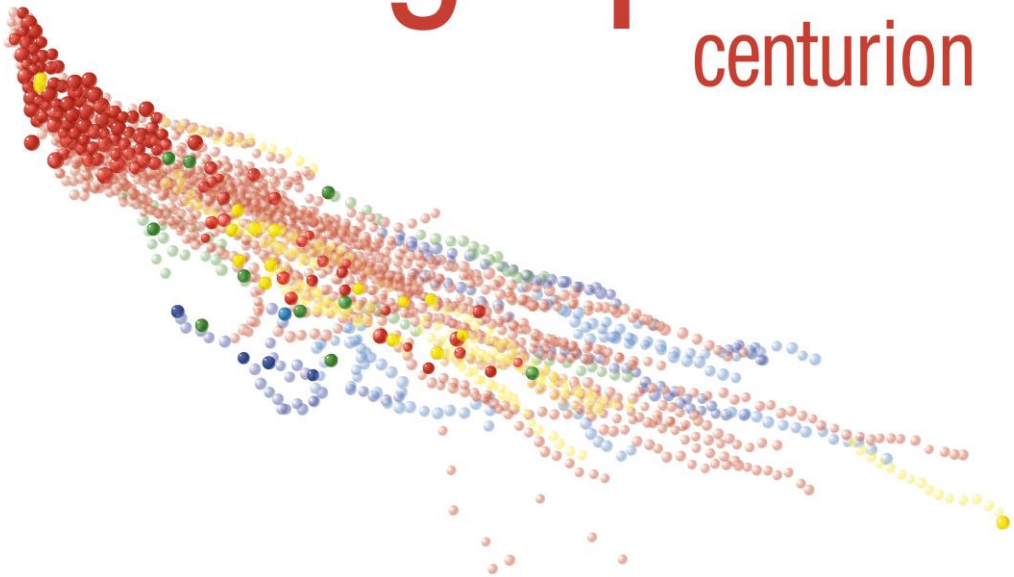
statgraphics®
centurion



Using Statgraphics and R for Text Mining

Presented by
Dr. Neil W. Polhemus

Statgraphics/R Interface

- The new interface between Statgraphics and R makes it possible to construct scripts and save them in StatFolios.

- Users can build generic StatFolios that access selected R procedures.

- Analysts can then take these StatFolios and edit them to meet their particular needs.

Example: Text Mining

- Refers to the process of extracting useful information from text.
- Usually we are looking for patterns or trends.
- Of particular interest is the frequency of occurrence of different terms or phrases.

Two Examples

- Example 1: Analyzing a column of text loaded into a Statgraphics datasheet.

- Example 2: Analyzing a directory containing multiple text documents.

Example 1: Analyzing a column of text

JeopardyAnswers.sgd

	Answer	Col_2	Col_3
1	Prism		
2	Bark		
3	Lever		
4	Primates		
5	Polymers		
6	graphical		
7	non [or no] rapid eye movement		
8	osteopathy		
9	the thyroid		
10	ribosomal		
11	twice a day		
12	international units		
13	solid rocket booster		
14	chlorofluorocarbons		
15	an anti-inflammatory drug		
16	boiling point		
17	infrared		
18	red blood cells		
19	a carnivorous plant		
20	low Earth orbit		

JeopardyAnswers B C

Preliminaries

- Download and install R by going to:

  https://cran.r-project.org/

- Start R and install the basic libraries:

  > install.packages(“installr”)

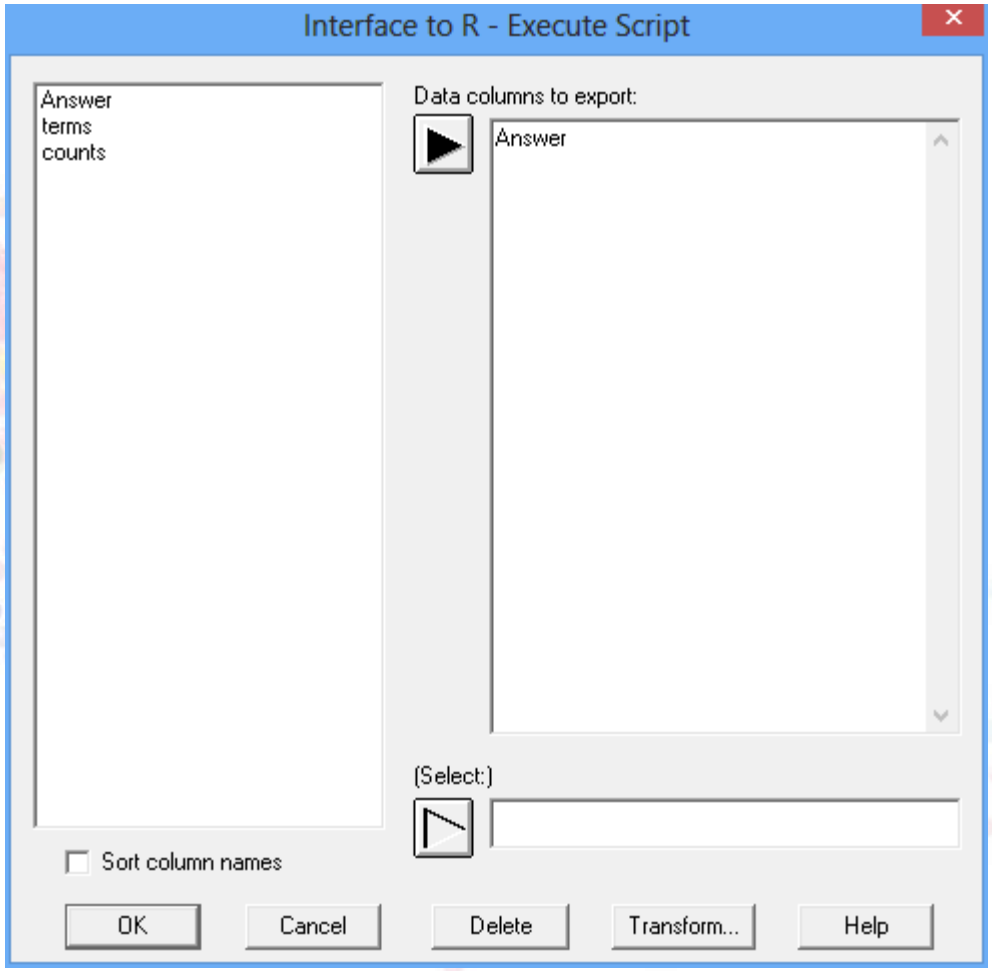
  > require(installr)

  > install.pandoc()

- Install the text mining library:

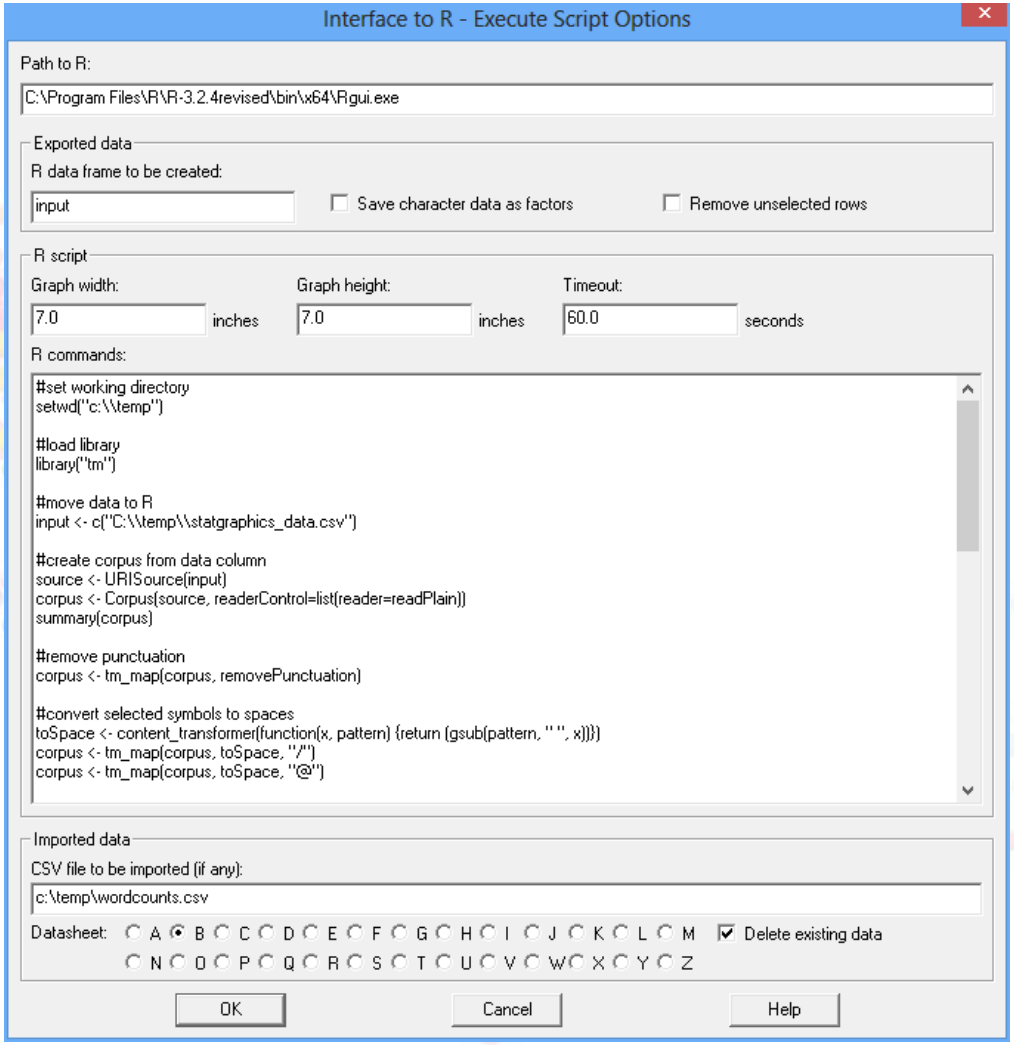
  >install.packages(“tm”)

StatFolio: MineTextColumn.sgp



statgraphics® centurion
statgraphics.com

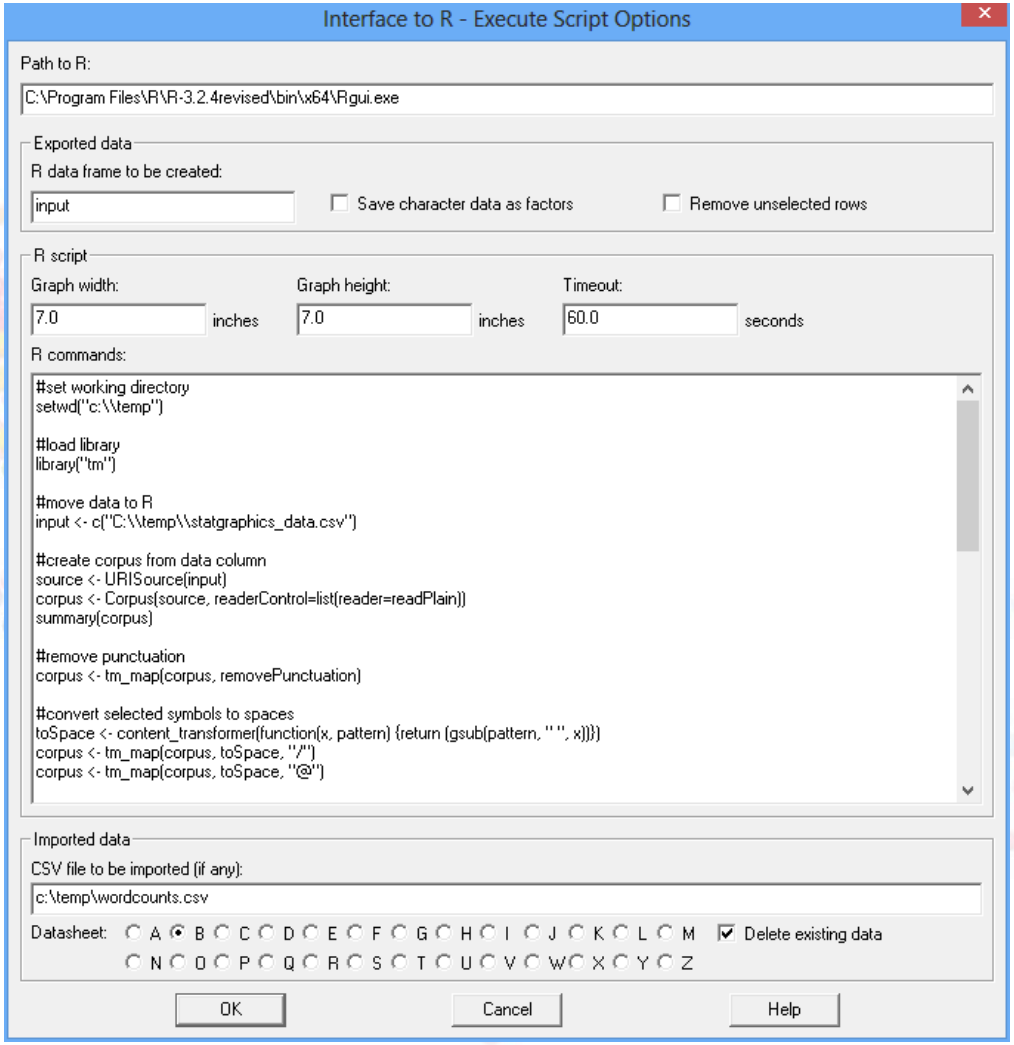
Analysis Options



statgraphics® centurion

statgraphics.com

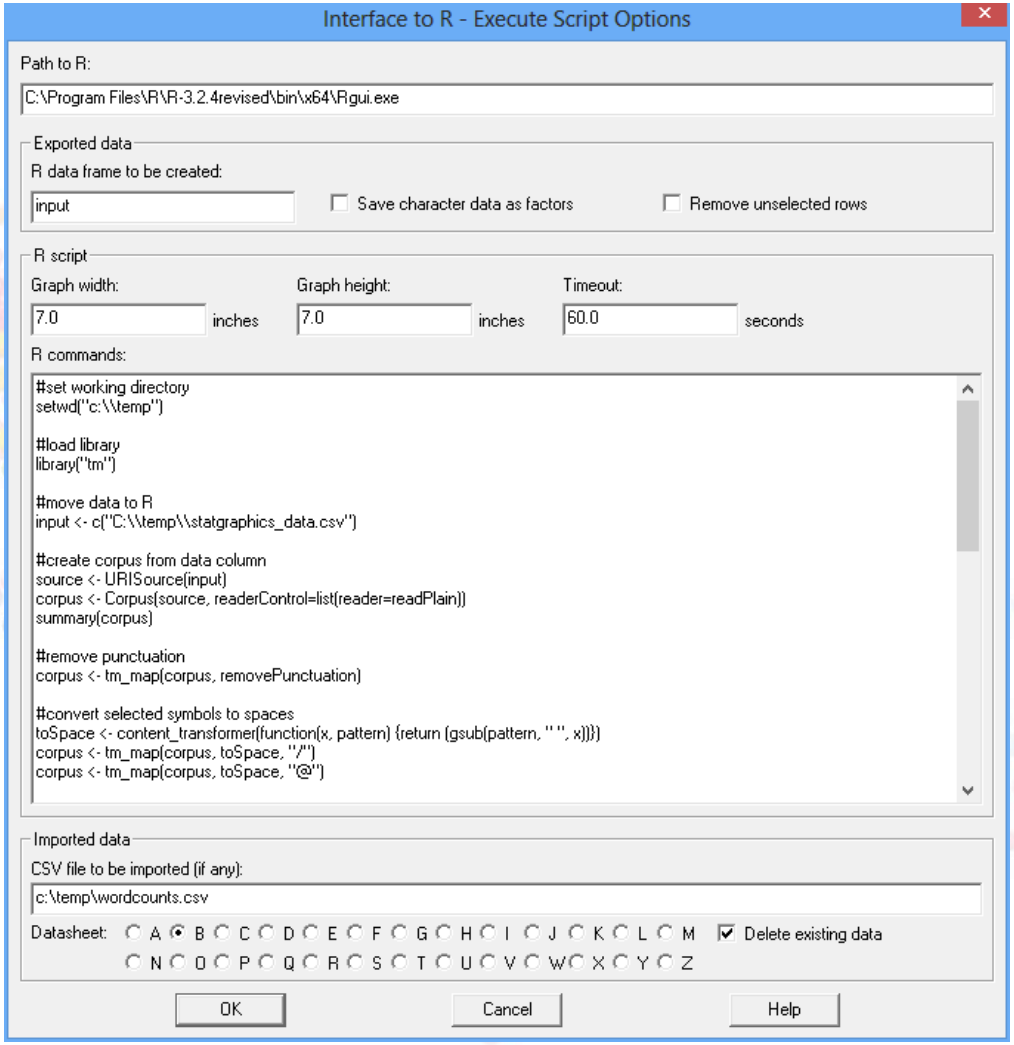
Specify Path to R



statgraphics® centurion

statgraphics.com

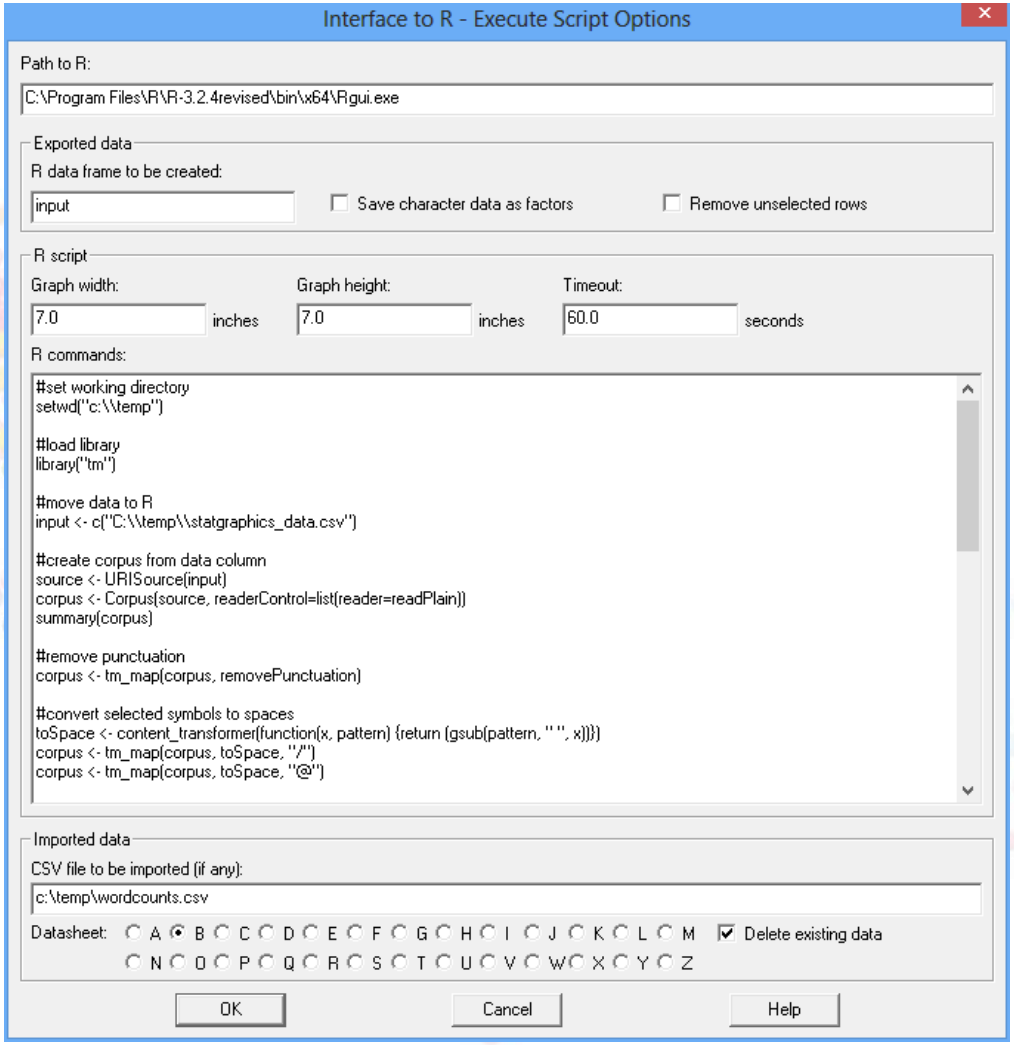
Specify Name of R Data Frame



statgraphics® centurion

statgraphics.com

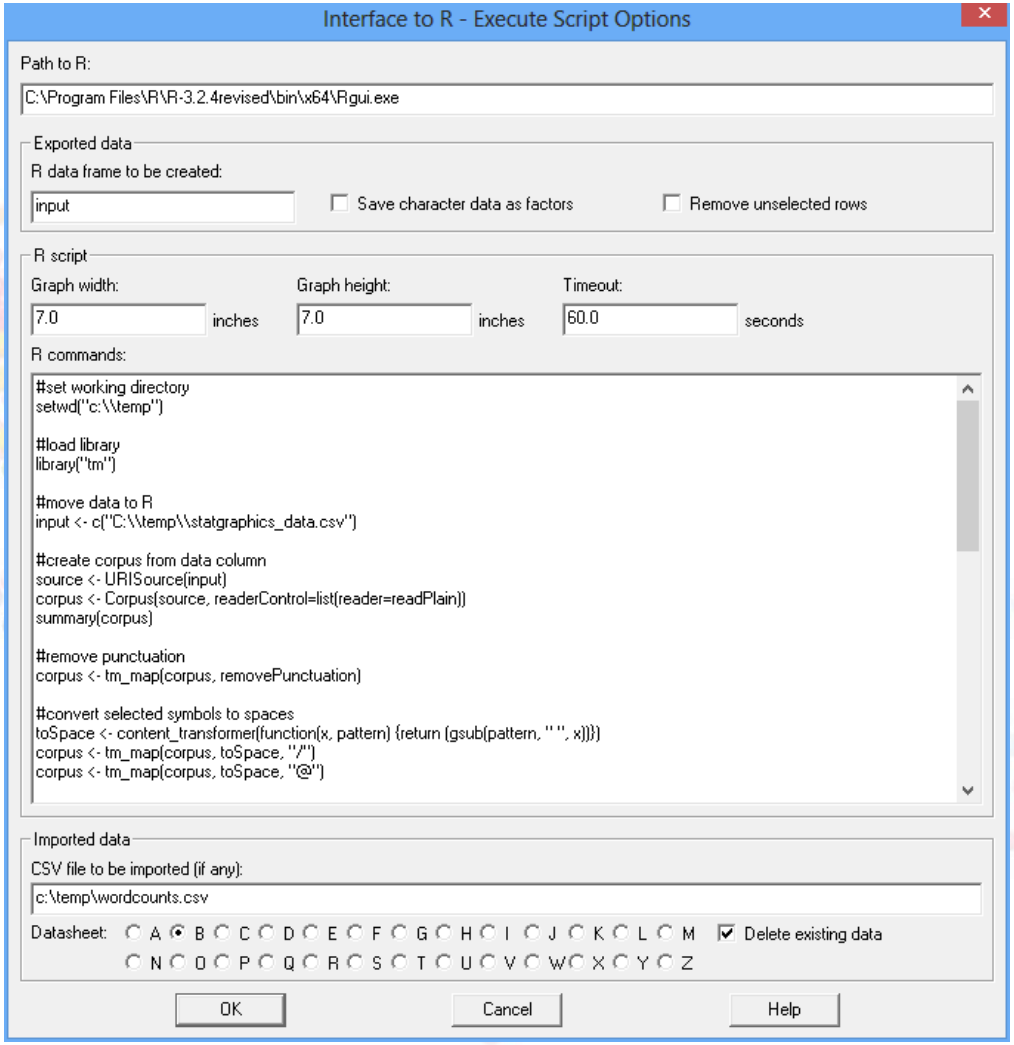
Set Size of Graphs and Timeout



statgraphics® centurion

statgraphics.com

Specify R Commands to Execute



statgraphics® centurion

statgraphics.com

Set Working Directory

```
#set working directory
setwd("c:\\temp")

#load library
library("tm")

#move data to R
input <- c("C:\\temp\\statgraphics_data.csv")

#create corpus from data column
source <- URISource(input)
corpus <- Corpus(source, readerControl=list(reader=readPlain))
summary(corpus)

#remove punctuation
corpus <- tm_map(corpus, removePunctuation)

#convert selected symbols to spaces
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "/")
corpus <- tm_map(corpus, toSpace, "@")
```

Load Text Mining Library

```
#set working directory
setwd("c:\\temp")

#load library
library("tm")

#move data to R
input <- c("C:\\temp\\statgraphics_data.csv")

#create corpus from data column
source <- URISource(input)
corpus <- Corpus(source, readerControl=list(reader=readPlain))
summary(corpus)

#remove punctuation
corpus <- tm_map(corpus, removePunctuation)

#convert selected symbols to spaces
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "/")
corpus <- tm_map(corpus, toSpace, "@")
```

Import the Data from Statgraphics

```
#set working directory
setwd("c:\\temp")

#load library
library("tm")

#move data to R
input <- c("C:\\temp\\statgraphics_data.csv")

#create corpus from data column
source <- URISource(input)
corpus <- Corpus(source, readerControl=list(reader=readPlain))
summary(corpus)

#remove punctuation
corpus <- tm_map(corpus, removePunctuation)

#convert selected symbols to spaces
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "/")
corpus <- tm_map(corpus, toSpace, "@")
```

Create a Corpus

```
#set working directory
setwd("c:\\temp")

#load library
library("tm")

#move data to R
input <- c("C:\\temp\\statgraphics_data.csv")

#create corpus from data column
source <- URISource(input)
corpus <- Corpus(source, readerControl=list(reader=readPlain))
summary(corpus)

#remove punctuation
corpus <- tm_map(corpus, removePunctuation)

#convert selected symbols to spaces
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "/")
corpus <- tm_map(corpus, toSpace, "@")
```

Remove Punctuation

```
#set working directory
setwd("c:\\temp")

#load library
library("tm")

#move data to R
input <- c("C:\\temp\\statgraphics_data.csv")

#create corpus from data column
source <- URISource(input)
corpus <- Corpus(source, readerControl=list(reader=readPlain))
summary(corpus)

#remove punctuation
corpus <- tm_map(corpus, removePunctuation)

#convert selected symbols to spaces
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "/")
corpus <- tm_map(corpus, toSpace, "@")
```

Convert Symbols to Spaces

```
#set working directory
setwd("c:\\temp")

#load library
library("tm")

#move data to R
input <- c("C:\\temp\\statgraphics_data.csv")

#create corpus from data column
source <- URISource(input)
corpus <- Corpus(source, readerControl=list(reader=readPlain))
summary(corpus)

#remove punctuation
corpus <- tm_map(corpus, removePunctuation)

#convert selected symbols to spaces
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "/")
corpus <- tm_map(corpus, toSpace, "@")
```

Remove Numbers

```
#remove numbers
corpus <- tm_map(corpus, removeNumbers)

#make all text lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Convert Text to Lowercase

```
#remove numbers
corpus <- tm_map(corpus, removeNumbers)

#make all text lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Remove Common Words

```
#remove numbers
corpus <- tm_map(corpus, removeNumbers)

#make all text lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Consolidate Words with Same Stem

```
#remove numbers
corpus <- tm_map(corpus, removeNumbers)

#make all text lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Remove Extra Whitespace

```
#remove numbers
corpus <- tm_map(corpus, removeNumbers)

#make all text lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Create Document-Term Matrix

```
#remove numbers
corpus <- tm_map(corpus, removeNumbers)

#make all text lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Create Frequency Matrix

```
#remove numbers
corpus <- tm_map(corpus, removeNumbers)

#make all text lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Create Word Cloud

```
#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))

#draw wordcloud
library("wordcloud")
library("ggplot2")
library("RColorBrewer")
wordcloud(names(freqr),freqr,scale=c(4,0.5),min.freq=4,max.words=1000,rot.per=0.1,random.order=TRUE,random.color=TRUE,color=

#send words and counts back to Statgraphics
results<-data.frame(term=names(freqr), count=freqr, row.names=NULL)
write.table(results,"C:\\Temp\\wordcounts.csv",dec=".",sep=",",row.names=FALSE)
```

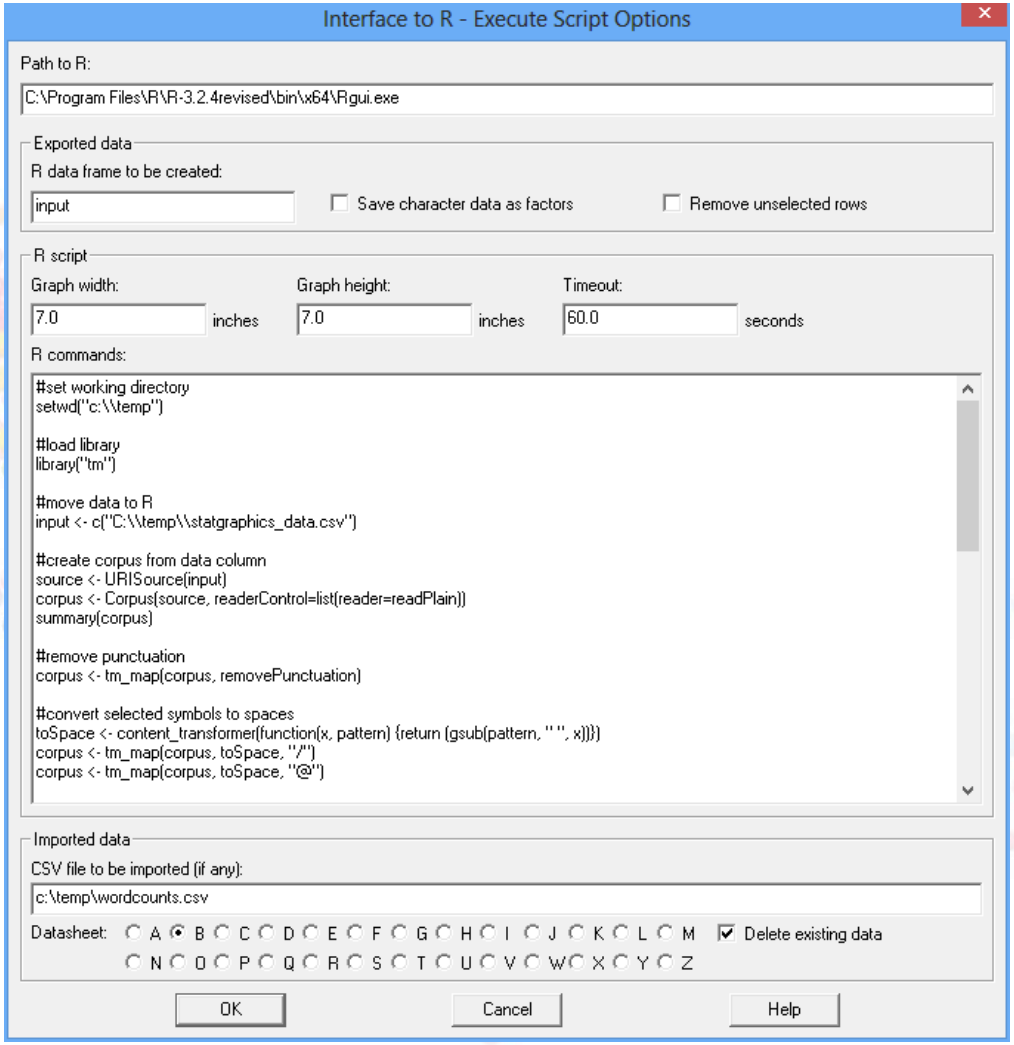
Wordcloud Options

- scale=c(4,0.5): range of the word sizes
- min.freq=4: minimum frequency to include word
- max.words=1000: maximum number of words
- rot.per=0.1: fraction of words shown vertically
- random.order=TRUE: randomize word order
- random.color=TRUE: randomize colors
- colors=brewer.pal(8,”Dark2”): color palette

Return Results to Statgraphics

```
#consolidate words with common stems
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove spaces
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))

#draw wordcloud
library("wordcloud")
library("ggplot2")
library("RColorBrewer")
wordcloud(names(freqr),freqr,scale=c(4,0.5),min.freq=4,max.words=1000,rot.per=0.1,random.order=TRUE,random.color=TRUE,color=

#send words and counts back to Statgraphics
results<-data.frame(term=names(freqr), count=freqr, row.names=NULL)
write.table(results,"C:\\Temp\\wordcounts.csv",dec=".",sep=",",row.names=FALSE)
```

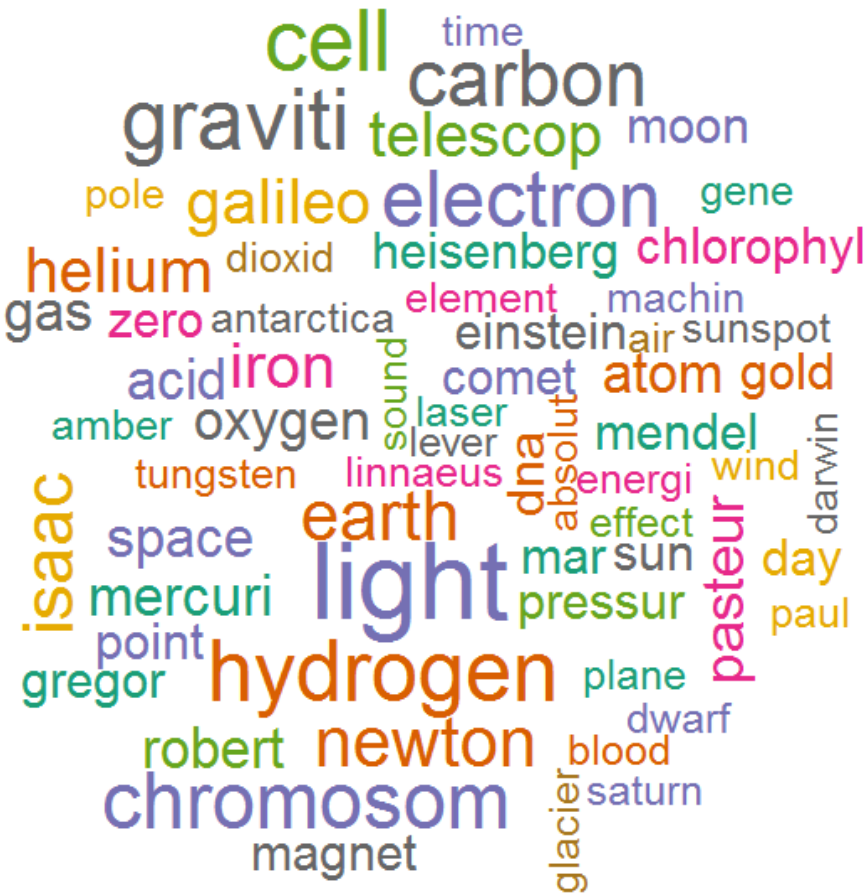
Specify File to Import



statgraphics® centurion

statgraphics.com

Word Cloud in Statgraphics



statgraphics®
statgraphics.com
centurion

Results

<in StatFolio>

	term	count	Col_3	Col_4	Col_5
1	aardvark	1			
2	absolut	4			
3	academi	1			
4	accept	2			
5	acid	6			
6	adenosin	1			
7	adren	1			
8	adrenalin	2			
9	adult	1			
10	africa	1			
11	african	1			
12	aftershock	1			
13	age	3			
14	agricultur	1			
15	aid	1			
16	air	4			
17	albert	2			
18	alexand	1			
19	alfr	1			
20	alga	1			
21	algebra	1			
22	allen	2			

JeopardyAnswers B C

Example 2: Mining documents

- Create a directory containing TXT documents that you wish to analyze.

- Example: 9 famous speeches

abrahamlincoln	8/11/2016 4:04 PM	Text Document	2 KB
frederickdouglas	8/11/2016 4:01 PM	Text Document	11 KB
johnfkennedy	8/11/2016 4:27 PM	Text Document	8 KB
lyndonbjohnson	8/11/2016 4:35 PM	Text Document	20 KB
martinlutherkingjr	8/11/2016 4:29 PM	Text Document	9 KB
patrickhenry	8/11/2016 2:47 PM	Text Document	6 KB
ronaldreagan	8/11/2016 5:00 PM	Text Document	16 KB
susanbanthony	8/11/2016 4:07 PM	Text Document	4 KB
winstonchurchill	8/11/2016 4:37 PM	Text Document	4 KB

StatFolio: MineTextDirectory.sgp

Interface to R - Execute Script Options

Path to R:

C:\Program Files\R\R-3.2.4revised\bin\x64\Rgui.exe

Exported data

R data frame to be created:

☐ Save character data as factors ☐ Remove unselected rows

R script

Graph width: 7.0 inches Graph height: 7.0 inches Timeout: 60.0 seconds

R commands:

```
#set working directory
setwd("c:\\temp")

#load text mining library
library("tm")

#specify source directory
source <- DirSource("C:\\Data\\webinar\\speeches")

#create corpus
corpus <- Corpus(source, readerControl=list(reader=readPlain))
summary(corpus)

#remove punctuation
corpus <- tm_map(corpus, removePunctuation)

# change selected characters to spaces
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "/")
corpus <- tm_map(corpus, toSpace, "@")

#remove numbers
```

Imported data

CSV file to be imported (if any):

c:\temp\wordcounts.csv

Datasheet: ○ A ◉ B ○ C ○ D ○ E ○ F ○ G ○ H ○ I ○ J ○ K ○ L ○ M ☑ Delete existing data
○ N ○ O ○ P ○ Q ○ R ○ S ○ T ○ U ○ V ○ W ○ X ○ Y ○ Z

OK Cancel Help

statgraphics® centurion

statgraphics.com

Define Directory with Documents

```
#set working directory
setwd("c:\\temp")

#load text mining library
library("tm")

#specify source directory
source <- DirSource("C:\\Data\\webinar\\speeches")

#create corpus
corpus <- Corpus(source, readerControl=list(reader=readPlain))
summary(corpus)

#remove punctuation
corpus <- tm_map(corpus, removePunctuation)

# change selected characters to spaces
toSpace <- content_transformer(function(x, pattern) {return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "/")
corpus <- tm_map(corpus, toSpace, "@")

#remove numbers
```

Remove Selected Words

```
#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#remove selected words
corpus <- tm_map(corpus, removeWords, c("can","may","will"))

#stem words
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove extra whitespace
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#remove words that are missing from 75% or more of the documents
dtm <- removeSparseTerms(dtm, 0.75)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Remove Sparse Terms

```
#remove common English words
corpus <- tm_map(corpus, removeWords, stopwords("en"))

#remove selected words
corpus <- tm_map(corpus, removeWords, c("can","may","will"))

#stem words
corpus <- tm_map(corpus, stemDocument, language = "en")

#remove extra whitespace
corpus <- tm_map(corpus, stripWhitespace)

#create document-term matrix
dtm <- DocumentTermMatrix(corpus)

#remove words that are missing from 75% or more of the documents
dtm <- removeSparseTerms(dtm, 0.75)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))
```

Fix Random Seed

```
dtm <- removeSparseTerms(dtm, 0.75)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))

#draw wordcloud libraries
library("wordcloud")
library("ggplot2")
library("RColorBrewer")

#set random seed so always get same result
set.seed(0)

#draw wordcloud
wordcloud(names(freqr),freqr,scale=c(4,0.5),min.freq=3,max.words=1000,rot.per=0.1,random.order=TRUE,random.color=TRUE,color=

#display some correlations
findAssocs(dtm,term=c("freedom","nation","vote"),0.8)

#send words and counts back to Statgraphics
results<-data.frame(term=names(freqr), count=freqr, row.names=NULL)
write.table(results,"C:\\Temp\\wordcounts.csv",dec=".",sep=",",row.names=FALSE)
```

Find Associations

```
dtm <- removeSparseTerms(dtm, 0.75)

#create frequency matrix
freqr <- colSums(as.matrix(dtm))

#draw wordcloud libraries
library("wordcloud")
library("ggplot2")
library("RColorBrewer")

#set random seed so always get same result
set.seed(0)

#draw wordcloud
wordcloud(names(freqr),freqr,scale=c(4,0.5),min.freq=3,max.words=1000,rot.per=0.1,random.order=TRUE,random.color=TRUE,color=

#display some correlations
findAssocs(dtm,term=c("freedom","nation","vote"),0.8)

#send words and counts back to Statgraphics
results<-data.frame(term=names(freqr), count=freqr, row.names=NULL)
write.table(results,"C:\\Temp\\wordcounts.csv",dec=".",sep=",",row.names=FALSE)
```

Word Cloud

statgraphics®
statgraphics.com centurion

Word Associations

```
#display some correlations
findAssocs(dtm,term=c("freedom","nation","vote"),0.8)

## $freedom
##    one moment   come  still brutal   sign togeth   lead  refus  given
##   0.97   0.93   0.92   0.91   0.90   0.89   0.89   0.88   0.85   0.84
##  today   note   year
##   0.84   0.82   0.81
##
## $nation
##   whose    need    soul  victim present america   wrong    rich command
##    0.90    0.88    0.88    0.88    0.87    0.86    0.86    0.85    0.84
##     man
##    0.81
##
## $vote
##     right   tonight      pass      race constitut democraci     elect
##      0.98      0.98      0.97      0.97      0.96      0.96      0.96
##      issu  opportun  american      came     civil  privileg     share
##      0.96      0.95      0.94      0.93      0.93      0.93      0.91
##   countri     equal     heart       use     everi      caus      hatr
##      0.90      0.90      0.90      0.90      0.89      0.88      0.88
##     peopl      root       men     among    fought      give    intend
##      0.88      0.88      0.87      0.86      0.86      0.86      0.86
##   poverti   histori      just      time      mani   violenc      help
##      0.86      0.85      0.85      0.85      0.84      0.84      0.82
##       law     might      live
##      0.82      0.82      0.81
```

References

- StatFolios and data files are at:
  www.statgraphics.com/webinars

- Primary manual for tm library is at:
  https://cran.r-project.org/web/packages/tm/tm.pdf

- Good tutorial on tm is at:
  https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf