigenvector term matrix

x

r

r

D:
r x r (r rank of matrix; strength of each 'concept') diagonal {D} matrix of singular eigenvalues, where
r= min(t,d)

x

d

r

V^T:
r x d (r concepts, d documents)
right-singular {RS}, rank-reduced eigenvector document matrix

Note: In R Principal Components (PCs) label the columns U and d where PC is the matrix product of U x diag(D)

Text Comments with Term Document Matrix (TDM) Stopwords removed and Term Principal Components ($U_{[t \times r]}$) Tables (after stemming)

Term\Document	X1	X2	X3
ccmt	0	0	1
dual	0	1	0
enjoyed	1	0	0
environment	0	1	0
except	0	0	1
food	1	0	1
good	0	2	1
good.	1	0	0
great.	0	0	1
location	0	0	1
mashed	0	0	1
meetings.	0	1	0
potatoes	0	0	1
screens.	0	1	0
sound,	0	1	0
the	0	0	3
this	1	0	0
thought	0	0	1
venue.	1	0	0
was	1	0	2
what	0	0	1

```
> dim(corpus2b.tdm)
[1] 22  3
```

Term\Principal Components	PC1	PC2	PC3
ccmt	0.269	-0.180	0.077
dual	-0.245	-0.237	0.201
enjoy	-0.023	0.417	0.329
environ	-0.245	-0.237	0.201
except	0.269	-0.180	0.077
food	0.245	0.237	0.684
good	-0.245	-0.237	0.297
great	0.269	-0.180	0.077
locat	0.269	-0.180	0.077
mash	0.269	-0.180	0.077
meetings	-0.245	-0.237	0.201
potato	0.269	-0.180	0.077
screens	-0.245	-0.237	0.201
sound	-0.245	-0.237	0.201
the	0.269	-0.180	0.077
thought	0.269	-0.180	0.077
venue	-0.023	0.417	0.291

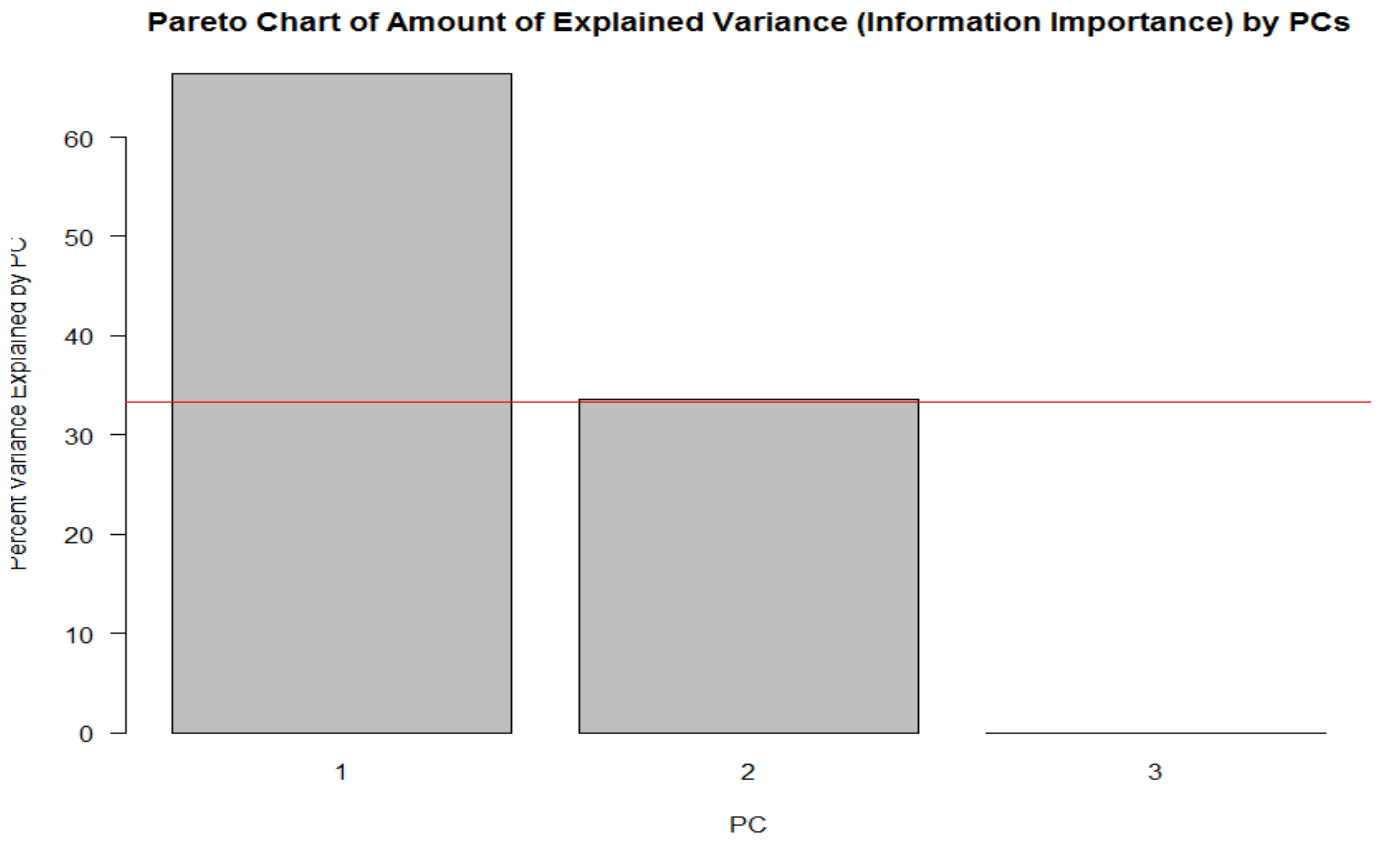
```
> dim(loadings)
[1] 17  3
```

Sample Output from R Summarizing the Right Singular Matrix ($\mathbf{V}^T$) of 3 Principal Components for 3 Documents

Docs	PC1	PC2	PC3
1	-0.3025826	2.752292	-3.885781e-16
2	-3.1976212	-1.562653	7.202572e-15
3	3.5002038	-1.189639	-6.578071e-15

Output from R Summarizing the Importance of the Principal Components from $D_{[r \times r]}$

```
Importance of components:
                            PC1    PC2      PC3
Standard deviation     3.3591 2.3908 6.791e-16
Proportion of Variance 0.6638 0.3362 0.000e+00
Cumulative Proportion  0.6638 1.0000 1.000e+00
```



SVD Formula Definition

$$TDM_{[t \times d]} = U_{[t \times r]} D_{[r \times r]} V^T_{[r \times d]}$$

$TDM_{[t \times d]}$

0	0	1
0	1	0
1	0	0
0	1	0
0	0	1
1	0	1
0	2	1
1	0	0
0	0	1
0	0	1
0	0	1
0	1	0
0	0	1
0	1	0
0	1	0
0	0	3
1	0	0
0	0	1
1	0	0
1	0	2
0	0	1

=

$U_{[t \times r]}$

0.269	-0.180	0.077
-0.245	-0.237	0.201
-0.023	0.417	0.329
-0.245	-0.237	0.201
0.269	-0.180	0.077
0.245	0.237	0.684
-0.245	-0.237	0.297
0.269	-0.180	0.077
0.269	-0.180	0.077
0.269	-0.180	0.077
-0.245	-0.237	0.201
0.269	-0.180	0.077
-0.245	-0.237	0.201
-0.245	-0.237	0.201
0.269	-0.180	0.077
0.269	-0.180	0.077
-0.023	0.417	0.291

$D_{[r \times r]}$

3.35914903
2.3908404
6.7911E-16

x

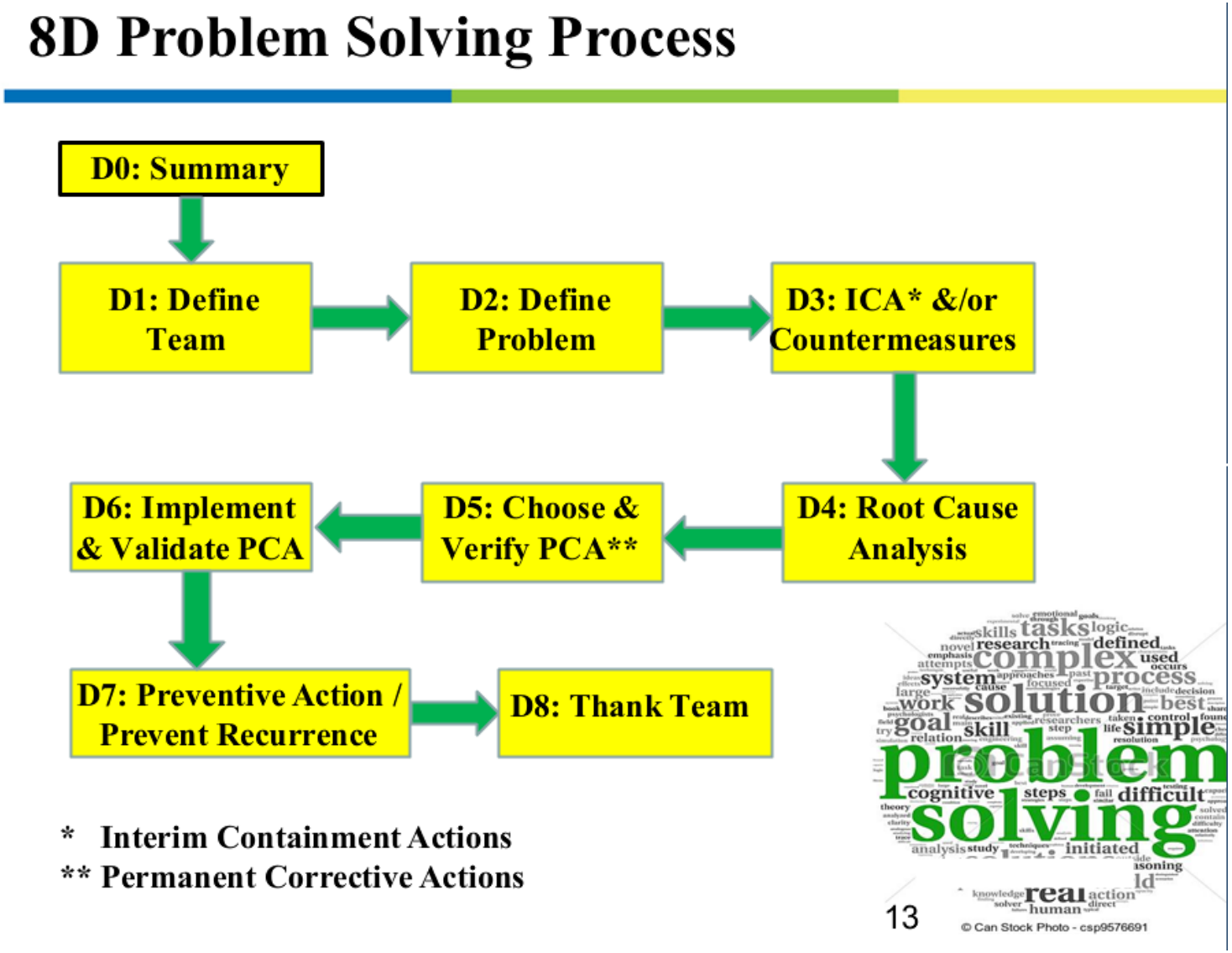
$V^T_{[r \times d]}$

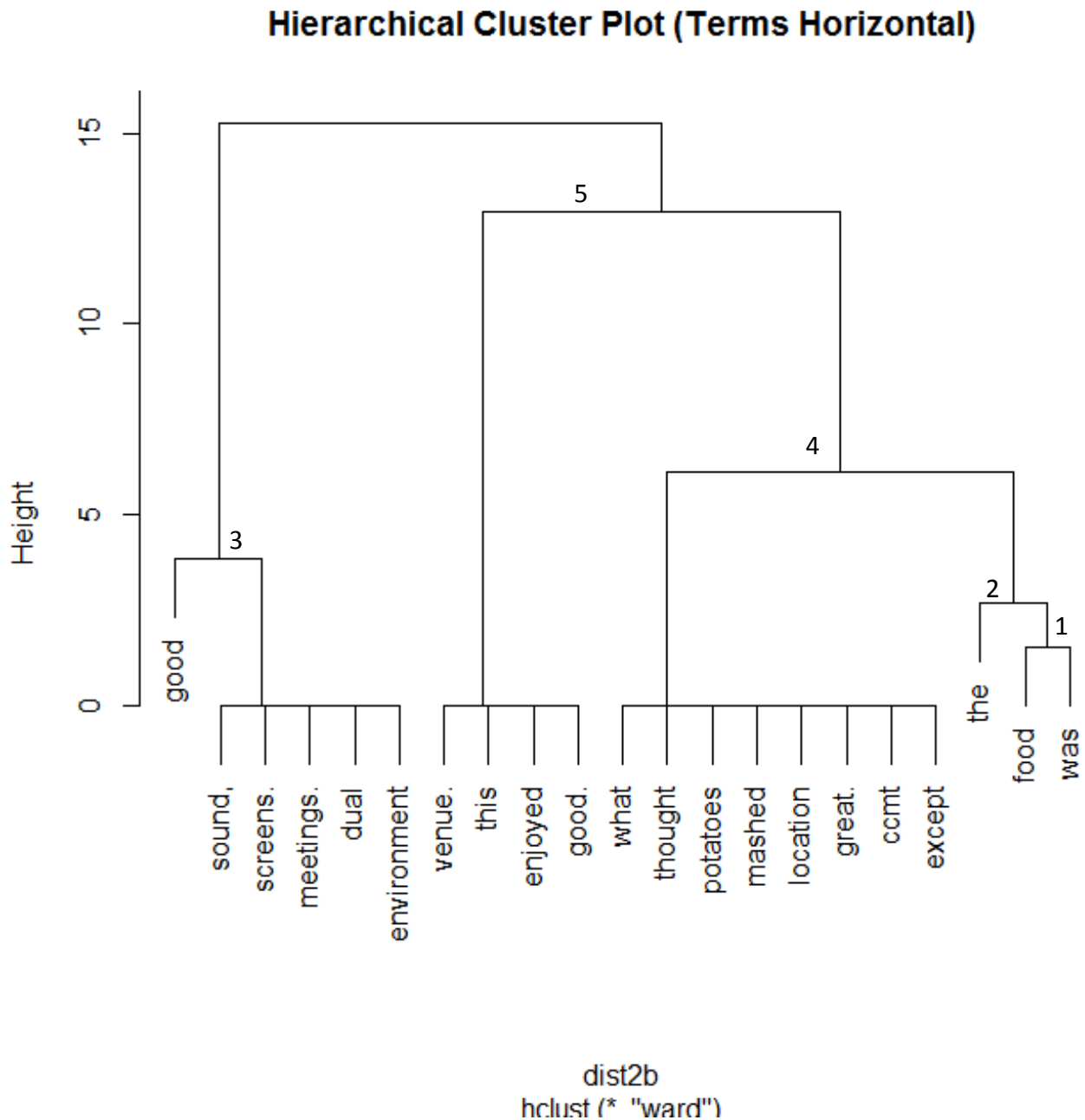
-0.30258	2.752292	-3.9E-16
-3.19762	-1.56265	7.2E-15
3.500204	-1.18964	-6.6E-15

Note: In R Principal Components (PCs) label the columns U and d where PC is the matrix product of U x diag(D)

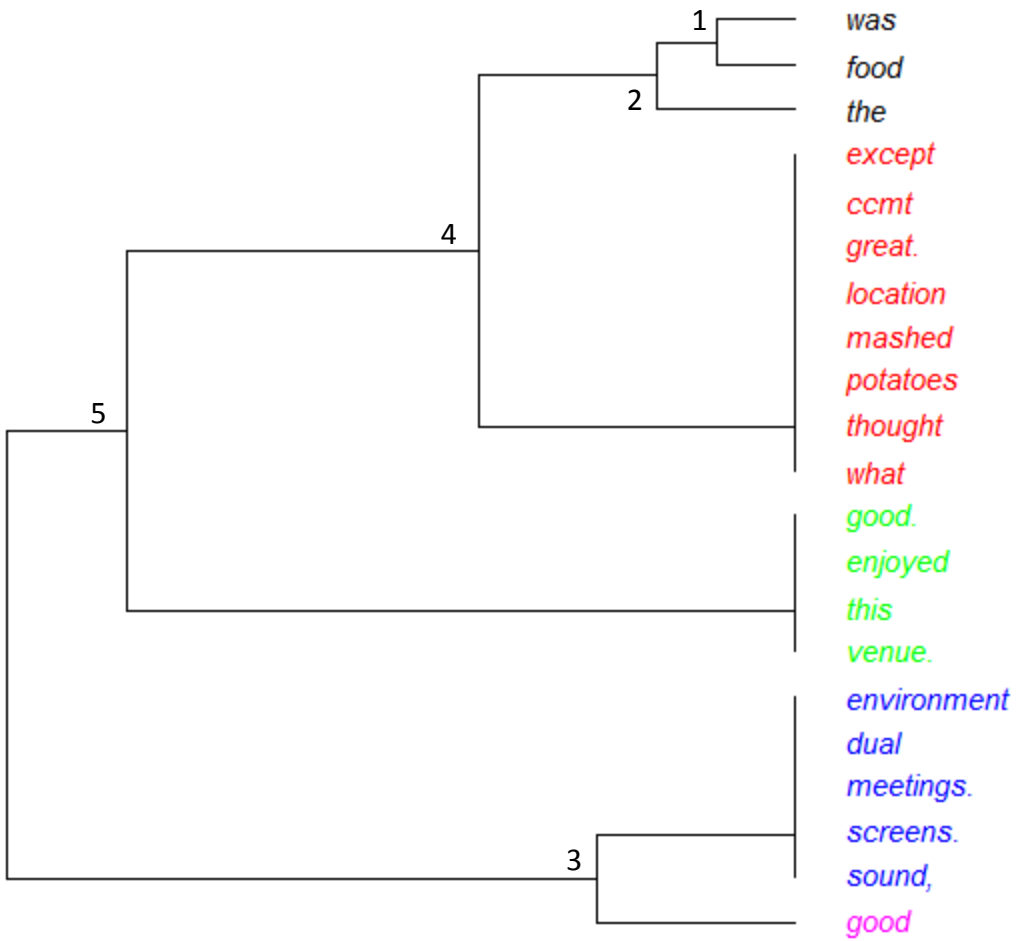
Word Cloud formed from the most-used terms by the Comments
(Pareto Analysis of Words)

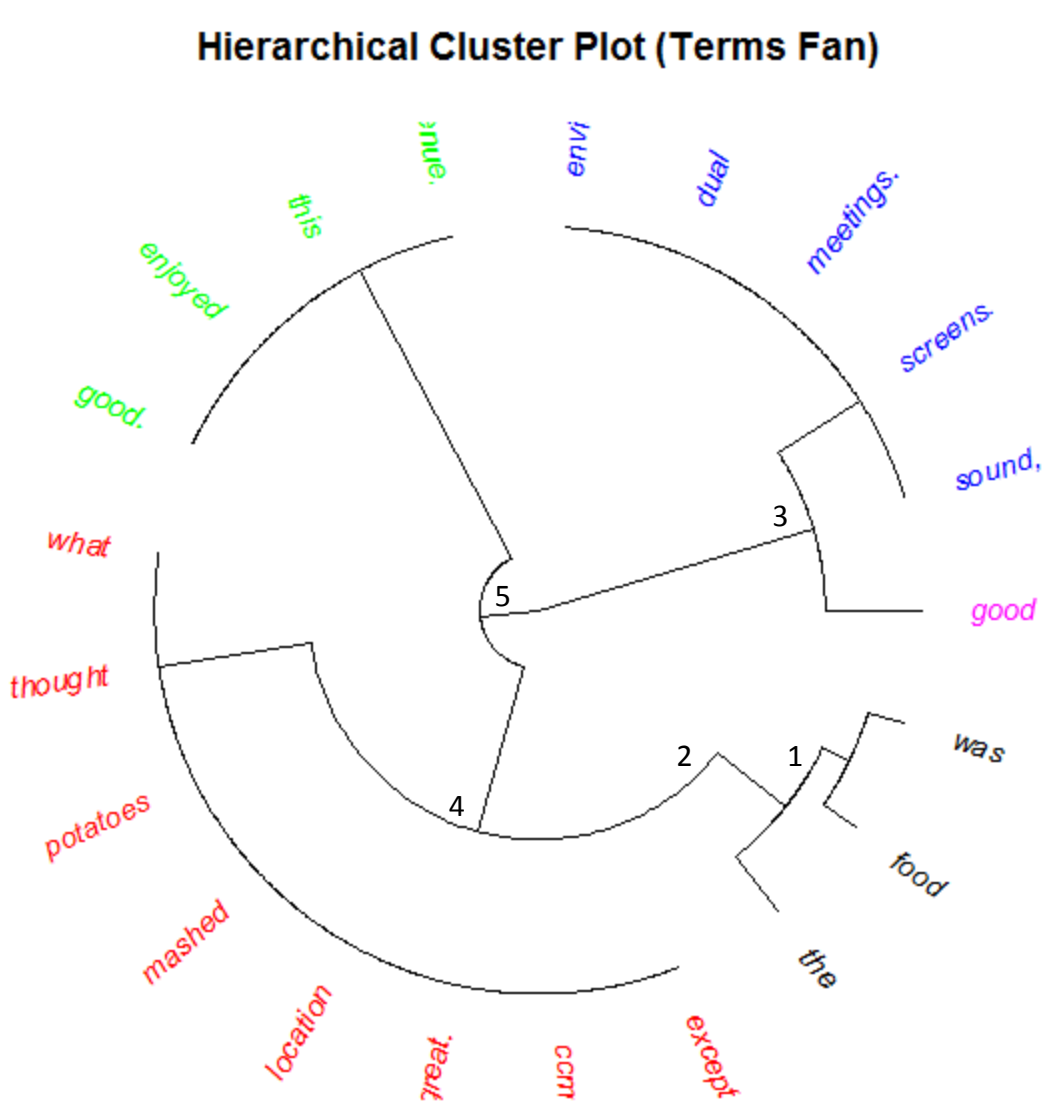
Example of a Word Cloud Depicted in the 8D (8 Disciplines) Problem Solving Process (Source: Jack B. ReVelle "8D Problem Solving", ASQ Webinar, 14Oct2014, http://asq.org/membership/members/gift/index.html)



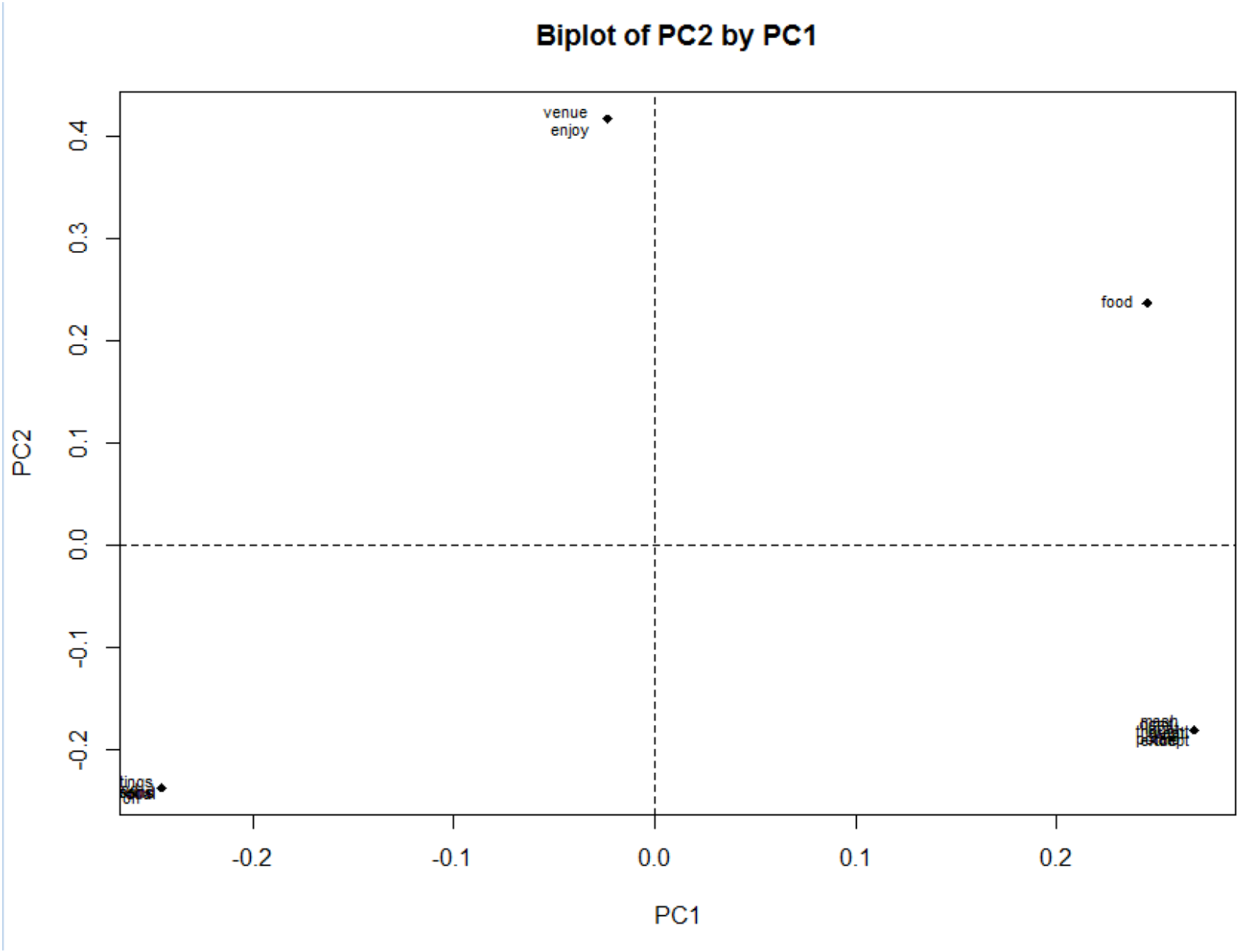

Hierarchical Cluster Plot (Terms Horizontal)
Height
0
5
10
15
3
5
4
2
1
good
sound,
screens.
meetings.
dual
environment
venue.
this
enjoyed
good.
what
thought
potatoes
mashed
location
great.
ccmt
except
the
food
was
dist2b
hclust (*, "ward")

Hiearchical Cluster Plot (Terms Vertical)

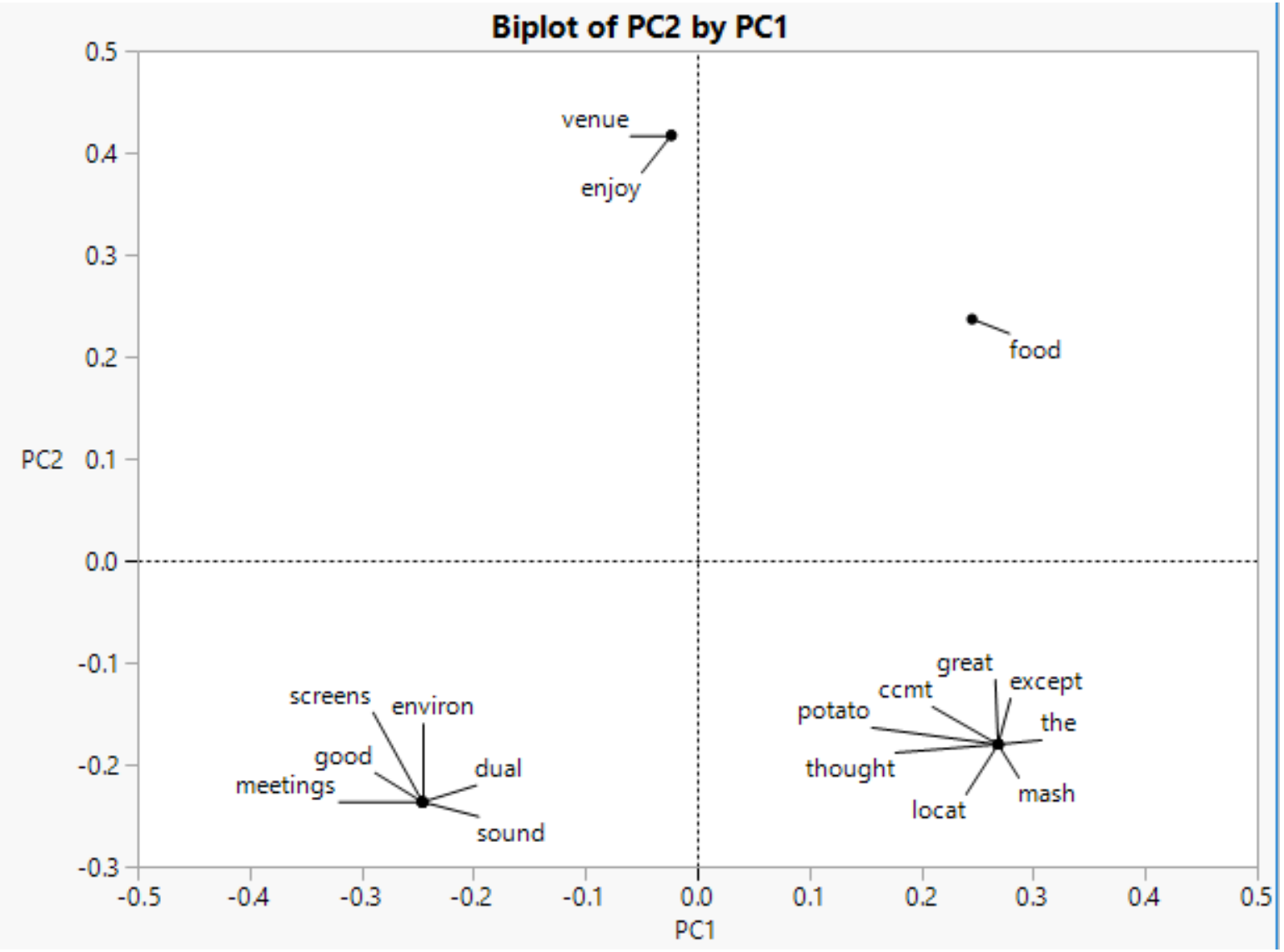



Hierarchical Cluster Plot (Terms Fan)
.nue
envi
dual
meetings.
screens.
sound,
good
was
food
the
except
ccm
reat.
location
mashed
potatoes
thought
what
good.
enjoyed
this
5
3
1
2
4

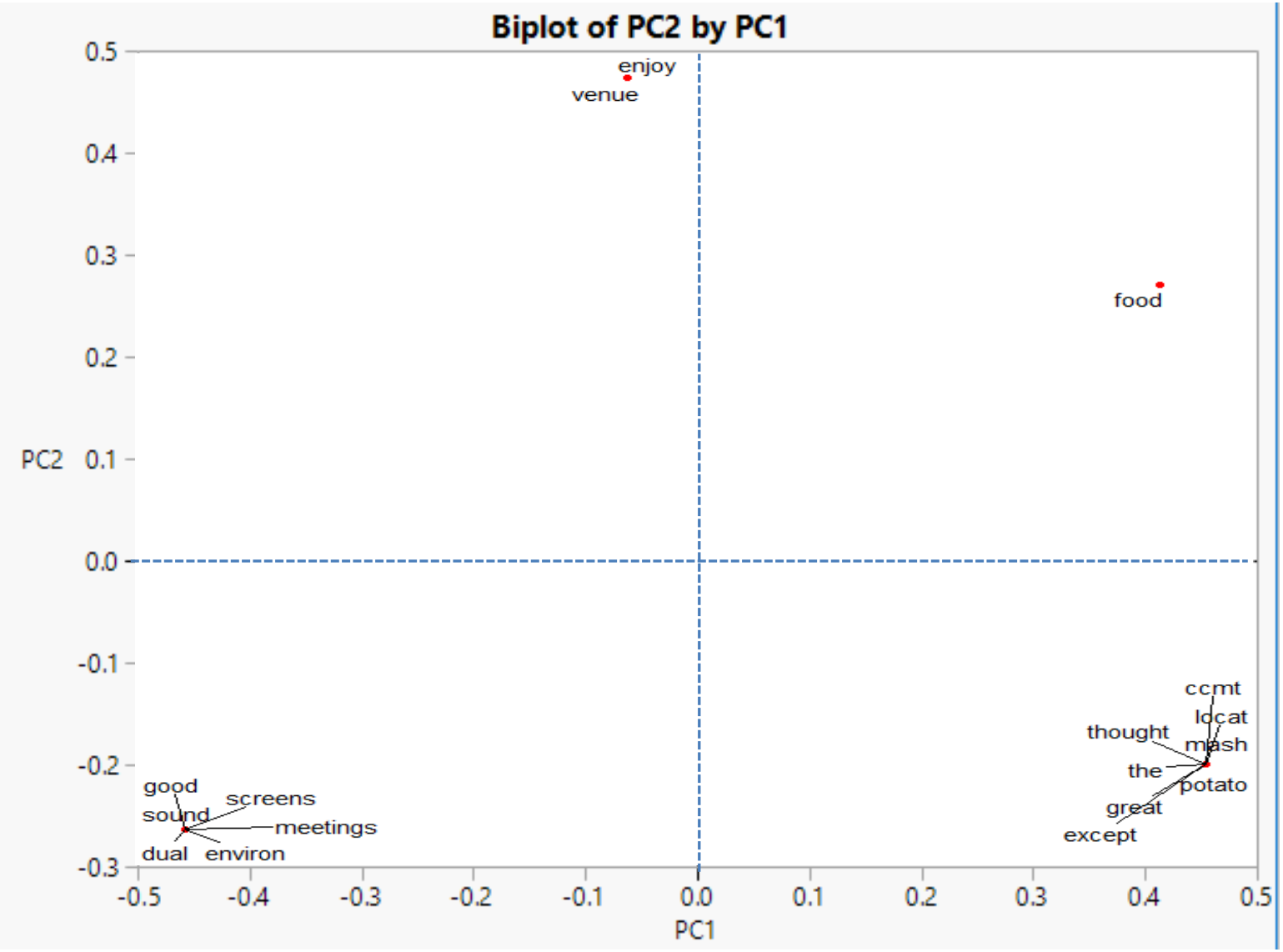
Latent Sematic Analysis of Themes formed by the Principal Component Bi-Plot of Terms



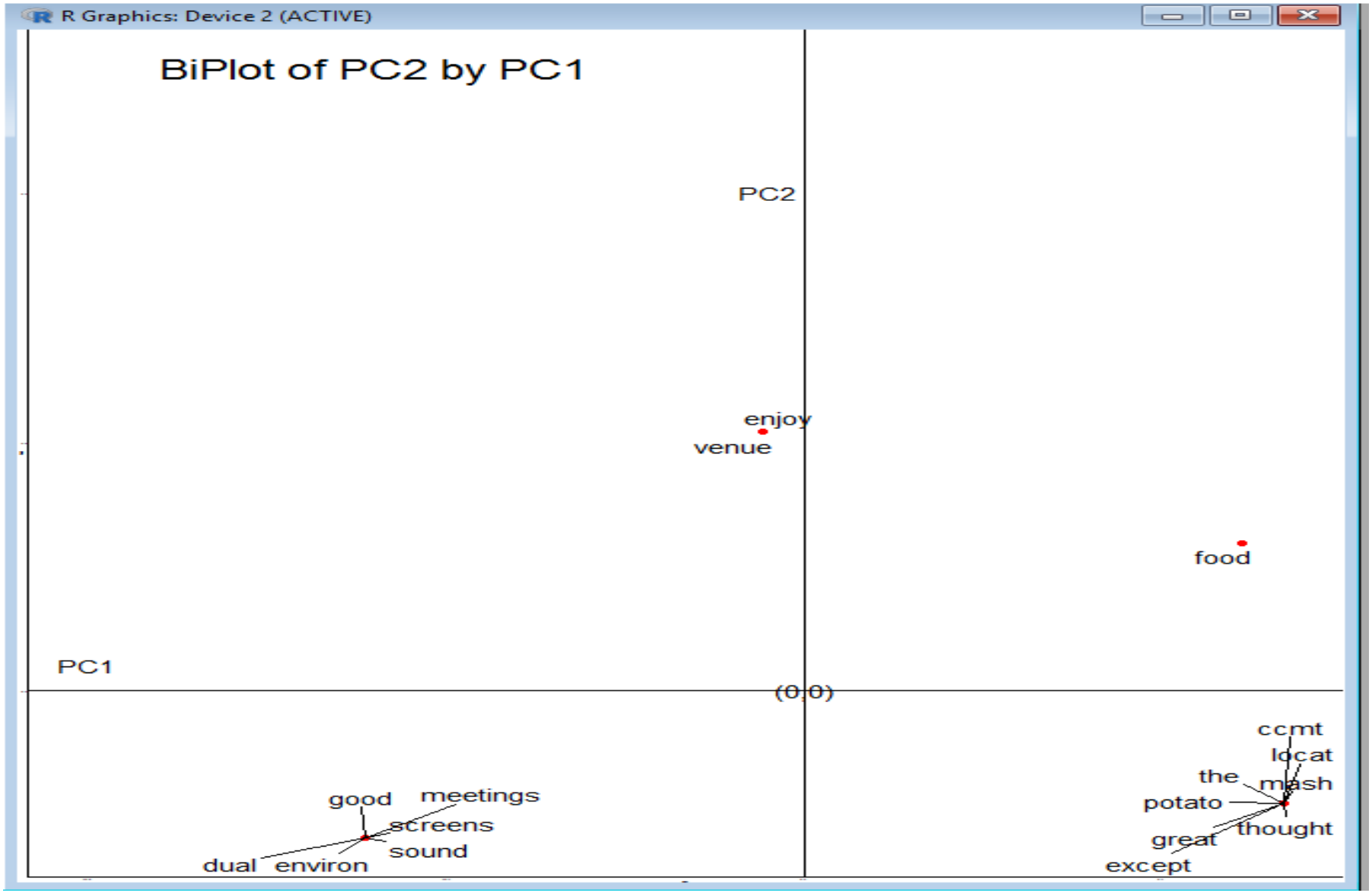
Latent Sematic Analysis of Themes formed by the Principal Component Bi-Plot of Terms Repelled from Vertices



Latent Sematic Analysis of Themes formed by the Principal Component Bi-Plot of Terms Repelled from Vertices (using a different randomization seed)



Latent Sematic Analysis of Themes formed by the Principal Component Bi-Plot of Terms Repelled from Vertices (using a different randomization seed)



How are these techniques useful?

- If someone gave you a collection of 20,000 paragraphed reports and asked you to summarize them in an hour, how would you do it?
- You could take a sample of 40 reports and put them into 40 clusters.
- Sample a document from each cluster to get a representative subset of documents.
- The techniques covered in this tutorial (and presentation to follow) provide a “quick” way of accomplishing the task in the allotted time.

Why use R?

- Open source software widely used by academicians and researchers for statistics and graphics.
- Has the latest, up-to-date statistical techniques and advanced graphics.
- Lots of documentation and tutorials on the internet.
- It's free and can be downloaded and installed on Windows, Mac, and Linux operating systems (Get R from https://www.r-project.org/).

Drawbacks using R

- Requires understanding and running code on an interpreter.
- Computes everything in memory (RAM) in 32-bit and 64-bit systems.
- Prone to error because newer techniques have not been fully test as done with commercial software.
- Has limited technical support despite its vast online user community.
- Some additional packages require upgraded versions of R for them to run

Questions?

Contact:

Melvin.Alexander@ssa.gov

About Mel Alexander

- Operations Research Analyst- Social Security Administration
- Analytician – Depts. of Diagnostic Radiology & Neurosurgery, University of Maryland Medical Center
- MS in Biostatistics – UNC, Chapel Hill
- Past Chair- ASQ Baltimore Section & Healthcare Division
- Former Member of ASQ's Certification Board (Quality Inspector, Recertification, and CQE Examinations) & Division Affairs Council (now Technical Communities Council)
- ASQ Fellow & CQE
- Authored "Expanding the Six Sigma Toolkit" series in ASQ's Six Sigma Forum Community and Book Reviews in *Journal of Quality Technology* and *Technometrics*; over 30 published papers in engineering, medical journals and technical conference proceedings
- Current Standing Review Board member for ASQ Quality Press and member of 10 Divisions (Audit, Biomedical, CPID, Education, FD&C, Government, Healthcare, Inspection, Six Sigma Forum, Statistics)

Text Comments with Filled Term Document Matrix (TDM) Solution

[Documents [Respondent]]
[[1]]
Enjoyed this venue.
Food was good.
[[2]]
Environment good for meetings. Good sound, dual screens.
[[3]]
The location at the CCMT is great. The food was good except for what I thought was mashed potatoes

Term\Document	X1	X2	X3
ccmt	0	0	1
dual	0	1	0
enjoyed	1	0	0
environment	0	1	0
except	0	0	1
food	1	0	1
good	0	2	1
good.	1	0	0
great.	0	0	1
location	0	0	1
mashed	0	0	1
meetings.	0	1	0
potatoes	0	0	1
screens.	0	1	0
sound,	0	1	0
the	0	0	3
this	1	0	0
thought	0	0	1
venue.	1	0	0
was	1	0	2
what	0	0	1

R code that Completed the Tutorial Part of the Program

```
SuggEx <- c("Enjoyed this venue.  Food was good.","Environment good for meetings.  Good sound, dual screens.",
"The location at the CCMT is great.  The food was good except for what I  thought was mashed potatoes")

library(wordcloud)
library(tm)
library(SnowballC)
# generates hierarchical cluster dendrograms
library(slam)


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

#remove new stopwords
TEXTFILE <-tm_map(TEXTFILE, removeWords)

#Document Term Matrix
corpus2a.dtm <- DocumentTermMatrix(corpus2.proc, control = list(wordLengths = c(3,10)))
dtmDataFrame <- as.data.frame(inspect(corpus2a.dtm))
# find frequent terms which occured more than 2 times
findFreqTerms(corpus2a.dtm, 1)
# Find Associations between words with high association (50%) with meetings
findAssocs(corpus2a.dtm, "meetings", 0.5)
#Term Document Matrix
corpus2b.tdm <- TermDocumentMatrix(TEXTFILE)
inspect(corpus2b.tdm)
dim(corpus2b.tdm)

# display the tdm
m <- as.matrix(corpus2b.tdm)

tdm.common <- removeSparseTerms(corpus2b.tdm,0.1)
#tdm.common
dim(tdm.common)
prcomponent <- prcomp(corpus2a.dtm,scale=TRUE, retx=TRUE,center=TRUE)
summary(prcomp(corpus2a.dtm,scale=TRUE,retx=TRUE,center=TRUE))
loadings <- prcomponent$rotation
dim(loadings)
```

R code that Completed the Tutorial Part of the Program (cont.)

```
loadingnames <- colnames(corpus2a.dtm)
scores <- prcomponent$x

eigvals <- prcomponent$sdev
var <- eigvals^2
var.percent <- var/sum(var) * 100
barplot(var.percent, xlab="PC", ylab="Percent Variance Explained by PC",
names.arg=1:length(var.percent), las=1, ylim=c(0, max(var.percent)),
col="gray", main="Pareto Chart of Amount of Explained Variance (Information Importance) by PCs")
abline(h=1/ncol(scores)*100, col="red")

# show sorted English language stopwords
sort(stopwords("english"), decreasing=FALSE)
#getTransformations()

# produce Word cloud of most frequent terms sorted by decreasing frequency
wordFreq <- sort(rowSums(m), decreasing=TRUE)
set.seed(350)
sugwordcloud <- wordcloud(words=names(wordFreq),freq=wordFreq, min.freq=100, colors=brewer.pal(6,"Dark2"),
random.order=FALSE)

# cluster corpus2b.tdm
tdm.corpus2b <- as.matrix(corpus2b.tdm)
dist2b <- dist(scale(tdm.corpus2b))
fit2b <- hclust(dist2b, method ="ward")
plot(fit2b, main = "Hierarchical Cluster Plot (Terms Horizontal)")


#use ape (analyses of phylogenetics and evolution) package to make more sophisticated dendrograms
library(ape)
#cut tree into 5 clusters
colors = c("red", "blue", "green", "black", "magenta")
clus5 = cutree(fit2b, 5)
plot(as.phylo(fit2b), main = "Hiearchical Cluster Plot (Terms Vertical)", edge.width=1,  tip.color= colors[clus5], edge.lty=1, cex = 0.9, label.offset = 0.5)
#fan
plot(as.phylo(fit2b), main = "Hierarchical Cluster Plot (Terms Fan)", cex = 0.9, type = "fan", tip.color= colors[clus5], label.offset = 1)

#library(ape)
#plot(as.phylo(fit2b), main = "Hiearchical Cluster Plot (Terms Vertical)", cex = 0.9, label.offset = 1)
#fan
#plot(as.phylo(fit2b), main = "Hierarchical Cluster Plot (Terms Fan)", cex = 0.9, type = "fan", label.offset = 1)

#get the two PC columns
term.coord <- loadings
head(term.coord[,1:2])
```

R code that Completed the Tutorial Part of the Program (cont.)

```
plot(term.coord[,1], term.coord[,2], pch = 18, main = "Biplot of PC2 by PC1",
   xlab="PC1", ylab="PC2")
abline(h=0,v=0, lty = 2)
text(jitter(term.coord[,1]), jitter(term.coord[,2]), labels=rownames(term.coord), cex=0.7, pos = 2)

biplot(scores[,1:2], loadings[,1:2], cex=0,
col=c("red", "blue", "green"), main = "Biplot of PC2 by PC1 with Term-Document Vectors")
abline(h=0,v=0, lty = 2)
text(jitter(loadings[,1]), jitter(loadings[,2]), labels=rownames(loadings), cex=0.7, pos = 3)

library(ggplot2)
df <-data.frame(term.coord[,1:2])
sp <- ggplot(df,aes(x=PC1, y=PC2))
sp <- sp + ggtitle("Biplot of PC2 by PC1")
sp <- sp + geom_point()
sp <- sp + geom_jitter()
sp <- sp + geom_text(hjust=0, vjust = 0, nudge_x = 0.05, nudge_y =0.1, aes(label=rownames(df)), size=2, check_overllap=TRUE, angle=45)
sp

# use ggrepel to repel overlapping terms
library(ggrepel)
set.seed(100)
df <- data.frame(term.coord[,c(1,2)])
ggplot(df) +
geom_point(aes(PC1,PC2), color = 'red') +
geom_hline(yintercept=0, colour="black") +
geom_vline(xintercept=0, colour="black") +
geom_text_repel(aes(PC1,PC2,label =rownames(df))) +
annotate(
"text", label = "BiPlot of PC2 by PC1",
x = -0.24, y = 1, size = 6, colour = "black"
) +
annotate('text',
            label='PC1',
            x=-0.4,y= 0.04, color="black") +
annotate('text',
            label='PC2',
            x=-0.02,y= 0.8, color="black") +
annotate('text',
            label='(0,0)',
            x=-0,y= 0, color="black") +
theme_classic(base_size = 0.5)

# export princippal components, eigens to excel
library("xlsx")
write.xlsx(loadings, "G:/Expand 6 Sigma/principal_components_tutorial.xlsx")
write.xlsx(m, "G:/Expand 6 Sigma/TDM_tutorial.xlsx")
write.xlsx(term.coord, "G:/Expand 6 Sigma/termPCs_Tutorial.xlsx")
write.xlsx(prcomponent$sdev, "G:/Expand 6 Sigma/Stdevs_Tutorial.xlsx")
write.xlsx(prcomponent$x, "G:/Expand 6 Sigma/Vtranspose_Tutorial.xlsx")
```