**######################################################################**

**########### R CODE: ProgramTest.docx #################################**

**######################################################################**

**rm(list = ls())**

**## read and clean data**

**data = read.csv('C:\\Users\\ledolter\\Desktop\\test.csv',header=FALSE,stringsAsFactors=F)**

**## Note that we read the data from the directory C:\\Users\\ledolter\\Desktop**

**## You need to change this and read from the appropriate directory**

**dim(data)**

**data[1:4,1]**

**dim(data)[1] ## number of speeches**

**for (i in 1:dim(data)[1]) {**

**txt=data[i,1]**

**txt=tolower(txt)**

**txt=gsub("[.]","", ignore.case = TRUE,txt)**

**txt=gsub("[,]","", ignore.case = TRUE,txt)**

**txt=gsub("[;]","", ignore.case = TRUE,txt)**

**txt=gsub("new york","new-york", ignore.case = TRUE,txt) ## the state of new-york**

**data[i,1]=txt**

**}**

**data[1:4,1]**

**## omit meta variables from the text**

**## speaker in meta2; determine length of each speech**

**## works if there are no missing values in meta2**

**len=dim(dim(data)[1])**

**meta1=dim(dim(data)[1])**

**meta2=dim(dim(data)[1])**

**for (i in 1:dim(data)[1]) {**

**txt=data[i,1]**

**temp=strsplit(txt, " ")[[1]]**

**len[i]=length(temp)-2**

**meta1[i]=temp[1]**

**meta2[i]=temp[2]**

**tempr=dim(len[i])**

**for (j in 1:len[i]) {**

**tempr[j]=temp[j+2]**

**}**

**data[i,1]=toString(tempr)**

**data[i,1]=gsub("[,]","", ignore.case = TRUE,data[i,1])**

**}**

**data[1:4,1]**

**len**

**hist(len)**

**boxplot(len)**

**quantile(len)**

**meta2**

**## omit meta variables from the text**

**## read and clean data**

**library(tm)**

**library(SnowballC) ## needed for stemming**

**## creating corpus**

**corpus = VCorpus(VectorSource(data[,1]),readerControl = list(reader = readPlain)) ## this is how to create corpus**

**corpus1 = tm\_map(corpus, stripWhitespace)**

**corpus2 = tm\_map(corpus1, content\_transformer(tolower))**

**corpus3 = tm\_map(corpus2, removePunctuation)**

**corpus4 = tm\_map(corpus3, removeNumbers)**

**corpus5 = tm\_map(corpus4, removeWords, stopwords("english"))**

**corp.dtm = DocumentTermMatrix(corpus5,control=list(stemming=FALSE)) ## no stemming is the default**

**corp.dtm**

**corp.tdm = TermDocumentMatrix(corpus5,control=list(stemming=FALSE))**

**corp.tdm**

**corps.dtm = DocumentTermMatrix(corpus5,control=list(stemming=TRUE))**

**corps.dtm**

**findFreqTerms(corp.dtm,1)**

**findFreqTerms(corps.dtm,1)**

**stopwords("english")**

**## adding your own stopwords**

**stopwords("english")**

**stopwordsnew1=c(stopwords("english"),"occasionally")**

**stopwordsnew1**

**stopwordsnew2=c("perhaps","never")**

**stopwordsnew2**

**## adding your own stopwords**

**## frequencies of words**

**## displaying frequencies**

**library(ggplot2)**

**dim(corp.dtm)**

**as.matrix(corp.dtm)**

**findFreqTerms(corp.dtm,1)**

**findFreqTerms(corp.dtm,2)**

**freq=colSums(as.matrix(corp.dtm))**

**ord=order(freq)**

**freq[head(ord)]**

**freq[tail(ord)]**

**freq=sort(colSums(as.matrix(corp.dtm)),decreasing=TRUE)**

**head(freq,20)**

**wf=data.frame(word=names(freq),freq=freq)**

**head(wf)**

**p=ggplot(subset(wf,freq>2),aes(word,freq))**

**p=p+geom\_bar(stat="identity")**

**p=p+theme(axis.text.x=element\_text(angle=45,hjust=1))**

**p**

**## displaying frequencies**

**## displaying word clouds**

**library(wordcloud)**

**set.seed(142)**

**wordcloud(names(freq),freq,min.freq=1)**

**set.seed(142)**

**dark2 = brewer.pal(6,"Dark2")**

**wordcloud(names(freq),freq,max.words=7,rot.per=0.2,colors=dark2)**

**## displaying word clouds**

**## frequencies of words**

**## finding associations**

**as.matrix(corp.dtm)**

**findAssocs(corp.dtm, "explanation", 0.5)**

**findAssocs(corp.dtm, "gentleman", 0.5)**

**## weightBin creates indicator variables for presence of term**

**Bcorp.dtm=weightBin(corp.dtm)**

**as.matrix(Bcorp.dtm)**

**findAssocs(Bcorp.dtm, "explanation", 0.5)**

**findAssocs(Bcorp.dtm, "gentleman", 0.5)**

**## finding associations**

**## sequence plotting: helpful as it shows whether certain terms occur together**

**vecg=as.matrix(corp.dtm)[,"gentleman"]**

**vecg**

**vece=as.matrix(corp.dtm)[,"explanation"]**

**vece**

**par(mfrow=c(1,1))**

**plot(vecg,type="l",lwd=7,xlab="document",ylab="frequency",ylim=c(0,max(c(vecg,vece))))**

**lines(vece,type="l",col=10,lwd=3)**

**## sequence plotting**

**## bigrams**

**BigramTokenizer = function(x)**

**unlist(lapply(ngrams(words(x), 2), paste, collapse = " "), use.names = FALSE)**

**bi.dtm = DocumentTermMatrix(corpus5, control = list(tokenize = BigramTokenizer))**

**bi.dtm**

**as.matrix(bi.dtm)**

**bi.tdm = TermDocumentMatrix(corpus5, control = list(tokenize = BigramTokenizer))**

**bi.tdm**

**as.matrix(bi.tdm)**

**## bigrams**

**## displaying bigram frequencies**

**findFreqTerms(bi.dtm,1)**

**findFreqTerms(bi.dtm,2)**

**freq=colSums(as.matrix(bi.dtm))**

**ord=order(freq)**

**freq[head(ord)]**

**freq[tail(ord)]**

**freq=sort(colSums(as.matrix(bi.dtm)),decreasing=TRUE)**

**head(freq,20)**

**wf=data.frame(word=names(freq),freq=freq)**

**head(wf)**

**p=ggplot(subset(wf,freq>1),aes(word,freq))**

**p=p+geom\_bar(stat="identity")**

**p=p+theme(axis.text.x=element\_text(angle=45,hjust=1))**

**p**

**## displaying bigram frequencies**