Assignment 1

Farayha Zaheer Alam

2022-04-26

will show the required code, analysis, and the visualization (scatterplots, graphs, etc) with respect to each of the question.

Loaded the following required libraries:

```
library("magrittr")
library("lubridate")
library("extrafont")
#library("hrbrthemes")
library("igraph")
library("ggplot2")
library(extrafont)
loadfonts(device = "win")
library(dplyr)
library(reshape2)
library(tidyr)
library(tidyverse)
library(plyr)
library(igraph)
library(igraphdata)
```

Pre-processing and Reading/extracting the required web forum data:

```
rm(list = ls())  #Clear the working environment
set.seed(31164943)
webforum <- read.csv("webforum.csv")
webforum <- webforum [sample(nrow(webforum), 20000), ] # 20000 rows extracted
```

Part A

Analyse activity and language on the forum over time:

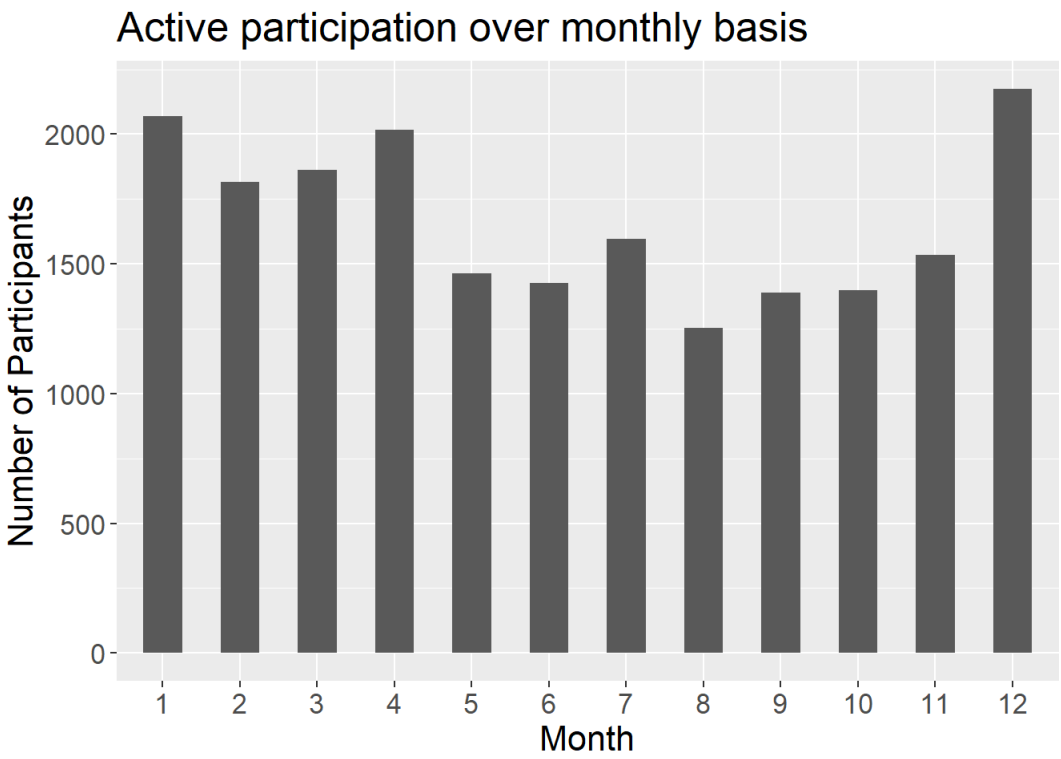
Pre-processing for monthly basis:

First of all, I will preprocess the existing data and include an attribute named “Month” for later analysis how active participants are on monthly basis.

```
Month <- month(as.POSIXlt(webforum$Date, format="%Y-%m-%d"))
webforum_new <- cbind(webforum, Month) # binding the "Month" column to new dataframe named 'webforum_new'
```

Visual Representation of Frequency of Active Participants on Monthly Basis:

```
webforum_new %>% ggplot( aes(x=as.factor(Month))) +
  stat_count(width=0.5) +  ggtitle("Active participation over monthly basis") +
  theme(
    text=element_text(size=16, family="TT Courier New")
  ) + labs(x="Month", y="Number of Participants")
```



The overall distribution is random, as seen in the histogram above, with a significant decrease noticed throughout the first 5-6 months and the lowest participation observed in the 8th month.Following that, the graph begins to surge again, with the highest participation in week 12. Overall, the linear trend line shows a decrease in the participation of authors over the observed period.

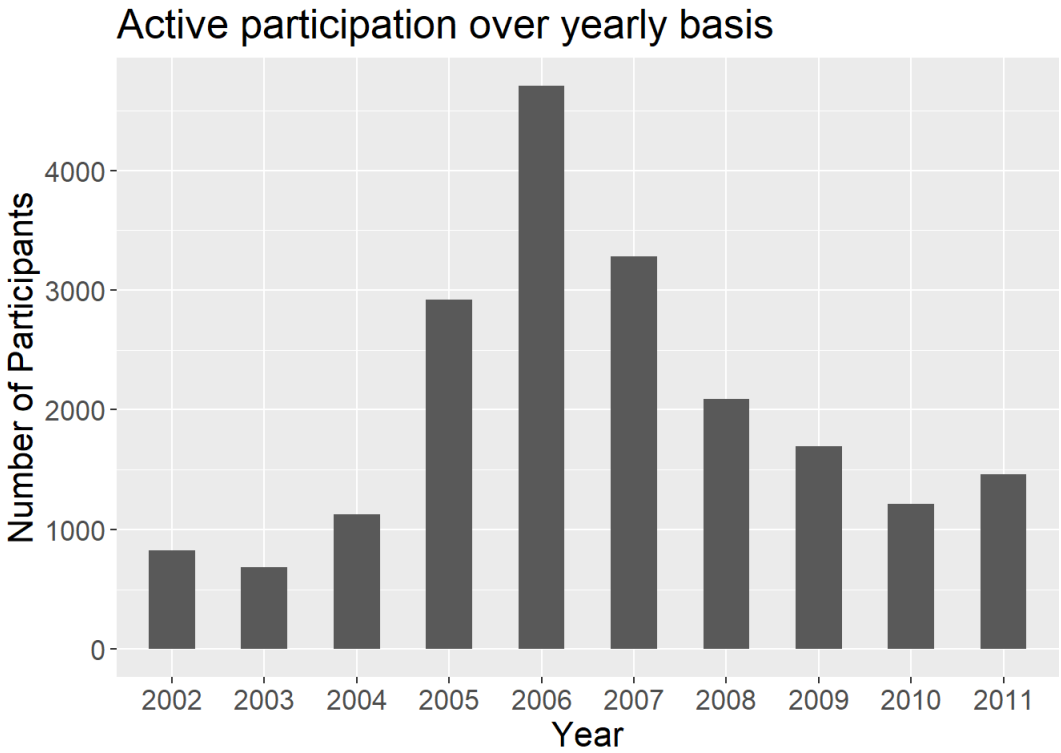
Pre-processing for yearly basis:

Moreover, another approach which can be adapted for this case is to check for the frequency of participation on the yearly basis. Thus, we can include an attribute named “Year” for later analysis how active participants are on yearly basis.

```
Year <- year(as.POSIXlt(webforum$Date, format="%Y-%m-%d"))
webforum_new <- cbind(webforum, Year)  #binding the "Year" column to new dataframe named 'webforum_new'
```

Visual Representation of Frequency of Active Participants on Yearly Basis:

```
webforum_new %>% ggplot( aes(x=as.factor(Year))) +
  stat_count(width=0.5) + ggtitle("Active participation over yearly basis") +
    theme(
    text=element_text(size=16, family="TT Courier New")
  ) + labs(x="Year", y="Number of Participants")
```



The overall distributions of the observations, as seen in the graph above, are normal, displaying as a bell-shaped figure. At first, there is a gradual rise in activity, with the largest peak in 2006, and then the trend line falls as the authors’ engagement rapidly diminishes.

The change in level of linguistic variables over the duration of the forum:

Pre-processing data:

Since the number of linguistic variable is in a huge number so plotting them all on one graph won't give a clear image of how the change in linguistic variable's respective value over a period of time. First of all, we'll extract the columns named "WC","Analytic", "Clout","Authentic",and "Tone" which will be gathered together under a variable named "linguistic_variable".Then, later on only 200 first observations will be extracted from each of the column for more closer analysis since noticing any sort of pattern in 20,000 observations is quite a hefty task.

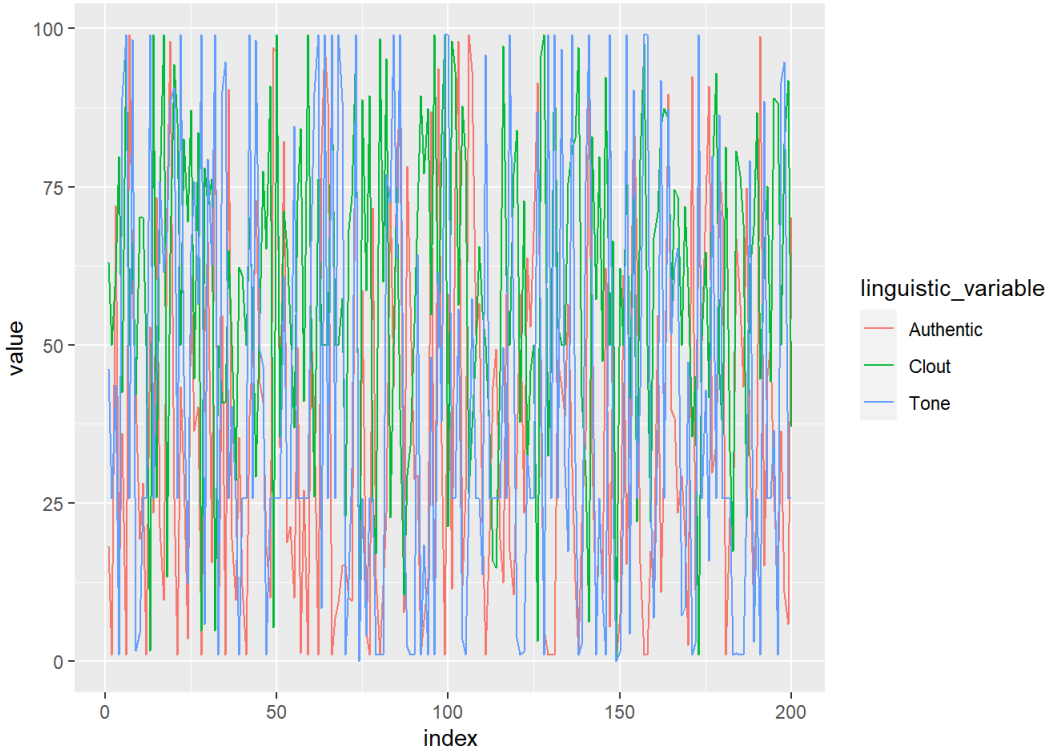
```
result1 = webforum %>% select("WC","Analytic", "Clout","Authentic",
        "Tone")

result1=gather(result1, key="linguistic_variable", value="value", 1:5)
wc=result1[1:200,]
wc$index=1:nrow(wc)
analytic=result1[20001:20200,]
analytic$index=1:nrow(analytic)
clout=result1[40001:40200,]
clout$index=1:nrow(clout)
authentic=result1[60001:60200,]
authentic$index=1:nrow(authentic)
tone=result1[80001:80200,]
tone$index=1:nrow(tone)
```

Since the initial plots with all free of the linguistic variable was messy so certain columns of dataframe are assigned to two different dataframes and two plots are made

Visual Representation of levels of change in Authentic, Clout, Tone:

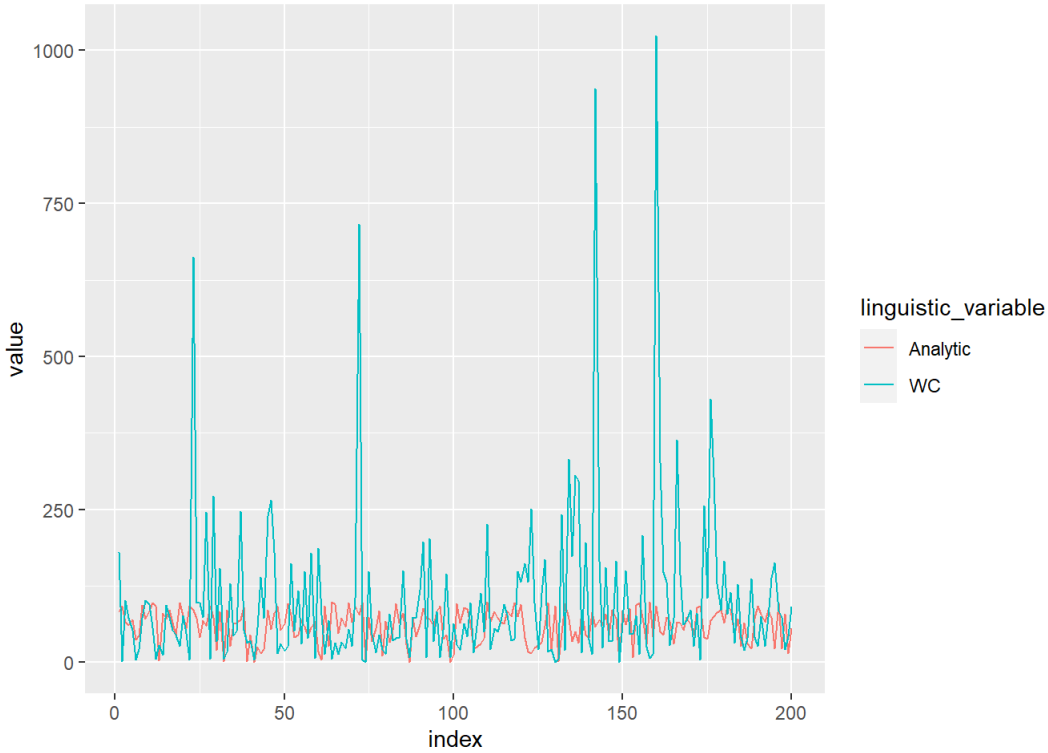
```
result1=rbind(authentic,clout, tone)
result2=rbind(wc, analytic)
ggplot(result1, aes(x=index,y=value, colour= linguistic_variable)) +
  geom_line()
```



200 numbers of dataset are extracted to have a more closer look of the behaviour of the available data with respect to variables. According to the graph above, All three of the linguistic variables are fluctuating throughout the extracted dataset and if u observe the dataset a bit more closely the higher the value of tone of a post in a forum it will have more clout (attention) whilst the authenticity observes the opposite trend for more interval.Thus, overall, both authenticity and clout observe higher values throughout the interval.

Visual Representation of levels of change in WC and Authentic:

```
ggplot(result2, aes(x=index,y=value, colour= linguistic_variable)) +
  geom_line()
```



Same extracted data is observed with respect to analytic and word count variables and according to the graph above,most of the time word count observes higher amount of value over the course of time whilst the analytic is considerably constant with little less to no fluctuation over the same observed dataset and the observed values for analytic are also quite low overall.

Pre-processing data:

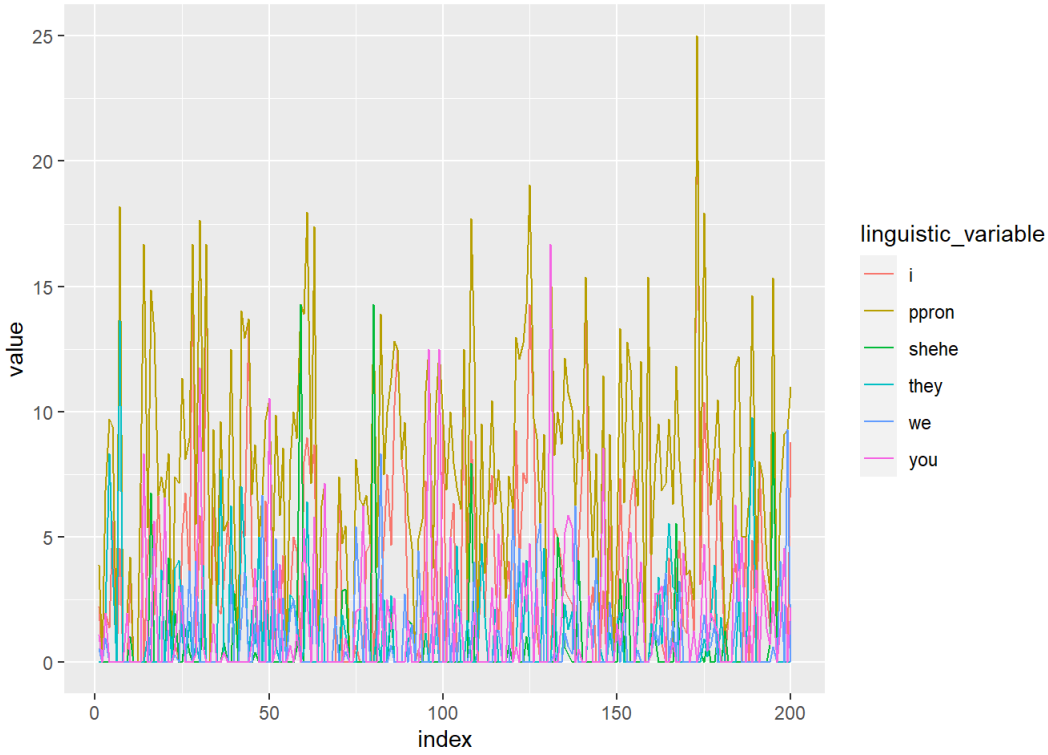
In the code below, ppron, i, we, you, shehe, and they are the linguistic variables which are grouped together and then first 200 observations are extracted in the similar manner.

```
result3 = webforum %>%
  select("ppron","i", "we","you",
         "shehe", "they")


result3=gather(result3, key="linguistic_variable", value="value", 1:6)
ppron=result3[1:200,]
ppron$index=1:nrow(ppron)
i=result3[20001:20200,]
i$index=1:nrow(i)
we=result3[40001:40200,]
we$index=1:nrow(we)
you=result3[60001:60200,]
you$index=1:nrow(you)
shehe=result3[80001:80200,]
shehe$index=1:nrow(shehe)
they=result3[100001:100200,]
they$index=1:nrow(they)
```

Visual Representation of levels of change in ppron, i, we, you, shehe, and hey:

```
result3=rbind(ppron,i,we,you,shehe,they)
ggplot(result3, aes(x=index,y=value, colour= linguistic_variable)) +
  geom_line()
```



On the same extracted data with respect to the variables in key given above (linguistic variables) ppron observes the highest values throughout the interval closely followed by you/i over the same course of index. Meanwhile, sheshe has mostly a downward trend with occasional higher value a times and same is the case with they which only witness 1-2 visible peaks over the same course of time. Lastly, we is one of the linguistic variable which shares quite low values over the forum in the same interval comparatively.

Pre-processing data:

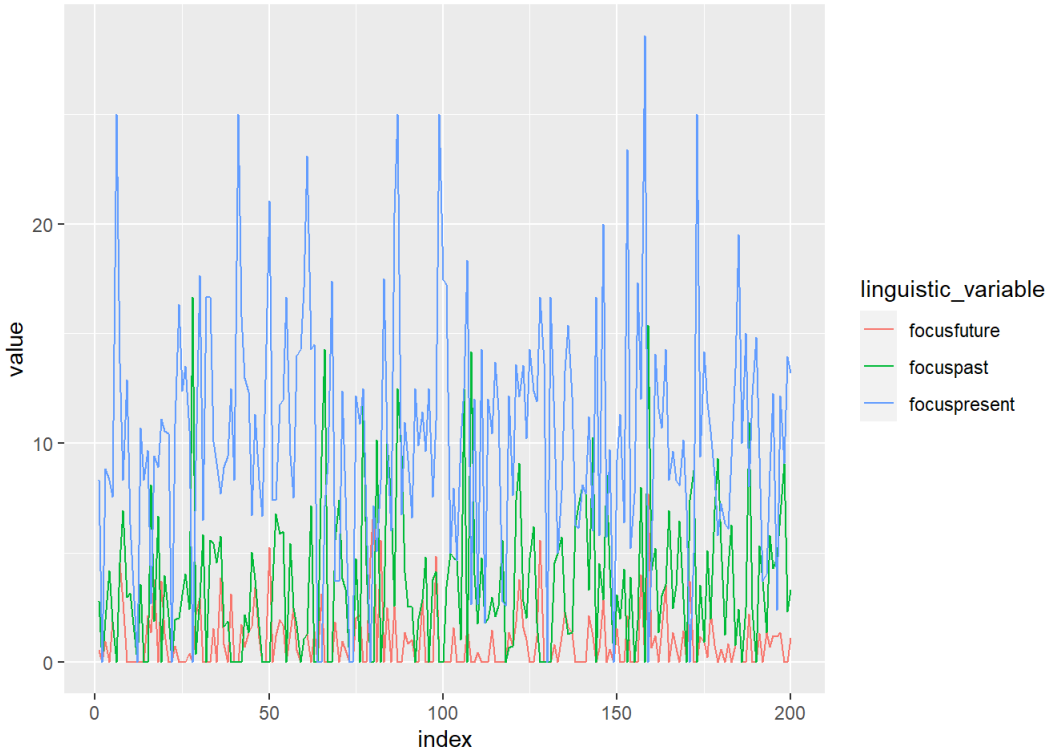
In the code below, focusfuture, focuspresent, and focuspast are the linguistic variables which are grouped together and then first 200 observations are extracted in the same way as above ones.

```
result4 = webforum %>%
  select("focusfuture","focuspresent", "focuspast")

result4=gather(result4, key="linguistic_variable", value="value", 1:3)
focusfuture=result4[1:200,]
focusfuture$index=1:nrow(focusfuture)
focuspresent=result4[20001:20200,]
focuspresent$index=1:nrow(focuspresent)
focuspast=result4[40001:40200,]
focuspast$index=1:nrow(focuspast)
```

Visual Representation of levels of change in focusfuture, focuspresent, and focuspast:

```
result4=rbind(focusfuture,focuspresent, focuspast)
ggplot(result4, aes(x=index,y=value, colour= linguistic_variable)) +
  geom_line()
```



Same extracted data is followed for this observation too and according to the graoh above,focuspast observes mostly a higher value throughout the forum dataset provided with a slight downfall during the end. Meanwhile, focuspast observes a mediocre value throughout the course of 200 dataset of forum being used and focusfuture is on the bottom-line with very less value in each of the 200 observation. This implies most of the forum posts have main focus on the current situation and a bit of past but little less to no focus on the future prospect.

Pre-processing data:

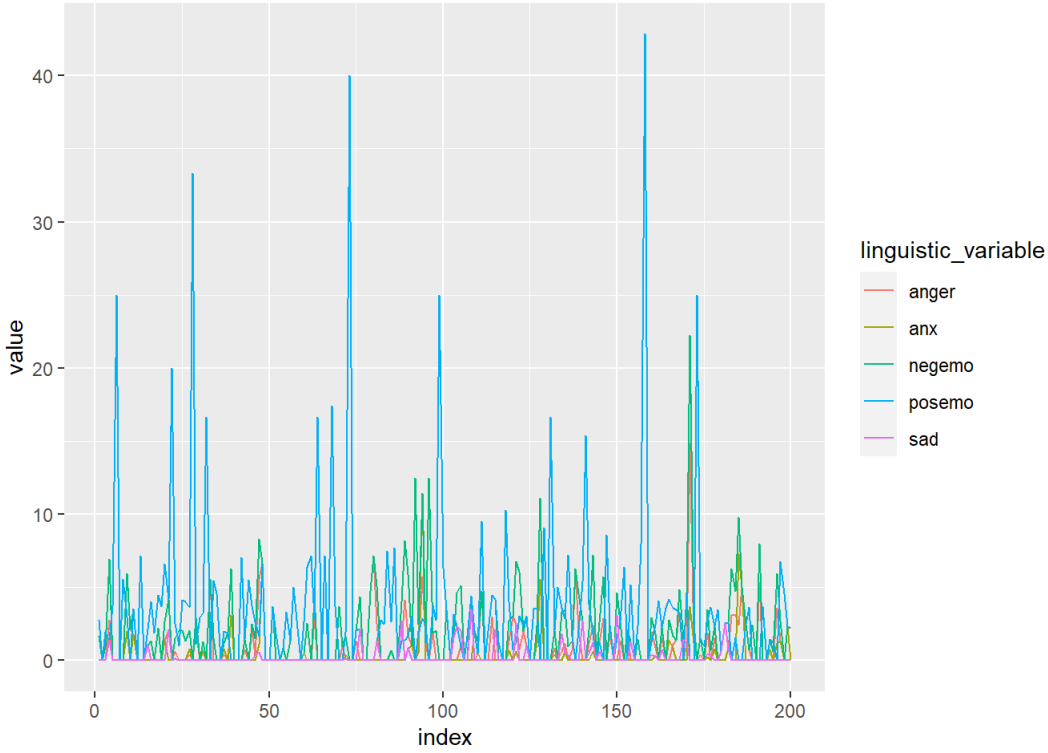
In the code below,anx,anger, sad, posemo,and negemo are the linguistic variables which are grouped together and then first 200 observations are extracted in the same way as above ones.

```
result5 = webforum %>%
  select("anx","anger", "sad", "posemo", "negemo")

result5=gather(result5, key="linguistic_variable", value="value", 1:5)
anx=result5[1:200,]
anx$index=1:nrow(anx)
anger=result5[20001:20200,]
anger$index=1:nrow(anger)
sad=result5[40001:40200,]
sad$index=1:nrow(sad)
posemo=result5[60001:60200,]
posemo$index=1:nrow(posemo)
negemo=result5[80001:80200,]
negemo$index=1:nrow(negemo)
```

Visual Representation of levels of change in anx,anger, sad, posemo, and negemo:

```
result5=rbind(anx,anger, sad, posemo, negemo)
ggplot(result5, aes(x=index,y=value, colour= linguistic_variable)) +
  geom_line()
```



Same extracted data is also used for the obersvation of the forum post with respect to these linguistic variables and posemo observed the highest value throughout the course of forum for this extracted dataset. Meanwhile, the other three(anx, negemo, sad, anger) are generally quite low during the whole course with only negemo being the prominent with some peaks during the interval.

Relationship between linguistic variables over the longer term:

We will use heatmap for determining the relationship between the lingusitic variables which will give us visual representation of the correlation between them.

Pre-processing data:

The following line of code extract the respective linguistic variable columns from the webforum dataset.

```
resultfinal = webforum %>%
  select("WC","Analytic", "Clout","Authentic",
         "Tone", "ppron","i", "we", "you", "shehe", "they","focuspresent", "focuspast", "focusfuture",
         "anx","anger", "sad", "posemo", "negemo")
```

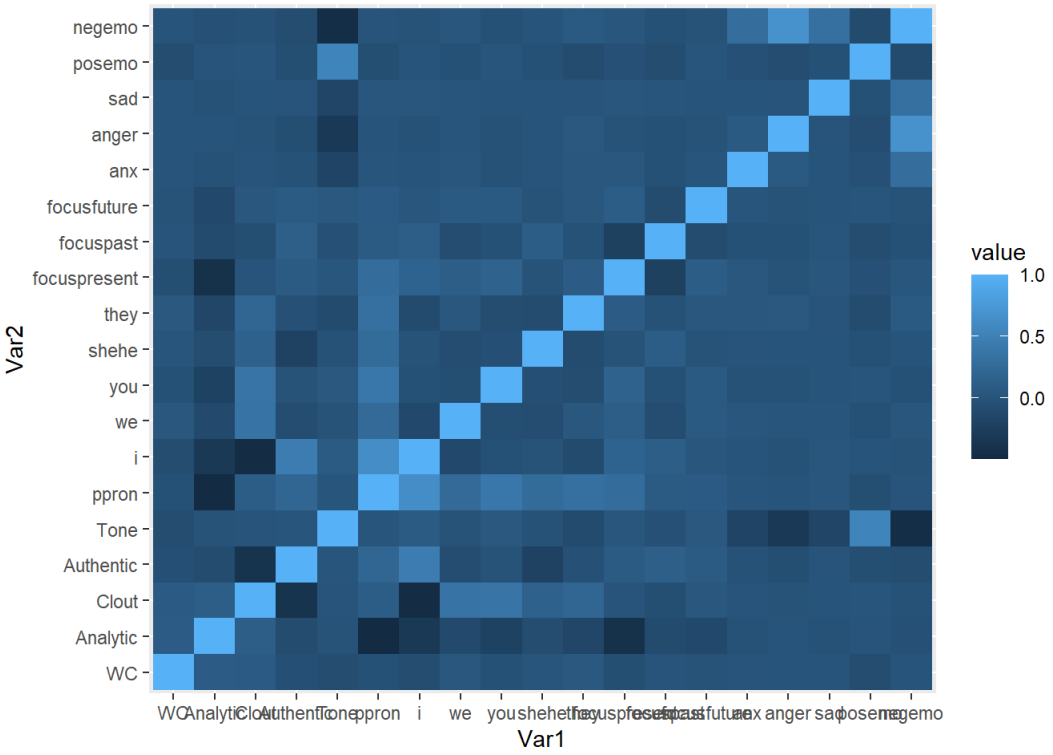
The following code follows the above approach:

```
# create correlation matrix
cormat <- round(cor(resultfinal),2)

# reshape required to melt the correlation matrix
melted_cormat <- reshape2::melt(cormat)
```

Visual Representation of the relationship/correlation between linguistic variables:

```
# visual representation
library(ggplot2)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()
```



According to the heatmap above, negemo shares high correlation with anx, anger, and sad respectively over the longer interval. Moreover, posemo and ppron are another linguistic variable which share a higher level of relationship/coorelation too.

Part B

Analyse the language used by threads:

Pre-processing data:

First of all, we've to pre-process the data and add a column for "Year" which represents the intervals for which we will judge the happiest thread. Later on, data is grouped by thread id and year and mean of all the linguistic variables for each thread id per year is calculated and asssigned to another data.

```
# convert time to year
Year <- year(as.POSIXlt(webforum$Date, format="%Y-%m-%d"))
result6 <- webforum %>% select('ThreadID','posemo', 'negemo', 'anx', 'anger', 'sad')
# binding year to the extracted columns from webforum
result6 <- cbind(result6, Year)
# grouped by thread id and year
result_m<-result6 %>% dplyr::group_by(Year, ThreadID) %>%
  dplyr::summarise(pos_emo_mean=mean(posemo), neg_emo_mean=mean(negemo),
            anx_mean=mean(anx),anger_mean=mean(anger), sad_mean=mean(sad))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

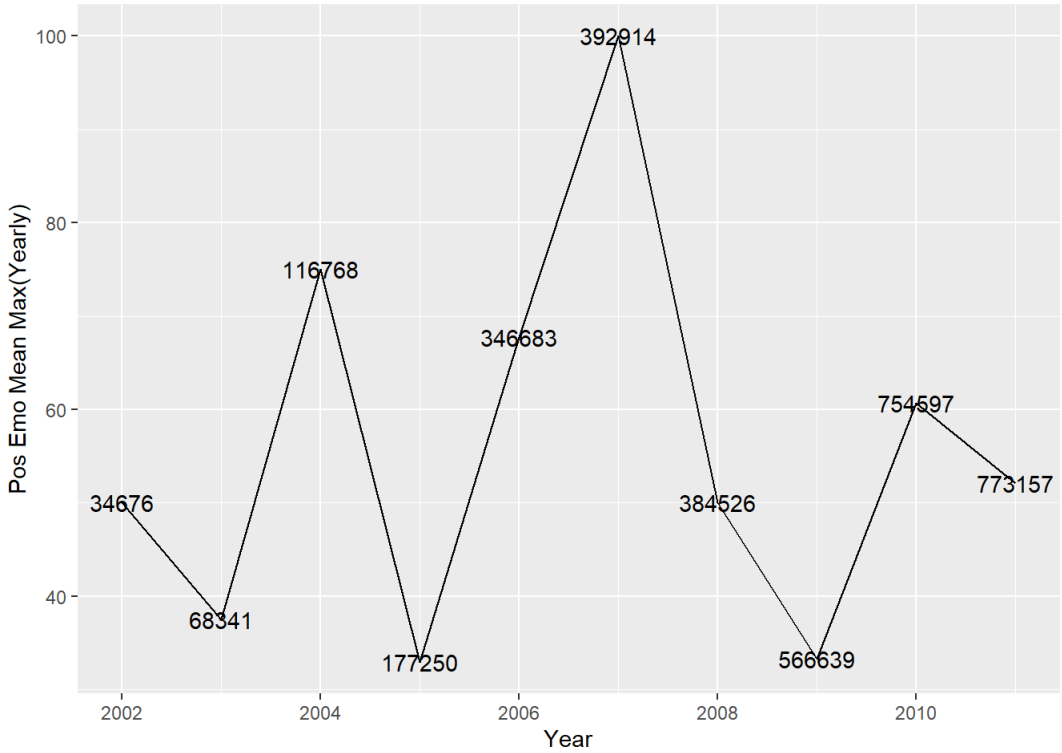
List of thread ids with max pos_emo_mean and the ones with min neg_emo_mean values calculated and assigned for each of the year.

```
# find max for pos_emo_mean
max_r <- result_m %>% dplyr::group_by(Year) %>%  slice(which.max(pos_emo_mean))

# find min for neg_emo_mean
min_r <- result_m %>% dplyr::group_by(Year) %>% slice(which.min(neg_emo_mean))
```

Visual Representation of happiest threads ids(based on pos_emo_mean):

```
ggplot(max_r, aes(x=Year,y=pos_emo_mean)) + geom_line() +
  geom_text(aes(label=ThreadID)) + labs(y='Pos Emo Mean Max(Yearly)')
```



The above figure depicts pos emotion mean (max value) for each of the year and the labels are representing thread id which had highest positive emotion in that time interval (year). For example, in year 2002, thread id 34676 had a pos emotion mean of about 50 which is the highest for that year and then the following year it was thread id 68341,and so on.

Visual Representation of saddest thread ids(based on neg_emo_mean)

```
ggplot(min_r, aes(x=Year,y=neg_emo_mean)) + geom_line() +
  geom_text(aes(label=ThreadID)) + labs(y='Neg Emo Mean Min(Yearly)')
```

For more clarity, the negative emotion mean (min value) for each of the year was also taken and is depicted in the graph above. Although, there are no thread ids for each respective year which have max pos emotion mean and lowest negative emotion mean but we can deduce our findings solely based on the previous graph which is showing threads ids with highest positive mean for each year meaning that they're more showing more optimism for that particular interval than any of the thread during that time.

Part C.1

Create a non-trivial social network of all authors who are posting over a particular time period:

Pre-processing data:

Year value can vary but we're keeping month to be february in this case checking author interaction for a particular month(february). Below is one of the approach(1) which can be adapted while looking for interaction of a particular month in some year. Same can be done

for months aswell but I have set values for both of them for analysis purposes.

```
# Approach(1)
#print("Year should be between 2002-2011")
#year_val<-readline("Enter year: ")
#year_val<-as.integer(year_val)

# setting values for month and year
month_val<-"2"
year_val<-2003
Month <- month(as.POSIXlt(webforum$Date, format="%Y-%m-%d"))
Year <- year(as.POSIXlt(webforum$Date, format="%Y-%m-%d"))

# not removing other columns from the data as they will be used in the manipulation for next part
extracted_info<-webforum
extracted_info<-cbind(Month, extracted_info)
extracted_info<-cbind(Year, extracted_info)
extracted_info<-extracted_info %>% filter(Year==year_val & Month=="2")


n<-length(unique(extracted_info$AuthorID))
upper_bound<-31
lower_bound<-1
# adding indexes next to the author id for later join operation
extracted_info$index<-as.numeric(factor(extracted_info$AuthorID, levels=unique(extracted_info$AuthorID),
                                 labels = c(1:n)))

# doing inner join and only extracting the author ids with index>=1 and <=31
extracted_info<-extracted_info %>% inner_join(extracted_info, extracted_info, by="ThreadID") %>% filter(AuthorID.x!=AuthorID.y)
extracted_info<-extracted_info[(extracted_info$index.y>=lower_bound & extracted_info$index.x<=upper_bound) & (extracted_info$index.y>=lower_bound
& extracted_info$index.y<=upper_bound),]


# reference for the below code: https://www.jessesadler.com/post/network-analysis-with-r/

# rename the AuthorID.x and AuthorID.y to label for helping in network analysis
source<-extracted_info %>% distinct(AuthorID.x) %>% dplyr::rename(label=AuthorID.x)
destination<-extracted_info %>% distinct(AuthorID.y) %>% dplyr::rename(label=AuthorID.y)

# join both of the columns to create a single dataframe
nodes<-full_join(source, destination, by="label") %>% rowid_to_column("id")
#nodes

# making empty undirected graph
g <- make_empty_graph(directed = FALSE)

# adding vertices to the graph referring to the author id
for (i in 1 : nrow(nodes)) {
  g <- add_vertices(g, 1, name =
                      as.character(nodes$label[i]))
  }

# finding the route_info by grouping the author id of x and y
route_info <-extracted_info %>% group_by(AuthorID.x, AuthorID.y) %>% tally()
#route_info

# renaming ids with from and to for AuthorID.x and AuthorID.y respectively
edges <- route_info %>%
 left_join(nodes, by = c("AuthorID.x" = "label")) %>%
 dplyr::rename(from = id)


edges <- edges %>%
 left_join(nodes, by = c("AuthorID.y" = "label")) %>%
 dplyr::rename(to = id)


edges<-select(edges, "from", "to", "n")
```

```
## Adding missing grouping variables: `AuthorID.x`
```

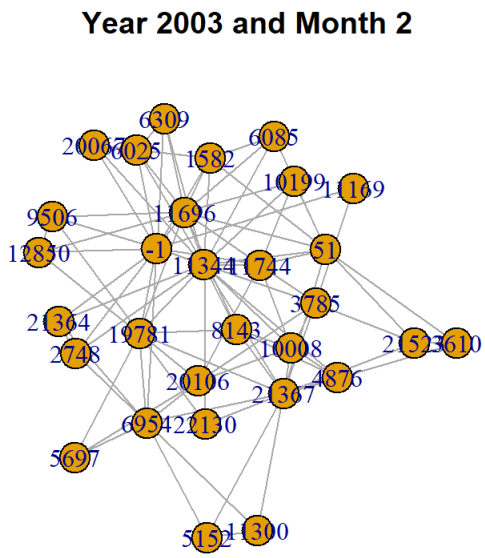
```
# selecting specific rows for later manipulation
edges<-edges[,2:4]

# adding edges to the graph
for (i in 1:nrow(edges)){
  from_node<-nodes[edges$from[i],]$label
  to_node<-nodes[edges$to[i],]$label
  edges
  g <- add_edges(g,
                 c(as.character(from_node),as.character(to_node)))

}
```

A non-trivial social network of all authors who are posting over a particular time period(Year 2003, Month 2)

```
g<-(igraph::simplify(g, remove.multiple=TRUE))
plot(g, main="Year 2003 and Month 2")
```



Part C.2

Identify the most important author in the social network you created:

```
# code referred from tutorial 6 and week 5 presentation slides
#library(igraphdata)
g_simplified<-(igraph::simplify(g, remove.multiple=TRUE, remove.loops = TRUE))
# Graph Summary
avg_g<-average.path.length(g_simplified)
diameter_g<- diameter(g_simplified)
clique_size<-table(sapply(cliques(g_simplified),length))

#calculate network centrality measures and combine in a dataframe
degree_coeff = as.table(degree(g_simplified))
betweenness_coeff = as.table(betweenness(g_simplified))
closeness_coeff = as.table(closeness(g_simplified))
eigenvector_coeff = as.table(evcent(g_simplified)$vector)
stats = as.data.frame(rbind(degree_coeff, betweenness_coeff, closeness_coeff, eigenvector_coeff))
stats = t(stats)
colnames(stats) = c("degree", "betweenness", "closeness", "eigenvector")
stats
```

```
##       degree betweenness  closeness eigenvector
## 11169      2   0.0000000 0.01515152   0.1409079
## 12850      4   0.0000000 0.01666667   0.3069063
## 3785       7  16.9992063 0.01923077   0.3640065
## 10008      7   6.3817460 0.01886792   0.4276039
## 11744      8  11.5992063 0.01923077   0.5494604
## 8143       6   3.2666667 0.01886792   0.4515245
## 5152       3   0.0000000 0.01515152   0.1376652
## 11696     13  31.8873016 0.02127660   0.7945896
## 21364      4   0.4111111 0.01785714   0.3021326
## 21367     13  68.0595238 0.02222222   0.6036273
## -1        15  68.4650794 0.02272727   0.8132435
## 20106      6   7.2000000 0.01724138   0.3067358
## 1582       7   4.3317460 0.01818182   0.5120467
## 6025       6   0.8333333 0.01724138   0.4571993
## 6085       4   0.0000000 0.01639344   0.3406379
## 21523      4   1.8539683 0.01612903   0.1996030
## 2748       4   0.4111111 0.01785714   0.3021326
## 6954       9  31.6428571 0.01960784   0.4088653
## 11344     18  89.7325397 0.02500000   1.0000000
## 22130      3   0.0000000 0.01785714   0.2696685
## 11300      3   0.0000000 0.01515152   0.1376652
## 19781     11  38.9357143 0.02127660   0.6493849
## 6309       4   0.0000000 0.01666667   0.3668612
## 5697       3   0.0000000 0.01515152   0.1633785
## 4876       4   0.0000000 0.01724138   0.2971671
## 10199      4   0.0000000 0.01694915   0.3451160
## 3610       3   0.0000000 0.01515152   0.1606916
## 20067      4   0.0000000 0.01666667   0.3330430
## 9506       4   0.0000000 0.01666667   0.3069063
## 51         9  24.9888889 0.01960784   0.5393069
```

The above table shows the graph summary along with the degree, betweenness, closeness centrality of each of the author id. As the questions asks to look for the most important author in the social network so basically we'll mainly focus on the closeness centrality meaning how closely it is connected to each of the other authors and also eigenvector centrality as its weight denotes the quality of the connection between the respective node and other nodes which can also provide strong evidence in regards to whatever our query is. If we look at the table above AuthorID of 11344 observes the highest eigenvector of 1 with the closeness centrality of 0.0.02500000 . Even though betwenness centrality(bc) isn't the prime focus but this respective author id supports a high bc too which says alot about its high hub potential. The author with id (-1) carries the second highest eigen vector also emphasizing on its quality connection but with comparatively less closeness centrality and betweenness centrality as well. We can ignore this finding and choose the first author with author id(11344) as the most important author of the social network.

Observe difference of linguistic variable between the most important and other authors:

For the second part, first we have to take the data from previous parts and reassign it to a new one for distinction and more clarity

```
author_ids<-nodes$label

dt_table<-data.frame(matrix(NA, length(author_ids), 7))
# feeding data into the first column
dt_table$X1 <- nodes$label
author_ids[1]
```

```
## [1] 11169
```

```
size<-nrow(dt_table)
for (i in 1:size){
  acquired_info<-extracted_info[extracted_info$AuthorID.x==author_ids[i],]
  start<-2
  for (j in 7:12){
    # performing t-testing to compare the linguistic variables of each author
    dt_table[i, start]=t.test(acquired_info[,j], acquired_info[,j+25], "two.sided")$p.value
    start=start+1
  }
}

colnames(dt_table)<-
  c("Author ID " ,"Wc p_values", "Analytics p_values", "Clout p_values", "Authentic p_values", "Tone p_values", "Ppron p_values")

dt_table
```

```
##    Author ID   Wc p_values Analytics p_values Clout p_values Authentic p_values
## 1      11169 8.105663e-01      3.520296e-01   1.227753e-01      2.967027e-01
## 2      12850 8.892652e-01      7.927531e-01   1.635294e-01      2.108381e-01
## 3       3785 5.844247e-02      9.182437e-01   1.201247e-01      3.355052e-01
## 4      10008 2.133546e-01      9.655121e-01   9.081958e-01      2.500085e-07
## 5      11744 1.513766e-02      7.575840e-01   5.435308e-03      7.631018e-01
## 6       8143 9.743109e-01      1.010328e-01   2.694208e-01      7.506431e-01
## 7       5152 1.013623e-02      6.388078e-01   3.979501e-01      4.838202e-03
## 8      11696 5.276460e-01      6.654605e-01   3.065796e-01      4.470256e-01
## 9      21364 7.972626e-04      7.025156e-01   1.453874e-03      2.502484e-02
## 10     21367 1.507107e-01      1.310198e-02   1.269133e-01      2.118538e-01
## 11        -1 7.624926e-01      7.346906e-02   1.167601e-01      3.834325e-01
## 12     20106 4.520158e-01      3.019793e-01   1.278716e-01      2.587683e-02
## 13      1582 8.020542e-06      5.060262e-07   8.297386e-07      9.928054e-02
## 14      6025 8.530709e-02      1.073753e-01   1.702325e-01      7.294788e-01
## 15      6085 2.219110e-01      1.635273e-02   2.184409e-03      1.481736e-03
## 16     21523 1.553154e-01      6.377502e-01   7.944397e-02      1.971603e-01
## 17      2748 5.500894e-01      4.078845e-01   4.412660e-02      1.654513e-01
## 18      6954 3.051978e-01      3.296406e-02   1.661448e-01      6.732566e-01
## 19     11344 1.116324e-01      1.299077e-03   6.739107e-01      3.493340e-05
## 20     22130 5.464465e-01      6.159258e-01   3.179895e-02      7.832547e-03
## 21     11300 2.128195e-01      5.051946e-01   6.105017e-01      5.587040e-01
## 22     19781 6.190972e-04      7.983369e-01   4.813035e-03      1.646894e-03
## 23      6309 3.295911e-03      3.478334e-01   3.616833e-01      6.684581e-01
## 24      5697 3.958210e-02      3.840399e-01   1.232464e-05      6.956538e-01
## 25      4876 2.178994e-02      2.055209e-02   2.646490e-03      9.743982e-01
## 26     10199 2.472649e-02      9.648188e-01   8.211557e-02      1.395170e-01
## 27      3610 7.040957e-04      5.833121e-01   2.626608e-02      2.462449e-03
## 28     20067 6.740493e-01      1.550468e-01   6.990629e-02      3.225320e-01
## 29      9506 2.788696e-01      2.680516e-01   7.664198e-02      9.080784e-01
## 30        51 3.672032e-01      9.109990e-01   1.247900e-01      1.889308e-03
##    Tone p_values Ppron p_values
## 1   0.1606665141   5.051951e-02
## 2   0.7489268736   5.532796e-01
## 3   0.0764880304   6.320933e-01
## 4   0.2254864397   6.532072e-01
## 5   0.2452950742   8.257362e-02
## 6   0.2819771936   5.003267e-01
## 7   0.3639282182   1.346661e-01
## 8   0.0349006341   2.168908e-04
## 9   0.2157571336   9.346454e-02
## 10  0.8093962452   6.080345e-01
## 11  0.6982935873   1.680167e-02
## 12  0.0031654993   1.402389e-01
## 13  0.1118244975   4.815428e-01
## 14  0.0045634698   1.336134e-01
## 15  0.0270128867   5.516050e-02
## 16  0.2209864105   1.803368e-01
## 17  0.9326941648   1.166656e-01
## 18  0.4920724688   9.845150e-01
## 19  0.0842954303   1.065853e-02
## 20  0.2099309327   4.299004e-01
## 21  0.1232292422   8.045296e-01
## 22  0.0078554690   1.335838e-06
## 23  0.0023569062   3.307774e-01
## 24  0.0171765656   2.673477e-01
## 25  0.2769857684   1.936087e-04
## 26  0.0031545449   1.335380e-02
## 27  0.2093505970   7.846676e-01
## 28  0.2409817319   9.177476e-03
## 29  0.3769917719   8.427757e-01
## 30  0.0004597814   1.728546e-01
```

The questions asks us about the difference of linguistic variable used by the most important author and other authors in the same social network. The above table depicts the p-value across different linguistic variables for each of the author. The first row refers to the author id (1169) p_values with respect to its linguistic variables. If we closely observe the p_values for linguistic variable for most important author (11344) Analytics, Authentic, and Ppron have a smaller p value indicating the difference of the usage of stated linguistic variable in posts between the important authors and other authors. Moreover, Tone's p value and WC's p value are relatively higher indicating that its usage by important author is similar to that of by other authors in their respective posts.