Juliet Chow - 32112602

Introduction

Assignment 1 asked to analyze data and output relevant graphs based on the given question from part A-C. webforum.csv file is being used as the source of the data. For every question, all data are being pre-processed based on their needs. It has been explained in detail throughout the questions in this report. In addition, all calculations provided in this report are calculated in R Script.

In this assignment, the three main components: activity, language, and social networks by participants are being analyzed.

A1

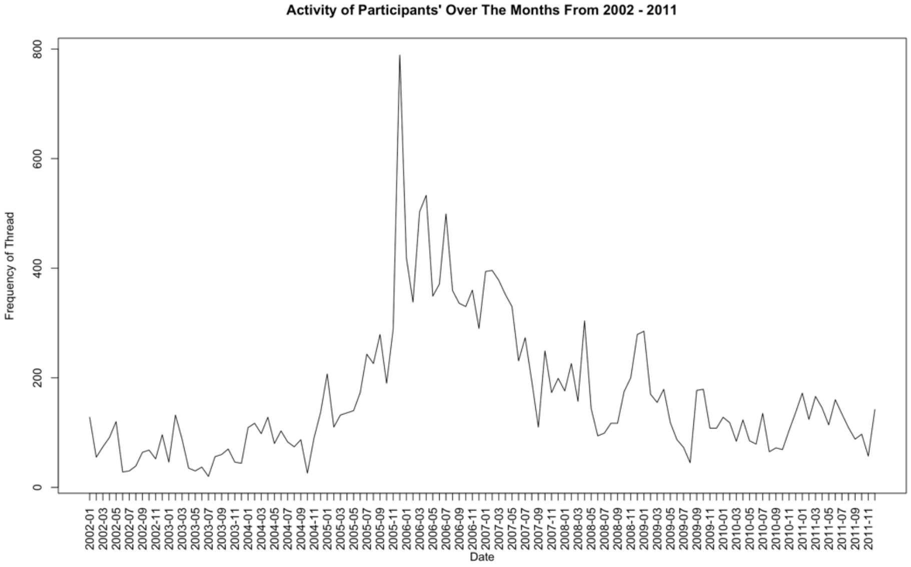
Data – Preprocessing:

Create a variable called YM that has Year-Month format with Date type. Observe the number of threads posted per month then get the total number of them per month and save it as a data frame called total_month.

Moreover, the calculations to get the average threads per year are calculated by getting the total number of threads per year and divide them by 12. Hence, saved it to a data frame called total_year.

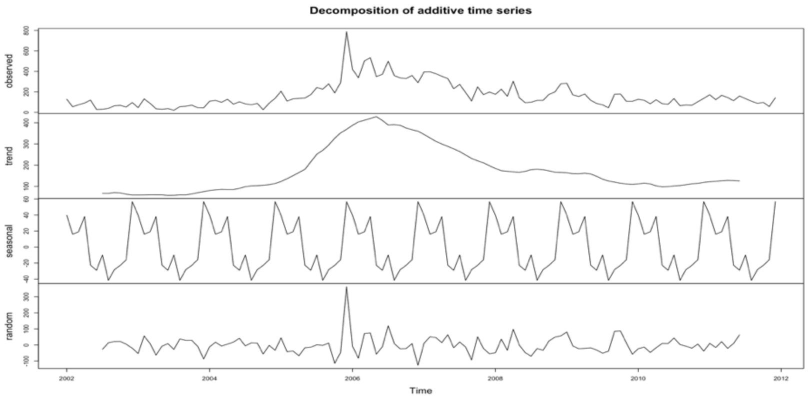
Analysis:

The term active is interpreted as the number of threads per month on the online forum.



Graph 1.1.1

Based on the graph, the participants were quite active over the periods, the number of threads fluctuated from 2002 to the end of 2004 in an upward trend with the average of around 55 until 94 threads per year and immediately soared up in January 2005. In December 2005, it reached its peak with almost 800 threads. Furthermore, the most active year was in 2006 with the average of 390 threads followed by 273 threads in the following year. From that point onwards, even though there were some fluctuations happening, the participants' activity started to decline until August 2009. Then, it started to have a significant increase in the following month. From that month to the end of the observed periods, the number of threads fluctuated around the range of 50 to 180 threads per month.



Graph 1.1.2

By looking at the trend graph from decomposed time series graph, there was not any trend over time as the participants' activity increased significantly from 2002 to the end of 2005, followed by a decreasing trend in the following year to the end of the observed period. On the other hand, the seasonal graph gives a crucial information such that for every year, participants are the least active in the middle of the year and mostly active at the end of the year.

A2

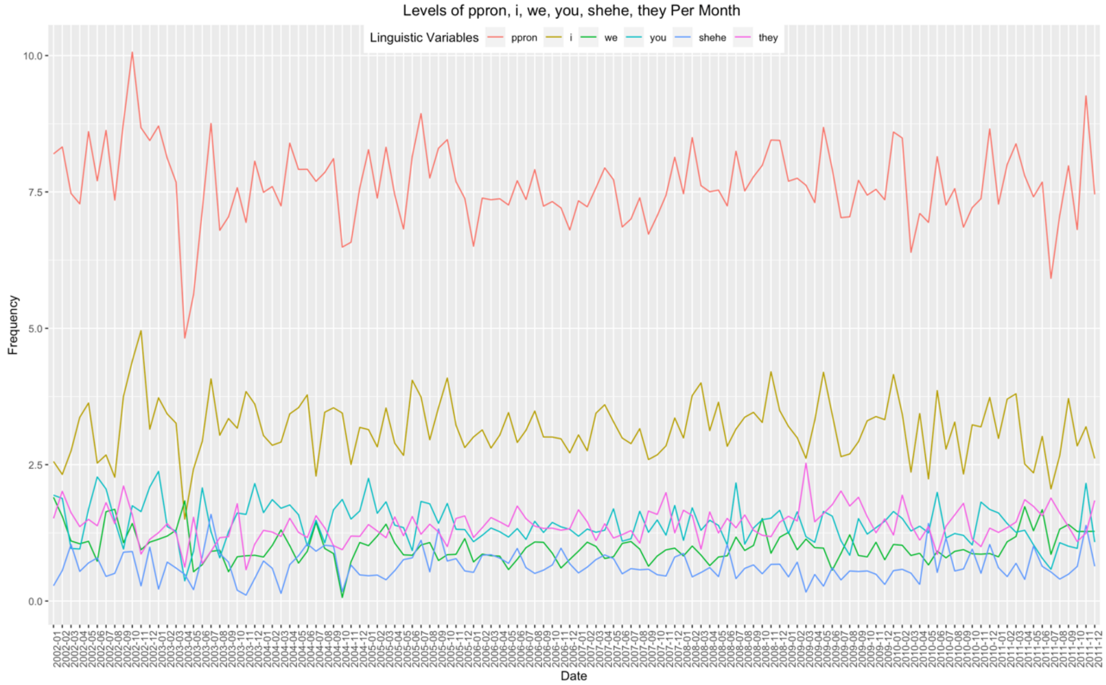
Data – Preprocessing:

Create avg_wc_summary (WC, Authentic, Analytic, Tone, Clout), avg_pron (ppron, i, we, you, shehe, they), avg_emo (posemo, negemo, anx, anger, sad), avg_focus (focuspast, focuspresent, focusfuture), avg_tone (Tone, posemo, negemo), avg_all (all the linguistic variables) variables that hold the average of all linguistic variables per month.

Analysis:

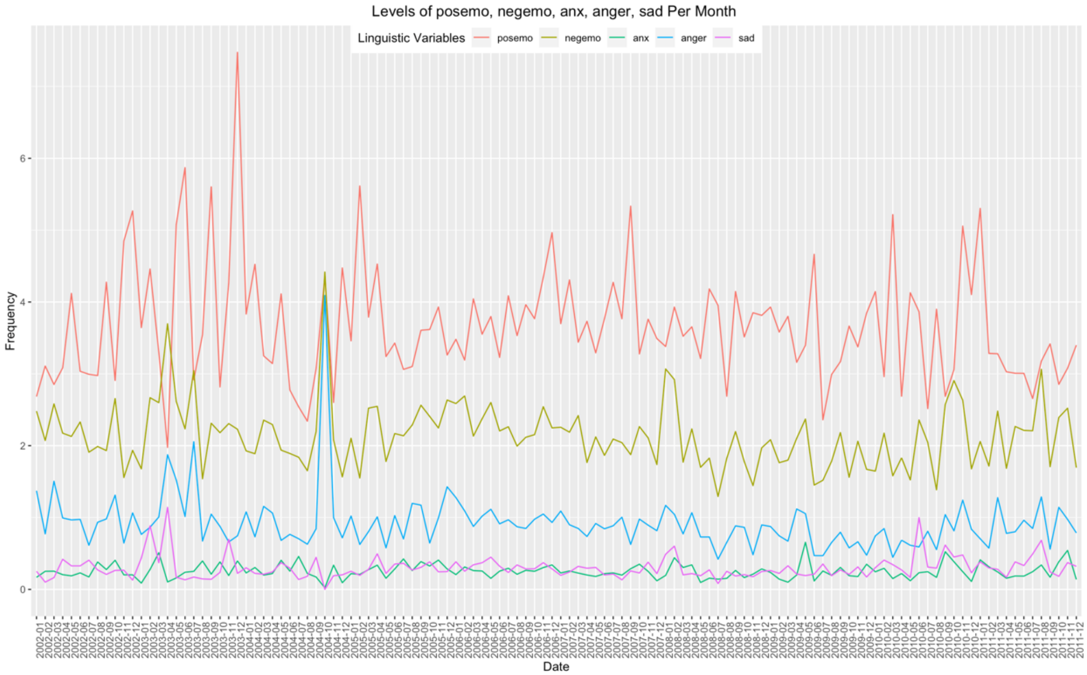
Having one or two graphs for each question is desirable, but for this question to have a better understanding and visibility of all linguistic variables, 5 graphs are created.

The levels of all the linguistic variables did change over time based on graph 1.2.1 until 1.2.4



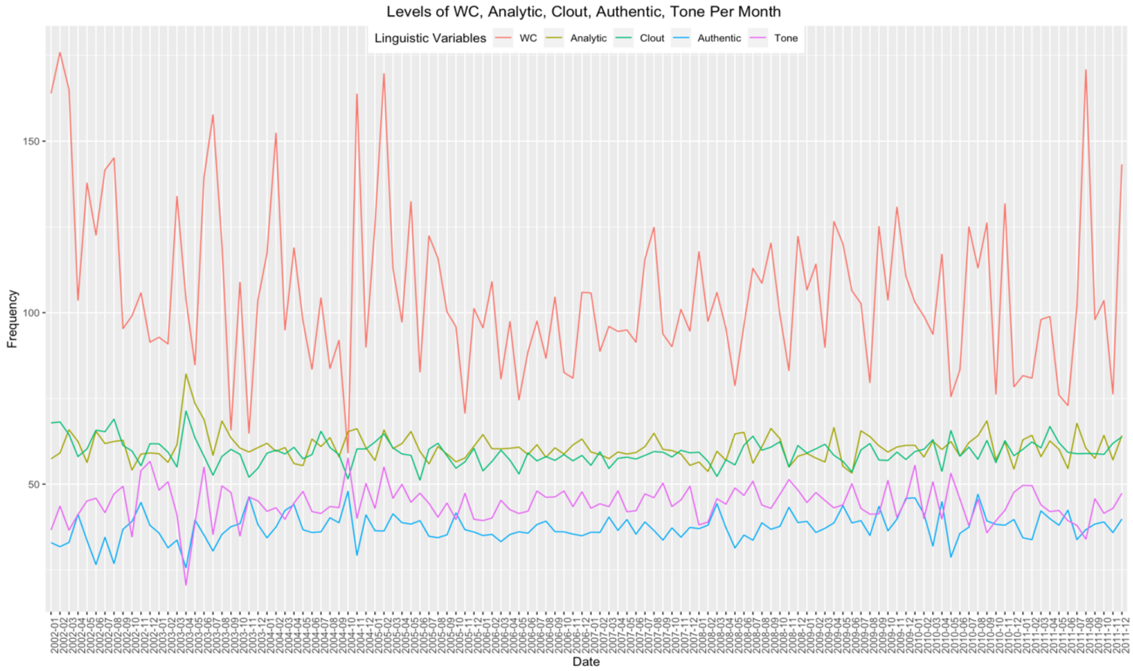
Graph 1.2.1

Over the longer term, there is a relationship between the linguistic variables. For graph 1.2.1, ppron had a similar trend with the other variables in that graph. It is definite that ppron has relationship with **i**, **we**, and **you** linguistic variables as ppron includes those words. Other than that, to prove that they do have a relationship with each other is by using summary of linear model. The significant level that is being used is α = 0.01 to have higher certainty of the relationships between linguistic variables. **i**, **we**, **you**, **shehe**, **they** variables have p-value less than 2e-16. Since this value is less than the significant level, they have a statistically significant relationship with ppron variable.



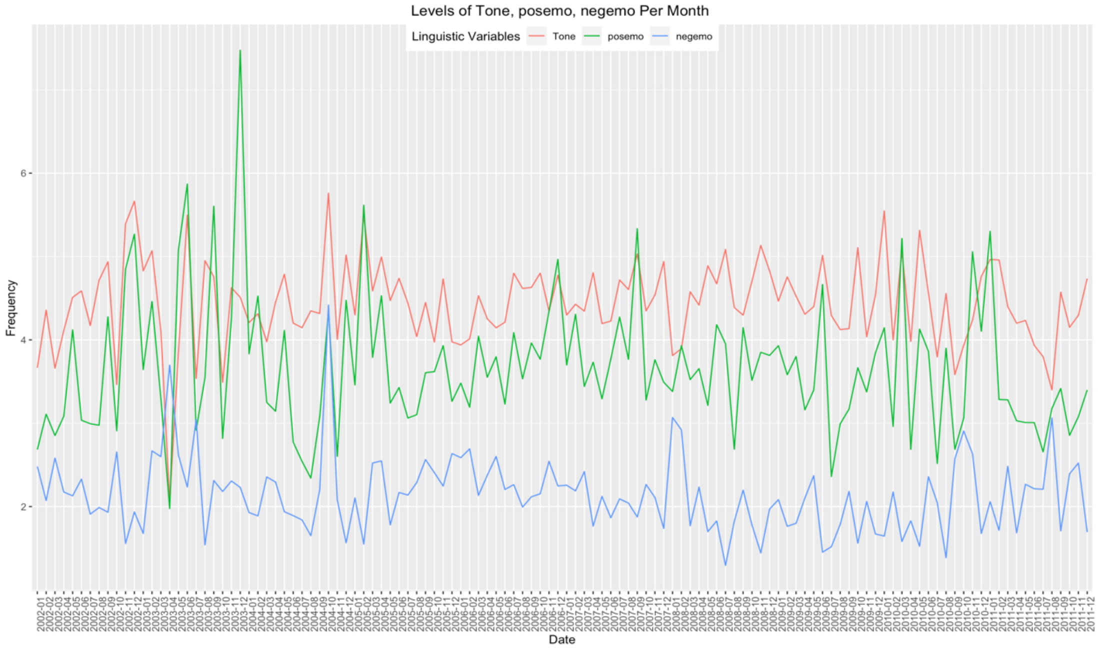
Graph 1.2.2

Furthermore, negemo expresses negative emotions, that are: anxiety, anger, and sadness. From graph 1.2.2, it is noticeable that negemo trend moved similarly with anxiety, sadness and in particular anger variables. Using the same techniques of the summary of linear model, the mentioned variables have p-values less than the significant level. Thus, negemo has a relationship with anxiety, anger, and sadness variables.



Graph 1.2.3

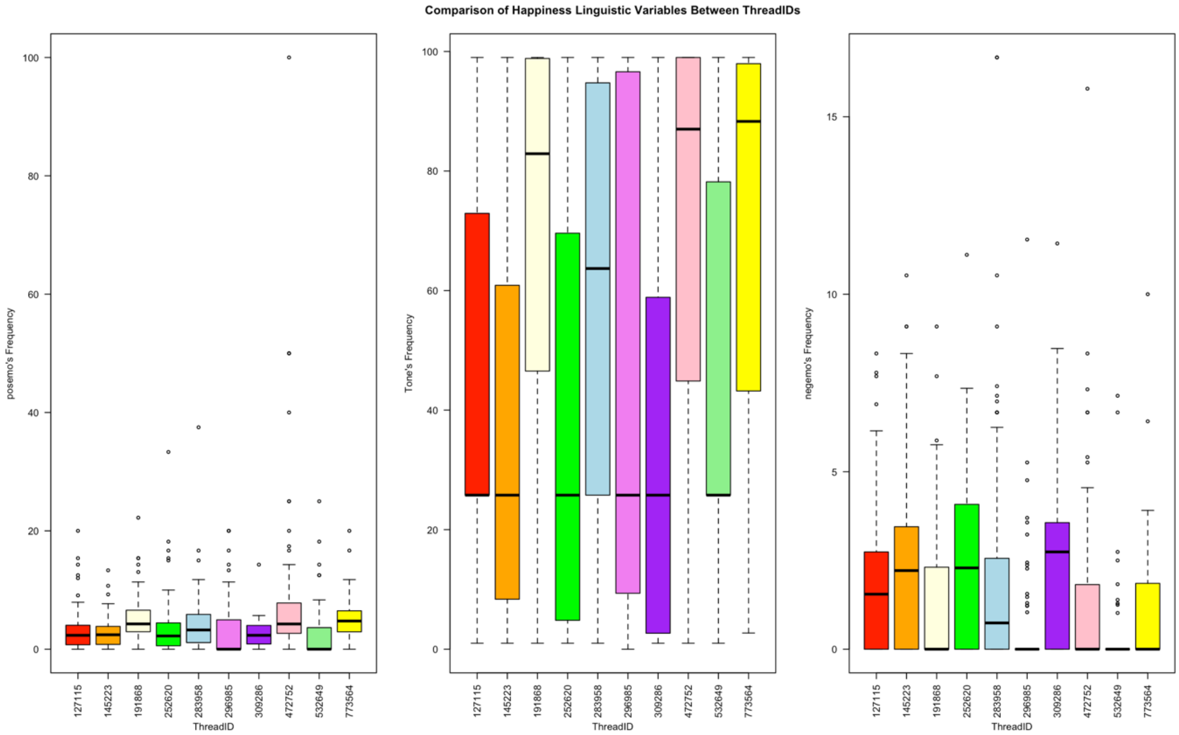
In addition, analytic variable seems to have a relationship with clout from graph 1.2.3. It can be seen from the peak of both variables that went in the same direction. From the summary of analytic linear model, clout has a p-value less than 0.01. Therefore, Clout and Analytic have a relationship with each other.



Graph 1.2.5

Moreover, another interesting part that was observed from the summary of negemo linear model is Tone variable. It also has a p-value less than 0.01, meaning that negemo has a relationship with Tone. To prove this, create a summary of linear model for Tone variable. It is discovered that not only negemo, but posemo variable is also related to Tone by having a p-value of 9.83e-11. By looking at graph 1.2.5, it is very apparent that Tone's trend over time went along alike negemo and posemo variables. Hence, Tone has a significant relationship with those two variables.

B1



Graph 2.1

Data – Preprocessing:
Firstly, count the number of ThreadIDs from 2002 – 2011. Then, take the top 10 ThreadIDs that has the highest number of threads in the forum and save it to a data frame called top10_threadID.

After that, select only ThreadID, posemo, negemo, and Tone columns from top10_threadID and save it again to that data frame.

Analysis:
The Top 10 most ThreadIDs are being chosen because those threads are assumed to have better data to be analyzed other than those who have lesser threads. To see which ThreadIDs are happier than other threads, posemo linguistic variable can be used as a guidance as it expresses positive emotions, for instance, happiness.

In addition, to choose which other related variables to posemo, the summary of posemo linear model should be used. Based on the summary of the linear model, negemo and Tone both are related to posemo as they have p-values less than the significant level of 0.01. Thus, posemo are related to both negemo and Tone.

Both posemo and Tone boxplot show that either 472752 or 773564 thread ids is the happiest than the other threads, as both of their medians are higher than the others. Moreover, based on the negemo boxplot, their median has the value 0 even though there are some negemo involved in those threads. To prove, which ones is the happiest threads, t-test will be chosen as hypothetical test.

The t-test compare the value of the linguistic variables of one of the ID to all the other IDs. Then, a data frame to compare the p-values for posemo, negemo, and Tone variables for both IDs (472752 and 773564) are created to show comparisons between them. It can be seen that thread ID of 472752 has smaller p-values for both posemo and Tone variables, and those p-values are less than the significant level of 0.01. This means that the evidence is sufficient to say that thread ID of 472752's posemo and Tone variables are greater than the other observed thread IDs. Thus, it is concluded that thread ID of 472752 is the happiest.

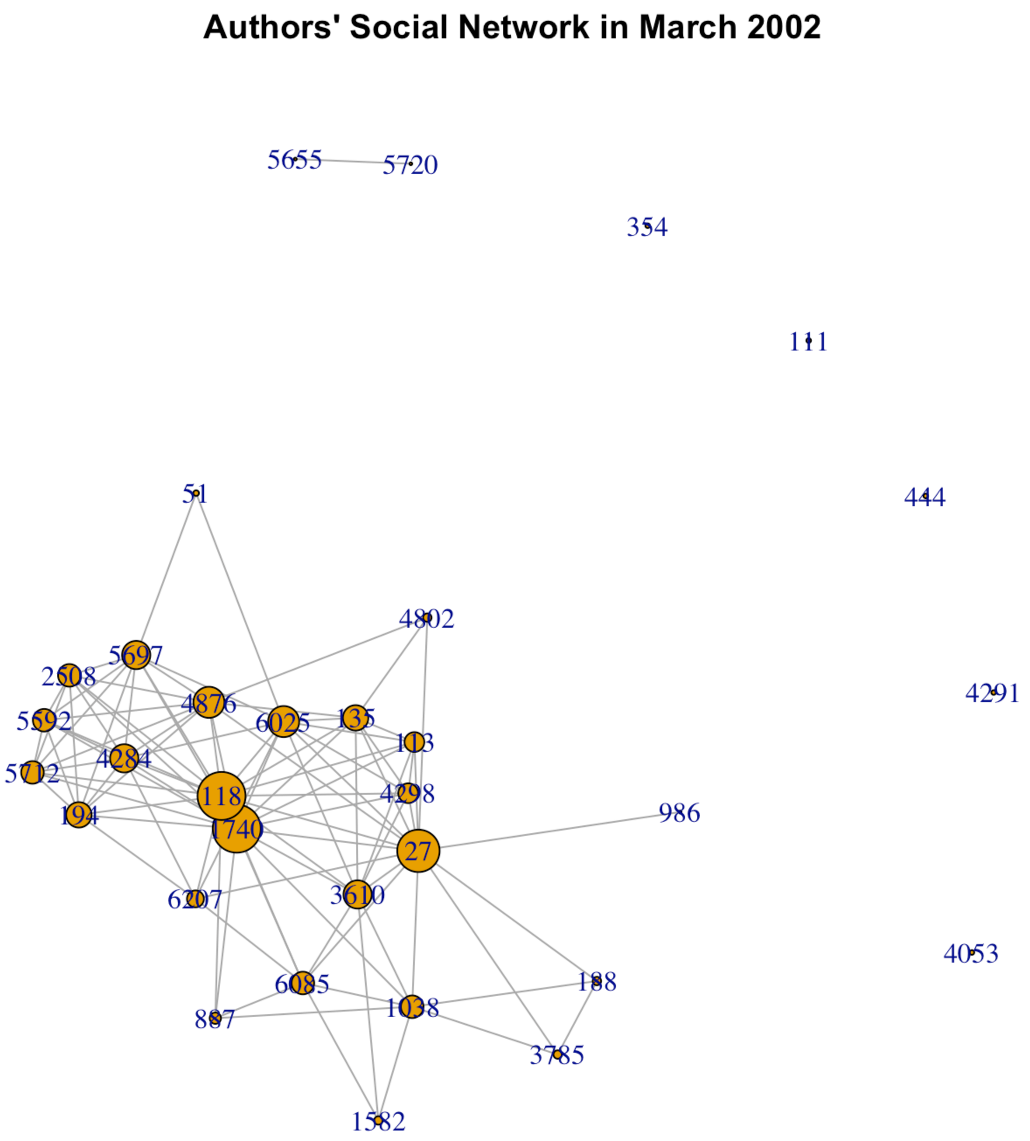
Therefore, it is possible to see whether particular threads are more optimistic than the others by observing the boxplot and hypothetical t-test.

C1

Data – Preprocessing:
Choose March 2002 as the time frame to be observed and get all the columns from webforum, save it to a social_network variable. Then, check if in that period there are more than 30 authors contributed. The number of authors contributed during that time frame are 32. After that, investigate if some authors have posted more than one thread. As March 2002 has surpassed all criteria mentioned in the question, it can be used as the time frame to be analyzed.

In addition, only ThreadID and AuthorID columns are involved in the social network graph. Thus, choose only those columns and save it into a data frame called thread_author. Besides that, it is needed to have the unique author IDs to be saved into another data frame called unique_authorID. Thus, the network graph can be created from both of those data frame.

Analysis:



Graph 3.1

The size of the circle of each authors' ID is different based on how strong the relationship between one author and another. Those who are not connected means that they are not contributing to the same threads as the other authors.

C2

Data – Preprocessing:

To compare the linguistic variables of the most important author in the social network. Take the AuthorID and all the other linguistic variables and save it as a data frame called comparison

Analysis:

Firstly, measure the graph's degree, closeness, eigenvector, and betweenness. For closeness measurement, the top 3 vertices which are the closest to all the other nodes in the network are 26, 29, and 1. Furthermore, betweenness measurement shows us that vertices 3, 1, 2 are in the top 3 list of it. Moreover, the vertices 1,2,3 consequently have the highest number of connections with other vertices. Lastly, the top 3 vertices that are more central in the graph are 2,1, and 3.

From the previous vertex importance measurement, it can be concluded that the author with vertices 1 is the most important author in the social network. After calculating it at the R Script, it is known that author ID of 1740 is the most important author.

To analyze why author ID of 1740 is the most important author is by their linguistic variables. It is discovered by using t-test that when posting threads, author ID of 1740 does not express his/her negative emotions and anxiety as much as the others. In addition, he/she also rarely used 'they' in his/her threads unlike the others. All the t-tests are reliable, as all the p-values are less than the significant level of 0.01. Therefore, the evidence is sufficient to support the statement that when posting threads, author ID of 1740 seldomly used 'they', expressing his/her negative emotions and anxiety unlike the others. That is why, he/she is the most important author in the social network.

Conclusion

From this assignment, the purpose of learning more about time series graph, box plots, social-network graph, hypothetical test, and linear model is achieved. In addition, it teaches how to analyze the graphs in a whole and using t-test or linear model to justify the analysis. Lastly, the three main components: activity, language, and social networks by participants have been analyzed thoroughly.

Appendix

```
# Set Working Directory Based on your devices
getwd()
setwd("/Users/Juliet/Documents/Monash_Uni/FIT3152/Assignment/1")

# Install Packages (If you do not have not these packages)
install.packages("ggplot2")
install.packages("reshape2")
install.packages("dplyr")

# Load the Packages
library(ggplot2)
library(reshape2)
library(dplyr)

# Create individual data based on requirements
rm(list =ls())
set.seed(321112602) # XXXXXXXX = your student ID
webforum <- read.csv("webforum.csv")
webforum <- as.data.frame(webforum [sample(nrow(webforum), 20000), ]) # 20000 rows

# A1
# Create a column that has Year Month Value with Date Format
YM <- format(as.Date(webforum$Date), "%Y-%m-%d")
YM <- format(as.Date(YM), "%Y-%m")
YM <- as.Date(paste(YM,"-01",sep=""))

# Count the number of participants' activities per month
tab_month <- table(cut(YM, 'month'))
total_month <- data.frame(Date = format(as.Date(names(tab_month)), "%Y-%m"), frequency = as.vector(tab_month))

# Count the average of the number of participants' activities per year
tab_year <- table(cut(YM, 'year'))
total_year <- data.frame(Date = format(as.Date(names(tab_year)), "%Y"), frequency = as.vector(tab_year))
total_year$frequency <- total_year$frequency/12
colnames(total_year) <- c("Date","AverageThreads")
total_year
```

```
# Order total_month to see the start date that is 2002-01
total_month[order(total_month$Date),]

# Plot time series graph
series <- ts(total_month$frequency, frequency = 12, start = c(2002))
plot(series, xaxt = "n", main = "Activity of Participants' Over The Months From
2002 - 2011", xlab = "", ylab = "Frequency of Thread")
tsp <- attributes(series)$tsp
dates <- seq(as.Date("2002-01-01"), by = "month", along = series)
axis(1, at = seq(tsp[1], tsp[2], along = series), labels = format(dates, "%Y-%m"),las
= 2)
mtext("Date", side=1, line= 4.1)
# Plot time series decomposition graph
decomposed_series <- decompose(series)
plot(decomposed_series)

#A2
# Average Per Month for WC, Analytic, Clout, Authentic, Tone
avg_wc_summary <- aggregate(cbind(webforum$WC, webforum$Analytic,
webforum$Clout, webforum$Authentic, webforum$Tone),
list(format(as.Date(webforum$Date), "%Y-%m")), mean)
colnames(avg_wc_summary) <- c("Date","WC",
"Analytic","Clout","Authentic","Tone")

# Average Per Month for ppron, i, we, you, shehe, they
avg_pron <- aggregate(cbind(webforum$ppron, webforum$i, webforum$we,
webforum$you, webforum$shehe, webforum$they) ,
list(format(as.Date(webforum$Date), "%Y-%m")), mean)
colnames(avg_pron) <- c("Date","ppron","i","we","you","shehe","they")

# Average Per Month for posemo, negemo, anx, anger, sad
avg_emo <- aggregate(cbind(webforum$posemo, webforum$negemo,
webforum$anx, webforum$anger, webforum$sad) ,
list(format(as.Date(webforum$Date), "%Y-%m")), mean)
colnames(avg_emo) <- c("Date","posemo","negemo","anx","anger","sad")

# Average Per Month for focuspast, focuspresent, focusfuture
avg_focus <- aggregate(cbind(webforum$focuspast, webforum$focuspresent,
webforum$focusfuture) , list(format(as.Date(webforum$Date), "%Y-%m")), mean)
colnames(avg_focus) <- c("Date","focuspast","focuspresent","focusfuture")

# Average Per Month for Tone, posemo, negemo
```

```
avg_tone <- aggregate(cbind((webforum$Tone/10), webforum$posemo,
webforum$negemo) , list(format(as.Date(webforum$Date), "%Y-%m")), mean)
colnames(avg_tone) <- c("Date","Tone","posemo","negemo")

# Average Per Month for All Linguistic Variables
avg_all <- aggregate(cbind(webforum$WC,webforum$Analytic, webforum$Clout,
webforum$Authentic, webforum$Tone,webforum$ppron, webforum$i,
webforum$we, webforum$you, webforum$shehe,
webforum$they,webforum$posemo, webforum$negemo, webforum$anx,
webforum$anger, webforum$sad,webforum$focuspast, webforum$focuspresent,
webforum$focusfuture) , list(format(as.Date(webforum$Date), "%Y-%m")), mean)
colnames(avg_all) <-
c("Date","WC","Analytic","Clout","Authentic","Tone","ppron","i","we","you","sh
ehe","they","posemo","negemo","anx","anger","sad","focuspast","focuspresent","f
ocusfuture")

# Graph for Levels of WC, Analytic, Clout, Authentic, Tone Per Month
melt_wc_summary <- melt(avg_wc_summary,id ="Date")
wc_summary <- ggplot(melt_wc_summary,aes(x = Date,y = value,colour = variable,
group = variable)) + geom_line()
wc_summary <- wc_summary + labs(title = "Levels of WC, Analytic, Clout,
Authentic, Tone Per Month ", x = "Date", y = "Frequency", fill =
guide_legend(title="Linguistic Variables"))
wc_summary <- wc_summary + theme(plot.title = element_text(hjust = 0.5),
axis.text.x = element_text(angle = 90), legend.position = c(0.5, 0.98),
legend.direction="horizontal")
wc_summary <- wc_summary +  guides(color=guide_legend(title="Linguistic
Variables"))
wc_summary

# Graph for Levels of ppron, i, we, you, shehe, they
melt_pron <- melt(avg_pron,id ="Date")
pron <- ggplot(melt_pron,aes(x = Date,y = value,colour = variable, group = variable))
+ geom_line()
pron <- pron + labs(title = "Levels of ppron, i, we, you, shehe, they Per Month ", x
= "Date", y = "Frequency", fill = guide_legend(title="Linguistic Variables"))
pron <- pron + theme(plot.title = element_text(hjust = 0.5), axis.text.x =
element_text(angle = 90), legend.position = c(0.5, 0.98),
legend.direction="horizontal")
pron <- pron +  guides(color=guide_legend(title="Linguistic Variables", nrow = 1))
pron
```

```
# Graph for Levels of posemo, negemo, anx, anger, sad
melt_emo <- melt(avg_emo,id ="Date")
emo <- ggplot(melt_emo,aes(x = Date,y = value,colour = variable, group = variable))
+ geom_line()
emo <- emo + labs(title = "Levels of posemo, negemo, anx, anger, sad Per Month ",
x = "Date", y = "Frequency", fill = guide_legend(title="Linguistic Variables"))
emo <- emo + theme(plot.title = element_text(hjust = 0.5), axis.text.x =
element_text(angle = 90), legend.position = c(0.5, 0.98),
legend.direction="horizontal")
emo <- emo +  guides(color=guide_legend(title="Linguistic Variables", nrow = 1))
emo

# Graph for Levels of focuspast, focuspresent, focusfuture
melt_focus <- melt(avg_focus,id ="Date")
focus <- ggplot(melt_focus,aes(x = Date,y = value,colour = variable, group =
variable)) + geom_line()
focus <- focus + labs(title = "Levels of focuspast, focuspresent, focusfuture Per
Month ", x = "Date", y = "Frequency", fill = guide_legend(title="Linguistic
Variables"))
focus <- focus + theme(plot.title = element_text(hjust = 0.5), axis.text.x =
element_text(angle = 90), legend.position = c(0.5, 0.98),
legend.direction="horizontal")
focus <- focus +  guides(color=guide_legend(title="Linguistic Variables", nrow =
1))
focus

# Graph for Levels of Tone, posemo, negemo
melt_tone <- melt(avg_tone,id ="Date")
tone <- ggplot(melt_tone,aes(x = Date,y = value,colour = variable, group = variable))
+ geom_line()
tone <- tone + labs(title = "Levels of Tone, posemo, negemo Per Month ", x = "Date",
y = "Frequency", fill = guide_legend(title="Linguistic Variables"))
tone <- tone + theme(plot.title = element_text(hjust = 0.5), axis.text.x =
element_text(angle = 90), legend.position = c(0.5, 0.98),
legend.direction="horizontal")
tone <- tone +  guides(color=guide_legend(title="Linguistic Variables", nrow = 1))
tone

# Relationship Between Linguistic Variables
# Relationship of Ppron with Other Linguistic Variables
ppron <- lm(ppron ~ .-Date, data = avg_all)
summary(ppron)
```

```
# Relationship of Negemo with Other Linguistic Variables
negemo <- lm(negemo ~ . -Date, data = avg_all)
summary(negemo)

# Relationship of Tone with Other Linguistic Variables
tone <- lm(Tone ~ . -Date, data = avg_all)
summary(tone)

# Relationship of Analytic with Other Linguistic Variables
analytic <- lm(Analytic ~ . -Date, data = avg_all)
summary(analytic)

# B
# Analyse Top 10 Most Thread In The Forum
# Get the total number of each thread in the forum
unique_threadID <- as.data.frame(tally(group_by(webforum, ThreadID)))
colnames(unique_threadID) <- c("ThreadID", "Count")

# Select top 10 ThreadID that have the most number of threads
unique_threadID <- unique_threadID[order(unique_threadID$Count, decreasing =
TRUE),]
top10_threadID <- unique_threadID[1][1:10,]
top10_threadID <- webforum[webforum$ThreadID %in% top10_threadID,]

# To Compare the Happiness of Each Thread, use posemo as our Independent
Variable
posemo <- lm(posemo ~ . -Date-Time-ThreadID-AuthorID, data = top10_threadID)
summary(posemo)

# Take only ThreadID, posemo, negemo, Tone columns
top10_threadID <- top10_threadID[, c('ThreadID', 'posemo','negemo','Tone')]

# Create a Box Plot for posemo, Tone, negemo
op <- par(mfrow=c(1, 3)) ## set par
boxplot(top10_threadID$posemo~top10_threadID$ThreadID,las = 2, xlab = "",
ylab = "posemo's Frequency", col = c("red","orange","light yellow","green","light
blue","violet","purple","pink","light green","yellow"))
title(xlab = "ThreadID", line = 4)
boxplot(top10_threadID$Tone~top10_threadID$ThreadID,las = 2, main =
"Comparison of Happiness Linguistic Variables Between ThreadIDs",xlab = "", ylab
= "Tone's Frequency", col = c("red","orange","light yellow","green","light
blue","violet","purple","pink","light green","yellow"))
title(xlab = "ThreadID", line = 4)
boxplot(top10_threadID$negemo~top10_threadID$ThreadID,las = 2, xlab = "",
ylab = "negemo's Frequency", col = c("red","orange","light yellow","green","light
blue","violet","purple","pink","light green","yellow"))
title(xlab = "ThreadID", line = 4)
par(op)

## Statistical Test
# posemo
t.test(top10_threadID[2][top10_threadID$ThreadID == 472752,],
top10_threadID[2][top10_threadID$ThreadID != 472752,], "greater", conf.level =
0.99)
p1 <- 0.0003868
t.test(top10_threadID[2][top10_threadID$ThreadID == 773564,],
top10_threadID[2][top10_threadID$ThreadID != 773564,], "greater", conf.level =
0.99)
p2 <- 0.01437

# negemo
t.test(top10_threadID[3][top10_threadID$ThreadID == 472752,],
top10_threadID[3][top10_threadID$ThreadID != 472752,], "less", conf.level = 0.99)
n1 <- 0.01739
t.test(top10_threadID[3][top10_threadID$ThreadID == 773564,],
top10_threadID[3][top10_threadID$ThreadID != 773564,], "less", conf.level = 0.99)
n2 <- 0.01162

# Tone
t.test(top10_threadID[4][top10_threadID$ThreadID == 472752,],
top10_threadID[4][top10_threadID$ThreadID != 472752,], "greater", conf.level =
0.99)
t1 <- 5.479e-09
t.test(top10_threadID[4][top10_threadID$ThreadID == 773564,],
top10_threadID[4][top10_threadID$ThreadID != 773564,], "greater", conf.level =
0.99)
t2 <- 3.015e-06

# Create data frame to compare the p-value for 472752 and 773564 ThreadID for the
observed linguistic variable
compare <- data.frame(ThreadID = c(472752, 773564), posemo = c(p1,p2), negemo
= c(n1,n2), Tone = c(t1,t2))
```

```
compare

# C1
# Install Packages
install.packages("igraph")
install.packages("igraphdata")

# Load Packages
library(igraph)
library(igraphdata)

# Choose over a month (2002-03) online activity
social_network <- webforum[webforum$Date >= as.Date("2002-03-01") & webforum$Date <= as.Date("2002-03-31"),]
# Check whether it has more than 30 authors
length(unique(social_network$AuthorID))
# Check if 1 author posted more than one thread
social_network[order(social_network$AuthorID),]

# Take only ThreadID and AuthorID columns
thread_author <- as.data.frame(social_network[,1:2])
colnames(thread_author) <- c("ThreadID","AuthorID")

# Create a separate data frame to store distinct author IDs
unique_authorID <- as.data.frame(unique(thread_author$AuthorID))
colnames(unique_authorID) <- "AuthorID"

# Make an empty graph
g <- make_empty_graph(directed = FALSE)

# Add vertices using “for loop”
for (i in 1 : nrow(unique_authorID)) {
  g <- add_vertices(g, 1, name = as.character(unique_authorID$AuthorID[i]))
}

# Loop through each Thread
for (k in unique(thread_author$ThreadID)){
temp = thread_author[(thread_author$ThreadID == k),]
 # Combine related Author IDs to make an edge list
if (nrow(temp) > 1) {
Edgelist = as.data.frame(t(combn(temp$AuthorID,2)))
colnames(Edgelist) = c("P1","P2")
for (i in 1:nrow(Edgelist)){
  g <- add_edges(g,c(as.character(Edgelist$P1[i]),as.character(Edgelist$P2[i])))
}}}

# Plot the graph
plot(g)

# Because there are some loop in the graph, we need to simplify the graph
g <- simplify(g)

# Plot the graph once more. It is noticeable that the size of the circle covered the
other author IDs
plot(g)

# Resize the size of the circle depending on the importance of each author ID
deg <- degree(g)
V(g)$size <- deg/1.7

# Plot the final social network graph
plot(g, main = "Authors' Social Network in March 2002")

# C2
# Which authors are the most important across the network graph?
order(closeness(g), decreasing = TRUE)
order(betweenness(g), decreasing = TRUE)
order(degree(g), decreasing = TRUE)
order(evcent(g)$vector, decreasing = TRUE)

# Most important Author for the graph is the one who has vertices 1 with 1740 as
his/her AuthorID
V(g)[1]

# Observed the language the most important authors than the others.
# Take the AuthorID and all of the other linguistic variables and save it as a data
frame of comparison
comparison <- as.data.frame(social_network[, c(2,5:23)])

# Analyze LIWC Variables that makes 1740 the most important author
# Statistical t-test
# they
t.test(comparison[12][comparison$AuthorID == 1740,],
comparison[12][comparison$AuthorID != 1740,], "less", conf.level = 0.99)
```

```
# negemo
t.test(comparison[14][comparison$AuthorID == 1740,],
comparison[14][comparison$AuthorID != 1740,], "less", conf.level = 0.99)
# anx
t.test(comparison[15][comparison$AuthorID == 1740,],
comparison[15][comparison$AuthorID != 1740,], "less", conf.level = 0.99)
```