IIT School of Applied Technology

ILLINOIS INSTITUTE OF TECHNOLOGY

information technology & management

527 Data Analytics

April 19,21 2016

Week 15 Presentation

Week 15 Topic: Agenda

- Additional Topics in Text Analytics
- Proxy statement analysis using Yahoo
  - QA Corpus
  - QA Term Document Matrix
  - QA Key Event Indicators (KEIs)

Week 13 Topic: Text Mining Applications - Unsupervised

- **Information retrieval**
  - finding documents with relevant content of interest
  - used for researching medical, scientific, legal, and news documents such as books and journal articles
- **Document categorization for organizing**
  - clustering documents into naturally occurring groups
  - extracting themes or concepts
- **Anomaly detection**
  - identifying unusual documents that might be associated with cases requiring special handling such as unhappy customers, fraud activity, and so on

Week 13 Topic: Text Mining Applications - Supervised

- Many typical predictive modeling or classification applications can be enhanced by incorporating textual data in addition to traditional input variables.
  - churning propensity models that include customer center notes, website forms, e-mails, and Twitter messages
  - hospital admission prediction models incorporating medical records notes as a new source of information
  - insurance fraud modeling using adjustor notes
  - sentiment categorization from customer comments
  - stylometry or forensic applications that identify the author of a particular writing sample

Week 15 Topic: Two General Goals in Text Mining

- Pattern Discovery (Unsupervised Learning)
  - Identify naturally occurring groups (classification*).
  - Derive convenient segments (clustering).
- Prediction (Supervised Learning)
  - Input variables are associated with values of a target variable.
  - Derive a model or set of rules that produces a predicted target value for a given set of inputs.

* Classification with a target variable is prediction. Classification refers to identifying "natural" groups, such as identifying different breeds within a species. Anthropology and other sciences try to find clear boundaries between groups to help define natural classification schemes. On the other hand, the same algorithms that can identify natural groups can be applied to any data set, even if no natural grouping exists.

Week 15 Topic: Text Mining Applications – Questions

- Stylometry (determining authorship)
  - Are documents created by more than one author? (Pattern discovery)
  - Who wrote a given document? (Prediction)
- Document categorization
  - Do the documents separate naturally into different categories? (Pattern discovery)
  - Can you assign a new document to a subject matter category? (Prediction)
- Information retrieval
  - Which documents are most relevant for a given information request? (Pattern discovery and prediction)
- Anomaly detection
  - Are there any unusual documents in the collection? (Pattern discovery and prediction)
  - What makes a document unusual? (Pattern discovery)
- Forensic linguistics
  - Can you identify the author of a manifesto? (Prediction)
  - This application area applies stylometry to crime investigation, and is related to anomaly detection for crime prevention.

Week 15 Topic: More on Stylometry

- *Stylometry* is defined as the use of linguistic style to characterize written language.

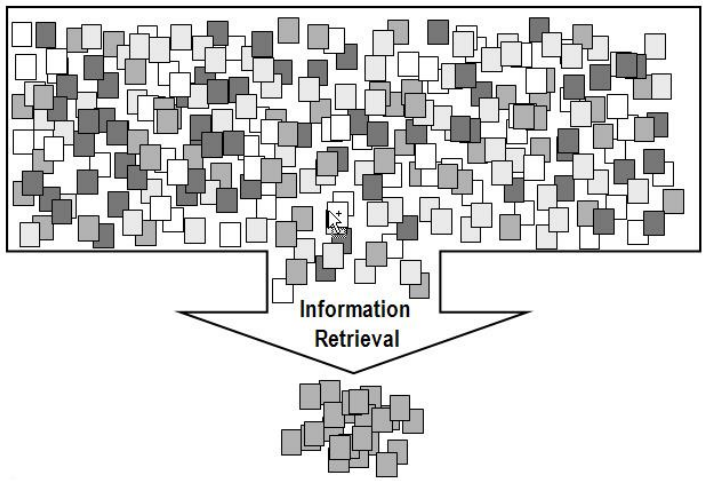
- Applications:
  - attributing authorship of anonymous or disputed literary works
  - detecting plagiarism
  - forensic linguistics

Forensic linguistics typically uses predictive modeling to score a document of unknown, but suspected, authorship. The score represents an estimate of the probability that the document was written by a suspect. The value of text mining applied to forensic linguistics is that suspects can be identified for investigation. The text mining results are rarely if ever used as evidence in prosecuting a suspect, although testimony might include a discussion of techniques in describing how the suspect was identified.

Week 15 Topic: More on Information Retrieval

"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."

– Manning, Raghavan, and Schütze (2008)



- One of the more publicized success stories in information retrieval concerns the discovery by Don Swanson (1988, 1991) that magnesium deficiency could be a source of migraine headaches. Swanson queried medical reports for articles about migraines and nutrition.
- For a given corpus of documents, information retrieval (IR) groups documents based on the similarity of contents. An IR query can be a Boolean query, a query based on latent semantic indexing, or a query based on some other method of quantifying document content. Documents that are most similar to the query are returned.

Week 15 Topic:
Text Filter Operation Examples

- "+*term*" - return all documents that have at least one occurrence of *term*.
- "-*term*" - return all documents that have zero occurrences of *term*.
- "*text string*" - return all documents that have at least one occurrence of the quoted text string.
- *"string1*string2"* - return all documents that have a term that begins with *string1*, ends with *string2*, and has text in between.
- ">#*term*" - return all documents that have *term* or any of the synonyms that were associated with *term*.

Week 15 Topic: Turning Text into Numbers

- Linear Algebraic approach e.g., Singular Value Decomposition (SVD), Vector Space Model (VSM), Latent Semantic Indexing (LSI), or Latent Semantic Analysis (LSA) to quantify terms/documents.
- Basic calculation per document:
  - Boolean counting (0-1) of terms
  - ***Frequency counting of terms***
  - Information theoretic counting of terms (logarithm of frequency counts)
- Adjusting for document size and corpus size term weights:
  - Entropy weights (Shannon information theory)
  - Inverse document frequency weights
  - Target-based weights

Week 15 Topic: Sparse, High Dimensional Vector Spaces

- After the frequency counts are obtained, you see that both terms and documents can be represented in vector spaces.
- However, in both cases, even after stemming and other filtering steps have been applied, you usually still face a very high-dimensional data set.
- In addition, the matrices of frequency counts are very sparse because many words appear only in just 1 or 2 documents. Typically, 90% or more of the cells in the matrices can be 0.
- Also, the frequency counts are highly skewed, as shown by Zipf's law. A small number of words occur many times.

Week 15 Topic: Addressing These Issues

The dimensionality and sparseness problems can be addressed by projecting the document and term vector spaces into a lower dimensional space by means of a key theorem from linear algebra referred to as the ***singular value decomposition (SVD).***

Before applying SVD, however, it has been found that weighting the raw document-term cell counts usually produces better text mining results. ***Weighting*** *also helps alleviate the problem of the skewness of the higher frequency terms by making them less influential.*

Week 15 Topic: Weighting of Terms

- The problem of skewed frequency counts can be addressed by applying weights to the frequencies.
- Weighting can be applied in two-tiers:
  - Local weights L_{ij} , also called frequency weights, are calculated for term i in document j.
  - Term weights G_i, also called global weights, are calculated for term i.
  - The final weight for each cell is the product $G_i L_{ij}$.

Frequency weights, which are often called local weights in the text mining and information retrieval literature, are the first step in transforming the raw cell counts.

Week 15 Topic: SVD Theorem

Reference: "Taming Text with the SVD"
ftp://ftp.sas.com/techsup/download/EMiner/TamingTextwiththeSVD.pdf
Theory Reference: http://www.math.iit.edu/~fass/477577_Chapter_2.pdf

The SVD theorem states that the term-document matrix (and, in fact, ***any*** rectangular matrix of real or complex values) can always be decomposed into the product of three matrices in the form $A = U\Sigma V^T$:

- Define A to be a term-document matrix with m terms and n documents. (Typically, $m > n$. That is, there are more terms than documents.)
- T signifies the transpose of a matrix.
- r is the rank of the matrix A.
- U is an m x r matrix satisfying the orthogonality condition $U^T U = I_{r \text{ x } r}$.
- $I_{r \text{ x } r}$ is an r x r identity matrix.
- Σ is an r x r diagonal matrix consisting of r positive "singular values"

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$$

- V is an r x n matrix satisfying the orthogonality condition $VV^T = I_{r \text{ x } r}$.
- The singular values σ_i can be thought of as providing a measure of importance used to decide how many dimensions to keep.

Week 15 Topic: SVD Document and Term Projections

The product or U and V with A produces the SVD projections of the original document vectors. This amounts to forming linear combinations of the original (possibly weighted) term frequencies for each document.

The transpose of the **U** matrix multiplied by the term-document frequency matrix produces a set of linear transformations of the original term frequencies per document. The term-document frequency matrix multiplied by the transpose of the V matrix produces a set of linear transformations of the original document frequencies per term.

$$U^T = \text{transpose of } U$$

$$U^T A = \text{SVD document vectors}$$

$$AV^T = \text{SVD term vectors}$$

Week 15 Topic: Linear Algebra and SVD in Text Mining

- The main data set in the analysis of free-form text consists of a term-document matrix.
- Assume at this point that all the natural language parsing and tokenization of terms, the application of start or stop lists, filtering, weighting, and so on, have been performed so that you can focus on the final version of the term-document matrix.
- Linear algebra includes the study of matrices and matrix properties.
- The rank is always less than or equal to the minimum of the number of documents and the number of terms.
- In actual practice, the rank of the term-document matrix will usually be in the thousands, so the SVD algorithm is used to dramatically reduce the dimensionality of the data.
- The SVD algorithm derives SVD dimensions in order of "importance" (based on the singular values σ_i).
- The number of SVD dimensions to keep is based on looking at these singular values and establishing a cut-off value k.

Prof. Gilbert Strang of MIT, a world expert on this topic, has referred to SVD as "The Fundamental Theorem of Linear Algebra."

Week 15 Topic:
Example – Corpus, Matrix, Weighted M

	Text
1	hamster dog cat walrus dog puppy dog kitten bear dog
2	dog mouse dog cat dog walrus dog seal dog otter
3	horse cat dog cat walrus cat bear cat cow
4	cow cat dog cat walrus cat seal cat otter pig
5	pig cat dog cat walrus cat seal cat tiger cat
6	walrus zebra walrus dog walrus cat walrus seal cow horse gopher
7	walrus kitten walrus seal hamster dog walrus cat walrus seal hamster
8	walrus tiger walrus dog walrus cat walrus seal cow horse
9	seal otter walrus dog seal cat seal walrus seal tiger seal
10	seal bear walrus dog seal cat seal walrus seal

Domestic Household:
- Cat, Dog, Hamster, Kitten, Puppy

Domestic Farm:
- Cow, Horse, Pig

Forest:
- Bear, Gopher, Mouse

Jungle:
- Tiger, Zebra

Marine:
- Otter, Seal, Walrus

Raw Frequency Counts

Term	DOC1	DOC2	DOC3	DOC4	DOC5	DOC6	DOC7	D
dog	4	5	1	1	1	1	1	
cat	1	1	4	4	5	1	1	
walrus	1	1	1	1	1	4	4	
seal	0	1	0	1	1	1	2	
cow	0	0	1	1	0	1	0	
bear	1	0	1	0	0	0	0	
otter	0	1	0	1	0	0	0	
horse	0	0	1	0	0	1	0	
tiger	0	0	0	0	1	0	0	
hamster	1	0	0	0	0	0	2	
kitten	1	0	0	0	0	0	1	
pig	0	0	0	1	1	0	0	

Term-Document Frequency Table from Text Miner: Term Weight=Entropy, Frequency Weight=Log

Term	DOC1	DOC2	DOC3	DOC4	DOC5	DOC6	DOC7	DOC8	DOC9	DOC10
dog	0.27118	0.30190	0.11679	0.11679	0.11679	0.11679	0.11679	0.11679	0.11679	0.11679
cat	0.11454	0.11454	0.26595	0.26595	0.29607	0.11454	0.11454	0.11454	0.11454	0.11454
walrus	0.07915	0.07915	0.07915	0.07915	0.07915	0.18379	0.18379	0.18379	0.12546	0.12546
seal	0.00000	0.20245	0.00000	0.20245	0.20245	0.20245	0.32088	0.20245	0.52333	0.47008
cow	0.00000	0.00000	0.39794	0.39794	0.00000	0.39794	0.00000	0.39794	0.00000	0.00000
bear	0.52288	0.00000	0.52288	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.52288
otter	0.00000	0.52288	0.00000	0.52288	0.00000	0.00000	0.00000	0.00000	0.52288	0.00000
horse	0.00000	0.00000	0.52288	0.00000	0.00000	0.52288	0.00000	0.52288	0.00000	0.00000
tiger	0.00000	0.00000	0.00000	0.00000	0.52288	0.00000	0.00000	0.52288	0.52288	0.00000
hamster	0.72357	0.00000	0.00000	0.00000	0.00000	0.00000	1.14682	0.00000	0.00000	0.00000
kitten	0.69897	0.00000	0.00000	0.00000	0.00000	0.00000	0.69897	0.00000	0.00000	0.00000
pig	0.00000	0.00000	0.00000	0.69897	0.69897	0.00000	0.00000	0.00000	0.00000	0.00000

Week 15 Topic:
SVD Objective - Dimension Reduction

- Algorithms process documents (parsing/filtering).
- A derived vector is associated with each document.
- The vector is typically too large and has too many zeros to work with directly, so transformation methods and dimensionality reduction techniques are applied to produce a more useful final vector representation for each document.
- Converting a document to a well-defined, structured vector permits application of any valid analytic technique to facilitate problem solving

You can think of the vector associated with each document as the score produced by each derived query. For example, if the dimensionality is set to 50, then 50 sets of query weights will be derived, and each document will produce 50 scores, 1 score for each derived query. The maximum dimensionality is the number of terms in the dictionary or vocabulary (start list) used for the analysis. The number of terms is usually in the thousands or hundreds of thousands.

Linear Algebraic approaches provide a methodology to reduce this maximum dimensionality down to a reasonable dimensionality that will still permit successful mining of the document collection.

Week 15 Topic: Example - Boolean Search

- If a term appears in a document, it receives a weight of 1, no matter how often it appears. Otherwise, the term receives a weight of 0. A query can be specified that assigns a weight to each term to represent how important it is in the query. The query could be Boolean as well, with a value of 1 for terms that are sought, and a value of 0 otherwise.
- To evaluate how well a document satisfies the query, the weight for each term in the document is multiplied by the corresponding weight for the term in the query. The products are added to give a total score for the document. This score represents how well a document satisfies the query.

Document	Term_1	Term_2	Term_3	Term_4	Term_5	Term_6
Doc_01	0	1	0	0	1	0
Doc_02	0	1	0	0	0	1
Doc_03	0	0	1	0	0	0
Doc_04	0	0	1	1	1	1
Doc_05	1	0	0	0	1	0
Doc_06	0	0	0	1	1	1
Doc_07	1	0	1	0	0	0
Doc_08	0	1	0	1	0	1
Doc_09	0	1	1	0	0	1
Doc_10	0	0	1	0	0	1
Doc_11	1	0	0	0	0	1
Doc_12	1	1	0	0	1	0

Document/Term Matrix

+

Document	Term_1	Term_2	Term_3	Term_4	Term_5	Term_6
Que_01	1	1	1	0	0	0
Que_02	0	0	0	1	1	1
Que_03	1	0	2	0	3	0

Query Matrix

Week 15 Topic:
Example - Boolean Search (cont.)

The largest value of the query occurs for the document (or documents) that most closely matches the query. This illustrates a Boolean search from information retrieval.

Document	Term_1	Term_2	Term_3	Term_4	Term_5	Term_6	Q1	Q2	Q3
Doc_01	0	1	0	0	1	0	1	1	3
Doc_02	0	1	0	0	0	1	1	1	0
Doc_03	0	0	1	0	0	0	1	0	2
Doc_04	0	0	1	1	1	1	1	3	5
Doc_05	1	0	0	0	1	0	1	1	4
Doc_06	0	0	0	1	1	1	0	3	3
Doc_07	1	0	1	0	0	0	2	0	3
Doc_08	0	1	0	1	0	1	1	2	0
Doc_09	0	1	1	0	0	1	2	1	2
Doc_10	0	0	1	0	0	1	1	1	2
Doc_11	1	0	0	0	0	1	1	1	1
Doc_12	1	1	0	0	1	0	2	1	4

Week 15 Topic: Example - SVD

Measures like cosine distance have turned out to be more useful in text mining practice. The idea is that if the angle between two concept vectors is small, the vectors probably represent the same concept, whereas if the angle is large, then two different concepts are probably being represented. If the angle between two document vectors is small, then the documents probably contain very similar information.

	doc	SVD Dimensions		
		_SVD_1	_SVD_2	_SVD_3
"Due to repetitive motion…"	i	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$
"Bilateral carpal tunnel…"	j	$x_{j,1}$	$x_{j,2}$	$x_{j,3}$

Euclidean Distance: $$D_{i,j} = \sqrt{\sum_{k=1}^{3}(x_{i,k} - x_{j,k})^2}$$

Cosine Distance: $$cos(i,j) = \sum_{k=1}^{3}(x_{i,k}x_{j,k}) / \sqrt{\sum_{k=1}^{3} x_{i,k}^2 \sum_{k=1}^{3} x_{j,k}^2}$$

Week 15 Topic: Example – SVD

Transform the vector from the "term" space to a "topic" space, which allows document of similar topics to situate close by each other even they use different terms. (e.g. document using the word "pet" and "cat" are map to the same topic based on their co-occurrence).

Text	Doc	_DO	_SVD_1	_SVD_2	_SVD_3	_SVD_4
hamster dog cat walrus dog puppy dog kitten bear dog	1	1	0.8627653478	-0.505604544	0.1045973187	-0.120193473
dog mouse dog cat dog walrus dog seal dog otter	2	2	0.7053602751	0.7088489841	-0.216564067	0.2366841568
horse cat dog cat walrus cat bear cat cow	3	3	0.7769577791	0.6295527058	0.6684175798	-0.156471947
cow cat dog cat walrus cat seal cat otter pig	4	4	0.5704071154	0.8213621142	-0.352265542	-0.404350098
pig cat dog cat walrus cat seal cat tiger cat	5	5	0.5790601121	0.8152848499	-0.370302268	-0.315265077
walrus zebra walrus dog walrus cat walrus seal cow horse gopher	6	6	0.6554941857	0.7552002201	0.4473725681	0.0172727027
walrus kitten walrus seal hamster dog walrus cat walrus seal hamster	7	7	0.8613923255	-0.507940215	-0.175857706	0.0515286037
walrus tiger walrus dog walrus cat walrus seal cow horse	8	8	0.603574935	0.797306276	0.3768541689	0.164752277
seal otter walrus dog seal cat seal walrus seal tiger seal	9	9	0.6440206232	0.7650081287	-0.333745451	0.5194389501
seal bear walrus dog seal cat seal walrus seal	10	10	0.9234475766	0.3837246061	0.1350170956	0.1617794521

Week 15 Topic: Housekeeping

- Site **course number** and **group number** in every submission file – Blackboard and Discussion
- If you have not joined a group and submitted work, I assume you have not completed Week 13 assignment at this time.
- Groups 7, 8, 9, and 10 – unless you post your code for sharing in Discussion, I will deduct an additional 1 point from Week 13.

Week 15 Topic: Code Sharing Highlights – 527-Group 7

Step 4: Create a String vector containing the sub string of the key column names present in the Summary Compensation Table

```
checkVector <-
c("principal","position","Principal","Position","Year",
"Salary","Bonus","Stock","awards","Awards","open","com
pensation","Compensation","Total","$","Option","option",
"Name","name")
```

Step 5: The following code iterates through all the table's column names present in the html.tables table vector and checks for the highest match with the checkvector mentioned above and provides the summary compensation table's index in the html.tables vector.

```
highestProbability<-0
for (i in 1:length(html.tables)) {
  matchs <- 0
  colNameVector <-colnames(html.tables[[i]])
  for (j in 1:length(colNameVector)) {
    for(n in 1:length(checkVector)){
      if(!is.null(colnames(html.tables[[i]][j])) &&
grepl(checkVector [n], colnames(html.tables[[i]][j])))
        matchs <- matchs+1
    }
  }
tempProbability <- matchs/length(checkVector)
  if(tempProbability>=highestProbability)
   {
    highestProbability=tempProbability
    requiredTable <- i
  }
}
```

Step 7: view the table to verify if we are pointing to correct table:

```
html.tables[[requiredTable]]
names(html.tables[[requiredTable]])
```

Step 6: Export the data frame to an excel file to clean the data and import it back using the following code.

```
write.table(html.tables[[requiredTable]],file="NikeDEF1
4A.csv",append = TRUE)
```

Week 15 Topic:
Code Sharing Highlights – 527-Group 2

```
library("assertthat", lib.loc="~/R/win-library/3.2")
library("BH", lib.loc="~/R/win-library/3.2")
library("bitops", lib.loc="~/R/win-library/3.2")
library("curl", lib.loc="~/R/win-library/3.2")
library("DBI", lib.loc="~/R/win-library/3.2")
library("dplyr", lib.loc="~/R/win-library/3.2")
library("htmltab", lib.loc="~/R/win-library/3.2")
library("httr", lib.loc="~/R/win-library/3.2")
library("jsonlite", lib.loc="~/R/win-library/3.2")
library("lazyeval", lib.loc="~/R/win-library/3.2")
library("magrittr", lib.loc="~/R/win-library/3.2")
library("mime", lib.loc="~/R/win-library/3.2")
library("NLP", lib.loc="~/R/win-library/3.2")
library("openssl", lib.loc="~/R/win-library/3.2")
library("plyr", lib.loc="~/R/win-library/3.2")
library("R6", lib.loc="~/R/win-library/3.2")
library("Rcpp", lib.loc="~/R/win-library/3.2")
library("RCurl", lib.loc="~/R/win-library/3.2")
library("slam", lib.loc="~/R/win-library/3.2")
library("stringi", lib.loc="~/R/win-library/3.2")
library("stringr", lib.loc="~/R/win-library/3.2")
library("tidyr", lib.loc="~/R/win-library/3.2")
library("tm", lib.loc="~/R/win-library/3.2")
library("XML", lib.loc="~/R/win-library/3.2")
library("boot", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("class", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
```

Week 15 Topic:
Code Sharing Highlights – 527-Group 2

```
library("cluster", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("codetools", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("compiler", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("datasets", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("foreign", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("graphics", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("grDevices", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("grid", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("KernSmooth", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("lattice", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("MASS", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("Matrix", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("methods", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("mgcv", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("nnet", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("parallel", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("rpart", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("spatial", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("splines", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("stats", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("stats4", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("survival", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("tcltk", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("tools", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("translations", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
library("utils", lib.loc="C:/Program Files/R/R-3.2.4revised/library")
```

Week 15 Topic:
Code Sharing Highlights – 527-Group 3

- Saved all the “html” links for the DEF 14A filings URL’s in a csv sheet named “Company_URL.csv”.
- Also we have a “csv” file with patterns of words we are searching for in the set of “html” documents. This file has words like “Principal”, ”Position” and ”Name”, combination of which would give us the Compensation table.
- Once we have the compensation table, we are writing it into a “csv” file for every year’s filing.

```
a <- "Company_URL.csv";
   b <- "Patterns.csv";
   Urls = scan(a, what = "",sep = "\n");
   Patterns = scan(b, what = "",sep = "\n");
   for (i in 1 :length(Urls)){
      data = htmlParse(Urls[i]);
      for(j in 1 : length(Patterns)){
         temp = as.character(paste("//table[contains(.,'", Patterns[j], "')]", sep=""))
         c = xpathSApply(data,temp);
         for(k in 1: length(c)){
            d = saveXML(c[[k]]);
            write(d,'temp.txt',append=TRUE);          }
         d = htmlParse('temp.txt');
         write('','temp.txt',append = FALSE);       }
      myXML = saveXML(d);
      table = readHTMLTable(myXML);
      dataframe = data.frame(table);
      df1 = data.frame(lapply(dataframe,function(x){gsub("ÃƒÂ,Ã,Â","",x)}));
      file = as.character(paste("file",i,".csv",sep = ""));
      write.csv(df1,file);    }
```

Week 15 Topic:
Code Sharing Highlights – 527-Group 4

```
> library("rvest")
> library("RCurl")
> library("XML")
> library("plyr")
> url <- "http://www.sec.gov/Archives/edgar/data/37996/000104746916011835/a2228102zdef14a.htm"
> compensationTableFord_2015_2013 <- url%>%
+ read_html()%>%
+ html_nodes(xpath = '/html/body/document/type/sequence/filename/description/text/div[77]/div/table')%>%
+ html_table(fill = TRUE)
> View(compensationTableFord_2015_2013)
> compensationTableFord_2015_2013 <- compensationTableFord_2015_2013[[1]]
> compensationTableFord_2015_2013 <-
compensationTableFord_2015_2013[6:8,c("X1","X4","X7","X10","X13","X16","X19","X22","X25","X28")]
> View(compensationTableFord_2015_2013)
> compensationTableFord_2015_2013 <- rename(compensationTableFord_2015_2013, c("X1"="Name and Principal
Position","X4"="Year","X7"="Salary($)","X10"="Bonus($)","X13"="Stock Awards($)","X16"="Option
Awards($)","X19"="Non-Equity Incentive Plan Compensation($)","X22"="Change in Pension value and Deferred
Compensation Earnings($)","X25"="All Other Compensation($)","X28"="Total($)"))
> View(compensationTableFord_2015_2013)
> write.csv(compensationTableFord_2015_2013,"Ford.csv")
> View(compensationTableFord_2015_2013)
```

Week 15 Topic:
Code Sharing Highlights – 527-Group 6

```
require(htmlTable)
library(htmltab)
library(plyr)
x <- scan("url.txt",what="",sep="\n")

for (i in 1:length(x)){  -- Create a for loop which opens
a text file and checks each URL
url <- x[i]    -- i values loops through the url
print(url)   -- prints the url
summaryTable <- htmltab(url,
  which =  "//*[text()[contains(.,'Principal')]
       and text()[contains(.,'Position')]]/ancestor::table")
as.data.frame(summaryTable) – This searchs for the
table with the tab principal,position
output <- as.data.frame(sapply(
          summaryTable,gsub,pattern='\\Â',replacement="
")) – Removes the 'A' and replaces
filename <-
paste(paste("summary",i,sep=""),".csv",sep="") --
write.csv(assign(paste("output",i,sep=""),
          output),file=filename,na = "", row.names =
FALSE)
}
```

```
>Data.frame(tablename)
```

The function data.frame() collections of variables which share many of the properties of matrices and of lists, used as main data structure.

```
> df4= data.frame (Morgan
2006)
>df4 = data.frame(lapply(df3,function(x){gsub("ÃƒÂ,Ã,
Â.","",x)}))
```

Week 15 Topic:
Code Sharing Highlights – 527-Group 10

```
#starts here
install.packages("statnet.common")
install.packages("plyr")
library("plyr")
library("XML")
library("statnet.common")
library("qdap")

mps<-
"http://www.sec.gov/Archives/edgar/data/24741/000130817914000058/lcorning2014_def14a.htm"

mps.doc <- htmlParse(mps)
# get all the tables in mps.doc as data frames
mps.tabs <- readHTMLTable(mps.doc)
length(mps.tabs)
```

```
# ... and the loop:
for (i in 1:length(mps.tabs)) {
 dat <-
data.frame(text1=sent_detect(head(mps.tabs[[i]][1,1])),
stringsAsFactors = FALSE)
  x<-Search(dat, "Executive Officer")
    if(identical(x, character(0)) ){
  }else{
    print(i)  }}
# The above loop will print the table numbers with
matching text.

mps.tabs[[197]]  # 197 is table Number
compensation.data1<-mps.tabs[[197]]
summarycompensation.table<-mps.tabs[[197]]
summarycompensation.table
compensation.data1 <- summarycompensation.table[2:4,
c("(a)","(b)","(c)","(e)(1)","(f)(2)")]
compensation.data2014 <-rename(compensation.data1,
c("(a)"="Name and Principal Position", "(b)"="Year",
"(c)"="Salary",
"(e)(1)"="Stock_Awards","(f)(2)"="Option_Awards"))
```

Week 15 Topic: Code Sharing Highlights – 529-Group 1

```
#Including useful libraries
library(XML)
library(plyr)
library(qdap)

#Storing the path of the HTML file into different variables
path  <-
"file:///C:/Users/Meet/Desktop/term4/529%20ADA/Exec_submission_1/Oracel/Oracle_2015.htm"
path1 <-
"file:///C:/Users/Meet/Desktop/term4/529%20ADA/Exec_submission_1/Oracel/Oracle_2014.htm"
path2 <-
"file:///C:/Users/Meet/Desktop/term4/529%20ADA/Exec_submission_1/Oracel/Oracle_2013.htm"
path3 <-
"file:///C:/Users/Meet/Desktop/term4/529%20ADA/Exec_submission_1/Oracel/Oracle_2012.htm"

#Creating a function which will find which table..
#..to look for and cleaning and storing it into a CSV file
fun.find.table <- function(path)
{
  #Reading HTML Tables from the given HTML File lying at the given URL
  all.tables <- readHTMLTable(path)

#Finding out which table to look for
  for (i in 1:length(all.tables))
  { #taking 1st Raw and 1st Column attribute of the particular table
  dat <- data.frame(text1=sent_detect(all.tables[[i]][1,1]), stringsAsFactors = FALSE)
  #Matching it with string "Name and Principal Position"
  x<-Search(dat, "Name and Principal Position")
  if(identical(x, character(0)) )    {
    #Don't return anything    }
  else    {
    #Clean the table before storing
    final.table <- as.data.frame(all.tables[i])
    #Removing special characters
    final.table <- data.frame(lapply(final.table, gsub, pattern = "Â", replacement = ""))

    #Removing NULL/Empty columns
    final.table <- na.omit(final.table)
    final.table <- final.table[, colwise(function(x){
      length (unique(x)) })(final.table)!=1]

    #Selecting only first 4 raws which has CEO information
    final.table <- final.table[1:4,]                } }
  #Write the data into CSV
  write.csv(final.table,file="final.2014.1.csv")
}
```

Week 15 Topic:
Code Sharing Highlights – 529-Group 1

```
#Function Call
fun.find.table(path)
fun.find.table(path1)
fun.find.table(path2)
fun.find.table(path3)


#Running the above function call for each and every file we
need data from..
#..will give us different csv file which we have merged in one
file.
# A snapshot is included in the zip file.
```

Week 15 Topic:
Code Sharing Highlights – 529-Group 7

```
#Installation of packages
install.packages("tm")
install.packages("devtools")
devtools::install_github("crubba/htmltable")
install.packages("xlsx")
#Assign library
library(htmltab)
library(tm)
library(xlsx)

#Create final summary table for each company
final.summary.table <-
data.frame(Company_Name=character(),
Name_and_Principal_Position=character(),
Year=character(),Salary=character(),
Stock_Awards=character(), Option_Awards=character(),
stringsAsFactors=FALSE)

#Set directory and save all DEF 14A files in this directory
using standard naming convention :: company name followed
by filing #year e.g Morgan_2015.htm
cname <- "E:/Sonali_MS/ITMD529/data_mining/dataset"
length(dir(cname))
dir(cname)
proxies <- Corpus(DirSource(cname))
```

```
#FOR loop is used to iterate through corpus in descending order so
latest filing's data will get loaded first and so on.
for(file_index in length(dir(cname)):1){
#Assigned filename to url and fetched company name from
filename
  url <- paste(cname,meta(proxies[[file_index]],"id"),sep = "/")
  comp_name<-
substr(meta(proxies[[file_index]],"id"),1,nchar(meta(proxies[[file_i
ndex]],"id"))-9)

  #Read all HTML tables and fetched summary compensation table
using keyword search
  tables = readHTMLTable(url)
  length(tables)   count<-0
  for(i in 1:length(tables)){    p<-tables[[i]]
    for(j in 1:length(p)){     if(sum(grepl("Principal", p[,j],ignore.case
= TRUE))>0 && sum(grepl("Position", p[,j],ignore.case =
TRUE))>0 && !is.null(p) && count ==0)        count<-count+1
if(sum(grepl("Year", p[,j],ignore.case = TRUE))>0 && !is.null(p)
&& count == 1)        count<-count+1       if(sum(grepl("Salary",
p[,j],ignore.case = TRUE))>0 && !is.null(p) && count == 2)
count<-count+1    }
#Assigned summary table to proxy.table
    if(count==3)    {proxy.table <- htmltab(doc = url, which = i)
    break}     count<-0  }
```

Week 15 Topic:
Code Sharing Highlights – 529-Group 7

```
 #Replaced garbage values with NA
  garbage_value<-c("Â—","Â—Â—","Â Â Â Â","Â Â Â","Â—Â Â","ÂÂ","Â","$","â€‹","â€\"*.","â€")
  for(i in 1:length(proxy.table)){
    for(x in garbage_value){
      proxy.table[,i]<-gsub(x,"",proxy.table[,i])
      proxy.table[,i][proxy.table[,i] == ""]<-NA
      proxy.table[,i][proxy.table[,i] == "$"]<-NA    }  }
#Remove unwanted columns and rename headers
  header_arr <- c("Name", "Year", "Salary", "Stock","Option")
  names(header_arr) <- c("Name_and_Principal_Position", "Year", "Salary", "Stock_Awards","Option_Awards")
  x<-""
  j<-1
  i<-1
  flag<-0
while(i != (length(proxy.table)+1)){
    for(x in header_arr){
      if(grepl(x,proxy.table[i],ignore.case = TRUE) | (grepl(x,names(proxy.table[i]),ignore.case = TRUE)))
      {  colnames(proxy.table)[i]<-names(header_arr[j])
                          flag=1
                         break      }
      j<-j+1     }
 if(flag == 0)
      proxy.table[,i]<-NULL
    else       i<-i+1       j<-1     flag<-0  }
```

```
  #Remove duplicated columns if any
  #Year column is never null so if year values doesn't appear in it then delete that column
  #as it's a garbage column
  year<-as.character(c(1994:2015))
  for(k in 1:3){
  if(sum(proxy.table[1:nrow(proxy.table),2] %in% year)==0 && names(proxy.table[2]) == "Year" | names(proxy.table[2]) == "Name_and_Principal_Position")
    proxy.table[,2]<-NULL   }

  #remove null rows
  proxy.table <- proxy.table[rowSums(is.na(proxy.table)) != ncol(proxy.table),]
  proxy.table <-proxy.table[proxy.table$Year %in% year,]
```

Week 15 Topic:
Code Sharing Highlights – 529-Group 7

```
 #Select CEO rows using first and second occurrence of
same year value
 value<-""
 for(i in 1:nrow(proxy.table)){
  if(proxy.table$Year[i] %in% year)
  {  first_occurance<-i
     value=proxy.table$Year[i]
     break    }  }
 second_occurance<-first_occurance+1
 i<-i+1
 while(i != nrow(proxy.table)){
  if(proxy.table$Year[i] == value | is.na(proxy.table$Year[i]))
  {  second_occurance<-i
     break }
  i<-i+1   }
 proxy.table <-
proxy.table[first_occurance:(second_occurance-1),]
```

```
#Remove null columns and add only required columns back
as it should match final table format
 proxy.table <- proxy.table[,colSums(is.na(proxy.table)) !=
nrow(proxy.table)]
 for(x in header_arr){
   if(sum(grepl(x,colnames(proxy.table),ignore.case = TRUE))
== 0){     if(x == "Salary")
       proxy.table[,"Salary"]<-""
     if(x == "Stock")
       proxy.table[,"Stock_Awards"]<-""
     if(x == "Option")
       proxy.table[,"Option_Awards"]<-""    }  }

 #Merge proxy.table into final table comparing year
 for(i in 1:nrow(proxy.table)){
   if(!as.integer(proxy.table$Year[i]) %in%
final.summary.table$Year)
   {       final.summary.table[sapply(final.summary.table,
is.factor)] <-
lapply(final.summary.table[sapply(final.summary.table,
is.factor)], as.character)
      final.summary.table <-
rbind(final.summary.table,c(Company_Name=comp_name,pr
oxy.table[i,]))
   }
 }
}
```

Week 15 Topic: Removing Sparse Terms

```
> dtms <- removeSparseTerms(qa_dtm, 0.1)
> dim(dtms)
[1] 5 336
> dim(dtm)
[1] 1546 5

> inspect(dtms)
<<DocumentTermMatrix (documents: 5, terms: 336)>>
Non-/sparse entries: 1680/0
Sparsity           : 0%
Maximal term length: 19
Weighting          : term frequency (tf)
              Terms
Docs           abl abstent access account act action addit adjourn admiss admit
  character(0)  2      3      1      8   6      2     3      4      3     2
  character(0)  2      7      1      6   6      2     3      3      3     2
  character(0)  2      8      1      7   5      1     5      1      3     2
  character(0)  3     12      2     12  12      4     9      2      6     4
  character(0)  2      7      2      8   6      2     4      1      3     3
```

Week 15 Topic: Removing Sparse Terms (cont.)

Docs	adopt	advanc	affirm	against	agent	allow	altern	although	alway	amend
character(0)	2	2	2	2	2	1	1	1	7	9
character(0)	2	2	2	4	2	1	1	1	6	2
character(0)	1	2	1	2	2	1	1	1	6	10
character(0)	4	4	2	6	3	2	2	2	11	4
character(0)	1	3	1	7	1	4	1	1	10	15

Terms

Docs	amount	and	anniversari	announc	annual	answer	appli	applic	appoint
character(0)	1	5	1	2	76	1	2	1	5
character(0)	1	2	1	2	70	1	2	2	4
character(0)	1	2	1	2	66	1	2	2	6
character(0)	2	4	2	3	132	2	2	3	10
character(0)	1	7	1	2	84	9	1	2	7

Week 15 Topic: Removing Sparse Terms (cont.)

Docs	Terms approv	are	assist	attend	attent	audit	author	avail	avenu	ballot	bank
character(0)	6	1	2	11	1	2	3	23	4	1	9
character(0)	3	1	2	12	1	1	3	22	2	1	8
character(0)	10	1	2	13	1	1	2	6	2	2	12
character(0)	12	1	4	26	2	2	6	41	5	4	23
character(0)	13	1	2	13	1	1	11	24	4	2	9

Docs	Terms bear	becom	begin	benefici	board	both	break	broker	brokerag	bulki
character(0)	2	1	2	16	12	1	7	16	2	1
character(0)	2	1	2	15	13	1	6	15	2	1
character(0)	2	1	2	17	12	1	6	23	2	1
character(0)	4	2	3	35	21	2	11	41	4	1
character(0)	2	1	2	19	15	1	10	18	2	1

…

Week 15 Topic: Removing Sparse Terms (cont.)

```
> freq <-colSums(as.matrix(dtms))
> freq
       abl   abstent    access   account
        11        37         7        41
       act    action    addit   adjourn
        35        11        24        11
    admiss     admit     adopt    advanc
        18        13        10        13
    affirm   against     agent     allow
         8        21        10         9
    altern  although     alway    amend
         6         6        40        40
    amount       and anniversari  announc
         6        20         6        11
    annual    answer     appli    applic
       428        14         9        10
   appoint    approv       are    assist
        32        44         5        12
…
```

Week 15 Topic: Frequencies of words

```
> table(freq)
freq
 5   6   7   8   9  10  11  12  13  14  16  17  18  19  20  21
 20  54  17   9  11  14  10  15  11   5   6   9   7   1   6   7
 22  23  24  25  26  27  28  29  30  31  32  34  35  36  37  38
  7   5   9   3   1   3   3   5   2   2   1   2   3   3   3   2
 40  41  42  43  44  45  46  47  48  49  50  51  53  57  58  60
  5   1   3   3   2   2   1   2   2   1   1   3   2   1   1   2
 61  62  63  66  67  68  70  71  73  75  76  81  82  84  86  88
  1   1   1   1   1   2   1   2   1   4   2   1   1   1   1   1
 89  93  96  97 100 102 103 109 112 113 116 127 129 143 149 172
  1   1   1   1   1   1   1   1   1   1   1   2   1   1   1   1
246 315 329 413 428 440 493 595 1549
  1   1   1   1   1   1   1   1   1
```

Week 15 Topic: Frequencies of words (cont.)

```
> findFreqTerms(dtm, lowfreq=50)

[1] "alignjustify" [2] "alignleft" [3] "alignleftfont" [4] "annual" [5] "availability" [6] "bank" [7]
"beneficial" [8] "bfonttdtrtable" [9] "board" [10] "border" [11] "broker" [12] "business" [13]
"can" [14] "cellpadding" [15] "cellspacing" [16] "companys" [17] "date" [18] "director" [19]
"directors" [20] "election" [21] "entitled" [22] "following" [23] "fontfamilyarial" [24]
"fontsizept" [25] "fonttdtrtable" [26] "fonttdtrtablenp" [27] "form" [28] "holder" [29]
"instructions" [30] "internet" [31] "materials" [32] "matters" [33] "may" [34] "meeting" [35]
"must" [36] "new" [37] "nominee" [38] "notice" [39] "person" [40] "present" [41] "proposal" [42]
"proposals" [43] "proxy" [44] "received" [45] "record" [46] "report" [47] "roman" [48]
"shareholder" [49] "shareholders" [50] "shares" [51] "size" [52] "sizebabfonttdntd" [53]
"sizebqbfonttdntd" [54] "sizebwhat" [55] "sizefonttdntd" [56] "sizenbspfonttdntd" [57] "solid"
[58] "statement" [59] "stock" [60] "stylebordercollapsecollapse" [61] "stylefontfamilyarial" [62]
"stylefontfamilytimes" [63] "stylefontsizeptmargintopptmarginbottomptnbsppntable" [64]
"stylefontsizepxmargintoppxmarginbottompxnbsppntable" [65]
"stylemargintoppxmarginbottompx" [66] "valignbottom" [67] "valignbottomfont" [68]
"valigntop" [69] "valigntopfont" [70] "vote" [71] "voted" [72] "votes" [73] "voting" [74] "width"
[75] "widthfont" [76] "widthntrntd" [77] "will" [78] "yahoo"
```

Week 15 Topic: Frequencies of words (cont.)

```
> table(freq)
freq
5 6 7 8 9 10 11 12 13 14 16 17 18 19 20 21 20 54 17 9 11 14 10 15 11 5 6 9 7 1 6 7 22 23 24 25
26 27 28 29 30 31 32 34 35 36 37 38 7 5 9 3 1 3 3 5 2 2 1 2 3 3 3 2 40 41 42 43 44 45 46 47 48
49 50 51 53 57 58 60 5 1 3 3 2 2 1 2 2 1 1 3 2 1 1 2 61 62 63 66 67 68 70 71 73 75 76 81 82 84
86 88 1 1 1 1 1 2 1 2 1 4 2 1 1 1 1 1 89 93 96 97 100 102 103 109 112 113 116 127 129 143 149
172 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 246 315 329 413 428 440 493 595 1549 1 1 1 1 1 1 1 1 1

> findFreqTerms(dtm, lowfreq=100)
[1] "alignleft" [2] "alignleftfont" [3] "annual" [4] "border" [5] "broker" [6] "cellpadding" [7]
"cellspacing" [8] "fontfamilyarial" [9] "fontsizept" [10] "internet" [11] "materials" [12] "may"
[13] "meeting" [14] "new" [15] "notice" [16] "proposals" [17] "proxy" [18] "record" [19] "roman"
[20] "shareholder" [21] "shareholders" [22] "shares" [23] "sizebabfonttdntd" [24]
"sizebqbfonttdntd" [25] "sizenbspfonttdntd" [26] "stylebordercollapsecollapse" [27]
"stylefontfamilyarial" [28] "stylefontfamilytimes" [29]
"stylefontsizepxmargintoppxmarginbottompxnbsppntable" [30] "valigntop" [31]
"valigntopfont" [32] "vote" [33] "voting" [34] "width" [35] "widthntrntd" [36] "will"
```

Week 15 Topic: Observing Correlations

```
> findAssocs(dtm, "yahoo", corlimit = .9)
$yahoo
matters annual 1.00 0.99
solicitation textindentfont 0.99 0.99
able acted 0.98 0.98
actions admissionb 0.98 0.98
alsonavailable alternatives 0.98 0.98
although amount 0.98 0.98
anniversary announced 0.98 0.98
anvoting arenconsidered 0.98 0.98
arennot assist 0.98 0.98
assisting attendance 0.98 0.98
attention audited 0.98 0.98
basis bear 0.98 0.98
become begin 0.98 0.98
benapproved boards 0.98 0.98
brokerage cause 0.98 0.98
choose class 0.98 0.98
…
```

Week 15 Topic: Plotting Frequent Words

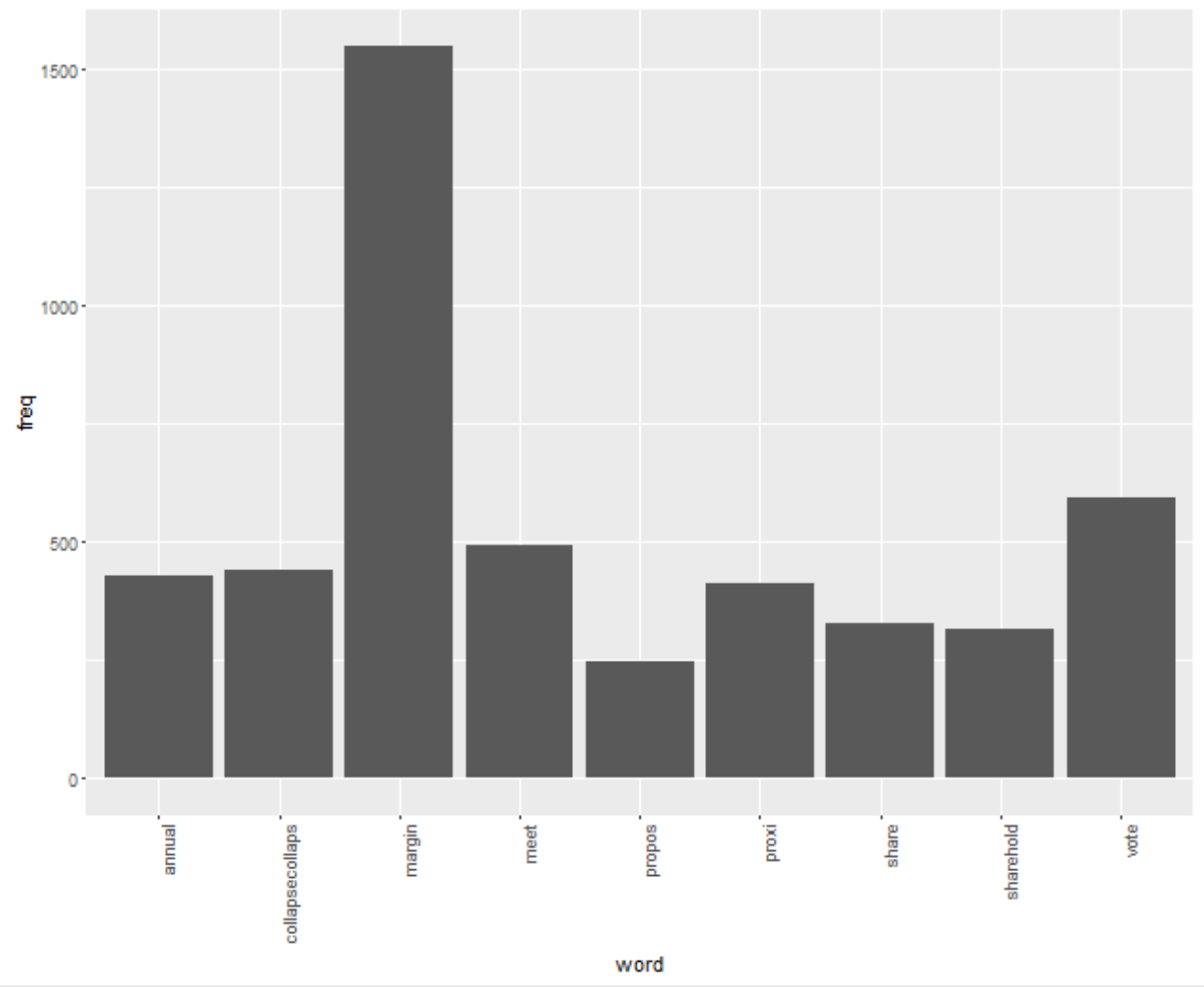
```
> freq <- sort(colSums(as.matrix(dtms)), decreasing = TRUE)
> head(freq, 14)
margin vote meet collapsecollaps annual 1549 595 493 440 428 proxi share sharehold propos
materi 413 329 315 246 172 director receiv yahoo instruct 149 143 129 127

> wf <- data.frame(word=names(freq), freq=freq)
> head(wf)
                     word freq
margin                margin 1549
vote                   vote  595
meet                   meet  493
collapsecollaps collapsecollaps  440
annual                annual  428
proxi                  proxi  413

>install.packages("ggplot2")
>library(ggplot2)
>subset(wf, freq>200) %>% ggplot(aes(word, freq)
```

Week 15 Topic: Plotting Frequent Words (cont.)

```
>install.packages("ggplot2")
>library(ggplot2)
>subset(wf, freq>200)
word freq
margin margin 1549
vote vote 595
meet meet 493
collapsecollaps collapsecollaps 440
annual annual 428
proxi proxi 413
share share 329
sharehold sharehold 315
propos propos 246
```

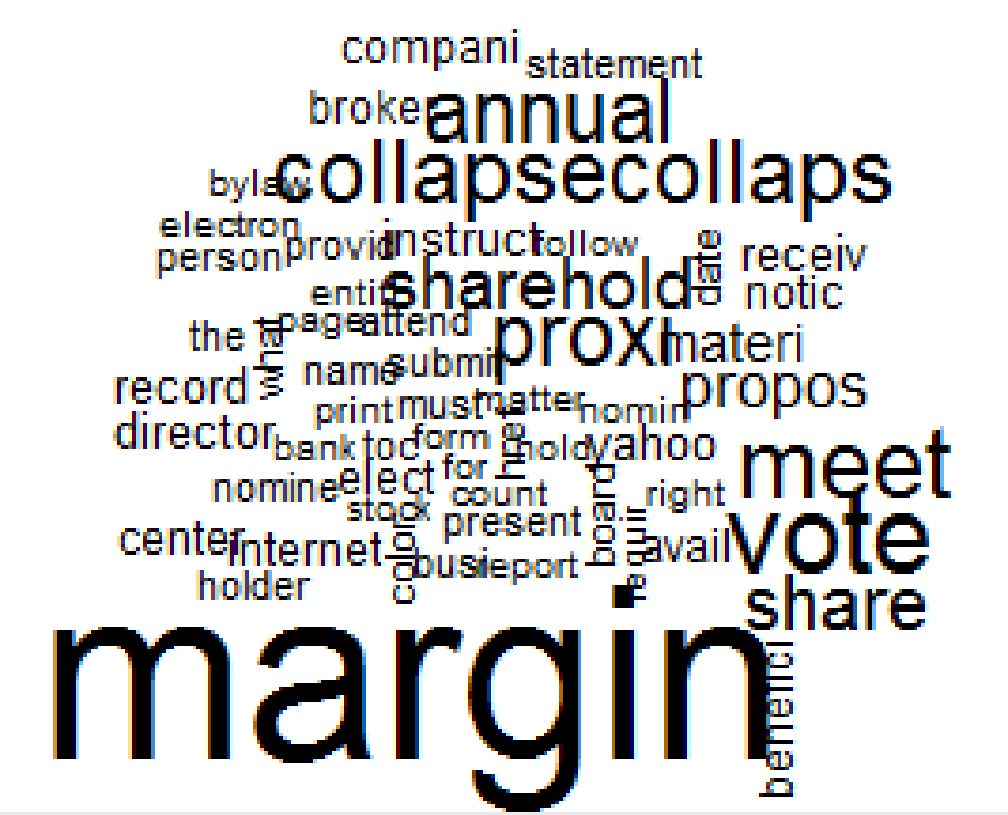
Week 15 Topic:
Plotting Frequent Words (cont.)

```
>install.packages("dplyr")
>library(dplyr)
> subset(wf, freq>200) %>%
ggplot(aes(word, freq))
+
geom_bar(stat="identity")
+
theme(axis.text.x=element_text(angle=45, hjust=1))
```



Week 15 Topic: Generating a Word Cloud

```
> install.packages("wordcloud")
> library(wordcloud)
> set.seed(123)
> wordcloud(names(freq), freq, min.freq=50)
```



Week 15 Topic:
Week 14 Assignment – due 21st

PART 2: Corpus/Term Document Matrix & Identifying KEIs

- Create a corpus of filings per company. The Corpus is to house the Q&A sections for each filing. Striped of punctuation, etc.
- Generate a Term Document Matrix of the Q&A content.
- Highlight any patterns or trends found in analyzing the results of the Q&A Term Document Matrix statistics.

Week 15 Topic:
Final Project Presentation – Due 26th

The final project should contain the following:

- Table of Content – well formatted (1 slide)
- Company/Industry Selection/Background/Overview (1 slide)

PART 1:

Per company (1-5 slides):

- Make sure to collect all available DEF 14A filings available for the companies. Make sure to chart the salary, option awards, and/or stock awards over the available years for the company.
- Highlight any stock trading activity changes/activites before and after filing.

For all companies/industry (1-3 slides):

- Make sure to chart the salary, option awards, and/or stock awards over the available years for the company.
- Highlight any stock trading activity changes before and after filing.

Week 15 Topic:
Final Project Presentation - Due 26th

PART 2:

Per company (1-5 slides):

- Highlight any patterns or trends found in analyzing the results of the Q&A Term Document Matrix statistics.

For all companies/industries (1-3 slides):

- Create a corpus of filings per company. The Corpus is to house the Q&A sections for each filing. Striped of punctuation, etc.
- Generate a Term Document Matrix of the Q&A content
- Highlight any patterns or trends found in analyzing the results of the Q&A Term Document Matrix statistics.
- Highlihgt any KEIs identified

Week 15 Topic:
Final Project Revisions - Due May 2nd

Resubmit Final Presentation with any updates and changes

Week 14 Topic: Discussion Topic Highlights

R for table parsing – alternative to readHTMLTable -- Using xpath:

```
proxy.table <- htmltab(doc = url, which = "//*[text()[contains(.,'Principal')] and
text()[contains(.,'Position')]]/ancestor::table")

> URL_YAHOO_2015 <- "https://www.sec.gov/***"
> Comp_Summ_YAHOO_2015_13 <- URL_YAHOO_2015 %>%
+ read_html() %>%
+ html_nodes(xpath='/html/body/document/type/sequence/filename/description/text/table
[286]') %>%
+ html_table(fill = TRUE)
> Comp_Summ_YAHOO_2015_13 <- Comp_Summ_YAHOO_2015_13[[1]]
> Comp_Summ_YAHOO_2015_13 <- Comp_Summ_YAHOO_2015_13[3:5, c("X1", "X4", "X8",
"X12", "X16", "X20", "X24", "X28", "X32")]
> Comp_Summ_YAHOO_2015_13 <- rename(Comp_Summ_YAHOO_2015_13, c("X1"="Name
and Principal Position", "X4"="Year", "X8"="Salary", "X12"= "Bonus", "X16"="Stock Awards",
"X20"="Option Awards","X24" = "Non-Equity Incentive Plan Compensation", "X28" = "All
Other Compensations" , "X32" = "Total Compensation"))
> write.csv(Comp_Summ_YAHOO_2015_13, file = "Yahoo.csv")
```

Week 14 Topic: Discussion Topic Highlights (cont.)

2) Cleansing table:

```
 i<-2
while(i != (length(proxy.table)+1))
{   proxy.table[,i] <- replace(as.character(proxy.table[,i]), grep("Â", substr(proxy.table[,i],
1,nchar("Â")) ,fixed=TRUE) , NA)
    proxy.table[,i] <- replace(as.character(proxy.table[,i]), grep("â€", substr(proxy.table[,i],
1,nchar("â€")) ,fixed=TRUE) , NA)
   proxy.table[,i][proxy.table[,i] == "$"]<-NA
   i<-i+1
}
```

Week 14 Topic: readLines Yahoo filings for 2010~2014

```
> yahoo_proxy2014 <- readLines("C:/Users/sshin/Desktop/Yahoo/yahoo_proxy2014.html")
> yahoo_proxy2013 <- readLines("C:/Users/sshin/Desktop/Yahoo/yahoo_proxy2013.html")
> yahoo_proxy2012 <- readLines("C:/Users/sshin/Desktop/Yahoo/yahoo_proxy2012.html")
> yahoo_proxy2011 <- readLines("C:/Users/sshin/Desktop/Yahoo/yahoo_proxy2011.html")
> yahoo_proxy2010 <- readLines("C:/Users/sshin/Desktop/Yahoo/yahoo_proxy2010.html")
```

Etc.

Better to create a loop per company or all companies…

Week 14 Topic:
Locating QA section Starting Lines

```
> qa_sentence_start <- "QUESTIONS AND ANSWERS ABOUT OUR PROXY MATERIALS"
> grep(qa_sentence_start, yahoo_proxy2014, ignore.case = TRUE) [1] 237 688 1041
```

**Take the third line number as the two are in the Table of Contents and Proposal sections*

```
> qa_sentence_start <- "QUESTIONS AND ANSWERS ABOUT THE PROXY MATERIALS"
> grep(qa_sentence_start, yahoo_proxy2013, ignore.case = TRUE) [1] 213 490
```

**Take the latter line number as the first is from Table of Contents*

```
> qa_sentence_start <- "QUESTIONS AND ANSWERS ABOUT THE PROXY MATERIALS"
> grep(qa_sentence_start, yahoo_proxy2012, ignore.case = TRUE) [1] 218 375
> grep(qa_sentence_start, yahoo_proxy2011, ignore.case = TRUE) [1] 206 401
> grep(qa_sentence_start, yahoo_proxy2010, ignore.case = TRUE) [1] 207
```

**Take the latter line number as the first is from Table of Contents*

Etc.

Week 14 Topic: Locating QA section Ending Lines

```
> qa_sentence_end <- "Your electronic delivery enrollment will be effective until you cancel it"
> grep(qa_sentence_end, yahoo_proxy2010, ignore.case = TRUE) [1] 813
> grep(qa_sentence_end, yahoo_proxy2011, ignore.case = TRUE) [1] 877
> grep(qa_sentence_end, yahoo_proxy2012, ignore.case = TRUE) [1] 904
> grep(qa_sentence_end, yahoo_proxy2013, ignore.case = TRUE) [1] 970

> qa_sentence_end <- "If you have questions about electronic delivery"
> grep(qa_sentence_end, yahoo_proxy2014, ignore.case = TRUE) [1] 1649
```

Week 14 Topic: Save QA section to file

```
> qa_section2014 <- yahoo_proxy2014[688:1649]
> cat(qa_section2014,
file="C:/Users/sshin/Desktop/Yahoo/qa_section/yahoo_proxyqa_section2014.txt", sep="n",
append = TRUE)
> qa_section2013 <- yahoo_proxy2013[490:970]
> cat(qa_section2013,
file="C:/Users/sshin/Desktop/Yahoo/qa_section/yahoo_proxyqa_section2013.txt", sep="n",
append = TRUE)
> qa_section2012 <- yahoo_proxy2012[375:904]
> cat(qa_section2012,
file="C:/Users/sshin/Desktop/Yahoo/qa_section/yahoo_proxyqa_section2012.txt", sep="n",
append = TRUE)
> qa_section2011 <- yahoo_proxy2011[401:877]
> cat(qa_section2011,
file="C:/Users/sshin/Desktop/Yahoo/qa_section/yahoo_proxyqa_section2011.txt", sep="n",
append = TRUE)
> qa_section2010 <- yahoo_proxy2010[207:813]
> cat(qa_section2010,
file="C:/Users/sshin/Desktop/Yahoo/qa_section/yahoo_proxyqa_section2010.txt", sep="n",
append = TRUE)
```

Week 14 Topic: Creating a QA section Corpus

```
> cname <- "C:/Users/sshin/Desktop/Yahoo/qa_section/"

> library(tm)
Loading required package: NLP
Warning messages: 1: package 'tm' was built under R version 3.2.3 2: package 'NLP' was built under R version 3.2.3

> qa_sections <- Corpus(DirSource(cname))
```

Week 14 Topic: Inspecting the QA Corpus

> inspect(qa_sections[1])

<<VCorpus>> Metadata: corpus specific: 0, document level (indexed): 0 Content: documents: 1 [[1]] <<PlainTextDocument>> Metadata: 7 Content: chars: 70329

> dir(cname)

[1] "yahoo_proxyqa_section2010.txt" "yahoo_proxyqa_section2011.txt" [3] "yahoo_proxyqa_section2012.txt" "yahoo_proxyqa_section2013.txt" [5] "yahoo_proxyqa_section2014.txt“

> class(qa_sections) [1] "VCorpus" "Corpus"

> class(qa_sections[[1]]) [1] "PlainTextDocument" "TextDocument"

> summary(qa_sections)

Length Class Mode

yahoo_proxyqa_section2010.txt 2 PlainTextDocument list

yahoo_proxyqa_section2011.txt 2 PlainTextDocument list

yahoo_proxyqa_section2012.txt 2 PlainTextDocument list

yahoo_proxyqa_section2013.txt 2 PlainTextDocument list

yahoo_proxyqa_section2014.txt 2 PlainTextDocument list

Week 14 Topic:
Inspecting the QA Corpus Content - Raw

**Define a function to view the Corpus files:*

> install.packages("magrittr")

> library(magrittr)

> viewDocs <- function(d,n) {d %>% extract2(n) %>% as.character() %>% writeLines()}

> viewDocs(qa_sections, 1)

<TD VALIGN="top"> <P STYLE="margin-left:1.00em; text-indent:-1.00em"><FONT STYLE="font-family:Times New Roman" SIZE="2"><A HREF="#toc25740_1">QUESTIONS AND ANSWERS ABOUT THE PROXY MATERIALS AND OUR 2010 ANNUAL MEETING OFnSHAREHOLDERS</A></FONT></P></TD>n<TD VALIGN="bottom"><FONT SIZE="1"> </FONT></TD>n<TD VALIGN="bottom" ALIGN="right"><FONT STYLE="font-family:Times New Roman" SIZE="2">1</FONT></TD></TR>n<TR>n<TD VALIGN="top"> <P STYLE="margin-left:1.00em; text-indent:-1.00em"><FONT STYLE="font-family:Times New Roman" SIZE="2"><A HREF="#toc25740_2">PROPOSAL NO. 1 ELECTION OF DIRECTORS</A></FONT></P></TD>n<TD VALIGN="bottom"><FONT SIZE="1"> </FONT></TD>n<TD VALIGN="bottom" ALIGN="right"><FONT STYLE="font-family:Times New Roman" SIZE="2">8</FONT></TD></TR>n<TR>n<TD VALIGN="top"> <P STYLE="margin-left:3.00em; text-indent:-1.00em"><FONT

….

Week 14 Topic: Transforming the QA Corpus - A

To lower case:

> qa_sections <- tm_map(qa_sections, content_transformer(tolower))

Remove Numbers:

> qa_sections <- tm_map(qa_sections, removeNumbers)

Remove Stop Words:

> qa_sections <- tm_map(qa_sections, removeWords, stopwords("english"))

> viewDocs(qa_sections, 1)

<td valign="top"> << href="#toc_">questions answers proxy materials annual meeting ofnshareholders</></></> </></td>n<td valign="bottom"> </></td>n<td valign="bottom" align="right"></></></td></></tr>n<tr>n<td valign="top"> << href="#toc_">proposal . election directors</></></> </></td>n<td valign="bottom"> </></td>n<td valign="bottom" align="right"><font style="font-family:times new roman“…

Week 14 Topic: List of stop words

```
> length(stopwords("english")) [1] 174
> stopwords("en")
[1] "i" "me" "my" "myself" "we" "our"
[7] "ours" "ourselves" "you" "your" "yours" "yourself"
[13] "yourselves" "he" "him" "his" "himself" "she"
[19] "her" "hers" "herself" "it" "its" "itself"
[25] "they" "them" "their" "theirs" "themselves" "what"
[31] "which" "who" "whom" "this" "that" "these"
[37] "those" "am" "is" "are" "was" "were"
[43] "be" "been" "being" "have" "has" "had"
[49] "having" "do" "does" "did" "doing" "would"
[55] "should" "could" "ought" "i'm" "you're" "he's"
[61] "she's" "it's" "we're" "they're" "i've" "you've"
[67] "we've" "they've" "i'd" "you'd" "he'd" "she'd"
[73] "we'd" "they'd" "i'll" "you'll" "he'll" "she'll"
[79] "we'll" "they'll" "isn't" "aren't" "wasn't" "weren't"
[85] "hasn't" "haven't" "hadn't" "doesn't" "don't" "didn't"
[91] "won't" "wouldn't" "shan't" "shouldn't" "can't" "cannot"
```

Week 14 Topic:
List of stop words (cont.)

[97] "couldn't" "mustn't" "let's" "that's" "who's" "what's"
[103] "here's" "there's" "when's" "where's" "why's" "how's"
[109] "a" "an" "the" "and" "but" "if"
[115] "or" "because" "as" "until" "while" "of"
[121] "at" "by" "for" "with" "about" "against"
[127] "between" "into" "through" "during" "before" "after"
[133] "above" "below" "to" "from" "up" "down"
[139] "in" "out" "on" "off" "over" "under"
[145] "again" "further" "then" "once" "here" "there"
[151] "when" "where" "why" "how" "all" "any"
[157] "both" "each" "few" "more" "most" "other"
[163] "some" "such" "no" "nor" "not" "only"
[169] "own" "same" "so" "than" "too" "very"

Week 14 Topic: Transforming the QA Corpus - B

Replacing certin expressions with spaces:

```
> toSpace <- content_transformer(function(x, pattern) gsub(pattern, " ",x))
> qa_sections <- tm_map(qa_sections, toSpace, "/|<|>|”|=|@|\\|:|;|-|\"")

> viewDocs(qa_sections, 1)
td valign "top" p style "margin left:.em text indent: .em" font style "font family:times new
roman" size "" href "#toc_" questions answers proxy materials annual meeting
ofnshareholders font p td n td valign "bottom" font size ""     font td n td valign
"bottom" align "right" font style "font family:times new roman" size "" font td tr n tr n td
valign "top" p style "margin left:.em text indent: .em" font style "font family:times new
roman" size "" href "#toc_" proposal  .  election directors font p td n td valign
"bottom" font size ""     font td n td valign "bottom" align "right" font style "font
family:times new roman" size "" font td tr n tr n td valign "top" p style "margin left:.em text
indent: .em" font style "font family:times new roman" size "" href "#toc_" voting standard font
p td n td valign "bottom" font size ""     font td n td valign "bottom" align "right"
font style "font family:times new roman" size "" font td tr n tr n td valign "top" p style
"margin left:.em text indent: .em" font style "font family:times new roman" size "" href …
```

Week 14 Topic: Transforming the QA Corpus – C

Remove Punctuations:

> qa_sections <- tm_map(qa_sections, removePunctuation)

*Punctuation characters: ! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~.

Strip white spaces:

> qa_sections <- tm_map(qa_sections, stripWhitespace)

> viewDocs(qa_sections, 1)

td valign top p style margin leftem text indent em font style font familytimes new roman size a href toc questions and answers about the proxy materials and our annual meeting ofnshareholders a font p td n td valign bottom font size nbsp nbsp font td n td valign bottom align right font style font familytimes new roman size font td tr n tr n td valign top p style margin leftem text indent em font style font familytimes new roman size a href toc proposalnbsp nonbsp election of directors a font p td n td valign bottom font size nbsp nbsp font td n td valign bottom align right font style font familytimes new roman size font td tr n tr n td valign top p style margin leftem text indent em font style font familytimes new roman size a href toc voting standard a font p td n td valign bottom font size nbsp nbsp font td n td…

Week 14 Topic: Transforming the QA Corpus - D

Remove known often words:

```
> qa_sections <- tm_map(qa_sections, removeWords, c("b", "q", "a", "i", "e", "font",
"style","n","trim","size","font", "can", "also", "e", "mail", "via", "td", "align","border",
"familytimes", "roman", "p", "tr", "nbsp", "with", "table", "cellspacing", "valign", "cellpadding",
"width", "top", "left", "sizepx", "telephone", "if", "may", "help", "us", "will", "please", "unless",
"visit", "thnbsp","toppx","bottompx", "nnn", "address", "nonbsp", "new", "bottom", "em"))
```

* *We remove more words after terms are identified in the matrix later…*

```
> viewDocs(qa_sections, 1)
```

margin leftem text indent href toc questions and answers about the proxy materials and our annual meeting ofnshareholders right margin leftem text indent href toc proposalnbsp election of directors right margin leftem text indent href toc voting standard right margin leftem text indent href toc nominees right margin leftem text indent href toc corporate governance right margin leftem text indent href toc director compensation right margin leftem text indent href toc proposalnbsp approval of amendments to the directors stock plan nn right margin leftem text indent href toc summary description directors plan right margin leftem text indent href toc aggregate past grants under directors plan right margin leftem text indent

Week 14 Topic: Transforming the QA Corpus - E

Specific Transformation:

```
> toString <- content_transformer(function(x, from, to) gsub(from, to, x))
> qa_sections <- tm_map(qa_sections, toString, "broker bank", "bb")
```

* *We will do more specific transformations after terms are identified in the matrix later…*

Week 14 Topic: Transforming the QA Corpus - F

Stemming:

> install.packages("SnowballC")

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/SnowballC_0.5.1.zip' Content type 'application/zip' length 3076512 bytes (2.9 MB) downloaded 2.9 MB package ‘SnowballC’ successfully unpacked and MD5 sums checked The downloaded binary packages are in C:\Users\sshin\AppData\Local\Temp\RtmpQBcYVv\downloaded_packages

> qa_sections <-tm_map(qa_sections, stemDocument)

**Stemming uses an algorithm that removes common word endings for English words, such as “es”, “ed” and “s”.*

> viewDocs(qa_sections, 1)

margin leftem text indent href toc question and answer about the proxi materi and our annual meet ofnsharehold right margin leftem text indent href toc proposalnbsp elect of director right margin leftem text indent href toc vote standard right margin leftem text indent href toc nomine right margin leftem text indent href toc corpor govern right margin leftem text indent href toc director compens right margin leftem text indent href toc proposalnbsp approv of amend to the director stock plan nn right margin leftem text indent href toc summari descript director plan …

Week 14 Topic: Create a TermDocumentMatrix

> qa_sections <-tm_map(qa_sections, PlainTextDocument)

**We do this because the latest version of tm has a "changes in tm 0.6.0 seems to have broken it. The problem is that the functions tolower and trim won't necessarily return TextDocuments (it looks like the older version may have automatically done the conversion). They instead return characters and the DocumentTermMatrix isn't sure how to handle a corpus of characters." from* http://stackoverflow.com/questions/24191728/documenttermmatrix-error-on-corpus-argument

```
> qa_dtm <- DocumentTermMatrix(qa_sections)

>qa_dtm
<<DocumentTermMatrix (documents: 5, terms: 965)>>
Non-/sparse entries: 2710/2115
Sparsity           : 44%
Maximal term length: 25
Weighting          : term frequency (tf)
```

Week 14 Topic: Inspecting the TermDocumentMatrix - A

```
> inspect(qa_dtm[1:5, 100:105])
<<DocumentTermMatrix (documents: 5, terms: 6)>>
Non-/sparse entries: 20/10
Sparsity           : 33%
Maximal term length: 9
Weighting          : term frequency (tf)

              Terms
Docs           basi bear bearingna becaus becom begin
  character(0)    0    2        1      0     1     2
  character(0)    0    2        1      0     1     2
  character(0)    0    2        1      1     1     2
  character(0)    2    4        0      0     2     3
  character(0)    0    2        0      0     1     2
```

Week 14 Topic: Inspecting the TermDocumentMatrix - B

```
> freq <- colSums((as.matrix(qa_dtm)))
> length(freq)
[1] 965
> ord <- order(freq)

> freq[head(ord)]
   abovenaddress   absentninstruct   acceler   additionnsharehold   affirmativenvot   aggreg
               1                 1         1                    1                 1        1
> freq[tail(ord)]
      annual collapsecollaps      sizept      meet      vote   margin
         428          440         478         493       595     1549
> head(table(freq), 15)
freq
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  16
326 108  54  54  55  65  21  11  17  18  13  15  13   6   6
> tail(table(freq), 15)
freq
 149  172  240  246  271  273  315  329  413  428  440  478  493  595 1549
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
```

Week 14 Topic: Conversion to Matrix and Save to CSV

```
> m<-as.matrix(qa_dtm)
> dim(m)
[1] 5 965

> write.csv(m,file= "C:/Users/sshin/Desktop/Yahoo/qa_section/qa_dtm.csv")
```

Week 14 Topic:
Week 13 Assignment – due today!

PART 1: Document/Table Parsing (Automation) & Trading Activity Analysis

- This is automation of Week 11 DEF 14A (Proxy) Filling – Executive Compensation Trend analysis. In essence, you are to automate Week 11 assignment using code. Using XML's readHTMLTable, RCurl, or other available functions in R.
- Make sure to collect all available DEF 14A filings available for the companies. Make sure to chart the salary, option awards, and/or stock awards over the available years for the company.
- Highlight any stock trading activity changes before and after filing.

Week 14 Topic: Text Mining Definitions

- **Corpus**

A collection of documents is called a *corpus*.

- **Tokens, Separators, and Terms**

A document consists of a set of tokens. A *token* is a contiguous string of characters that does not contain a separator. A *separator* is a special character such as a blank or mark of punctuation. A *term* is a token or a sequence of tokens (such as *White House*) with a specific meaning in a given language.

Week 14 Topic: Text Extraction by Increasing Complexity

1. Token extraction
2. Term extraction (token + language ⇒ term)
3. Concept extraction (nouns, noun phrases)
4. Entity extraction (associates nouns with entities – for example, Person: Mr. White, Location: White House)
5. Atomic fact extraction (associates nouns with verbs, that is, subject ⇒ action – for example, terrorist ⇒ bombed)
6. Complex fact extraction (natural language understanding)

Week 14 Topic: Contents of a Document

- A document consists of the following elements:
  - letters
  - words
  - sentences
  - paragraphs
  - punctuation
  - possible structural items (chapters, sections)

- The elements of a document can be counted and compared across documents.

Week 14 Topic: Zipf's Law

Let $t_1, t_2, \ldots, t_n$ be the terms in a document collection arranged in order from most frequent to least frequent.

Let $f_1, f_2, \ldots, f_n$ be the corresponding frequencies of the terms. The frequency f_k for term t_k is proportional to $^1/_k$.

- Zipf's law and its variants help quantify the importance of terms in a document collection. (Konchady 2006)
- "The product of the frequency of words (f) and their rank (r) is approximately constant."
- In practice, Zipf's Law is derived as a Power Law, with free parameters that can be estimated based on the document collection.
- The general formula is shown here: $f_k = C/(\omega + k)^\theta$
- where C is a constant such that, for given ω and θ, $\sum_{k=1}^{n} f_k = T$, the total number of words in the document collection. The parameters ω and θ are estimated for a given document collection.
- Application of Zipf's Law permits identification of important terms for purposes such as describing concepts or topics. You can see the results of Zipf's Law in text mining applications (for example, in the list of terms used to define a topic). Along with methods such as Hidden Markov Models (HMM), the implementation is often hidden from the user. Only the results of the methodology are visible.

Week 14 Topic: Zipf's Law Relevance to Text Mining

- Often, a few, very frequent terms are not good discriminators.
  - *stop* words, for example, the, and, an, or, of
  - often words that are described in linguistics as *closed-class* words, which is a grammatical class that does not get new members
- Typically, there is the following in a document collection:
  - a high number of infrequent terms
  - an average number of average frequency terms
  - a low number of high frequency terms
- Terms that are neither high nor low frequency are the most informative.

Week 14 Topic:
7 Text Mining Practices

http://datamininglab.com/images/pdfs/PracticalTextMining_Excerpt.pdf:

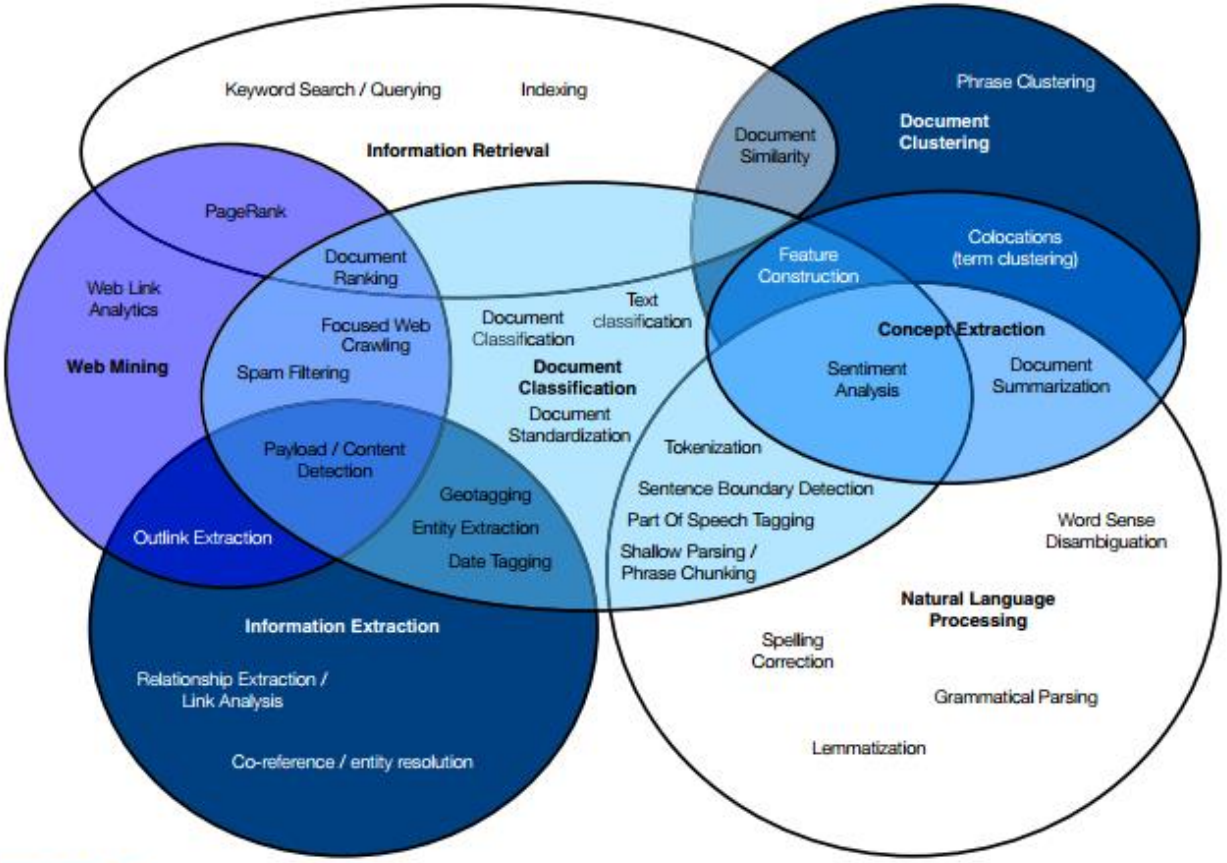


FIGURE 2.3
Visualizing the seven text mining practice areas (ovals) and how specific text mining tasks (labels within ovals) exist at their intersections.

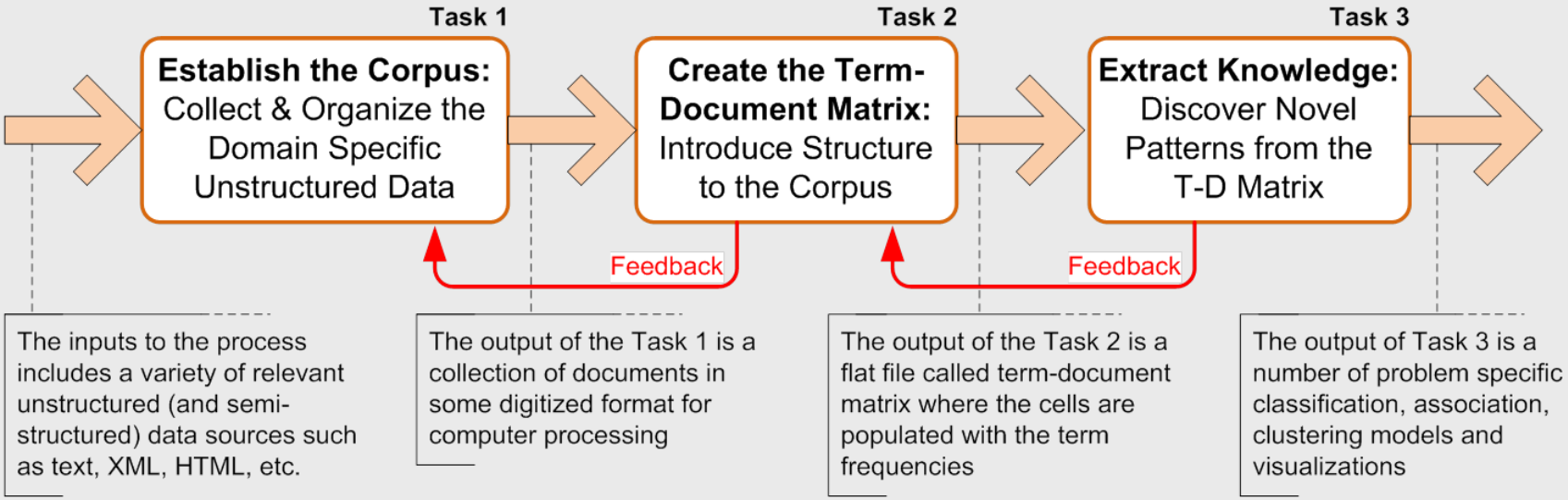
Week 14 Topic: Key Event Indicator (KEI)

For the final project, we define KEIs as indicators that can be used as alerts or notifications of unusual events that may prompt an investigation for action. For example:

- Unusual amount of Questions and Answers discussions. Defined by over a certain percentage from the average character count of the section.
- Unique and/or low frequency terms in the Questions and Answer section with inferred factual context.
- Entity references to executive position changes that may (with some inference) affect company stock prices.
- Entity references to company structure changes that may (with some inference) affect company stock prices.
- Etc.

Week 14 Topic:
Three Step Text Mining Process

www.washburn.edu/faculty/boncella/PPT/CH05.ppt:



The three-step text mining process

Week 14 Topic: Install tm

> library(tm)

Loading required package: NLP Warning messages: 1: package 'tm' was built under R version 3.2.3 2: package 'NLP' was built under R version 3.2.3

> getSources()

[1] "DataframeSource" "DirSource" "URISource" "VectorSource" "XMLSource" [6] "ZipSource"

> getReaders()

[1] "readDOC" "readPDF" "readPlain" [4] "readRCV1" "readRCV1asPlain" "readReut21578XML" [7] "readReut21578XMLasPlain" "readTabular" "readTagged" [10] "readXML"

Week 14 Topic:
Set Up – Document Repository

```
> cname <- "C:/Users/sshin/Desktop/Yahoo"

> cname
[1] "C:/Users/sshin/Desktop/Yahoo"

> length(dir(cname))
[1] 6

> dir(cname)
[1] "0001193125-13-187918.txt" "0001193125-14-172132.txt" "0001193125-15-156926.txt" [4] "yahoo_proxy2013.html" "yahoo_proxy2014.html" "yahoo_proxy2015.html"
```

Week 14 Topic: Create a Corpus

```
> proxies <- Corpus(DirSource(cname))

> proxies
<<VCorpus>> Metadata: corpus specific: 0, document level (indexed): 0 Content: documents: 6

> class(proxies)
[1] "VCorpus" "Corpus"

> class(proxies[[1]])
[1] "PlainTextDocument" "TextDocument"

> summary(proxies)
Length Class Mode 0001193125-13-187918.txt 2 PlainTextDocument list 0001193125-14-172132.txt 2 PlainTextDocument list 0001193125-15-156926.txt 2 PlainTextDocument list yahoo_proxy2013.html 2 PlainTextDocument list yahoo_proxy2014.html 2 PlainTextDocument list yahoo_proxy2015.html 2 PlainTextDocument list
```

Week 14 Topic: Installing stringi

Install:

> install.packages("stringi")

Open library:

>library(stringi)

Getting help:

>library(help=stringi)

```
stri_extract_all          Extract Occurrences of a Pattern
stri_extract_all_boundaries
                          Extract Text Between Text Boundaries
stri_flatten              Flatten a String
stri_info                 Query Default Settings for 'stringi'
stri_install_check        Installation-Related Utilities [DEPRECATED]
stri_isempty              Determine if a String is of Length Zero
stri_join                 Concatenate Character Vectors
stri_length               Count the Number of Code Points
stri_list2matrix          Convert a List to a Character Matrix
stri_locale_info          Query Given Locale
stri_locale_list          List Available Locales
stri_locale_set           Set or Get Default Locale in 'stringi'
stri_locate_all           Locate Occurrences of a Pattern
stri_locate_all_boundaries
                          Locate Specific Text Boundaries
stri_match_all            Extract Regex Pattern Matches, Together with
                          Capture Groups
stri_numbytes             Count the Number of Bytes
stri_opts_brkiter         Generate a List with BreakIterator Settings
stri_opts_collator        Generate a List with Collator Settings
stri_opts_fixed           Generate a List with Fixed Pattern Search
                          Engine's Settings
stri_opts_regex           Generate a List with Regex Matcher Settings
stri_order                Ordering Permutation and Sorting
stri_pad_both             Pad (Center/Left/Right Align) a String
stri_rand_lipsum          A Lorem Ipsum Generator
stri_rand_shuffle         Randomly Shuffle Code Points in Each String
stri_rand_strings         Generate Random Strings
stri_read_lines           [DRAFT API] Read Text Lines from a Text File
stri_read_raw             [DRAFT API] Read Whole Text File as Raw
stri_replace_all          Replace Occurrences of a Pattern
stri_replace_na           Replace Missing Values in a Character Vector
stri_reverse              Reverse Each String
stri_split                Split a String By Pattern Matches
stri_split_boundaries     Split a String at Specific Text Boundaries
stri_split_lines          Split a String Into Text Lines
```

Week 14 Topic: Locating text boundaries

http://finzi.psych.upenn.edu/library/stringi/html/stri_locate_boundaries.html:

Description

These functions locate specific text boundaries (like character, word, line, or sentence boundaries). stri_locate_all_* locate all the matches. On the other hand, stri_locate_first_* andstri_locate_last_* give the first or the last matches, respectively.

Usage

- stri_locate_all_boundaries(str, omit_no_match = FALSE, ..., opts_brkiter = NULL)
- stri_locate_last_boundaries(str, ..., opts_brkiter = NULL)
- stri_locate_first_boundaries(str, ..., opts_brkiter = NULL)
- stri_locate_all_words(str, omit_no_match = FALSE, locale = NULL)
- stri_locate_last_words(str, locale = NULL)
- stri_locate_first_words(str, locale = NULL)

Week 13 Topic: Yahoo DEF 14A filings

- Yahoo has 18 filings from 1997 to 2015.

EDGAR Search Results

https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK=0001011006&type=def+14a&dateb=&owner=exclude&count=40

Home | Latest Filings | Previous Page

U.S. Securities and Exchange Commission

EDGAR Search Results

EDGAR Search Results BETA View

SEC Home » Search the Next-Generation EDGAR System » Company Search » *Current Page*

YAHOO INC CIK#: 0001011006 (see all company filings)
SIC: 7373 - SERVICES-COMPUTER INTEGRATED SYSTEMS DESIGN
State location: CA | State of Inc.: **DE** | Fiscal Year End: 1231
(Assistant Director Office: 3)
Note: Ownership filings are available at this link. Reports containing extracted ownership data are temporarily unavailable.

Business Address
YAHOO! INC.
701 FIRST AVENUE
SUNNYVALE CA 94089
4083493300

Mailing Address
701 FIRST AVENUE
SUNNYVALE CA 94089

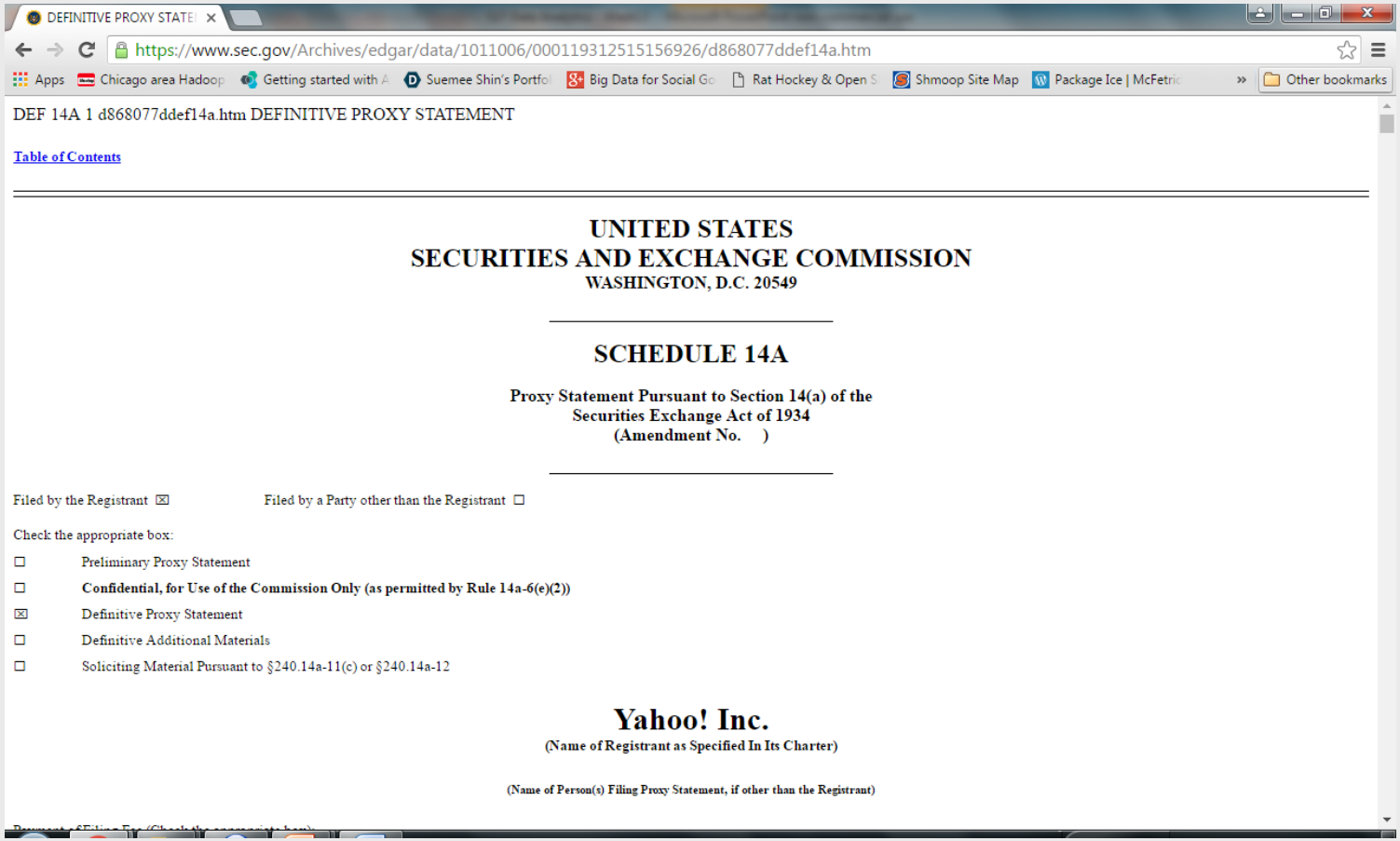
Filter Results: Filing Type: def 14a | Prior to: (YYYYMMDD) | Ownership? include exclude only | Limit Results Per Page 40 Entries | Search | Show All

Items 1 - 18 RSS Feed

Filings	Format	Description	Filing Date	File/Film Number
DEF 14A	Documents	Other definitive proxy statements Acc-no: 0001193125-15-156926 (34 Act) Size: 2 MB	2015-04-29	000-28018 15813744
DEF 14A	Documents	Other definitive proxy statements Acc-no: 0001193125-14-172132 (34 Act) Size: 2 MB	2014-04-30	000-28018 14798942
DEF 14A	Documents	Other definitive proxy statements Acc-no: 0001193125-13-187918 (34 Act) Size: 1 MB	2013-04-30	000-28018 13799046
DEF 14A	Documents	Other definitive proxy statements Acc-no: 0001193125-12-258870 (34 Act) Size: 1 MB	2012-06-04	000-28018 12886969
DEF 14A	Documents	Other definitive proxy statements Acc-no: 0001193125-11-118858 (34 Act) Size: 1 MB	2011-04-29	000-28018 11795903
DEF 14A	Documents	Other definitive proxy statements Acc-no: 0001193125-10-099552 (34 Act) Size: 1 MB	2010-04-29	000-28018 10782547
DEF 14A	Documents	Other definitive proxy statements Acc-no: 0001193125-09-092231 (34 Act) Size: 1 MB	2009-04-29	000-28018 09780142
DEF 14A	Documents	Other definitive proxy statements Acc-no: 0000891618-07-000262 (34 Act) Size: 1 MB	2007-04-30	000-28018 07802049
DEF 14A	Documents	Other definitive proxy statements	2006-04-14	000-28018

Week 13 Topic:
Yahoo DEF 14A - html version

https://www.sec.gov/Archives/edgar/data/1011006/000119312515156926/d868077ddef14a.htm:



DEF 14A 1 d868077ddef14a.htm DEFINITIVE PROXY STATEMENT

Table of Contents

UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
WASHINGTON, D.C. 20549

SCHEDULE 14A

Proxy Statement Pursuant to Section 14(a) of the Securities Exchange Act of 1934
(Amendment No.)

Filed by the Registrant ☒ Filed by a Party other than the Registrant ☐

Check the appropriate box:

☐ Preliminary Proxy Statement
☐ **Confidential, for Use of the Commission Only (as permitted by Rule 14a-6(e)(2))**
☒ Definitive Proxy Statement
☐ Definitive Additional Materials
☐ Soliciting Material Pursuant to §240.14a-11(c) or §240.14a-12

Yahoo! Inc.
(Name of Registrant as Specified In Its Charter)

(Name of Person(s) Filing Proxy Statement, if other than the Registrant)

Week 13 Topic:
Yahoo DEF 14A - txt version

https://www.sec.gov/Archives/edgar/data/1011006/000119312515156926/0001193125-15-156926.txt:

https://www.sec.gov/Archives/edgar/data/1011006/000119312515156926/0001193125-15-156926.txt

```
<SEC-DOCUMENT>0001193125-15-156926.txt : 20150429
<SEC-HEADER>0001193125-15-156926.hdr.sgml : 20150429
<ACCEPTANCE-DATETIME>20150429170414
ACCESSION NUMBER:               0001193125-15-156926
CONFORMED SUBMISSION TYPE:      DEF 14A
PUBLIC DOCUMENT COUNT:          25
CONFORMED PERIOD OF REPORT:     20150624
FILED AS OF DATE:               20150429
DATE AS OF CHANGE:              20150429
EFFECTIVENESS DATE:             20150429

FILER:

        COMPANY DATA:
                COMPANY CONFORMED NAME:                 YAHOO INC
                CENTRAL INDEX KEY:                      0001011006
                STANDARD INDUSTRIAL CLASSIFICATION:     SERVICES-COMPUTER INTEGRATED SYSTEMS DESIGN [7373]
                IRS NUMBER:                             770398689
                STATE OF INCORPORATION:                 DE
                FISCAL YEAR END:                        1231

        FILING VALUES:
                FORM TYPE:              DEF 14A
                SEC ACT:                1934 Act
                SEC FILE NUMBER:        000-28018
                FILM NUMBER:            15813744

        BUSINESS ADDRESS:
                STREET 1:               YAHOO! INC.
                STREET 2:               701 FIRST AVENUE
                CITY:                   SUNNYVALE
                STATE:                  CA
                ZIP:                    94089
                BUSINESS PHONE:         4083493300

        MAIL ADDRESS:
                STREET 1:               701 FIRST AVENUE
                CITY:                   SUNNYVALE
                STATE:                  CA
                ZIP:                    94089
</SEC-HEADER>
<DOCUMENT>
<TYPE>DEF 14A
<SEQUENCE>1
```

Week 13 Topic: Yahoo's Executive Compensation Table

Target Table in HTML View:

Summary Compensation Table—2012–2014

The following table presents 2012–2014 summary compensation information for our Named Executive Officers. As required by SEC rules, stock awards (RSUs) and option awards are shown as compensation for the year in which they were granted (even if they have multi-year vesting schedules), and are valued based on their grant date fair values for accounting purposes. Accordingly, the table includes stock and option awards granted in the years shown even if they were scheduled to vest in later years, and even if they were subsequently forfeited (such as upon the executive's termination). Therefore, the stock and option columns do *not* report whether the officer realized a financial benefit from the awards (such as by vesting in stock or exercising options).

Name and Principal Position	Year	Salary ($)(1)	Bonus ($)(1)	Stock Awards ($)(2)(3)(4)	Option Awards ($)(2)(3)	Non-Equity Incentive Plan Compensation ($)(5)	All Other Compensation ($)(6)	Total ($)(2)
Marissa A. Mayer	2014	1,000,000	0	11,752,355(7)	28,194,288(7)	1,108,800	28,065	42,083,508
Chief Executive Officer	2013	1,000,000	2,250	8,312,316	13,847,283	1,700,000	73,863	24,935,712
	2012	454,862	0	35,000,002	0	1,120,000	40,540	36,615,404
Ken Goldman	2014	600,000	0	2,813,080	9,327,427	300,000	4,549	13,045,056
Chief Financial Officer	2013	600,000	0	2,597,612	2,290,527	500,000	4,615	5,992,754
	2012	116,667	100,000	7,262,357	0	0	29	7,479,053
David Filo	2014	1	0	0	0	0	0	1
Co-Founder and Chief Yahoo	2013	1	0	0	0	0	0	1
	2012	1	0	0	0	0	0	1
Ronald S. Bell	2014	600,000	0	3,282,107	0	300,000	4,549	4,186,656
General Counsel	2013	600,000	0	3,896,386	0	450,000	4,615	4,951,001
	2012	442,763	206,800	558,300	0	443,200	4,424	1,655,487
Henrique de Castro(8)	2014	27,083	0	0	0	0	1,177,157	1,204,240
Former Chief Operating Officer	2013	600,000	0	0	10,307,359	0	37,001	10,944,360
	2012	84,092	1,100,000	37,999,991	0	0	29	39,184,112

(1) Salary and bonus columns include amounts earned in, or awarded for performance during, the specified year (even if paid out early in the following year).

(2) As required by SEC rules, the stock and option award columns present the aggregate grant date fair value of equity awards granted during the years shown as computed for accounting purposes in accordance with FASB ASC 718. As a result, the stock and option columns (as well as the total column) include awards that have not yet vested, awards that were granted but later forfeited (such as upon the executive's termination), and performance-based awards that failed to vest; therefore, these columns are *not* intended as presentations of pay actually realized by the executive. For information on the assumptions used in the grant date fair value computations, refer to Note 14—"Employee Benefits" in the Notes to Consolidated Financial Statements in our 2014 Form 10-K.

71

Week 13 Topic: Yahoo's Executive Compensation Table

Target Table in Source Code View:

```
6002
6003 <div style="width:97%; margin-top:1.5%; margin-left:1.5%; margin-right:-1.25%">
6004  <P STYLE="margin-top:0pt; margin-bottom:0pt; font-size:9pt; font-family:arial" ALIGN="right">EXECUTIVE COMPENSATION </P>
6005  <p STYLE="margin-top:0pt;margin-bottom:0pt ; font-size:8pt"> </P></div>
6006  <p STYLE="margin-top:0pt;margin-bottom:0pt ; font-size:8pt"> 
6007 </P> <P STYLE="margin-top:0pt; margin-bottom:0pt; font-size:16pt; font-family:arial"><FONT COLOR="#7300ff"><B><A NAME="toc868077_29"></A>COMPENSATION TABLES </B></FONT>
     </P> <P STYLE="font-size:6pt;margin-top:0pt;margin-bottom:0pt"> </P>
6008 <P STYLE="line-height:1.0pt;margin-top:0pt;margin-bottom:2pt;border-bottom:1.00pt solid #000000"> </P> <P STYLE="margin-top:12pt; margin-bottom:0pt; text-indent:6%;
     font-size:10pt; font-family:arial" ALIGN="justify">The tables on the following
6009 pages present compensation information regarding our Chief Executive Officer, Marissa A. Mayer; our Chief Financial Officer, Ken Goldman; our co-founder and Chief Yahoo,
     David Filo; and our General Counsel, Ronald S. Bell. As required by SEC rules,
6010 the tables also include our former Chief Operating Officer, Henrique de Castro, whose service ended during 2014. These five individuals are our &#147;Named Executive
     Officers.&#148; We did not have any other executive officers in 2014. </P>
6011 <P STYLE="margin-top:12pt; margin-bottom:0pt; text-indent:6%; font-size:10pt; font-family:arial" ALIGN="justify">As required by SEC rules, in these tables performance-
     based awards are treated as having been granted in the year in which their
6012 performance goals were established (and if an award has multiple performance periods, the portion relating to each period is treated as a separate grant). </P> <P
     STYLE="margin-top:18pt; margin-bottom:0pt; font-size:16pt; font-family:arial"><FONT
6013  COLOR="#7300ff"><B>Summary Compensation Table&#151;2012&#150;2014 </B></FONT></P> <P STYLE="font-size:6pt;margin-top:0pt;margin-bottom:0pt"> </P>
6014 <P STYLE="line-height:1.0pt;margin-top:0pt;margin-bottom:2pt;border-bottom:1.00pt solid #000000"> </P> <P STYLE="margin-top:12pt; margin-bottom:0pt; text-indent:6%;
     font-size:10pt; font-family:arial" ALIGN="justify"><I></I>The following table
6015 presents 2012&#150;2014 summary compensation information for our Named Executive Officers. As required by SEC rules, stock awards (RSUs) and option awards are shown as
     compensation for the year in which they were granted (even if they have
6016 multi-year vesting schedules), and are valued based on their grant date fair values for accounting purposes. Accordingly, the table includes stock and option awards
     granted in the years shown even if they were scheduled to vest in later years, and
6017 even if they were subsequently forfeited (such as upon the executive&#146;s termination). Therefore, the stock and option columns do <I>not</I> report whether the officer
     realized a financial benefit from the awards (such as by vesting in stock or
6018 exercising options). <I> </I></P> <P STYLE="font-size:12pt;margin-top:0pt;margin-bottom:0pt"> </P>
6019 <TABLE CELLSPACING="0" CELLPADDING="0" WIDTH="100%" BORDER="0" STYLE="BORDER-COLLAPSE:COLLAPSE; font-family:arial; font-size:8pt" ALIGN="center">
6020
6021
6022 <TR>
6023 <TD WIDTH="36%"></TD>
6024 <TD VALIGN="bottom" WIDTH="1%"></TD>
```

ITM - 527

Week 13 Topic: Extract tables from HTML

Using readHTMLTable:

> install.packages("XML")

>library(XML)

> proxy.yahoo.2015.HTML.tables <- readHTMLTable("C:/Users/Desktop/yahoo_proxy2015.html")

> proxy.yahoo.2015.HTML.page[6003:6020] ← *to view Summary Compensation Table section in HTML*

Week 13 Topic:
Select the Executive Compensation Table

> compensation.table <- proxy.yahoo.2015.HTML.tables[[306]] ← *to view Summary Compensation Table in the array. Table # = 306 for 2015 Yahoo DEF 14A filing*

Filter

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16
1																
2	Name and Principal Position	Â	Year	Â	Â	Salary($)(1)	Â	Â	Bonus($)(1)	Â	Â	StockAwards($)(2)(3)(4)	Â	Â	OptionAwards($)(2)(3)	Â
3	Marissa A. Mayer	Â	Â	2014	Â Â	Â	Â	1,000,000	Â Â	Â	Â	0	Â Â	Â	Â	11,752,355
4	Chief Executive Officer	Â	Â	2013	Â Â	Â	Â	1,000,000	Â Â	Â	Â	2,250	Â Â	Â	Â	8,312,316
5		Â	Â	2012	Â Â	Â	Â	454,862	Â Â	Â	Â	0	Â Â	Â	Â	35,000,002
6	Ken Goldman	Â	Â	2014	Â Â	Â	Â	600,000	Â Â	Â	Â	0	Â Â	Â	Â	2,813,080
7	Chief Financial Officer	Â	Â	2013	Â Â	Â	Â	600,000	Â Â	Â	Â	0	Â Â	Â	Â	2,597,612
8		Â	Â	2012	Â Â	Â	Â	116,667	Â Â	Â	Â	100,000	Â Â	Â	Â	7,262,357
9	David Filo	Â	Â	2014	Â Â	Â	Â	1	Â Â	Â	Â	0	Â Â	Â	Â	0
10	Co-Founder and Chief Yahoo	Â	Â	2013	Â Â	Â	Â	1	Â Â	Â	Â	0	Â Â	Â	Â	0
11		Â	Â	2012	Â Â	Â	Â	1	Â Â	Â	Â	0	Â Â	Â	Â	0
12	Ronald S. Bell	Â	Â	2014	Â Â	Â	Â	600,000	Â Â	Â	Â	0	Â Â	Â	Â	3,282,107

Week 13 Topic:
Sub-select relevant table data, rename

> compensation.table.data <- compensation.table[3:5, c("V1", "V4","V8","V12","V16")]

	V1	V4	V8	V12	V16
3	Marissa A. Mayer	2014	1,000,000	0	11,752,355
4	Chief Executive Officer	2013	1,000,000	2,250	8,312,316
5		2012	454,862	0	35,000,002

> compensation.table.data <-rename(compensation.table.data, c("V1"="Name and Principal Position", "V4"="Year", "V8"="Salary", "V12"="Stock_Awards", "V16"="Option_Awards"))

	Name and Principal Position	Year	Salary	Stock_Awards	Option_Awards
3	Marissa A. Mayer	2014	1,000,000	0	11,752,355
4	Chief Executive Officer	2013	1,000,000	2,250	8,312,316
5		2012	454,862	0	35,000,002

Week 13 Topic: Other methods

Using getURL:

> install.packages("RCurl")

>library(RCurl)

> proxy.yahoo.2015.HTML.page <- getURL("https://www.sec.gov/Archives/edgar/data/1011006/000119312515156926/d868077ddef14a.htm")

…

Using readLines:

> proxy.yahoo.2015.HTML.page <- readLines("https://www.sec.gov/Archives/edgar/data/1011006/000119312515156926/d868077ddef14a.htm")

>length(proxy.yahoo.2015.HTML.page)

[1] 10261

Week 13 Topic: Installing plyr

Install:

> install.packages("plyr")

Open library:

>library(plyr)

Getting help:

>library(help=plyr)

```
m_ply                   Call function with arguments in array or data
                        frame, discarding results.
maply                   Call function with arguments in array or data
                        frame, returning an array.
mapvalues               Replace specified values with new values, in a
                        vector or factor.
match_df                Extract matching rows of a data frame.
mdply                   Call function with arguments in array or data
                        frame, returning a data frame.
mlply                   Call function with arguments in array or data
                        frame, returning a list.
mutate                  Mutate a data frame by adding new or replacing
                        existing columns.
name_rows               Toggle row names between explicit and implicit.
ozone                   Monthly ozone measurements over Central
                        America.
plyr                    plyr: the split-apply-combine paradigm for R.
plyr-deprecated         Deprecated Functions in Package plyr
progress_text           Text progress bar.
progress_time           Text progress bar with time.
progress_tk             Graphical progress bar, powered by Tk.
progress_win            Graphical progress bar, powered by Windows.
r_ply                   Replicate expression and discard results.
raply                   Replicate expression and return results in a
                        array.
rbind.fill              Combine data.frames by row, filling in missing
                        columns.
rbind.fill.matrix       Bind matrices by row, and fill missing columns
                        with NA.
rdply                   Replicate expression and return results in a
                        data frame.
rename                  Modify names by name, not position.
revalue                 Replace specified values with new values, in a
                        factor or character vector.
```

Week 13 Topic: Installing RCurl

Install:

>install.package("RCurl")

Open library:

>library(RCurl)

Getting help:

>library(help=RCurl)

```
Index:

AUTH_ANY                Constants for identifying Authentication
                        Schemes
CFILE                   Create a C-level handle for a file
CURLHandle-class        Class "CURLHandle" for synchronous HTTP
                        requests
CurlFeatureBits         Constants for libcurl
HTTP_VERSION_1_0        Symbolic constants for specifying HTTP and SSL
                        versions in libcurl
MultiCURLHandle-class   Class "MultiCURLHandle" for asynchronous,
                        concurrent HTTP requests
base64                  Encode/Decode base64 content
basicHeaderGatherer     Functions for processing the response header of
                        a libcurl request
basicTextGatherer       Cumulate text across callbacks (from an HTTP
                        response)
binaryBuffer            Create internal C-level data structure for
                        collecting binary data
chunkToLineReader       Utility that collects data from the HTTP reply
                        into lines and calls user-provided function.
clone                   Clone/duplicate an object
coerce,numeric,NetrcEnum-method
                        Internal functions
complete                Complete an asynchronous HTTP request
curlError               Raise a warning or error about a CURL problem
curlEscape              Handle characters in URL that need to be
                        escaped
curlGlobalInit          Start and stop the Curl library
curlOptions             Constructor and accessors for CURLOptions
                        objects
curlPerform             Perform the HTTP query
curlSetOpt              Set values for the CURL options
curlVersion             Information describing the Curl library
dynCurlReader           Dynamically determine content-type of body from
                        HTTP header and set body reader
```

Week 13 Topic: Installing RCurl (cont.)

Install:

>install.package("RCurl")

Open library:

>library(RCurl)

Getting help:

>library(help=RCurl)

```
fileUpload                  Specify information about a file to upload in
                            an HTTP request
findHTTPHeaderEncoding
                            Find the encoding of the HTTP response from the
                            HTTP header
ftpUpload                   Upload content via FTP
getBinaryURL                Download binary content
getBitIndicators            Operate on bit fields
getCurlErrorClassNames
                            Retrieve names of all curl error classes
getCurlHandle               Create libcurl handles
getCurlInfo                 Access information about a CURL request
getFormParams               Extract parameters from a form query string
getURIAsynchronous          Download multiple URIs concurrently, with
                            inter-leaved downloads
getURL                      Download a URI
guessMIMEType               Infer the MIME type from a file name
httpPUT                     Simple high-level functions for HTTP PUT and
                            DELETE
merge.list                  Method for merging two lists by name
mimeTypeExtensions          Mapping from extension to MIME type
postForm                    Submit an HTML form
reset                       Generic function for resetting an object
scp                         Retrieve contents of a file from a remote host
                            via SCP (Secure Copy)
url.exists                  Check if URL exists
|,BitwiseValue,BitwiseValue-method
                            Classes and coercion methods for enumerations
                            in libcurl
```

Week 13 Topic: Installing stringr

Install:

>*install.package("stringr")*

Open library:

>*library(stringr)*

Getting help:

>*library(help=stringr)*

```
Index:

case                    Convert case of a string.
invert_match            Switch location of matches to location of
                        non-matches.
modifiers               Control matching behaviour with modifier
                        functions.
str_c                   Join multiple strings into a single string.
str_conv                Specify the encoding of a string.
str_count               Count the number of matches in a string.
str_detect              Detect the presence or absence of a pattern in
                        a string.
str_dup                 Duplicate and concatenate strings within a
                        character vector.
str_extract             Extract matching patterns from a string.
str_length              The length of a string.
str_locate              Locate the position of patterns in a string.
str_match               Extract matched groups from a string.
str_order               Order or sort a character vector.
str_pad                 Pad a string.
str_replace             Replace matched patterns in a string.
str_replace_na          Turn NA into "NA"
str_split               Split up a string into pieces.
str_sub                 Extract and replace substrings from a character
                        vector.
str_subset              Keep strings matching a pattern.
str_trim                Trim whitespace from start and end of string.
str_wrap                Wrap strings into nicely formatted paragraphs.
stringr                 Fast and friendly string manipulation.
word                    Extract words from a sentence.

Further information is available in the following vignettes in directory
'C:/Users/sshin/Documents/R/R-3.2.2/library/stringr/doc':
```

Week 13 Topic: Installing XML

Install:

>install.package("XML")

Open library:

>library(XML)

Getting help:

>library(help=XML)

```
Index:

Doctype                     Constructor for DTD reference
Doctype-class               Class to describe a reference to an XML DTD
ExternalReference-class
                            Classes for working with XML Schema
SAXState-class              A virtual base class defining methods for SAX
                            parsing
XMLAttributes-class         Class '"XMLAttributes"'
XMLCodeFile-class           Simple classes for identifying an XML document
                            containing R code
XMLInternalDocument-class
                            Class to represent reference to C-level data
                            structure for an XML document
XMLNode-class               Classes to describe an XML node object.
[.XMLNode                   Convenience accessors for the children of
                            XMLNode objects.
[<-.XMLNode                 Assign sub-nodes to an XML node
addChildren                 Add child nodes to an XML node
addNode                     Add a node to a tree
append.xmlNode              Add children to an XML node
asXMLNode                   Converts non-XML node objects to XMLTextNode
                            objects
asXMLTreeNode               Convert a regular XML node to one for use in a
                            "flat" tree
catalogLoad                 Manipulate XML catalog contents
catalogResolve              Look up an element via the XML catalog
                            mechanism
coerce,XMLHashTreeNode,XMLHashTree-method
                            Transform between XML representations
compareXMLDocs              Indicate differences between two XML documents
docName                     Accessors for name of XML document
dtdElement                  Gets the definition of an element or entity
                            from a DTD.
```

Week 13 Topic: Installing XML (cont.)

Install:

>install.package("XML")

Open library:

>library(XML)

Getting help:

>library(help=XML)

getXMLErrors	Get XML/HTML document parse errors
isXMLString	Facilities for working with XML strings
length.XMLNode	Determine the number of children in an XMLNode object.
libxmlVersion	Query the version and available features of the libxml library.
makeClassTemplate	Create S4 class definition based on XML node(s)
names.XMLNode	Get the names of an XML nodes children.
newXMLDoc	Create internal XML node or document object
newXMLNamespace	Add a namespace definition to an XML node
parseDTD	Read a Document Type Definition (DTD)
parseURI	Parse a URI string into its elements
parseXMLAndAdd	Parse XML content and add it to a node
print.XMLAttributeDef	Methods for displaying XML objects
processXInclude	Perform the XInclude substitutions
readHTMLList	Read data in an HTML list or all lists in a document
readHTMLTable	Read data from one or more HTML tables
readKeyValueDB	Read an XML property-list style document
readSolrDoc	Read the data from a Solr document
removeXMLNamespaces	Remove namespace definitions from a XML node or document
saveXML	Output internal XML Tree
setXMLNamespace	Set the name space on a node
startElement.SAX	Generic Methods for SAX callbacks
supportsExpat	Determines which native XML parsers are being used.
toHTML	Create an HTML representation of the given R object, using internal C-level nodes
toString.XMLNode	Creates string representation of XML node
xmlApply	Applies a function to each of the children of an XMLNode
xmlAttributeType	The type of an XML attribute for element from the DTD

Week 13 Topic: Installing tm

Install:

>install.package("tm")

Open library:

>library(tm)

Getting help:

>library(help=tm)

```
Index:

Corpus                  Corpora
DataframeSource         Data Frame Source
DirSource               Directory Source
Docs                    Access Document IDs and Terms
MC_tokenizer            Tokenizers
PCorpus                 Permanent Corpora
PlainTextDocument       Plain Text Documents
Reader                  Readers
Source                  Sources
TermDocumentMatrix      Term-Document Matrix
TextDocument            Text Documents
URISource               Uniform Resource Identifier Source
VCorpus                 Volatile Corpora
VectorSource            Vector Source
WeightFunction          Weighting Function
XMLSource               XML Source
XMLTextDocument         XML Text Documents
ZipSource               ZIP File Source
Zipf_plot               Explore Corpus Term Frequency Characteristics
acq                     50 Exemplary News Articles from the
                        Reuters-21578 Data Set of Topic acq
c.VCorpus               Combine Corpora, Documents, Term-Document
                        Matrices, and Term Frequency Vectors
content_transformer     Content Transformers
crude                   20 Exemplary News Articles from the
                        Reuters-21578 Data Set of Topic crude
findAssocs              Find Associations in a Term-Document Matrix
findFreqTerms           Find Frequent Terms
getTokenizers           Tokenizers
getTransformations      Transformations
inspect                 Inspect Objects
meta                    Metadata Management
plot.TermDocumentMatrix
```

Week 13 Topic: Installing tm (cont.)

Install:

>install.package("tm")

Open library:

>library(tm)

Getting help:

>library(help=tm)

```
plot.TermDocumentMatrix
                        Visualize a Term-Document Matrix
readDOC                 Read In a MS Word Document
readPDF                 Read In a PDF Document
readPlain               Read In a Text Document
readRCV1                Read In a Reuters Corpus Volume 1 Document
readReut21578XML        Read In a Reuters-21578 XML Document
readTabular             Read In a Text Document
readTagged              Read In a POS-Tagged Word Text Document
readXML                 Read In an XML Document
read_dtm_Blei_et_al     Read Document-Term Matrices
removeNumbers           Remove Numbers from a Text Document
removePunctuation       Remove Punctuation Marks from a Text Document
removeSparseTerms       Remove Sparse Terms from a Term-Document Matrix
removeWords             Remove Words from a Text Document
stemCompletion          Complete Stems
stemDocument            Stem Words
stopwords               Stopwords
stripWhitespace         Strip Whitespace from a Text Document
termFreq                Term Frequency Vector
tm_filter               Filter and Index Functions on Corpora
tm_map                  Transformations on Corpora
tm_reduce               Combine Transformations
tm_term_score           Compute Score for Matching Terms
weightBin               Weight Binary
weightSMART             SMART Weightings
weightTf                Weight by Term Frequency
weightTfIdf             Weight by Term Frequency - Inverse Document
                        Frequency
writeCorpus             Write a Corpus to Disk

Further information is available in the following vignettes in
directory 'C:/Users/sshin/Documents/R/R-3.2.2/library/tm/doc':

extensions: Extensions (source, pdf)
tm: Introduction to the tm Package (source, pdf)
```