1. **Preliminary Phase**:

First step is to load the respective documents (DEF 14A FILINGS) to be extracted and processed in a folder for that corresponding company on a yearly basis.

For instance, we create a folder called datasets/txt.

**cname <- file.path (".", "datasets", "txt")**

We are ready to load files from the directory by using **Dirsource ()**

Following code shows the process: - library(tm)

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

Reading corpus pdf documents by the following code:-

**docs <- Corpus(DirSource(cname), readerControl=list(reader=readPDF))**

Now that loading of the data is done, Next would be to read the datasets.

1. We need to use the readLines() function to read the datasets, or alternatively to read the HTML tags, we can use XML package library.

For Instance : **readLines ( “ex.data”,-n).**

(In the place of ex.data, we should give the url of the link from which we would want the data to be read.)

n : The (maximal) number of lines to read. Negative values indicate that one should read up to the end of input on the connection.

1. Once the datasets is read, we need to scan through the dataset and get the required/correct data using a function caled grep() comand. Grep functions search for matches to argument patterm within each element of a character. With this function, we can find the required table ( Execution Compensation table)

For Instance, the general syntax is

**grep [-options] patterns [filename].**

1. Once the required table is found, we need to use the readHTMLTable() function to read the tables. Once the table is read, we need to remove the noises , to do so we need to clean the data which is done using TM package.
2. **Data Import**: The tm package supports text, pdf, Microsoft word and XML formats, we will either use text formatting or Even the Xml type formatting for this purpose.

Let us assume that we save the files in pdf format and then want to bring them to use in Text format,

We use the following code:

**System ("for f in \*.pdf; do pdftotext -enc ASCII7 -nopgbrk $f; done")**

We can list the sources by function: - getsources(). In addition we can read document in say pdf format by the function: - getreaders() ##”readPDF”

We can (and should) inspect the documents using inspect(). This will assure us that data has been loaded properly and as we expect.

1. **Preprocessing/Transformations**: This step allows us to make sure that there is no discrepancy in our data i.e. Numbers, capitalization, common words, punctuation, cases, removing stop words as they lack any analytic value which is otherwise preparing our texts for high quality text analysis. Normally tm\_map () is the function used to perform this step.

We use the following code:-

getTransformations()

**## [1] "removeNumbers" "removePunctuation" "removeWords"**

**## [4] "stemDocument" "stripWhitespace"**

1. **Creating Term document Matrices and saving to CSV**: A document term matrix is simply a matrix with documents as the rows and terms as the columns and a count of the frequency of words as the cells of the matrix. We use DocumentTermMatrix()

To create the matrix:

**dtm <- DocumentTermMatrix(docs)**

**Convert the dtm matrix into a simple matrix:**

**m <- as.matrix (dtm)**

**Once converted into a standard matrix the usual write.csv () can be used to write the data to file.**

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

1. **Exploring Data** Finally after exporting this data into excel**.** We check for consistency of this data if a particular record is present for two different documents for two consecutive years. We consolidate the data by the same company for its filing as the same CEO. However we make sure that we add the consolidated data for unique rows only. Finally we summarize this data by aggregations or charts in R to get the desired output.