FIT3152asm\_1\_final\_markdown

knitr::opts\_chunk$set(echo = TRUE,include=T, message =FALSE)

Import needed library

library(tidyverse)  
library(lubridate)  
library(extrafont) # for changing the font stlye of the graph

Read the data

rm(list = ls())  
set.seed(31084222)  
data <- read.csv("C:/Users/sjsa3/Desktop/Shared\_with\_Mac/year2\_sem1/FIT3152/Assignment\_FIT3152\_2021/webforum.csv")  
  
data <- data[sample(nrow(data),20000),] #20000 rows

Clean the data

#define Min-Max normalisation method  
min\_max\_normalisation <- function(x) {  
 (x - min(x)) / (max(x) - min(x))  
 }

Change the font style

changeFont <- function(){  
 theme\_classic() + theme(text=element\_text(family="Times New Roman", face="bold", size=12)) #Times New Roman, 12pt, Bold  
}

data$Date <- as.Date(data$Date)  
  
  
#check if there is any missing values  
sum(is.na(data))

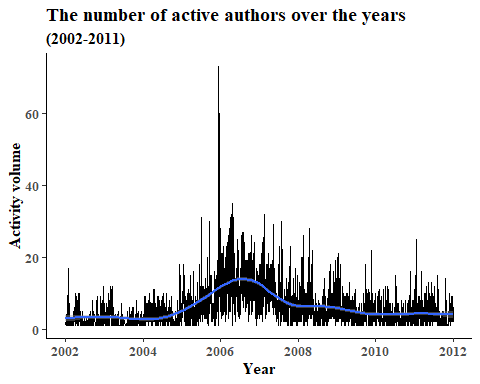
## [1] 0

data\_tidy <- data %>%  
 mutate(month = month(Date, label = TRUE, abbr = TRUE),   
 wday = wday(Date, label = TRUE, abbr = TRUE, week\_start = 1),  
 year = year(Date),  
 day = day(Date),  
 hour = hour(hm(data$Time)))  
 data\_tidy1 <- data\_tidy  
#create a function for normalisation  
normalise\_data\_tidy <- function(x){  
 #apply Min-Max normalisation to all numeric columns  
data\_tidy\_norm <- as.data.frame(lapply(x[,5:19], min\_max\_normalisation))  
return(data\_tidy\_norm)  
}  
data\_tidy\_norm <- normalise\_data\_tidy(data\_tidy)

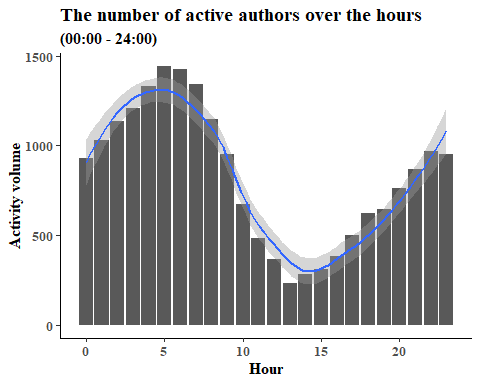
==============================================================================================================

* Q1
  1. How active are participants, are there periods where this increases or decreases?

over\_df = data\_tidy %>% group\_by(Date) %>% summarise(count =n())  
ggplot(over\_df,  
 aes(x = Date,   
 y = count)) +  
 geom\_line() +  
 stat\_smooth()+labs(  
 title = "The number of active authors over the years",  
 subtitle = "(2002-2011)",  
 x = "Year",  
 y = "Activity volume"  
 )+changeFont()

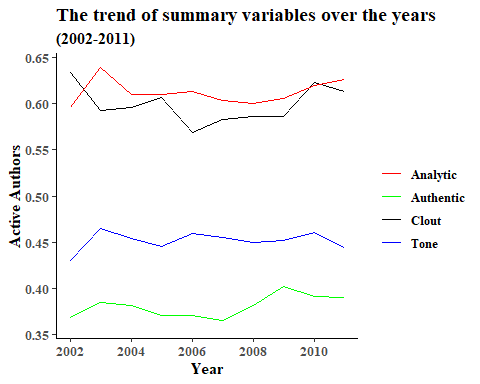


library(extrafont)  
#Week-Day   
library(lubridate)  
hour\_df = data\_tidy %>% group\_by(hour) %>% summarise(count=n())  
ggplot(hour\_df,  
 aes(x = hour,   
 y = count))+labs(  
 title = "The number of active authors over the hours",  
 subtitle = "(00:00 - 24:00)",  
 x = "Hour",  
 y = "Activity volume"  
 ) +   
 geom\_col()+theme\_classic()+geom\_smooth()+changeFont()

 ——————————————————————————————– b Looking at the linguistic variables, 1 do these change over time?

grp\_yr = data.frame(data\_tidy\_norm ) %>% cbind(year = data\_tidy$year)  
  
grp\_yr = grp\_yr %>% group\_by(year) %>%   
 summarise(count =n(),Tone = mean(Tone, na.rm = TRUE),  
 WC = mean(WC,na.rm = TRUE),  
 Analytic = mean(Analytic,na.rm = TRUE),  
 Clout = mean(Clout,na.rm = TRUE),  
 Authentic = mean(Authentic,na.rm = TRUE),  
 WP = mean(WPS,na.rm = TRUE),  
 i = mean(i,na.rm = TRUE),  
 we = mean(we,na.rm = TRUE),  
 you = mean(you,na.rm = TRUE),  
 they = mean(they,na.rm = TRUE),  
 number = mean(number,na.rm = TRUE),  
 affect = mean(affect,na.rm = TRUE),  
 posemo = mean(posemo,na.rm = TRUE),  
 negemo = mean(negemo,na.rm = TRUE),  
 anx = mean(anx,na.rm = TRUE)) %>%   
 arrange(desc(count))

ggplot(data = grp\_yr)+geom\_line(aes(year,Tone,colour = "Tone"))+  
  
 geom\_line(aes(year,Authentic,colour = "Authentic"))+  
  
 geom\_line(aes(year,Clout, colour = "Clout"))+  
  
 geom\_line(aes(year,Analytic,colour ="Analytic"))+  
 scale\_colour\_manual("",values =  
 c( "Tone"="blue",   
 "Authentic"="green",   
 "Clout" = "black",   
 "Analytic" = "red"))+   
 ylim(0.36,.64)+labs(  
 title = "The trend of summary variables over the years",  
 subtitle = "(2002-2011)",  
 x = "Year",  
 y = "Active Authors"  
 )+theme\_minimal()+changeFont()

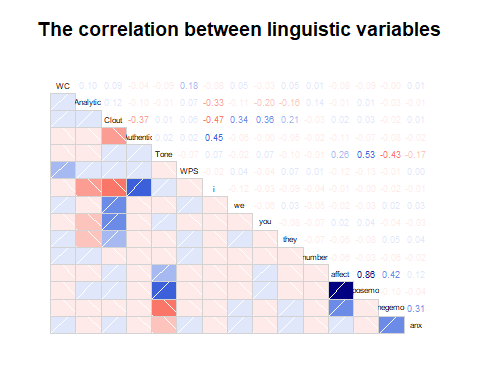


Since Clout’s change has been most turbulent, it is worth know which variables define this variable. We will use linear regression model to know it ——————————————————————————————– c Is there a relationship between variables? ——————————————————————————————–

library(corrgram)

## Warning: package 'corrgram' was built under R version 4.0.4

corrgram(data\_tidy\_norm,upper.panel=panel.cor, main= "The correlation between linguistic variables")



Regression model

library(Hmisc)

## Warning: package 'Hmisc' was built under R version 4.0.4

library(psych)

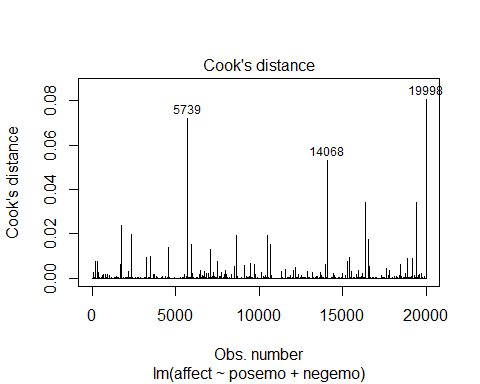
## Warning: package 'psych' was built under R version 4.0.4

library(car)

fit <- lm(affect ~ posemo +negemo, data=data\_tidy)  
summary(fit)

##   
## Call:  
## lm(formula = affect ~ posemo + negemo, data = data\_tidy)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.2158 -0.0470 -0.0430 -0.0405 9.0313   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0429685 0.0028567 15.04 <2e-16 \*\*\*  
## posemo 0.9994324 0.0003455 2893.03 <2e-16 \*\*\*  
## negemo 1.0017282 0.0006154 1627.78 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2979 on 19997 degrees of freedom  
## Multiple R-squared: 0.998, Adjusted R-squared: 0.998   
## F-statistic: 5.106e+06 on 2 and 19997 DF, p-value: < 2.2e-16

# crPlots(fit)  
  
# Eliminate extreme values  
cutoff <- 4/((nrow(data\_tidy)-length(fit$coefficients)-2)) # Cook's D plot, cutoff as 4/(n-k-1)  
plot(fit, which=4, cook.levels=cutoff) # identify D values > cutoff



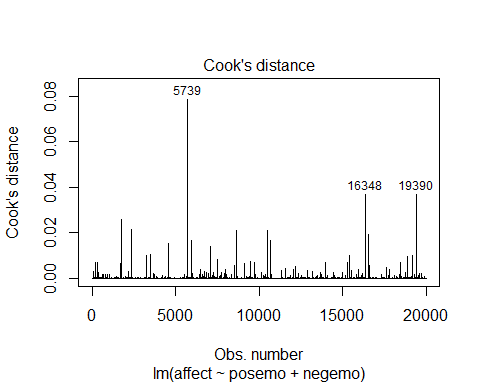
data\_tidy <- data\_tidy[-which(rownames(data\_tidy) # Row names discovered in 2 rounds  
 %in% c("19998", "5379", "14068")),]

After refiting the model 1

fit <- lm(affect ~ posemo+negemo, data=data\_tidy)  
summary(fit)

##   
## Call:  
## lm(formula = affect ~ posemo + negemo, data = data\_tidy)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.1947 -0.0462 -0.0429 -0.0400 7.1009   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0429051 0.0027416 15.65 <2e-16 \*\*\*  
## posemo 0.9993468 0.0003316 3013.91 <2e-16 \*\*\*  
## negemo 1.0015181 0.0005907 1695.57 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2859 on 19994 degrees of freedom  
## Multiple R-squared: 0.9982, Adjusted R-squared: 0.9982   
## F-statistic: 5.542e+06 on 2 and 19994 DF, p-value: < 2.2e-16

# crPlots(fit)  
  
# Eliminate extreme values  
cutoff <- 4/((nrow(data\_tidy)-length(fit$coefficients)-2)) # Cook's D plot, cutoff as 4/(n-k-1)  
plot(fit, which=4, cook.levels=cutoff) # identify D values > cutoff



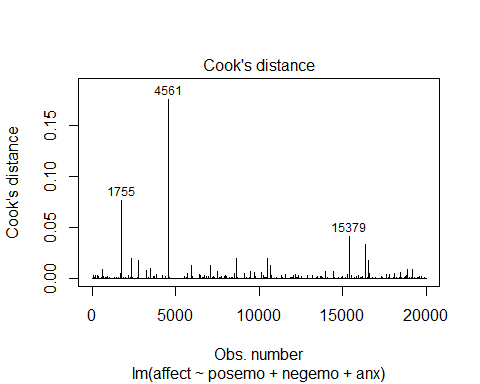
data\_tidy <- data\_tidy[-which(rownames(data\_tidy) # Row names discovered in 2 rounds  
 %in% c("19390", "16438", "5739")),]

After refiting the model 2

fit <- lm(affect ~ posemo+negemo+anx, data=data\_tidy)  
summary(fit)$adj.r.squared # R2=81%, F=139.5

## [1] 0.9982738

# crPlots(fit)  
  
# Eliminate extreme values  
cutoff <- 4/((nrow(data\_tidy)-length(fit$coefficients)-2)) # Cook's D plot, cutoff as 4/(n-k-1)  
plot(fit, which=4, cook.levels=cutoff) # identify D values > cutoff



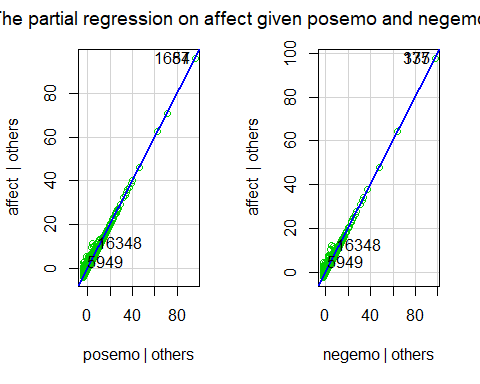
data\_tidy <- data\_tidy[-which(rownames(data\_tidy) # Row names discovered in 2 rounds  
 %in% c("4561", "1755", "15379")),]

After refiting the model 3

fit <- lm(affect ~ posemo+negemo, data=data\_tidy)  
summary(fit)$adj.r.squared # R2=81%, F=139.5

## [1] 0.9983227

avPlots(fit, main = "The partial regression on affect given posemo and negemo",col = carPalette()[7])



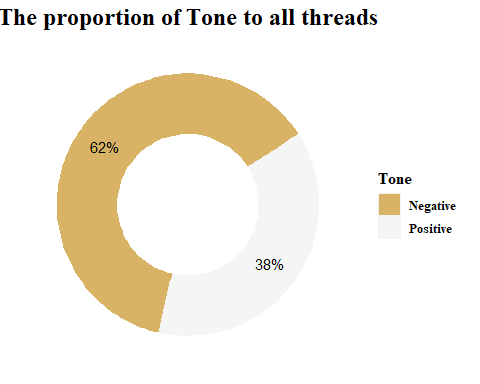
============================================================================================================== - Q2

Analyse the language used by groups. Some starting points:  
  
a Threads indicate groups of participants communicating on the same topic. Describe the  
threads present in your data.  
--------------------------------------------------------------------------------------------

df\_tone <- data\_tidy %>% group\_by(ThreadID) %>% summarise(Tone = median(Tone))   
summary(df\_tone$Tone)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 25.77 34.93 43.94 66.07 99.00

df\_2 <- data\_tidy %>%   
 group\_by(ThreadID) %>%  
 summarise(Tone = median(Tone, na.rm = TRUE))  
df\_2 = df\_2 %>% mutate(Tone = ifelse(Tone >50 , "Positive", "Negative"))%>%   
 group\_by(Tone) %>% summarise(count =n())  
  
df\_for\_donut\_chart <- df\_2   
  
  
# Compute percentages  
df\_for\_donut\_chart$fraction <- df\_for\_donut\_chart$count / sum(df\_for\_donut\_chart$count)  
  
  
ggplot(df\_for\_donut\_chart, aes(x=2,y=fraction,fill=Tone)) +  
geom\_col()+  
 coord\_polar(theta="y",start = 1) +  
 geom\_text(aes(label= paste0(round(fraction\*100),"%")),  
 position = position\_stack(vjust = .5))+  
 theme(panel.background = element\_blank(),  
 axis.line = element\_blank(),  
 axis.text = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.title = element\_blank(),  
 plot.title = element\_text(hjust = .5,size = 18)  
 )+  
 labs(title = "The proportion of Tone to all threads") +  
 scale\_fill\_brewer(palette="BrBG") +  
 xlim(0.5,2.5)+   
 theme(text=element\_text(family="Times New Roman", face="bold", size=12))

 1. Sentiments : are most the thread +ve ? 2. Pronoun : What are the most used pronoun? 3. Structure : What are the mean of WC and WPS?

b By analysing the linguistic variables for all or some of the threads, is it possible to see a  
difference in the language used by different groups?  
--------------------------------------------------------------------------------------------  
  
Find out the languages used between the most postivie and negative threads.  
 1. find out the 10 most active threads

data\_tidy\_norm <- data\_tidy\_norm %>% cbind(ThreadID = data$ThreadID)  
df\_active\_10 <- data\_tidy %>% group\_by(ThreadID)%>% summarise(count = n()) %>% arrange(desc(count))  
df\_active\_10 <- head(df\_active\_10,10)  
df\_active\_10

## # A tibble: 10 x 2  
## ThreadID count  
## <int> <int>  
## 1 283958 136  
## 2 252620 127  
## 3 127115 114  
## 4 472752 109  
## 5 145223 95  
## 6 532649 79  
## 7 309286 71  
## 8 191868 69  
## 9 296985 65  
## 10 249001 56

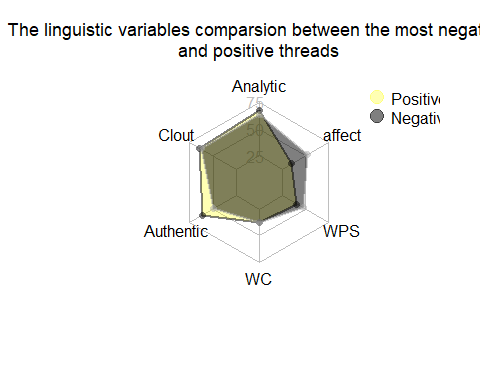
df <- data\_tidy1 %>% filter(data$ThreadID %in% df\_active\_10$ThreadID ) %>% arrange(ThreadID)

2. In the most active threads, find out the most most postivie and negative threads. The data set is called data\_nega\_pose

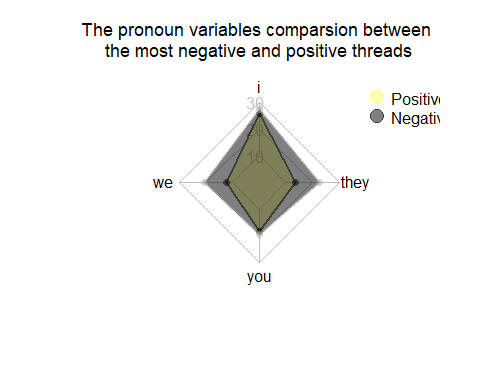
data\_nega\_pose <- df %>% group\_by(ThreadID) %>% summarise(mean = mean(Tone))  
data\_nega\_pose1 <- data\_nega\_pose  
data\_nega\_pose = data\_nega\_pose %>% mutate(emo = ifelse(mean >50 , "Positive", "Negative"))%>%   
 group\_by(emo) %>% arrange(desc(mean))  
  
  
#It is noticed that Thread 472752 is the most Positive and 309286 is the most Negative  
#most positve  
data.mostPose <- data\_tidy %>% filter(ThreadID =="472752" )  
data.mostPose <- data.mostPose[5:19]  
data.mostPose1 <- data.mostPose  
  
#most negative  
data.mostNega <- data\_tidy %>% filter(ThreadID =="309286" )  
data.mostNega <- data.mostNega[5:19]  
data.mostNega1 <- data.mostNega

Based on the data collected from the most negative and positive, we can make a Radarchart to visualise the difference

library(fmsb)  
data.mostPose <- data.mostPose %>% select(Analytic,Clout,Authentic,WC,WPS,affect)  
  
  
data.mostPose <- as.data.frame(lapply(data.mostPose[,], min\_max\_normalisation))  
data.mostPose <- data.mostPose %>% summarise(Analytic\_m = mean(Analytic)\*100,  
 Clout\_m = mean(Clout)\*100,  
 Authentic\_m = mean(Authentic)\*100,  
 WC\_m = mean(WC)\*100,  
 WPS\_m = mean(WPS)\*100,  
 affect\_m = mean(affect)\*100  
 )  
  
# negative  
data.mostNega <- data.mostNega %>% select(Analytic,Clout,Authentic,WC,WPS,affect)  
  
  
data.mostNega <- as.data.frame(lapply(data.mostNega[,], min\_max\_normalisation))  
data.mostNega <- data.mostNega %>% summarise(Analytic\_m = mean(Analytic)\*100,  
 Clout\_m = mean(Clout)\*100,  
 Authentic\_m = mean(Authentic)\*100,  
 WC\_m = mean(WC)\*100,  
 WPS\_m = mean(WPS)\*100,  
 affect\_m = mean(affect)\*100  
 )  
  
radar\_data\_Pose <- data.mostNega %>% rbind(data.mostPose )  
  
  
  
radar\_data\_Pose <-data.frame( Analytic = c(75, 0 , data.mostPose[1,1],data.mostNega[1,1] ),  
 Clout = c(75, 0 , data.mostPose[1,2],data.mostNega[1,2] ),  
 Authentic = c(75, 0 , data.mostPose[1,3],data.mostNega[1,3] ),  
 WC = c(75, 0 , data.mostPose[1,4],data.mostNega[1,4] ),  
 WPS = c(75, 0 , data.mostPose[1,5],data.mostNega[1,5] ),  
 affect = c(75, 0 , data.mostPose[1,6],data.mostNega[1,6] ),  
 row.names = c("max","min","Positive","Negative")  
 )  
#defien the colors filled  
colors\_fill <- c(scales::alpha("yellow", 0.3),scales::alpha("black", 0.5))  
#define the line colors  
colors\_line <- c(scales::alpha("black", 0.5),scales::alpha("darkgrey", 0.5))  
  
 radarchart(radar\_data\_Pose, axistype = 1,  
 seg = 2,  
 # Customize the polygon  
 pfcol =colors\_fill, plwd = 2, plty = 1,  
 # Customize the grid  
 cglcol = "grey", cglty = 1, cglwd = 0.8,  
 pcol = colors\_line,  
 # Customize the axis  
 axislabcol = "grey",  
 caxislabels = c(25, 50, 75),  
 palcex = 1.5)   
  
 title(main = "The linguistic variables comparsion between the most negative \nand positive threads",  
 cex.main = 1.1,   
 font.main= 1,  
 cex.sub = 0.75, font.sub = 1, col.sub = "green",  
 col.lab ="darkblue")  
   
   
legend( x=1.3,y=1.3,legend = row.names( radar\_data\_Pose[3:4,] ) ,  
 bty = "n",  
 pch = 20,  
 col=colors\_fill,  
 cex=1,  
 pt.cex=3)



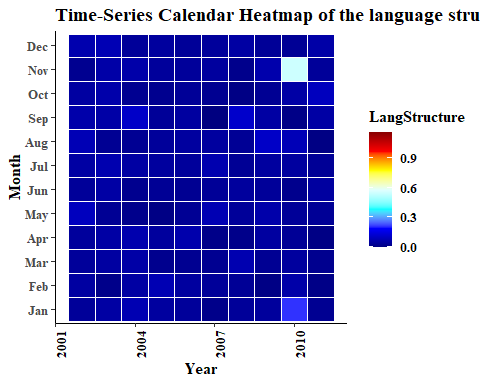
library(fmsb)  
data.mostPose1 <- data.mostPose1 %>% select(i ,we,they, you)  
data.mostPose1 <- as.data.frame(lapply(data.mostPose1[,], min\_max\_normalisation))  
data.mostPose1 <- data.mostPose1 %>% summarise(i\_m = mean(i)\*100,  
 we\_m = mean(we)\*100,  
 you\_m = mean(you)\*100,  
 they\_m = mean(they)\*100  
 )  
  
# negative  
data.mostNega1 <- data.mostNega1 %>% select(i ,we,they, you)  
data.mostNega1 <- as.data.frame(lapply(data.mostNega1[,], min\_max\_normalisation))  
data.mostNega1 <- data.mostNega1 %>% summarise(i\_m = mean(i)\*100,  
 we\_m = mean(we)\*100,  
 you\_m = mean(you)\*100,  
 they\_m = mean(they)\*100  
 )  
  
  
radar\_data\_Pose <- data.mostNega1 %>% rbind(data.mostPose1 )  
radar\_data\_Pose <-data.frame( i = c(30, 0 , data.mostPose1[1,1],data.mostNega1[1,1] ),  
 we = c(30, 0 , data.mostPose1[1,2],data.mostNega1[1,2] ),  
 you = c(30, 0 , data.mostPose1[1,3],data.mostNega1[1,3] ),  
 they= c(30, 0 , data.mostPose1[1,4],data.mostNega1[1,4] ),  
 row.names = c("max","min","Positive","Negative")  
 )  
  
 radarchart(radar\_data\_Pose, axistype = 1,  
 seg = 2,  
 # Customize the polygon  
 pfcol =colors\_fill, plwd = 2, plty = 1,  
 # Customize the grid  
 cglcol = "grey", cglty = 1, cglwd = 0.8,  
 pcol = colors\_line,  
 # Customize the axis  
 axislabcol = "grey",  
 caxislabels = c(10, 20, 30))   
  
 title(main = "The pronoun variables comparsion between \nthe most negative and positive threads",  
 cex.main = 1.1,   
 font.main= 1,  
 cex.sub = 0.75, font.sub = 1, col.sub = "green",  
 col.lab ="darkblue")  
   
   
legend( x=1.3,y=1.3,legend = row.names( radar\_data\_Pose[3:4,] ) ,  
 bty = "n",  
 pch = 20,  
 col=colors\_fill,  
 cex=1,  
 pt.cex=3)



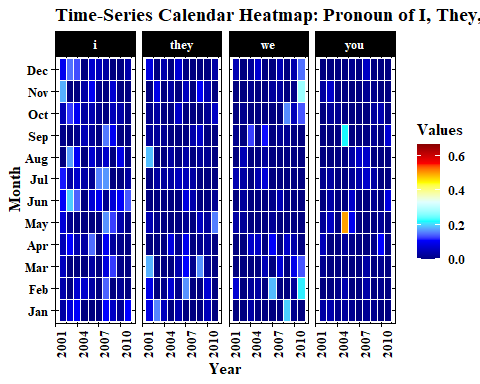
What about the pronoun they use?

c Does the language used within threads (or between threads) change over time? How  
 consistent or variable is the language used within threads?  
--------------------------------------------------------------------------------------------

data\_tidy\_norm <- normalise\_data\_tidy(data\_tidy1)  
data\_tidy\_norm <- data\_tidy\_norm %>% cbind(ThreadID = data\_tidy1$ThreadID, Date =data\_tidy1$Date )  
  
  
data\_tidy\_norm <- data\_tidy\_norm %>% select(Analytic,Clout,Tone,Authentic,WC,WPS,affect,ThreadID,Date, i ,we,they, you )%>%   
 mutate(LangStructure = data\_tidy\_norm[,1] + data\_tidy\_norm[,6] )  
  
grp\_yr = data.frame(data\_tidy\_norm ) %>% cbind(year = data\_tidy1$year, month=data\_tidy1$month )  
  
grp\_yr <- grp\_yr %>% group\_by(ThreadID,year,month) %>% summarise(count =n(),LangStructure = mean(LangStructure,na.rm = TRUE),)%>%   
 arrange(desc(year))   
  
# visualise it  
mycol <- c("navy", "blue", "cyan", "lightcyan", "yellow", "red", "red4")  
ggplot(grp\_yr, aes(x= year , y=month, color =LangStructure )) +  
 geom\_tile (aes(fill=LangStructure),colour = "white" )+labs(  
 title = "Time-Series Calendar Heatmap of the language struture",  
 x = "Year",  
 y = "Month"  
 )+   
 scale\_fill\_gradientn(colours = mycol)+theme\_bw()+  
 changeFont()+  
 theme(axis.text.x = element\_text(, color="BLACK", angle=90))



data\_tidy\_norm <- data\_tidy\_norm %>% select(ThreadID,Date, i ,we,they, you )   
  
  
grp\_yr <- data.frame(data\_tidy\_norm ) %>% cbind(year = data\_tidy1$year, month=data\_tidy1$month )  
  
grp\_yr <- grp\_yr %>% group\_by(ThreadID,year,month) %>% summarise(count =n(),i=median(i),  
 we=median(we),  
 you=median(you),  
 they = median(they))%>%   
 arrange(desc(year))   
  
grp\_yr <-grp\_yr %>% pivot\_longer(cols = c(`i`,`we`,`you`,`they`), names\_to = "Pronoun", values\_to= "Values")  
  
mycol <- c("navy", "blue", "cyan", "lightcyan", "yellow", "red", "red4")  
ggplot(grp\_yr, aes(x= year , y=month, color = Values)) +   
 geom\_tile (aes(fill=Values),colour = "white" ) +   
 scale\_fill\_gradientn(colours = mycol)+  
 facet\_grid(~Pronoun)+  
 theme\_linedraw()+  
 theme(axis.text.x = element\_text(, color="BLACK", angle=90))+labs(  
 title = "Time-Series Calendar Heatmap: Pronoun of I, They, We, You",  
 x = "Year",  
 y = "Month"  
 ) +theme(text=element\_text(family="Times New Roman", face="bold", size=12))



==============================================================================================================  
Q3 Challenge: Social networks online=

#filter data for social network analysis   
filter\_dt <- function(yr,month1){  
df <- data\_tidy1 %>% filter(year == yr, month == month1)  
df <- df %>% select(ThreadID, AuthorID)  
df <- inner\_join(df, df, by = "ThreadID")  
df3 <- apply(df, 2, as.character) #AuthorID as character will become vertex ID  
df3 <- as.tibble(df3)  
df3 <- df3 %>% rename(sources = AuthorID.x, destinations= AuthorID.y)  
df3 <- df3 %>% filter(sources != destinations)  
# df3 <- df3 %>% distinct()  
return(df3)  
}

2002 feb

library(igraph)  
#set a class of table so that I can return multiple   
#objects for the function of tabulate\_dt  
setClass(Class = "Table",representation (edges ="list",  
 nodes ="list"))  
  
tabulate\_dt <- function(df3){  
  
sources <- df3 %>%  
 distinct(sources) %>%  
 rename(label = sources)  
  
## take destinication from letters and make it as "destinations" and renamed as "label"  
destinations <- df3 %>%  
 distinct(destinations) %>%  
 rename(label = destinations)  
  
  
##To create a single dataframe with a column with the unique locations   
nodes <- full\_join(sources, destinations, by = "label")  
nodes <- nodes %>% rowid\_to\_column("id")  
  
per\_route <- df3 %>%   
 group\_by(sources, destinations) %>%   
 summarise(weight = n()) %>%   
 ungroup()  
  
  
edges <- per\_route %>%   
 left\_join(nodes, by = c("sources" = "label")) %>%   
 rename(from = id)  
  
edges <- edges %>%   
 left\_join(nodes, by = c("destinations" = "label")) %>%   
 rename(to = id)  
  
edges <- select (edges, from, to, weight)  
  
#this will return an instance of this class -- Table  
return(new("Table",edges=edges,nodes=nodes))  
}

create function for centrality summary of node within the network

network\_centrality <- function(routes\_igraph){  
degree\_table <- as.table( degree(routes\_igraph))  
betweenness\_table <- as.table( betweenness(routes\_igraph))  
closeness\_table <- as.table( closeness(routes\_igraph))  
eigenvector\_table <- as.table( evcent(routes\_igraph)$vector)  
  
#merge table  
bt\_degree <- merge(degree\_table, betweenness\_table,by= "Var1" )  
bt\_degree <- bt\_degree %>% rename(Betweenness = Freq.x, Degree=Freq.y , id=Var1)  
  
cls\_eig <- merge(closeness\_table,eigenvector\_table, by= "Var1" )  
cls\_eig <- cls\_eig %>% rename(Closeness = Freq.x, Eigenvector=Freq.y,id=Var1)  
  
network\_summary <- merge(bt\_degree,cls\_eig,by= "id")  
network\_summary <- merge(nodes,network\_summary,by="id") %>% select(-id)  
  
#round all numbers in 2 digits  
network\_summary <- network\_summary %>%   
 mutate(Degree = round(Degree,2),   
 Closeness =round( Closeness,2),   
 Eigenvector = round(Eigenvector,2))  
  
network\_summary <- network\_summary %>% arrange(desc(Betweenness,Degree,Eigenvector,Closeness) )  
return(network\_summary)  
}

network for 2002 feb

df3 <- filter\_dt(2002,"Feb")

## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.  
## Please use `as\_tibble()` instead.  
## The signature and semantics have changed, see `?as\_tibble`.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

feb\_table <- tabulate\_dt(df3)  
# access your information of feb\_table and make them as dataframe  
edges <- as.data.frame(feb\_table@edges)  
nodes <- as.data.frame(feb\_table@nodes)  
  
routes\_igraph <- graph\_from\_data\_frame(d = edges, vertices=nodes, directed = F)  
  
network\_centrality(routes\_igraph)

## Warning in closeness(routes\_igraph): At centrality.c:2617 :closeness centrality  
## is not well-defined for disconnected graphs

## label Betweenness Degree Closeness Eigenvector  
## 1 1038 20 49.50 0.01 1.00  
## 2 27 10 1.50 0.01 0.67  
## 3 4876 10 0.50 0.01 0.72  
## 4 4574 8 0.00 0.01 0.54  
## 5 118 8 11.50 0.01 0.33  
## 6 4919 8 1.07 0.01 0.46  
## 7 3785 6 16.40 0.01 0.28  
## 8 5697 4 0.00 0.01 0.31  
## 9 1740 4 11.00 0.01 0.24  
## 10 986 4 12.00 0.01 0.16  
## 11 931 2 0.00 0.00 0.00  
## 12 186 2 0.00 0.01 0.12  
## 13 51 2 0.00 0.01 0.03  
## 14 444 2 0.00 0.00 0.00  
## 15 5103 2 0.00 0.00 0.00  
## 16 3610 2 0.00 0.01 0.04  
## 17 1390 2 0.00 0.00 0.00

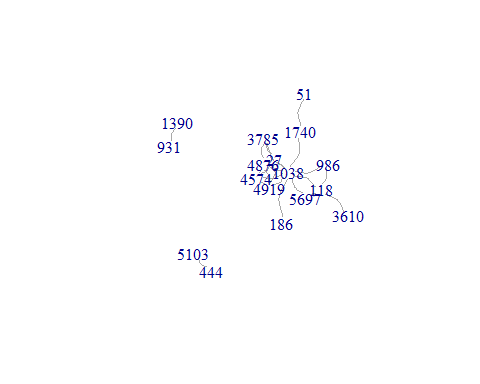
#count the vertexes and nodes  
vcount(routes\_igraph)

## [1] 17

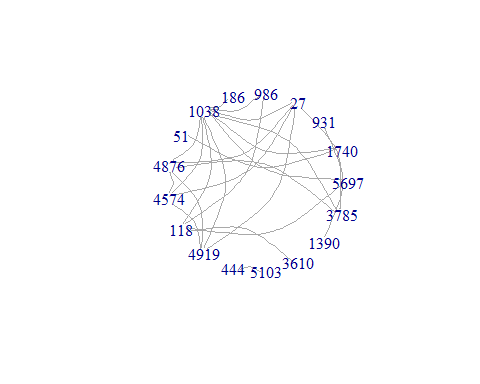
ecount(routes\_igraph)

## [1] 48

plot(routes\_igraph,   
 vertex.shape="none",  
 edge.curved=TRUE)



plot(routes\_igraph, layout = layout\_in\_circle(routes\_igraph),   
 vertex.shape="none",  
 edge.curved=TRUE)



# change the degree size  
# plot(routes\_igraph, layout = layout\_with\_graphopt,  
# edge.arrow.size = 0.2,  
# vertex.size=deg\*3,  
# vertex.color=rgb(0.1,0.7,0.8,0.5) )

network for 2002 march

df3 <- filter\_dt(2002,"Mar")  
mar\_table <- tabulate\_dt(df3)  
# access your information of feb\_table and make them as dataframe  
edges <- as.data.frame(mar\_table@edges)  
nodes <- as.data.frame(mar\_table@nodes)  
  
routes\_igraph <- graph\_from\_data\_frame(d = edges, vertices = nodes, directed = F)  
network\_centrality(routes\_igraph)

## Warning in closeness(routes\_igraph): At centrality.c:2617 :closeness centrality  
## is not well-defined for disconnected graphs

## label Betweenness Degree Closeness Eigenvector  
## 1 118 28 83.64 0.01 0.99  
## 2 5697 26 81.04 0.01 1.00  
## 3 1740 18 2.51 0.01 0.88  
## 4 111 12 20.00 0.01 0.71  
## 5 113 12 4.00 0.01 0.29  
## 6 6025 12 4.00 0.01 0.29  
## 7 4251 12 4.00 0.01 0.29  
## 8 4298 12 4.00 0.01 0.29  
## 9 2508 10 46.66 0.01 0.40  
## 10 194 10 0.00 0.01 0.70  
## 11 986 8 13.00 0.01 0.12  
## 12 6207 8 20.00 0.01 0.12  
## 13 1038 6 24.00 0.01 0.02  
## 14 6085 6 26.00 0.01 0.15  
## 15 887 6 0.00 0.01 0.19  
## 16 1162 6 0.00 0.01 0.12  
## 17 3610 4 4.00 0.01 0.10  
## 18 4291 4 19.40 0.01 0.06  
## 19 42 2 0.00 0.01 0.00  
## 20 5720 2 0.00 0.01 0.01  
## 21 4284 2 0.00 0.01 0.05  
## 22 27 2 0.00 0.00 0.00  
## 23 2748 2 0.00 0.01 0.04  
## 24 1946 2 0.00 0.00 0.00

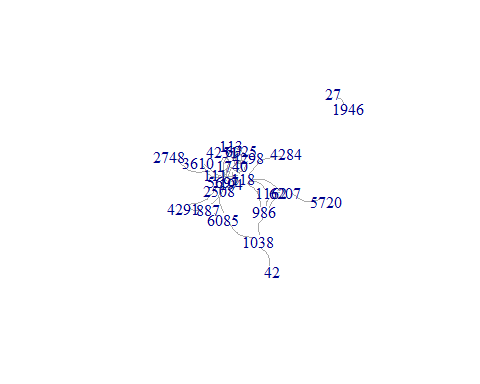
#count the vertexes and nodes  
vcount(routes\_igraph)

## [1] 24

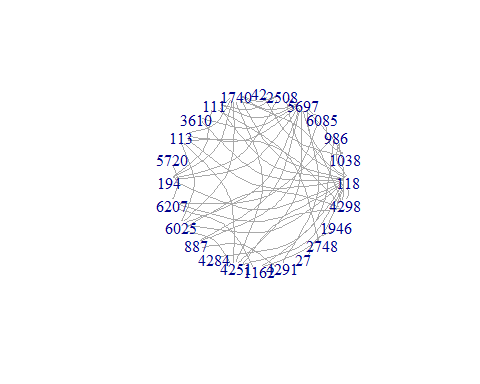
ecount(routes\_igraph)

## [1] 106

plot(routes\_igraph,   
 vertex.shape="none",  
 edge.curved=TRUE)



plot(routes\_igraph, layout = layout\_in\_circle(routes\_igraph),   
 vertex.shape="none",  
 edge.curved=TRUE)



# change the degree size  
# plot(routes\_igraph, layout = layout\_with\_graphopt,  
# edge.arrow.size = 0.2,  
# vertex.size=deg\*3,  
# vertex.color=rgb(0.1,0.7,0.8,0.5) )