FIT3152asm\_1\_final\_markdown

Import needed library

library(tidyverse)  
library(lubridate)

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

# remove outlines  
outliers <- function(x) {  
   
 Q1 <- quantile(x, probs=.25)  
 Q3 <- quantile(x, probs=.75)  
 iqr = Q3-Q1  
   
 upper\_limit = Q3 + (iqr\*1.5)  
 lower\_limit = Q1 - (iqr\*1.5)  
   
 x > upper\_limit | x < lower\_limit  
}  
  
remove\_outliers <- function(new\_df, cols = names(new\_df)) {  
 for (col in cols) {  
 new\_df <- new\_df[!outliers(new\_df[[col]]),]  
 }  
 new\_df  
}

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)))  
  
  
  
#scale the data  
data\_tidy\_scale <- data.frame( scale(data\_tidy[,5:19],center=TRUE,scale=TRUE) )

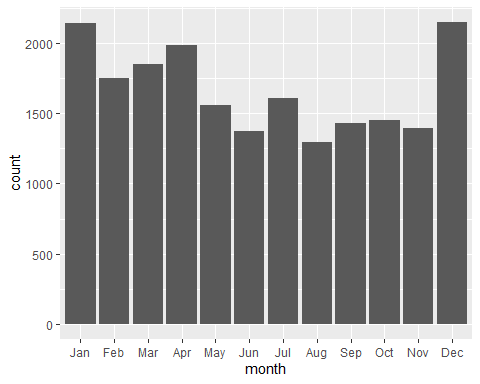
==============================================================================================================

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

#Month  
month\_df = data\_tidy %>% group\_by(month) %>% summarize(count =n())

## `summarise()` ungrouping output (override with `.groups` argument)

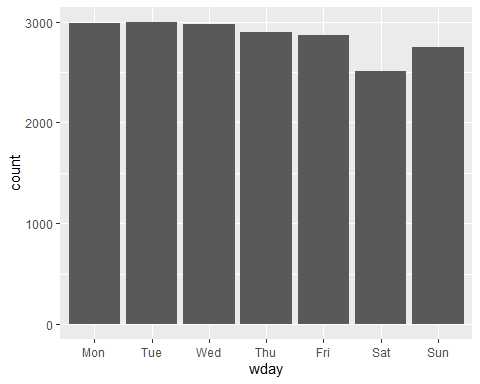
ggplot(month\_df,  
 aes(x = month,   
 y = count)) +   
 geom\_col()



#Week-Day   
wday\_df = data\_tidy %>% group\_by(wday) %>% summarize(count=n())

## `summarise()` ungrouping output (override with `.groups` argument)

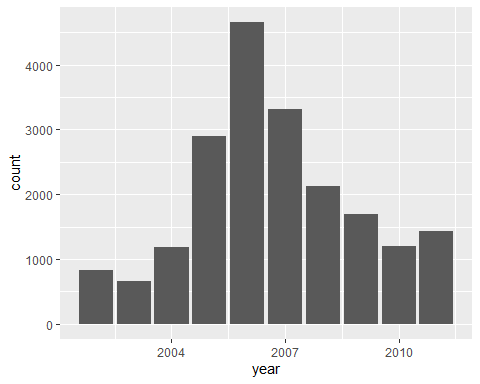
ggplot(wday\_df,  
 aes(x = wday,   
 y = count)) +   
 geom\_col()



#year  
year\_df = data\_tidy %>% group\_by(year) %>% summarize(count =n())

## `summarise()` ungrouping output (override with `.groups` argument)

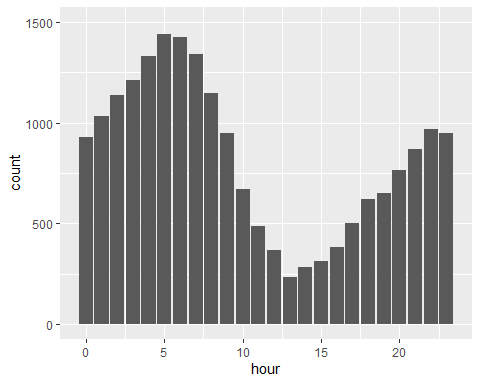
ggplot(year\_df,  
 aes(x = year,   
 y = count)) +   
 geom\_col()



hour\_df = data\_tidy %>% group\_by(hour,Date) %>% summarize(count =n())

## `summarise()` regrouping output by 'hour' (override with `.groups` argument)

ggplot(hour\_df,  
 aes(x = hour,   
 y = count,group=Date  
 )) +   
 geom\_col()+ylim(0,1500)



#day  
day\_df = data\_tidy %>% group\_by(day) %>% summarize(count =n())

## `summarise()` ungrouping output (override with `.groups` argument)

ggplot(day\_df,  
 aes(x = day,   
 y = count)) +   
 geom\_col()



data\_day <- data\_tidy %>%   
 group\_by(Date,hour) %>%  
 summarise(Count = n())

## `summarise()` regrouping output by 'Date' (override with `.groups` argument)

data\_day

## # A tibble: 14,799 x 3  
## # Groups: Date [3,220]  
## Date hour Count  
## <date> <dbl> <int>  
## 1 2002-01-01 20 1  
## 2 2002-01-02 5 1  
## 3 2002-01-02 8 2  
## 4 2002-01-04 1 1  
## 5 2002-01-04 4 1  
## 6 2002-01-04 16 1  
## 7 2002-01-04 20 1  
## 8 2002-01-05 19 1  
## 9 2002-01-06 7 1  
## 10 2002-01-06 18 1  
## # ... with 14,789 more rows

data\_day <- data\_day %>%  
 mutate(year = year(Date))

data\_day %>%  
 mutate(wday = wday(Date, label = TRUE, abbr = TRUE, week\_start = 1)) %>%   
 group\_by(wday) %>%  
 summarise(m = mean(n()))

## `summarise()` ungrouping output (override with `.groups` argument)

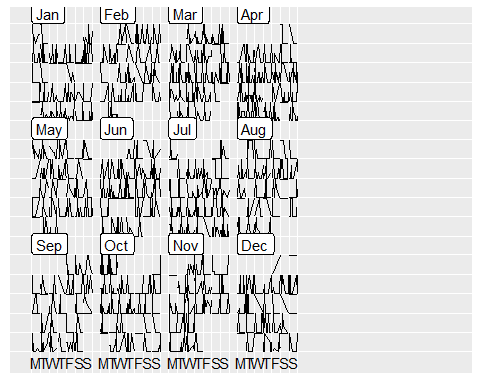
## # A tibble: 7 x 2  
## wday m  
## <ord> <dbl>  
## 1 Mon 2146  
## 2 Tue 2175  
## 3 Wed 2173  
## 4 Thu 2169  
## 5 Fri 2110  
## 6 Sat 1936  
## 7 Sun 2090

--------------------------------------------------------------------------------------------  
b Is there a trend over time?

library(sugrrants)

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

a = data\_day %>% filter(year == 2006)  
data\_tidy\_calendar <-   
 frame\_calendar(a,  
 x =hour,   
 y = Count,   
 date = Date,   
 )  
  
p1 <- ggplot(data\_tidy\_calendar,  
 aes(x = .hour,   
 y = .Count,   
 group = Date)) +  
 geom\_line()  
prettify(p1)

 ——————————————————————————————– c Looking at the linguistic variables, 1 do these change over time? ——————————————————————————————– 2 Is there a relationship between variables? ——————————————————————————————–

============================================================================================================== - 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.  
--------------------------------------------------------------------------------------------

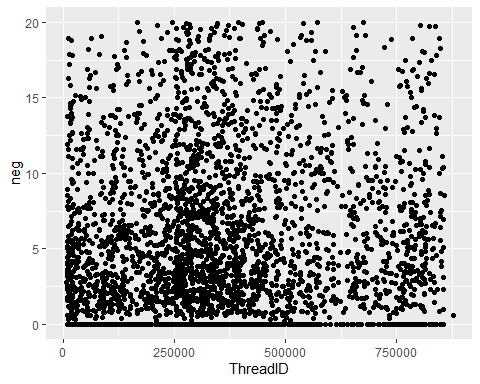
df <- data\_tidy %>%   
 group\_by(ThreadID) %>%  
 summarise(pos = sum(posemo, na.rm = TRUE),  
 neg = sum(negemo, na.rm = TRUE),  
 anxiety = sum(anx, na.rm =TRUE))

## `summarise()` ungrouping output (override with `.groups` argument)

How negative the treads overall

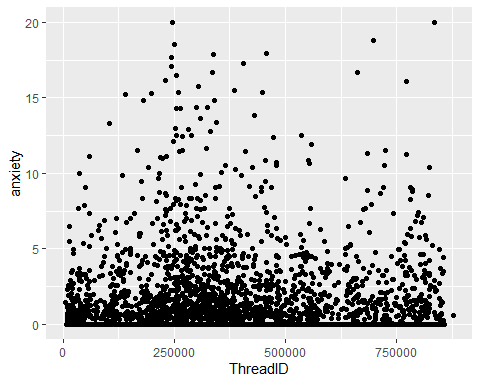
ggplot(df,aes(x = ThreadID, y = neg))+ geom\_point() + ylim(0,20)

## Warning: Removed 594 rows containing missing values (geom\_point).

 How anxious the treads overall

ggplot(df,aes(x = ThreadID, y = anxiety))+ geom\_point() + ylim(0,20)

## Warning: Removed 13 rows containing missing values (geom\_point).



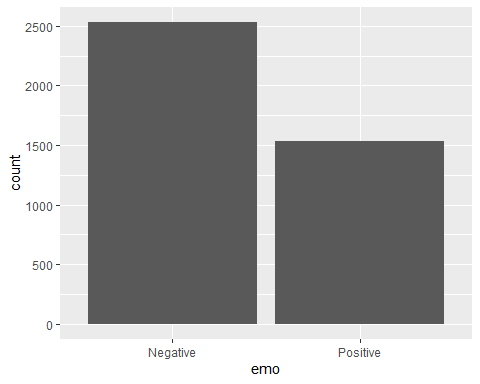
df\_2 <- data\_tidy %>%   
 group\_by(ThreadID) %>%  
 summarise(Tone = median(Tone, na.rm = TRUE))

## `summarise()` ungrouping output (override with `.groups` argument)

df\_2 = df\_2 %>% mutate(emo = ifelse(Tone >50 , "Positive", "Negative"))%>%   
 group\_by(emo) %>% summarize(count =n())

## `summarise()` ungrouping output (override with `.groups` argument)

ggplot(df\_2, aes(x= emo, y =count))+ geom\_col()



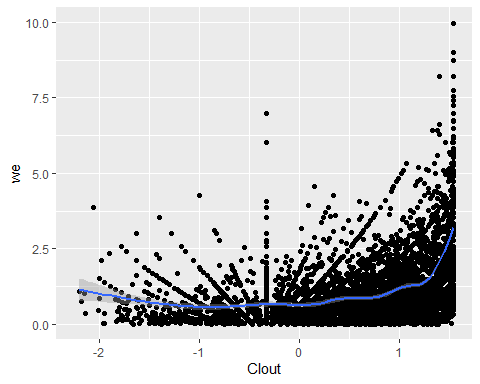
b By analysing the linguistic variables for all or some of the threads, is it possible to see a  
difference in the language used by different groups?  
--------------------------------------------------------------------------------------------  
c Does the language used within threads (or between threads) change over time? How  
 consistent or variable is the language used within threads?  
--------------------------------------------------------------------------------------------

ggplot(data\_tidy\_scale, aes(Clout,we)) + geom\_point() +ylim(0,10) + geom\_smooth()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 14355 rows containing non-finite values (stat\_smooth).

## Warning: Removed 14355 rows containing missing values (geom\_point).

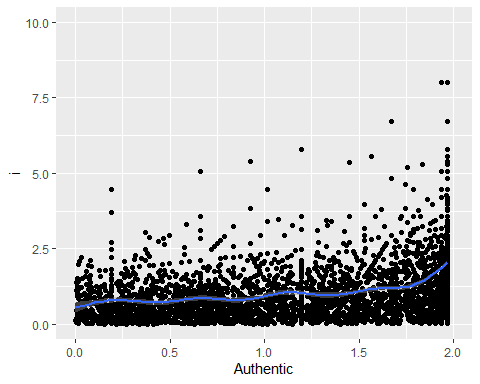


ggplot(data\_tidy\_scale, aes(Authentic,i)) + geom\_point() +ylim(0,10) + geom\_smooth() +xlim(0,2)

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 15040 rows containing non-finite values (stat\_smooth).

## Warning: Removed 15040 rows containing missing values (geom\_point).



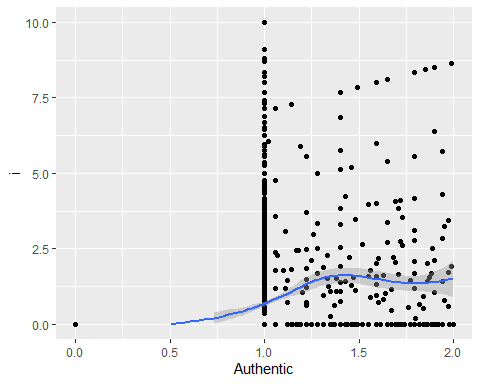
ggplot(data\_tidy, aes(Authentic,i)) + geom\_point() +ylim(0,10) + geom\_smooth() +xlim(0,2)

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 17559 rows containing non-finite values (stat\_smooth).

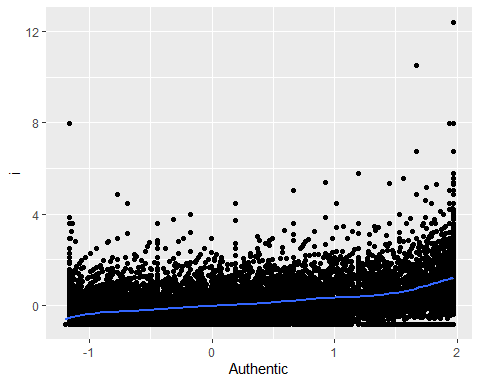
## Warning: Removed 17559 rows containing missing values (geom\_point).

## Warning: Removed 20 rows containing missing values (geom\_smooth).



ggplot(data\_tidy\_scale, aes(Authentic,i)) + geom\_point() + geom\_smooth()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



ggplot(data\_tidy, aes(Authentic,i)) + geom\_point()+ geom\_smooth()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

