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

**####################### ANALYSIS (SECTION 5.3) ######################################**

**################### Reading the data from file PrelimData.RData #########################**

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

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

**####################### ANALYSIS (SECTION 5.3.2) #####################################**

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

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

**library(tm)**

**library(slam)**

**library(wordcloud)**

**## LOADING DATA (39th Congress)**

**load("C:\\Johannes Ledolter\\2020March01Book\\Chapter5WEB\\PrelimData.RData")**

**## creating the corpus**

**corpus <- VCorpus(VectorSource(data),readerControl = list(reader = readPlain))**

**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**

**dim(corp.dtm)**

**## cutting out short letters**

**quantile(len)**

**indexlen=len>=32 ## can pick a length to omit short letters**

**nnn=dim(corp.dtm)[1]**

**number=dim(nnn)**

**for (i in 1:nnn) {**

**number[i]=i**

**}**

**corp.dtm=corp.dtm[number[indexlen],]**

**dim(corp.dtm)**

**meta2=meta2[number[indexlen]]**

**meta3=meta3[number[indexlen]]**

**meta4=meta4[number[indexlen]]**

**meta5=meta5[number[indexlen]]**

**meta6=meta6[number[indexlen]]**

**group=group[number[indexlen]]**

**len=len[number[indexlen]]**

**## cutting out short letters**

**## simplifying the corpus and its dtm**

**dim(corp.dtm)**

**corpStripped.dtm=removeSparseTerms(corp.dtm,0.999)**

**## terms that show up in at least 0.1 percent of documents (for 50,000 documents: 50,000 x 0.001 = 50)**

**dim(corpStripped.dtm)**

**## Check for documents with zero words as stripping words may lead a document with no words**

**## empty documents need to be omitted**

**dim(corpStripped.dtm)**

**index=row\_sums(corpStripped.dtm)==0**

**nnn=dim(corpStripped.dtm)[1]**

**number=dim(nnn)**

**for (i in 1:nnn) {**

**number[i]=i**

**}**

**number[index]**

**## do the following only if there are empty documents**

**corpStripped.dtm=corpStripped.dtm[-number[index],]**

**dim(corpStripped.dtm)**

**meta2=meta2[-number[index]]**

**meta3=meta3[-number[index]]**

**meta4=meta4[-number[index]]**

**meta5=meta5[-number[index]]**

**meta6=meta6[-number[index]]**

**group=group[-number[index]]**

**len=len[-number[index]]**

**## STRATIFICATION ON FREQUENT SPEAKERS**

**## look at the kkk most frequent speakers**

**kkk=30**

**frequent=dim(kkk)**

**tt=sort(table(meta2),decreasing=TRUE)**

**for(i in 1:kkk) {**

**frequent[i]=labels(tt[i])**

**}**

**frequent**

**labels(corpStripped.dtm)$Terms**

**BcorpStripped.dtm=weightBin(corpStripped.dtm) ## to show presence/absence**

**## individual terms such as "rfg", "declarindep", "civilwar", "labor", "indian", "war"**

**## you can pick others of interest**

**rfg=dim(kkk)**

**for (i in 1:kkk) {**

**pp=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"rfg"])**

**rfg[i]=100\*mean(pp) ## calculate proportion**

**}**

**rfg=round(rfg,1)**

**declarindep=dim(kkk)**

**for (i in 1:kkk) {**

**pp=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"declarindep"])**

**declarindep[i]=100\*mean(pp) ## calculate proportion**

**}**

**declarindep=round(declarindep,1)**

**civilwar=dim(kkk)**

**for (i in 1:kkk) {**

**pp=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"civilwar"])**

**civilwar[i]=100\*mean(pp) ## calculate proportion**

**}**

**civilwar=round(civilwar,1)**

**labor=dim(kkk)**

**for (i in 1:kkk) {**

**pp=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"labor"])**

**labor[i]=100\*mean(pp)**

**}**

**labor=round(labor,1)**

**indian=dim(kkk)**

**for (i in 1:kkk) {**

**pp=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"indian"])**

**indian[i]=100\*mean(pp)**

**}**

**indian=round(indian,1)**

**war=dim(kkk)**

**for (i in 1:kkk) {**

**pp=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"war"])**

**war[i]=100\*mean(pp)**

**}**

**war=round(war,1)**

**## joint terms (AND) indicating that both terms have to be in the same document**

**warANDindian=dim(kkk)**

**for (i in 1:kkk) {**

**p1=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"war"])**

**p2=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"indian"])**

**pp=p1\*p2**

**warANDindian[i]=100\*mean(pp)**

**}**

**warANDindian=round(warANDindian,1)**

**## one of several terms (either OR). Document gets flagged if there is at least one word present**

**indianlike=dim(kkk)**

**for (i in 1:kkk) {**

**p1=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"agent"])**

**p2=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"indian"])**

**p3=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"chief"])**

**pp=p1+p2+p3**

**for (j in 1:length(pp)) {**

**if(pp[j]>=1)pp[j]=1**

**}**

**indianlike[i]=100\*mean(pp)**

**}**

**indianlike=round(indianlike,1)**

**warlike=dim(kkk)**

**for (i in 1:kkk) {**

**p1=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"war"])**

**p2=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"army"])**

**p3=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"navy"])**

**p4=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"naval"])**

**p5=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"peace"])**

**p6=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"soldier"])**

**pp=p1+p2+p3+p4+p5+p6**

**for (j in 1:length(pp)) {**

**if(pp[j]>=1)pp[j]=1**

**}**

**warlike[i]=100\*mean(pp)**

**}**

**warlike=round(warlike,1)**

**slaverylike=dim(kkk)**

**for (i in 1:kkk) {**

**p1=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"slave"])**

**p2=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"negro"])**

**p3=as.matrix(BcorpStripped.dtm[meta2==frequent[i],"slavery"])**

**pp=p1+p2+p3**

**for (j in 1:length(pp)) {**

**if(pp[j]>=1)pp[j]=1**

**}**

**slaverylike[i]=100\*mean(pp)**

**}**

**slaverylike=round(slaverylike,1)**

**ttt=sort(tt,decreasing=TRUE)**

**ttt[1]**

**frequency=dim(kkk)**

**for (i in 1:kkk) {**

**frequency[i]=ttt[[i]]**

**}**

**frequency**

**table1=data.frame(frequent,frequency,war,warlike,indian,indianlike,warANDindian,slaverylike,labor,rfg)**

**print(table1)**

**## proportion of speeches that contain rfg**

**pc=100\*(sum(BcorpStripped.dtm[,"rfg"])/dim(BcorpStripped.dtm)[1])**

**pc**

**table2=data.frame(frequent,frequency,rfg,declarindep,civilwar)**

**print(table2)**

**## plotting results**

**par(mfrow=c(2,4))**

**ylim=c(0,0.50)**

**plot(war,type="b",ylim=ylim,xlab="speaker",ylab="war")**

**plot(warlike,type="b",ylim=ylim,xlab="speaker",ylab="war like")**

**plot(indian,type="b",ylim=ylim,xlab="speaker",ylab="indian")**

**plot(indianlike,type="b",ylim=ylim,xlab="speaker",ylab="indian like")**

**plot(warANDindian,type="b",ylim=ylim,xlab="speaker",ylab="war AND indian")**

**plot(slaverylike,type="b",ylim=ylim,xlab="speaker",ylab="slavery like")**

**plot(labor,type="b",ylim=ylim,xlab="speaker",ylab="labor")**

**plot(rfg,type="b",ylim=ylim,xlab="speaker",ylab="rfg")**

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

**plot(rfg,type="b",xlab="speaker",ylab="rep form gov")**

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

**####################### ANALYSIS (SECTION 5.3.1) #####################################**

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

**## CONSTRUCTING THE ADJACENCY TABLE**

**## Here we look at all letters (also short letters)**

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

**library(tm)**

**library(slam)**

**library(stm)**

**## LOADING DATA (39th Congress)**

**load("C:\\Johannes Ledolter\\2020March01Book\\Chapter5WEB\\PrelimData.RData")**

**meta2**

**## frequency bar graph of speakers with more than 750 speeches**

**hh=sort(table(meta2),decreasing=FALSE)**

**hh1=hh[hh>750]**

**plot(hh1,las=2,cex.axis=0.8,ylab="frequency",xlab="",main="Frequency Bar Graph for speakers with more than 750 speeches")**

**## selecting the speakers with nu of speeches > 750**

**u=sort(table(meta2),decreasing = FALSE)**

**u**

**v=row.names(u)[u>750]**

**v**

**v=sort(v)**

**v**

**speaker=dim(length(meta2))**

**for (i in 1:length(meta2)) {**

**speaker[i]="nafs"**

**## nafs is designating any speaker that has not spoken more than 750 times**

**## nfas: Not A Frequent Speaker**

**for (j in 1:length(v)) {**

**if(meta2[i]==v[j]) speaker[i]=v[j]**

**}**

**}**

**speaker**

**table(speaker)**

**label=row.names(table(speaker))**

**label**

**m=length(table(speaker))**

**m**

**t=length(meta2)**

**t**

**tend=t-1**

**## look for the next one (adjacent)**

**sumorig=matrix(nrow=m,ncol=m)**

**for (i in 1:m) {**

**for (j in 1:m) {**

**sumorig[i,j]=0**

**for (k in 1:tend) {**

**if(speaker[k]==label[i] & speaker[k+1]==label[j]) sumorig[i,j]=sumorig[i,j]+1**

**}**

**}**

**}**

**## setting diagonal elements zero**

**for (i in 1:m) {**

**sumorig[i,i]=0**

**}**

**sum**

**dim(sumorig)**

**rownames(sumorig)=label**

**colnames(sumorig)=label**

**sumorig**

**label**

**## with weights within the next three periods**

**## sumorig=matrix(nrow=m,ncol=m)**

**## for (i in 1:m) {**

**## for (j in 1:m) {**

**## sumorig[i,j]=0**

**## tend=t-3**

**## for (k in 1:tend) {**

**## if(speaker[k]==label[i] & speaker[k+1]==label[j]) sumorig[i,j]=sumorig[i,j]+1**

**## if(speaker[k]==label[i] & speaker[k+2]==label[j]) sumorig[i,j]=sumorig[i,j]+1**

**## if(speaker[k]==label[i] & speaker[k+3]==label[j]) sumorig[i,j]=sumorig[i,j]+1**

**## }**

**## }**

**## }**

**## rownames(sumorig)=label**

**## colnames(sumorig)=label**

**## sumorig**

**## label**

**## delete selective names (as we are not interested whether "secretary" follows "clerk")**

**ind=c(1:m)**

**speakerstodelete=c(ind[label=="UNKNOWN"],ind[label=="nafs"],ind[label=="president"],ind[label=="presiding"],ind[label=="secretary"],ind[label=="chairman"],ind[label=="speaker"],ind[label=="clerk"])**

**delete=-speakerstodelete**

**sum=sumorig[,delete]**

**sum=sum[delete,]**

**lab1=label[delete]**

**dim(sum)**

**rownames(sum)=lab1**

**colnames(sum)=lab1**

**sum**

**lab1**

**senind=c(ind[lab1=="foot"], ind[lab1=="wade"], ind[lab1=="sumner"], ind[lab1=="trumbull"], ind[lab1=="collamer"], ind[lab1=="foster"], ind[lab1=="harlan"], ind[lab1=="chandler"], ind[lab1=="doolittle"], ind[lab1=="clark"], ind[lab1=="anthony"], ind[lab1=="saulsbury"], ind[lab1=="grimes"], ind[lab1=="howe"], ind[lab1=="mcdougall"], ind[lab1=="lane"], ind[lab1=="nesmith"], ind[lab1=="cowan"], ind[lab1=="sherman"], ind[lab1=="pomeroy(S)"], ind[lab1=="howard"], ind[lab1=="ramsey"], ind[lab1=="sprague"], ind[lab1=="conness"], ind[lab1=="hendricks"], ind[lab1=="buckalew"], ind[lab1=="willey"], ind[lab1=="vanwinkle"], ind[lab1=="brown"], ind[lab1=="riddle"], ind[lab1=="stewart"], ind[lab1=="nye"], ind[lab1=="cragin"], ind[lab1=="yates"], ind[lab1=="fessenden"], ind[lab1=="norton"], ind[lab1=="guthrie"], ind[lab1=="creswell"], ind[lab1=="stockton"], ind[lab1=="poland"], ind[lab1=="kirkwood"], ind[lab1=="edmunds"], ind[lab1=="fowler"], ind[lab1=="fogg"], ind[lab1=="cattell"], ind[lab1=="henderson(S)"], ind[lab1=="wilson(S)"], ind[lab1=="johnson(S)"], ind[lab1=="davis(S)"], ind[lab1=="harris(S)"], ind[lab1=="williams(S)"], ind[lab1=="dixon(S)"], ind[lab1=="morgan(S)"], ind[lab1=="wright(S)"], ind[lab1=="ross(S)"], ind[lab1=="patterson(S)"], ind[lab1=="morrill(S)"],ind[lab1=="frelinghuysen"])**

**senind**

**length(senind)**

**#### consider only speakers from the HOUSE**

**delete=-senind**

**sumhouse=sum[,delete]**

**sumhouse=sumhouse[delete,]**

**dim(sumhouse)**

**lab2=lab1[delete]**

**dim(sumhouse)**

**rownames(sumhouse)=lab2**

**colnames(sumhouse)=lab2**

**sumhouse**

**lab2**

**m=dim(sumhouse)[1]**

**rsum=rowSums(sumhouse)**

**csum=colSums(sumhouse)**

**ratio=matrix(nrow=m,ncol=m)**

**for (i in 1:m) {**

**for (j in 1:m) {**

**ratio[i,j]=sumhouse[i,j]/rsum[i]**

**}**

**}**

**rownames(ratio)=lab2**

**colnames(ratio)=lab2**

**ratio**

**ratio=round(ratio,dig=2)**

**ratio**

**comb=matrix(nrow=m,ncol=m)**

**for (i in 1:m) {**

**for (j in 1:m) {**

**comb[i,j]=toString(c(lab2[i],lab2[j]))**

**}**

**}**

**## draw network (directed) with sumhouse**

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

**library(igraph) ## load the package**

**letters <- graph.adjacency(sumhouse,mode="directed",weighted=TRUE,diag=FALSE,add.colnames=NULL,add.rownames=NA)**

**letters**

**V(letters)**

**E(letters)**

**degree(letters)**

**degree(letters,mode="in")**

**degree(letters,mode="out")**

**lsym=degree(letters,mode="out")/degree(letters)**

**lsym**

**set.seed(1234)**

**plot(letters)**

**set.seed(1234)**

**plot.igraph(letters,vertex.label=V(letters)$name,layout=layout.fruchterman.reingold,edge.color="black",edge.curved=TRUE)**

**set.seed(1234)**

**plot.igraph(letters,vertex.label=V(letters)$name,layout=layout.fruchterman.reingold,edge.color="black",edge.width=E(letters)$weight/15,edge.curved=TRUE)**

**nuconn=E(letters)$weight**

**nuconn**

**qq=quantile(nuconn)**

**qq**

**nuconncode=dim(length(nuconn))**

**for (i in 1:length(nuconn)) {**

**nuconncode[i]=4**

**if(nuconn[i]<qq[4])nuconncode[i]=3**

**if(nuconn[i]<qq[3])nuconncode[i]=2**

**if(nuconn[i]<qq[2])nuconncode[i]=1**

**}**

**nuconncode**

**E(letters)$weight=nuconncode**

**set.seed(1234)**

**plot.igraph(letters,vertex.label=V(letters)$name,layout=layout.fruchterman.reingold,edge.color="black",edge.width=nuconncode,edge.curved=TRUE)**

**#### consider only speakers from the SENATE**

**delete=c(1:length(lab1))[-senind]**

**delete=-delete**

**sumsenate=sum[,delete]**

**sumsenate=sumsenate[delete,]**

**lab2=lab1[delete]**

**lab2**

**rownames(sumsenate)=lab2**

**colnames(sumsenate)=lab2**

**sumsenate**

**m=length(lab2)**

**rsum=rowSums(sumsenate)**

**csum=colSums(sumsenate)**

**ratio=matrix(nrow=m,ncol=m)**

**for (i in 1:m) {**

**for (j in 1:m) {**

**ratio[i,j]=sum[i,j]/rsum[i]**

**}**

**}**

**rownames(ratio)=lab2**

**colnames(ratio)=lab2**

**ratio**

**ratio=round(ratio,dig=2)**

**ratio**

**## draw network (directed) with sumsenate**

**library(igraph) ## load the package**

**letters <- graph.adjacency(sumsenate,mode="directed",weighted=TRUE,diag=FALSE,add.colnames=NULL,add.rownames=NA)**

**letters**

**V(letters)**

**E(letters)**

**degree(letters)**

**degree(letters,mode="in")**

**degree(letters,mode="out")**

**lsym=degree(letters,mode="out")/degree(letters)**

**lsym**

**set.seed(1234)**

**plot(letters)**

**set.seed(1234)**

**plot.igraph(letters,vertex.label=V(letters)$name,layout=layout.fruchterman.reingold,edge.color="black",edge.curved=TRUE)**

**set.seed(1234)**

**plot.igraph(letters,vertex.label=V(letters)$name,layout=layout.fruchterman.reingold,edge.color="black",edge.width=E(letters)$weight/50,edge.curved=TRUE)**

**nuconn=E(letters)$weight**

**nuconn**

**qq=quantile(nuconn)**

**qq**

**sumsenatered=matrix(nrow=m,ncol=m)**

**sumsenatered=sumsenate**

**for (i in 1:m) {**

**for (j in 1:m) {**

**if(sumsenatered[i,j]<qq[3])sumsenatered[i,j]=0**

**}**

**}**

**rownames(sumsenatered)=lab2**

**colnames(sumsenatered)=lab2**

**sumsenatered**

**letters1 <- graph.adjacency(sumsenatered,mode="directed",weighted=TRUE,diag=FALSE,add.colnames=NULL,add.rownames=NA)**

**set.seed(1234)**

**plot.igraph(letters1,vertex.label=V(letters1)$name,layout=layout.fruchterman.reingold,edge.color="black",edge.width=E(letters1)$weight/50,edge.curved=TRUE)**