# Data Analytics Lesson 10 – Text Mining

Dr. Jeffrey Strickland

9/12/2018

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

## Introduction

This is our first lesson on text analytics, so we will do some necessary but basic preprocessing to prepare for our analysis. This includes converting the text to lower case, removing numbers and stop-words, combining words that need to stay together (like "data science"), and putting our text into a dataframe.

The text we will use is a collect of texts, specifically a few blogs I have written. The complete set of blogs comprise what we call a "corpus," which is Latin for "body" or "body of texts" in this instance. You may have heard "corpus" used in city names like Corpus Christi, which literally means "the body of Christ."

## Load the R packages

Load the following R packages for text mining and then load your texts into R.

```
library(tm)

## Loading required package: NLP

library(wordcloud2)
library(yaml)
library(NLP)
library(tm)
library(SnowballC)
library(ggplot2)

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

On your PC, create a folder "Data_Analytics" on your C: drive or use a folder you already have to download the "text.zip" from https://github.com/stricje1/VIT_University/tree/master/Predictive_Modeling/data. Then use the following code chunk to load your data into R Studio (the path you should use has been commented out):

## Load the Data

```
cname <-
file.path("C:/Users/jeff/Documents/VIT_Course_Material/Data_Analytics_2018/da
ta", "text")
#cname <- file.path("C:/Users/username/Documents/Data_Analytics/data",
"text")
cname

## [1]
"C:/Users/jeff/Documents/VIT_Course_Material/Data_Analytics_2018/data/text"

dir(cname)

##  [1] "A one-eyed man in the kingdom of the blind.txt"
##  [2] "All Things Data.txt"
##  [3] "Analytics and Statistics.txt"
##  [4] "Analytics is it more than a buzzword.txt"
##  [5] "Bayesian networks.txt"
##  [6] "Big Data Analytics and Human Resources.txt"
##  [7] "Big Data The Good the Bad and the Ugly.txt"
##  [8] "Call Center Analytics.txt"
##  [9] "Classification Trees using R.txt"
## [10] "Clouds clouds and more clouds.txt"
## [11] "Cluster Models.txt"
## [12] "Cyber-Threat Risk Assessment using R.txt"
## [13] "Data Scientist are Dead Long Live Data Science.txt"
## [14] "Do you like my Ensemble.txt"
## [15] "Free SAS.txt"
## [16] "Getting the Question Right.txt"
## [17] "What are Association Rules in Analytics.txt"
## [18] "Where_did_all_the_Teaching_Go.txt"
## [19] "Where_did_all_the_Thinking_Go.txt"
## [20] "Why_Stand_Many_Have_Fallen.txt"

docs <- Corpus(DirSource(cname))
```

Now examine the data you loaded using

```
summary(docs)

##                                                     Length
## A one-eyed man in the kingdom of the blind.txt     2
## All Things Data.txt                                 2
## Analytics and Statistics.txt                        2
## Analytics is it more than a buzzword.txt            2
## Bayesian networks.txt                               2
## Big Data Analytics and Human Resources.txt          2
## Big Data The Good the Bad and the Ugly.txt         2
## Call Center Analytics.txt                          2
## Classification Trees using R.txt                   2
## Clouds clouds and more clouds.txt                  2
## Cluster Models.txt                                 2
## Cyber-Threat Risk Assessment using R.txt           2
## Data Scientist are Dead Long Live Data Science.txt 2
## Do you like my Ensemble.txt                        2
## Free SAS.txt                                       2
## Getting the Question Right.txt                     2
## What are Association Rules in Analytics.txt        2
## Where_did_all_the_Teaching_Go.txt                  2
## Where_did_all_the_Thinking_Go.txt                  2
## Why_Stand_Many_Have_Fallen.txt                     2
##                                                    Class             Mode
## A one-eyed man in the kingdom of the blind.txt     PlainTextDocument list
## All Things Data.txt                                PlainTextDocument list
## Analytics and Statistics.txt                       PlainTextDocument list
## Analytics is it more than a buzzword.txt           PlainTextDocument list
## Bayesian networks.txt                              PlainTextDocument list
## Big Data Analytics and Human Resources.txt         PlainTextDocument list
## Big Data The Good the Bad and the Ugly.txt         PlainTextDocument list
## Call Center Analytics.txt                          PlainTextDocument list
## Classification Trees using R.txt                   PlainTextDocument list
## Clouds clouds and more clouds.txt                  PlainTextDocument list
## Cluster Models.txt                                 PlainTextDocument list
## Cyber-Threat Risk Assessment using R.txt           PlainTextDocument list
## Data Scientist are Dead Long Live Data Science.txt PlainTextDocument list
## Do you like my Ensemble.txt                        PlainTextDocument list
## Free SAS.txt                                       PlainTextDocument list
## Getting the Question Right.txt                     PlainTextDocument list
## What are Association Rules in Analytics.txt        PlainTextDocument list
## Where_did_all_the_Teaching_Go.txt                  PlainTextDocument list
## Where_did_all_the_Thinking_Go.txt                  PlainTextDocument list
## Why_Stand_Many_Have_Fallen.txt                     PlainTextDocument list
```

```
inspect(docs)
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 20
##
##
A one-eyed man in the kingdom of the blind.txt
##
A one-eyed man in the kingdom of the blind:\nPredicting the
Unpredictable\nâ\200⍰Almost nobodyâ\200\231s competent, Paul. Itâ\200\231s
enough to make you cry to see how bad most people are at their jobs. If you
can do a half-assed job of anything, youâ\200\231re a one-eyed man in the
kingdom of the blind.â\200\235 â\200⍰Kurt Vonnegut, Player
```

```
Piano\nAbstract\nThis article is about Predictive Modeling. It explores the
appropriateness of modeling in general and predictive modeling in particular,
as well as examining some pitfalls. Modeling is the process of formulating
and abstracting a representation of a real problem, based on simplifying
assumptions. Thus, no model is an exact representation of reality. Said a
different way, a model cannot fully represent a complex problem, but can
provide some insight into the problem and assist decision makers with
applying solutions.
```

## Corpus Preprocessing

Next, convert the text to lowercase and inspect your work:

```
docs <- tm_map(docs, tolower)
inspect(docs[1])

## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
##
A one-eyed man in the kingdom of the blind.txt
## a one-eyed man in the kingdom of the blind:\npredicting the
unpredictable\nâ\200⍰almost nobodyâ\200\231s competent, paul. itâ\200\231s
enough to make you cry to see how bad most people are at their jobs. if you
can do a half-assed job of anything, youâ\200\231re a one-eyed man in the
kingdom of the blind.â\200\235 â\200⍰kurt vonnegut, player
piano\nabstract\nthis article is about predictive modeling. it explores the
appropriateness of modeling in general and predictive modeling in particular,
as well as examining some pitfalls. modeling is the process of formulating
and abstracting a representation of a real problem, based on simplifying
assumptions. thus, no model is an exact representation of reality. said a
different way, a model cannot fully represent a complex problem, but can
provide some insight into the problem and assist decision makers with
applying solutions.
```

Next, remove unnecessary words from the text:

```
docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, removeWords, c("can", "should", "would", "figure",
"using", "will", "use", "now", "see", "may", "given", "since", "want",
"next", "like", "new", "one", "might", "without"))
```

Now, combine words that should stay together

```
for (j in seq(docs))
{
docs[[j]] <- gsub("data analytics", "data_analytics", docs[[j]])
docs[[j]] <- gsub("predictive models", "predictive_models", docs[[j]])
docs[[j]] <- gsub("predictive analytics", "predictive_analytics", docs[[j]])
docs[[j]] <- gsub("data science", "data_science", docs[[j]])
docs[[j]] <- gsub("operations research", "operations_research", docs[[j]])
docs[[j]] <- gsub("chi-square", "chi_square", docs[[j]])
}
```

## Create Document Matrices

In these setps we will prepare the documents for analysis. First we will put the text into a term-doucment matrix and view it:

```
tdm <- TermDocumentMatrix(docs)
tdm

## <<TermDocumentMatrix (terms: 3971, documents: 20)>>
## Non-/sparse entries: 7178/72242
## Sparsity           : 91%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

Second, create document-term matrix and view it:

```
dtm <- DocumentTermMatrix(docs)
dtm

## <<DocumentTermMatrix (documents: 20, terms: 3971)>>
## Non-/sparse entries: 7178/72242
## Sparsity           : 91%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

Next, organizes the terms by their frequency:

```
freq <- colSums(as.matrix(dtm))
length(freq)

## [1] 3971

ord <- order(freq)
```

Now, put it into a matrix and save it to your working directory:

```
m <- as.matrix(dtm)
dim(m)

## [1]   20 3971

write.csv(m, file="dtm.csv")
```

Remove sparse terms. This makes a matrix that is a maximum of 10% empty space.

```
dtms <- removeSparseTerms(dtm, 0.1)
inspect(dtms)
```

```
## <<DocumentTermMatrix (documents: 20, terms: 0)>>
## Non-/sparse entries: 0/0
## Sparsity           : 100%
## Maximal term length: 0
## Weighting          : term frequency (tf)
## Sample             :
##                                                      Terms
## Docs
##   A one-eyed man in the kingdom of the blind.txt
##   All Things Data.txt
##   Analytics and Statistics.txt
##   Analytics is it more than a buzzword.txt
##   Bayesian networks.txt
##   Big Data Analytics and Human Resources.txt
##   Big Data The Good the Bad and the Ugly.txt
##   Call Center Analytics.txt
##   Classification Trees using R.txt
##   Clouds clouds and more clouds.txt
##   Cluster Models.txt
##   Cyber-Threat Risk Assessment using R.txt
##   Data Scientist are Dead Long Live Data Science.txt
##   Do you like my Ensemble.txt
##   Free SAS.txt
##   Getting the Question Right.txt
##   What are Association Rules in Analytics.txt
##   Where_did_all_the_Teaching_Go.txt
##   Where_did_all_the_Thinking_Go.txt
##   Why_Stand_Many_Have_Fallen.txt
```

Next, we check some of the frequency counts. There are a lot of terms, so for now, we just check out some of the most and least frequently occurring words, as well as check out the frequency of frequencies.

```
freq[head(ord)]

##     abstract  abstracting abstractions      abusive      acquire
##            1            1            1            1            1
##       affair
##            1

freq[tail(ord)]

##     dendrogram classification          model           true      analytics
##             64             66             67             68            104
##           data
##            198

head(table(freq), 50)

## freq
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 2047  688  369  201  151   85   62   63   44   36   22   25   17   17   18
##   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
##   10    8   12   11   11    2    3    5    4    4    3    5    2    4    3
##   32   33   34   35   36   37   38   42   43   44   49   51   54   55   56
##    2    4    1    2    2    2    6    1    1    2    1    1    1    1    1
##   57   59   62   64   66
##    2    2    1    1    1
```

```
tail(table(freq), 50)
```

```
## freq
##   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22
## 151  85  62  63  44  36  22  25  17  17  18  10   8  12  11  11   2   3
##  23  24  25  26  27  28  29  30  32  33  34  35  36  37  38  42  43  44
##   5   4   4   3   5   2   4   3   2   4   1   2   2   2   6   1   1   2
##  49  51  54  55  56  57  59  62  64  66  67  68 104 198
##   1   1   1   1   1   2   2   1   1   1   1   1   1   1
```

```
freq <- colSums(as.matrix(dtms))
freq <- sort(colSums(as.matrix(dtm)), decreasing=TRUE)
head(freq, 14)
```

```
##           data      analytics           true          model classification
##            198            104             68             67             66
##     dendrogram           tree       branches     clustering         models
##             64             62             59             59             57
##           used        members       analysis           leaf
##             57             56             55             54
```

```
findFreqTerms(dtm, lowfreq=150)
```

```
## [1] "data"
```

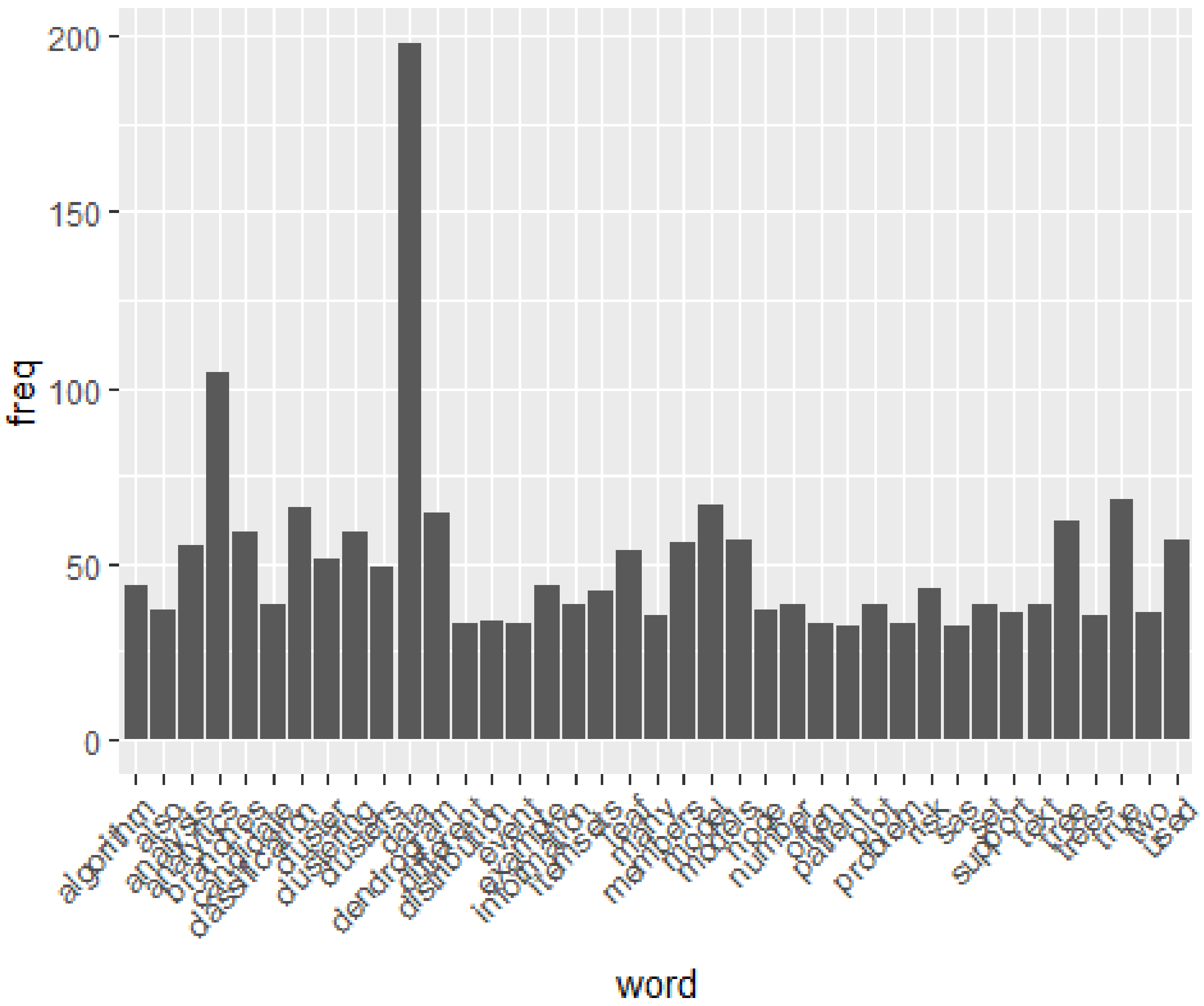
## Visualizing the Results

Now, we plot words that appear at least 50 times.

```
wf <- data.frame(word=names(freq), freq=freq)
head(wf)
```

```
##                          word freq
## data                     data  198
## analytics           analytics  104
## true                     true   68
## model                   model   67
## classification classification   66
## dendrogram         dendrogram   64
```

```
p <- ggplot(subset(wf, freq>30), aes(word, freq))
p <- p + geom_bar(stat="identity")
p <- p + theme(axis.text.x=element_text(angle=45, hjust=1))
p
```



## Find correlations

Now, we find correlations in the text.

```
findAssocs(dtm, c("question" , "analysis"), corlimit=0.98) # specifying a
correlation limit of 0.98

## $question
##     achieve   behaviors     brainer        cart    currency        dave
##        0.99        0.99        0.99        0.99        0.99        0.99
##      detect     deviceÃ    dialogue    director  downstream   expertise
##        0.99        0.99        0.99        0.99        0.99        0.99
##    failures     forever       horse  investment        john        keys
##        0.99        0.99        0.99        0.99        0.99        0.99
##     knowing     mention      months  phenomenon     realize      recipe
##        0.99        0.99        0.99        0.99        0.99        0.99
## reiterated         ret    robinson       rolls       roske      roskeÃ
##        0.99        0.99        0.99        0.99        0.99        0.99
##        seen      slides      staffs       stake       stems    temporal
##       0.99       0.99       0.99       0.99       0.99       0.99
##     timing      twice      vince     wallet
##       0.99       0.99       0.99       0.99
##
## $analysis
## numeric(0)
```

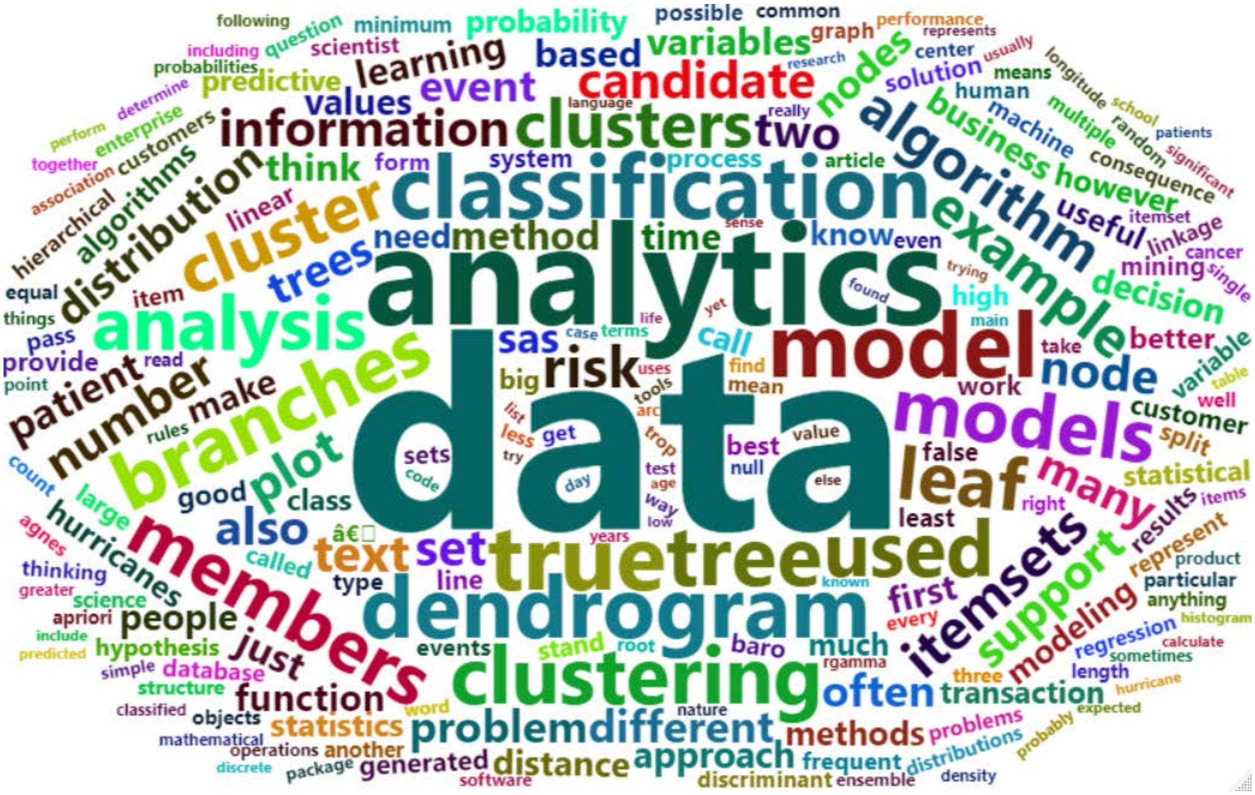
```
findAssocs(dtms, "contrast", corlimit=0.90) # specifying a correlation limit
of 0.95
```

```
## $contrast
## numeric(0)
```

## Using Wordclouds to Visualize Results

Plot words using a wordcloud that occur at least 50 times.

```
wordcloud2(subset(wf, freq>10))
```


following
question
minimum
probability
possible
common
performance
represents
including
scientist
learning
based
variables
graph
center
usually
probabilities
predictive
candidate
research
solution
means
longitude
determine
values
event
nodes
human
school
language
really
perform
enterprise
information
clusters
two
algorithm
business
machine
multiple
random
patients
customers
together
think
form
system
process
article
however
consequence
significant
association
algorithms
classification
useful
itemset
hierarchical
distribution
linear
cluster
need
method
time
know
even
example
linkage
sense
mining
cancer
single
trees
trying
equal
item
analytics
found
high
decision
things
analysis
yet
life
main
pass
sas
case
terms
call
model
take
better
provide
read
risk
find
variable
table
point
patient
uses
mean
node
work
number
make
big
tools
customer
well
branches
rules
list
arc
models
split
count
less
get
trop
best
value
false
statistical
plot
sets
try
null
leaf
many
large
good
code
day
test
age
else
items
class
data
way
right
results
agnes
hurricanes
low
least
represent
also
set
years
product
thinking
called
text
truetreeused
particular
members
type
line
known
first
greater
science
people
itemsets
anything
histogram
apriori
dendrogram
every
support
include
hypothesis
just
events
stand
root
baro
much
modeling
regression
calculate
predicted
simple
database
rgamma
sometimes
structure
clustering
three
length
function
often
transaction
hurricane
classified
word
nature
expected
objects
statistics
problemdifferent
methods
problems
probably
mathematical
operations
another
approach
frequent
distributions
discrete
package
generated
distance
density
software
discriminant
ensemble