Project 1

Yiding Xie UNI: yx2443

HappyDB is a corpus of 100,000 crowd-sourced happy moments via Amazon's Mechanical Turk. The goal of my project is to look deeper into the datasets and to draw any insights on the causes that make us happy. Several Natural language processing and text mining techniques (such as) are used in my project to derive interesting findings in this collection of happy moments.

Step 0 - Load all the required libraries

```
packages.used=c("plyr","tm","tidytext","tidyverse","DT","wordcloud","scales","wordcloud2",
                "gplots","sentimentr","ngram","dplyr","qdap","syuzhet","ggplot2","topicmodels")

# check packages that need to be installed.
packages.needed=setdiff(packages.used,
                        intersect(installed.packages()[,1],
                                  packages.used))
# install additional packages
if(length(packages.needed)>0){
  install.packages(packages.needed, dependencies = TRUE)
}

library(tm)
library(tidytext)
library(tidyverse)
library(DT)
library(wordcloud)
library(scales)
library(wordcloud2)
library(gplots)
library(sentimentr)
library(ngram)
library(dplyr)
library(qdap)
library(syuzhet)
library(ggplot2)
library(topicmodels)
```

Step 1 - Data Import & Preparation

Step 1.1 - Load the processed text data along with demographic information on contributors

We use the processed data (cleaned and with all stop words removed) for our analysis and combine it with the demographic information available.

```
hm_data <- read_csv("../output/processed_moments.csv")

urlfile<-'https://raw.githubusercontent.com/rit-public/HappyDB/master/happydb/data/demographic.csv'
demo_data <- read_csv(urlfile)
```

Step 1.2 - Combine both the data sets and keep the required columns for analysis

We select a subset of the data that satisfies my project need.

```
hm_data <- hm_data %>%
  inner_join(demo_data, by = "wid") %>%
  select(wid,
         original_hm,
         baseform_hm,
         num_sentence,
         gender,
         marital,
         parenthood,
         reflection_period,
         age,
         country,
         ground_truth_category,
         predicted_category,
         text) %>%
  mutate(count = sapply(hm_data$text, wordcount))
```

Step 1.3 - Bag of Words

Create a bag of words using the text data, generate word_count datasets (grouped by predicted_category, gender, marital status, and reflection_period separately), and then sort each word_count sets

```
bag_of_words <-  hm_data %>%
  unnest_tokens(word, text)

word_count <- bag_of_words %>%
  count(word, sort = TRUE)

word_count_by_category <- bag_of_words %>%
  group_by(predicted_category) %>%
  count(word, sort = TRUE)

word_count_by_gender <- bag_of_words %>%
  group_by(gender) %>%
  count(word, sort = TRUE)

word_count_by_marital <- bag_of_words %>%
  group_by(marital) %>%
  count(word, sort = TRUE)

word_count_by_reflection <- bag_of_words %>%
  group_by(reflection_period) %>%
  count(word, sort = TRUE)
```

Step 2 - Word Cloud

Step 2.1 - Word Cloud on the whole Dataset (cleaned words)

We collected the top 100 words with most appearanaces in the entire dataset. From the graph, words like "friend", "time", "family", and "home" etc, tend to appear more frequently.

```r
wordcloud(words = word_count$word, freq = word_count$n, min.freq = 1,
          max.words=100, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8,"Dark2"))
```

Step 2.2 - Word Cloud vs. Grouped Data

Then I want to dig deeper into the dataset. By utilizing word cloud and bar charts, I am able to examine the following relations: (1) Most frequent words vs Predicted_Category (2) Most frequent words vs Gender (3) Most frequent words vs Marital Status (4) Most frequent words vs Reflection_Period

(1) Most frequent words vs Predicted_Category: In the "achievement" category, words like "job" and "received" ranked top; while for "bonding" category, words like "friend" are the hottest, and for "nature" category, words like "weather" and "rain" appeared the most.

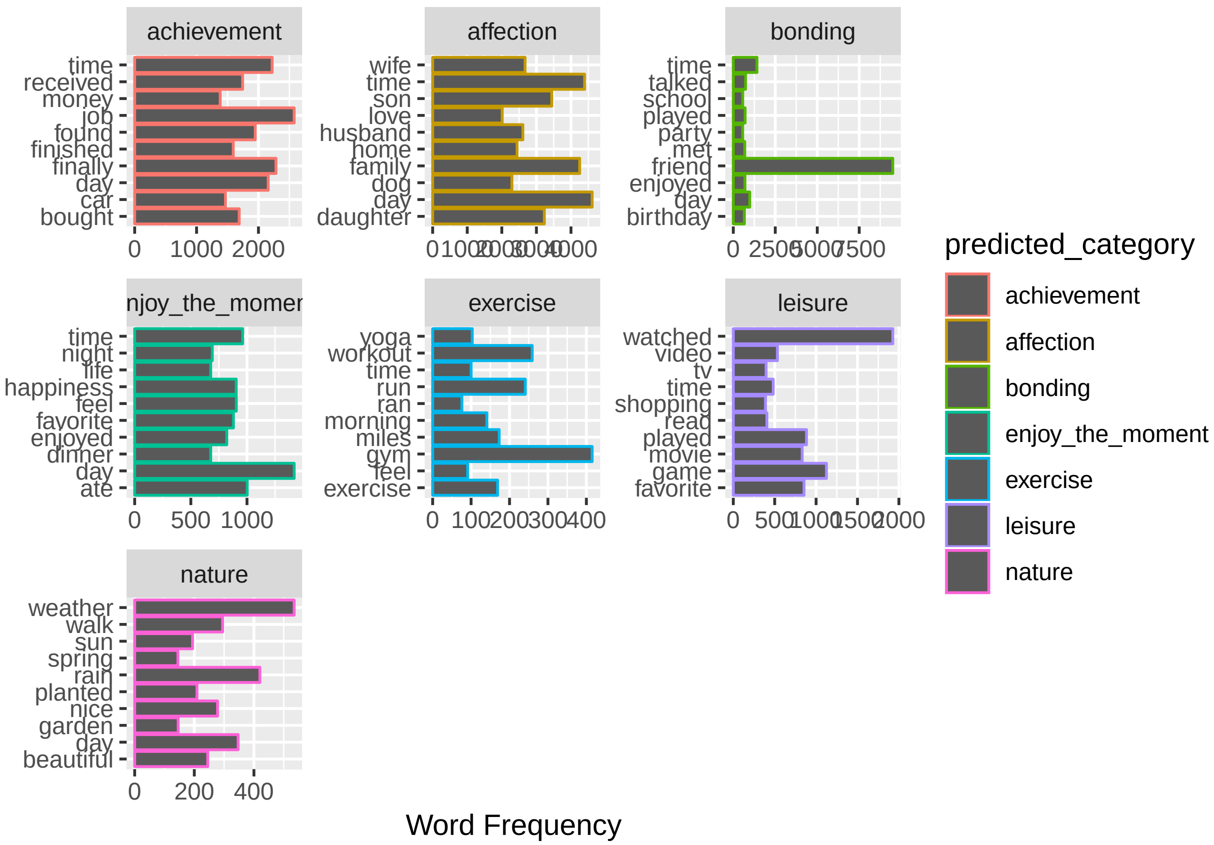
(2) Most frequent words vs Gender For both male and female, most frequent words are both "friend", "day" and "time".

(3) Most frequent words vs Marital Status The most obvious fact is that for people who are divorced or single, word "friend" tended to appear a lot more frequent than that for married people.

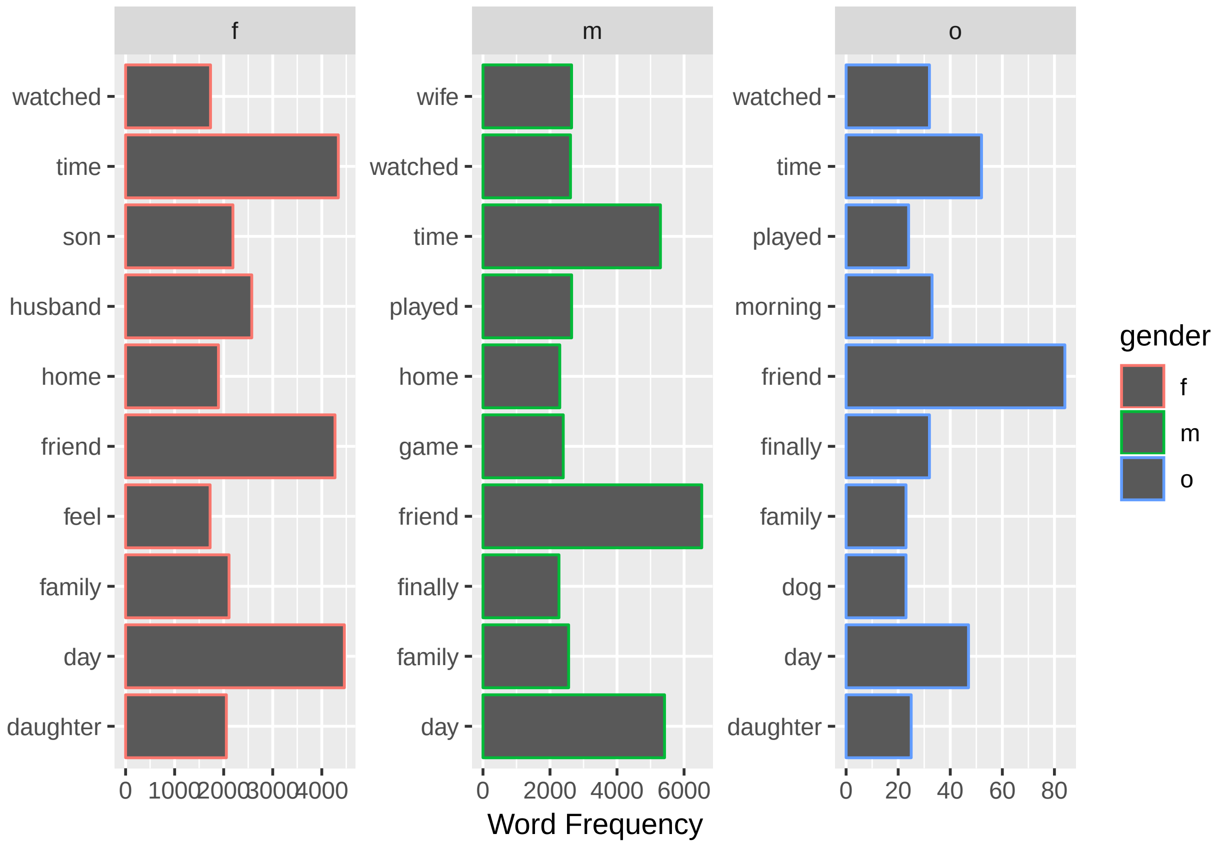
(4) Most frequent words vs Reflection_Period For both 24h and 3m data, most frequent words are both "friend", "day" and "time". However, the difference is that for 24h data, words that tend to be memorized in a short term are also very hot, such as "watched", "morning", "dinner"; while for 3m data, words that represent more significant events tend to show up more, such as "job", "home", and "birthday".

```r
# By Predicted Category
word_count_by_category %>%
  slice(1:10) %>%
  mutate(word = reorder(word, n)) %>%
```

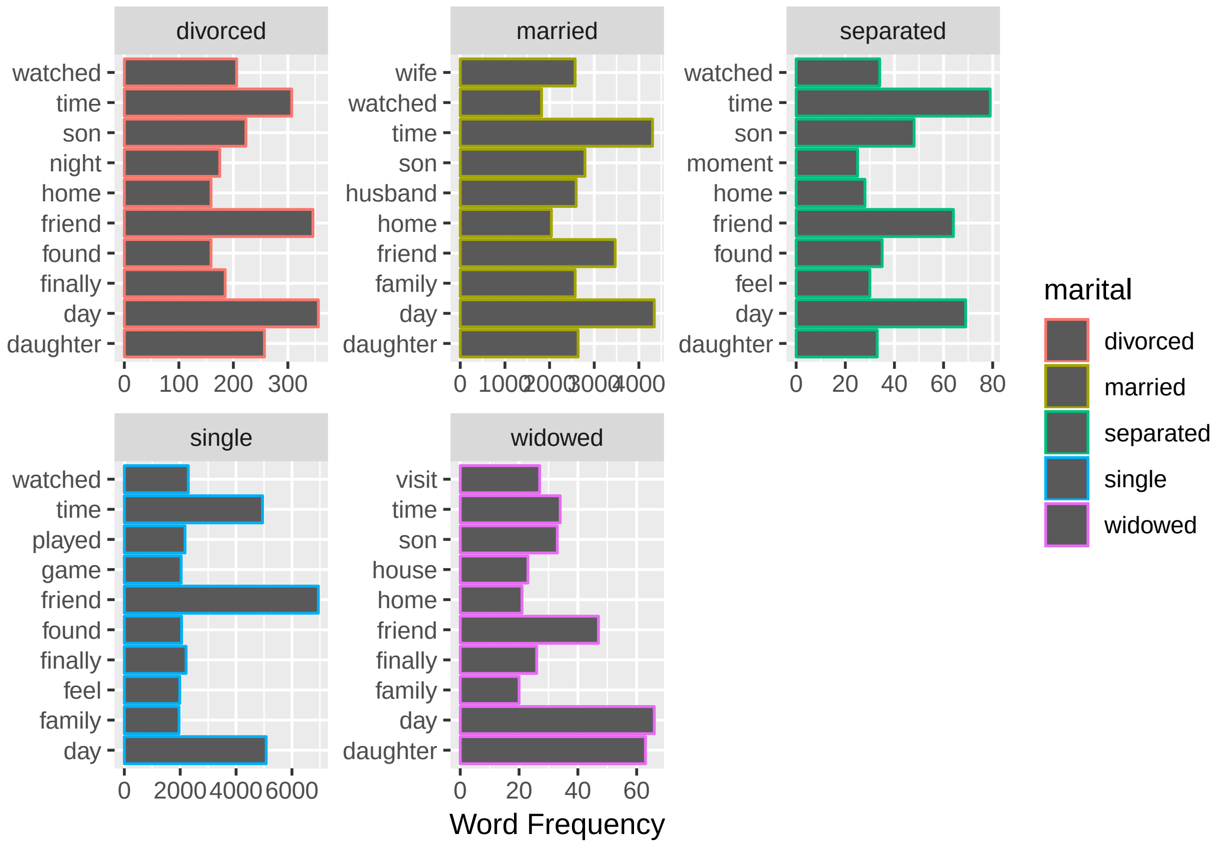
```r
  ggplot(aes(x = word, y = n, color = predicted_category)) + geom_col() +  facet_wrap(~predicted_catego
      ylab("Word Frequency")+ coord_flip()
```



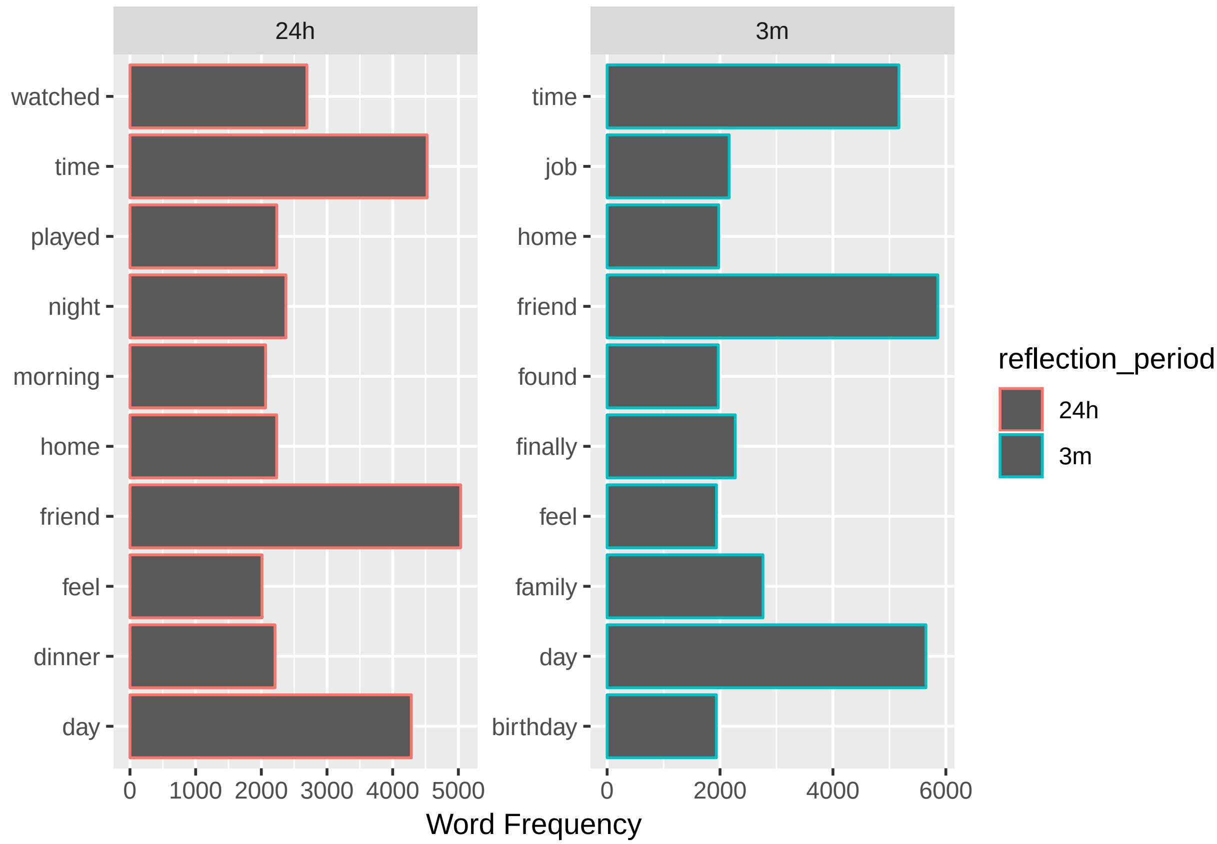

```r
# By Gender
word_count_by_gender[!is.na(word_count_by_gender$gender),] %>%
  slice(1:10) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n, color = gender)) + geom_col() +  facet_wrap(~gender, scales = "free") + xl
```




```
# By Marital
word_count_by_marital[!is.na(word_count_by_marital$marital),] %>%
  slice(1:10) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n, color = marital)) + geom_col() +  facet_wrap(~marital, scales = "free") +
```




```
# By Reflection Period
word_count_by_reflection[!is.na(word_count_by_reflection$reflection_period),] %>%
  slice(1:10) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n, color = reflection_period)) + geom_col() +  facet_wrap(~reflection_period
```



Step 3 - Sentiment Value Analysis

The term "sentiment value", is a numerical value that was created in the R package, "Syuzhet". This value weighs the emotional intensity of text, and it is part of the sentiment analysis method.

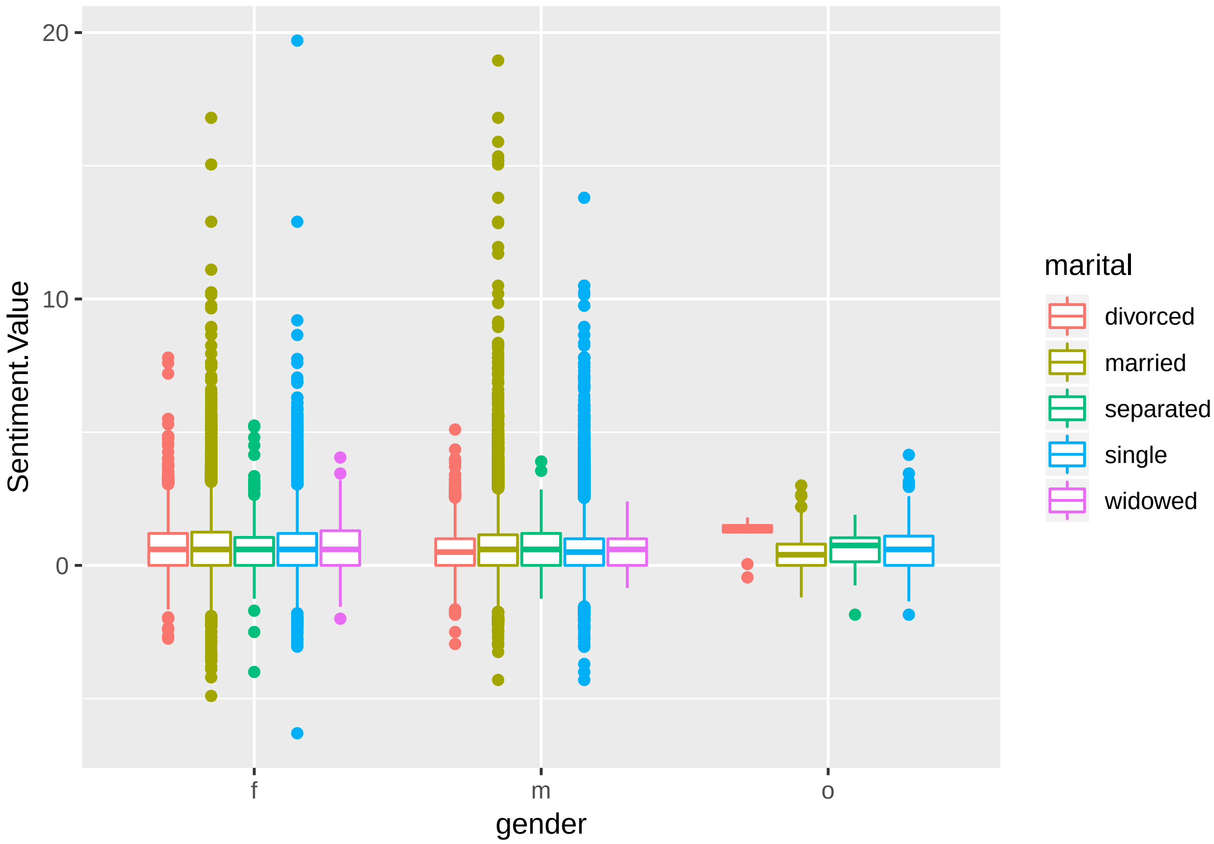
Then I want to dig deeper into the dataset. By utilizing boxplots, I am able to examine the following relations: (1) Sentiment Value vs. Gender & Marital Status (2) Sentiment Value vs. 10 Countries with the most word entries (3) Sentiment Value vs. Age groups

(1) Sentiment Value vs. Gender & Marital Status The mean and medians of the sentiment value do not vary much among the individual demographic groups. However, I do notice more extreme values for people who are either married or single. The other thing worth mentioning is that, most sentiment values are positive.

(2) Sentiment Value vs. 10 Countries with the most word entries The 10 countries with the most word entires are: AUS, BRA, CAN, GBR, IND, MEX, PHL, USA, VEN, VNM. And IND and VEN seem to have sider IQRs compared to everyone else (Especially for USA: Quite narrow IQR, with over 78000 records). This probably means American people's sentiments are pretty consistent compared to other nations. I also notice IND and USA seem to have more extreme values, and this is probably due to the fact that they have collected a lot more data from IND and USA (both more than 10000 records), while most countries have less than 100.

(3) Sentiment Value vs. Age groups The age groups are binned into an interval of 10. The sentiment value itself does not tell much story, but I saw some extreme outliers. For example, there are couple records who were submitted by people who are over 200 years old. Also, a vast majority of the data records were contributed by people who are in their 20s through 40s.

```r
hm_data$Sentiment.Value <- get_sentiment(hm_data$text)

# Sentiment Value vs. Gender & Marital Status
ggplot(hm_data[(!is.na(hm_data$gender))&(!is.na(hm_data$marital)),], aes(x = gender, y = Sentiment.Valu
```

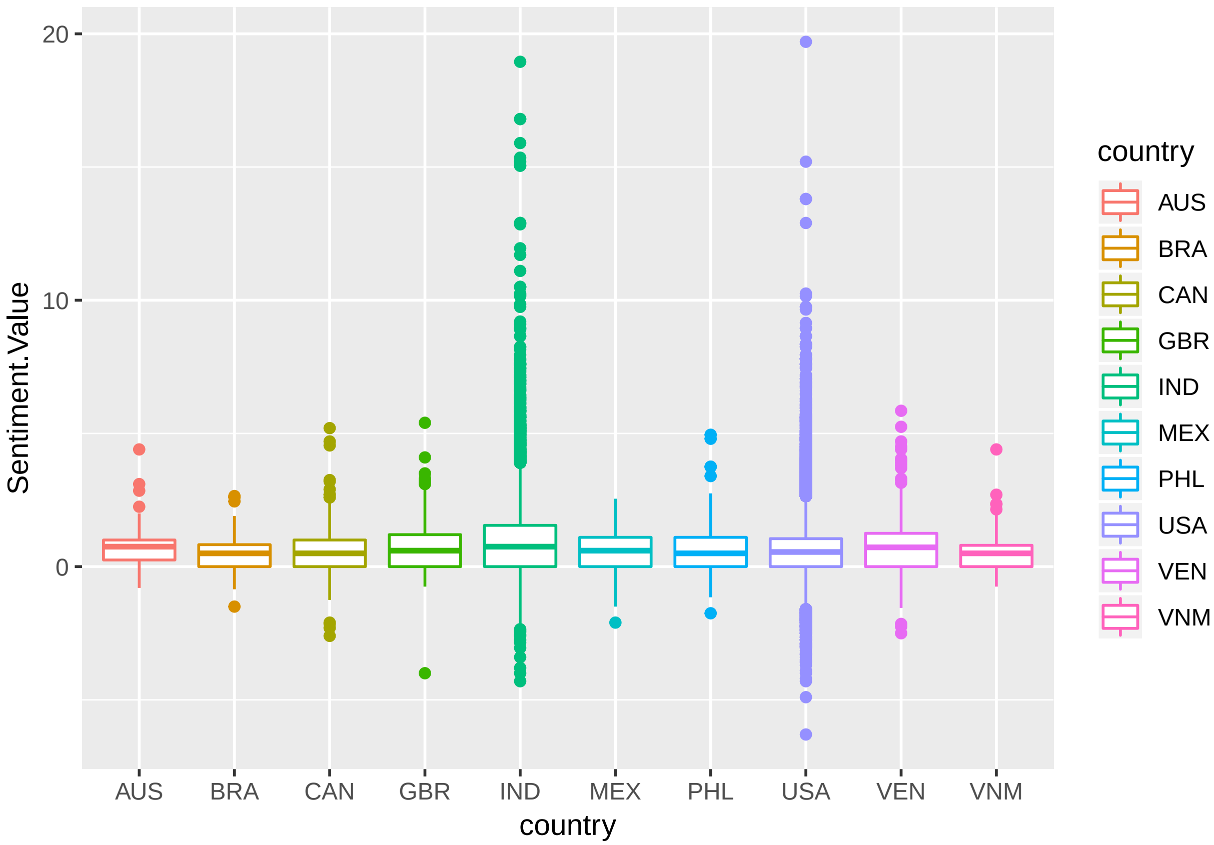



```r
# Sentiment Value vs. 10 Countries with the most word entries
country.int <- tail(names(sort(table(hm_data$country))), 10)
table(hm_data$country)
```

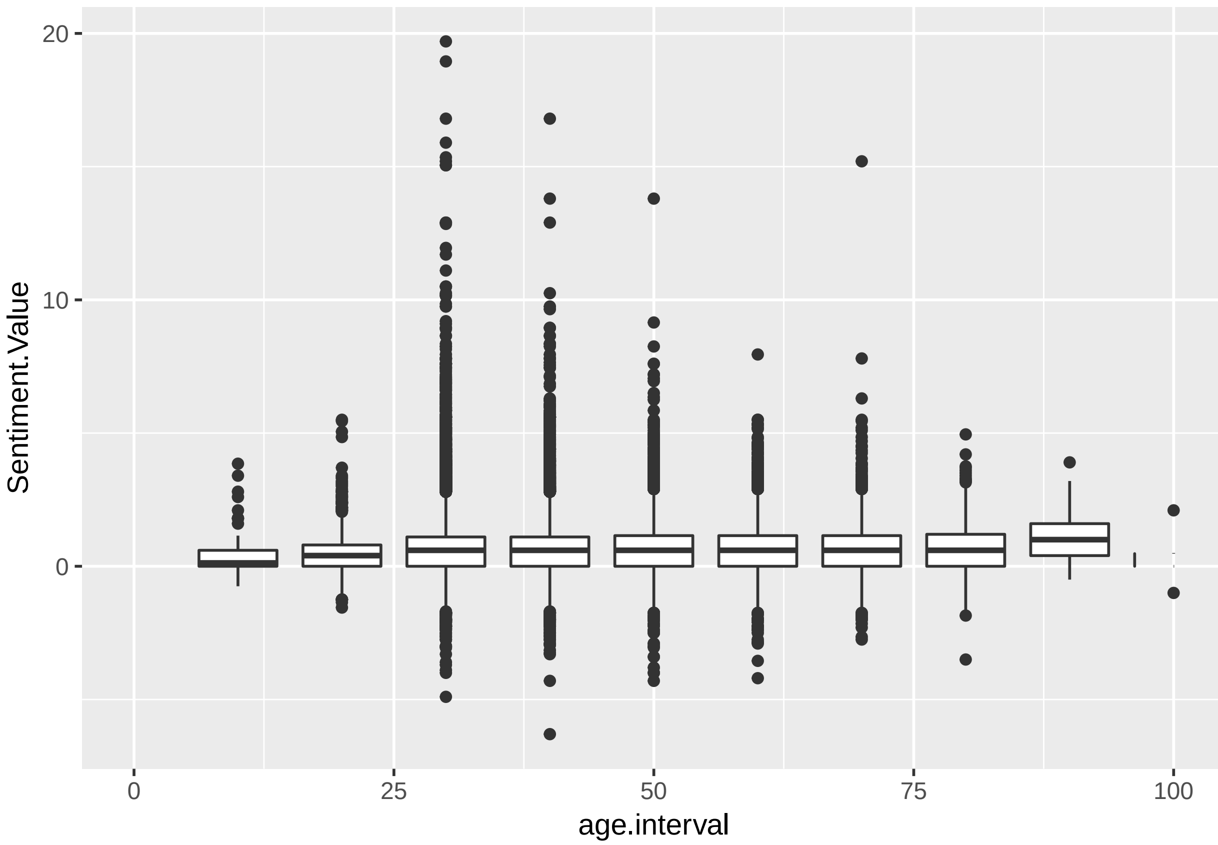
```
##
##   AFG   ALB   ARE   ARG   ARM   ASM   AUS   AUT   BEL   BGD   BGR   BHS
##    11    48    36     6    15    13   117    17    12    69    67     3
##   BRA   BRB   CAN   CHL   COL   CRI   CYP   CZE   DEU   DNK   DOM   DZA
##   123     6   555     6    32     3     3     6    84    51    51    12
##   ECU   EGY   ESP   EST   ETH   FIN   FRA   GBR   GHA   GMB   GRC   GTM
##     3    57    23     6     3    21    51   364     3     6    42     6
##   HKG   HRV   IDN   IND   IRL   IRQ   ISL   ISR   ITA   JAM   JPN   KAZ
##     3     6    90 16713    30     3     9     3    36    60    15     3
##   KEN   KNA   KOR   KWT   LKA   LTU   LVA   MAC   MAR   MDA   MEX   MKD
##    33     9     6    18    12    42     3    18     6    36   150   104
##   MLT   MUS   MYS   NGA   NIC   NLD   NOR   NPL   NZL   PAK   PER   PHL
##     9     3    15    81    15    15     3     6    36    39    34   279
##   POL   PRI   PRT   ROU   RUS   SAU   SGP   SLV   SRB   SUR   SVN   SWE
##    15    30    84    46    30     3    24     3    96     3     6    27
##   TCA   THA   TTO   TUN   TUR   TWN   UGA   UKR   UMI   URY   USA   VEN
##     6    90    30     3    51     9    18     3    15    42 78941   588
##   VIR   VNM   ZAF   ZMB
```

```
##      3    125     21      3
```

```
ggplot(subset(hm_data, country %in% country.int), aes(x = country, y = Sentiment.Value, color = country)
```




```
# Sentiment Value vs. Age groups
hm_data$age <- as.integer(hm_data$age)
x.interval <- seq(0,250,10)
xx.interval <- seq(0,100,25)
hm_data$age.interval <- findInterval(hm_data$age, x.interval)
ggplot(hm_data[!is.na(hm_data$age),], aes(x = age.interval, y = Sentiment.Value, group = age.interval))
```



Step 4 - Topic Modeling

Step 4.1 - LDA Method

Here I used the topicmodels package written by Bettina Gruen and Kurt Hornik. Specifically, we'll use the LDA function with the Gibbs sampling option mentioned in class. Also, since the LDA function has a fairly large number of parameters, I mainly stuck to the ones given in class/tutorial website, but I scaled down the number of burnin, iteration, and thins in order to speed up the sampling process.

```
dtm <- VCorpus(VectorSource(hm_data$text)) %>% DocumentTermMatrix()
rowTotals <- slam::row_sums(dtm)
dtm <- dtm[rowTotals > 0, ]

#Set parameters for Gibbs sampling
burnin <- 800
iter <- 400
thin <- 100
seed <-list(2003,5,63,100001,765)
nstart <- 5
best <- TRUE

#Number of topics
k <- 10

#Run LDA using Gibbs sampling
ldaOut <- LDA(dtm, k, method="Gibbs", control=list(nstart=nstart,
                                                 seed = seed, best=best,
```

```
                                                   burnin = burnin, iter = iter,
                                                   thin=thin))
#write out results
ldaOut.topics <- as.matrix(topics(ldaOut))
table(c(1:k, ldaOut.topics))   # Total number per each topic
```

```
##
##     1     2     3     4     5     6     7     8     9    10
## 16915 12008 11531 10769  5740 11026  8602  9522  7759  6514
```

```
#top 10 terms in each topic
ldaOut.terms <- as.matrix(terms(ldaOut,10))

#probabilities associated with each topic assignment
topicProbabilities <- as.data.frame(ldaOut@gamma)

terms.beta=ldaOut@beta
terms.beta=scale(terms.beta)
topics.terms=NULL
for(i in 1:k){
  topics.terms=rbind(topics.terms, ldaOut@terms[order(terms.beta[i,], decreasing = TRUE)[1:10]])
}

ldaOut.terms   #top 10 terms in each topic
```

```
##       Topic 1     Topic 2    Topic 3   Topic 4      Topic 5
##  [1,] "found"     "day"      "time"    "night"      "feel"
##  [2,] "bought"    "son"      "family"  "morning"    "moment"
##  [3,] "received"  "daughter" "enjoyed" "dog"        "life"
##  [4,] "car"       "event"    "visit"   "hours"      "happiness"
##  [5,] "money"     "school"   "house"   "home"       "people"
##  [6,] "shopping"  "mother"   "home"    "love"       "live"
##  [7,] "purchased" "college"  "spend"   "girlfriend" "person"
##  [8,] "buy"       "told"     "brother" "cat"        "makes"
##  [9,] "free"      "excited"  "trip"    "sleep"      "positive"
## [10,] "store"     "class"    "weekend" "husband"    "experience"
##       Topic 6    Topic 7  Topic 8    Topic 9     Topic 10
##  [1,] "dinner"   "friend" "watched"  "finally"   "walk"
##  [2,] "birthday" "job"    "played"   "finished"  "beautiful"
##  [3,] "wife"     "talked" "game"     "started"   "park"
##  [4,] "surprise" "called" "favorite" "completed" "run"
##  [5,] "lunch"    "met"    "movie"    "weeks"     "weather"
##  [6,] "husband"  "phone"  "won"      "book"      "drive"
##  [7,] "food"     "baby"   "fun"      "ive"       "taking"
##  [8,] "eat"      "sister" "video"    "project"   "nice"
##  [9,] "ate"      "meet"   "team"     "read"      "rain"
## [10,] "mom"      "girl"   "tickets"  "hard"      "bike"
```

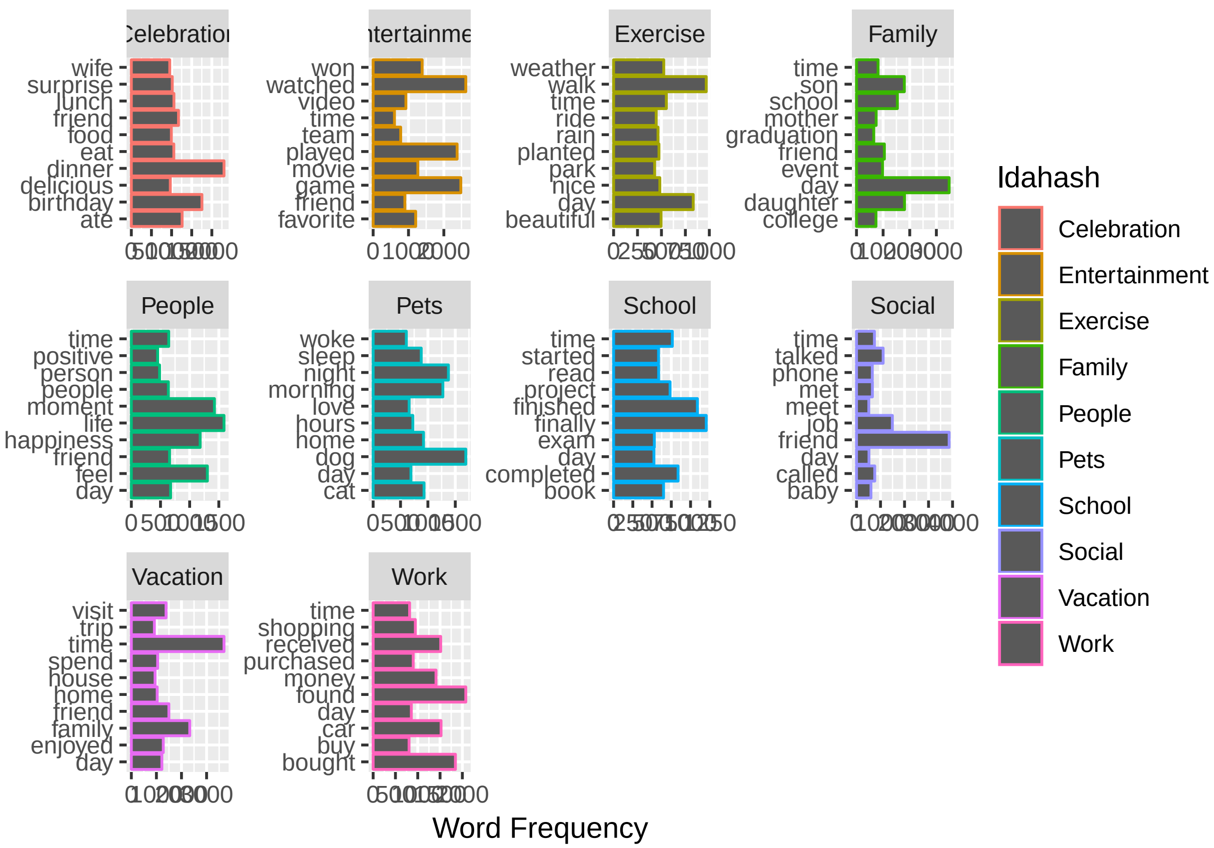
Step 4.2 - Come up with 10 topics

I set the topic numbers to be 10. I manually tag them as “Work”, “Family”,“Vacation”,“Pets”,“People”,“Celebration”,“Social”,“E Because Topic 2 contains the key words: “Son”, “Daughter”, and “Brother”, and Topic 10 contains “Walk”, “Park”, and “Run”,etc. Based on the most popular terms and the most salient terms for each topic, we assign a hashtag to each topic.

```
topics.hash=c("Work","Family","Vacation","Pets","People","Celebration","Social","Entertainment","School

hm_data <- hm_data[rowTotals > 0, ]
hm_data$ldatopic <- as.vector(ldaOut.topics)
hm_data$ldahash <- topics.hash[ldaOut.topics]
colnames(topicProbabilities) <- topics.hash

hm_data.df <- cbind(hm_data, topicProbabilities)
```

Some Visualization here:

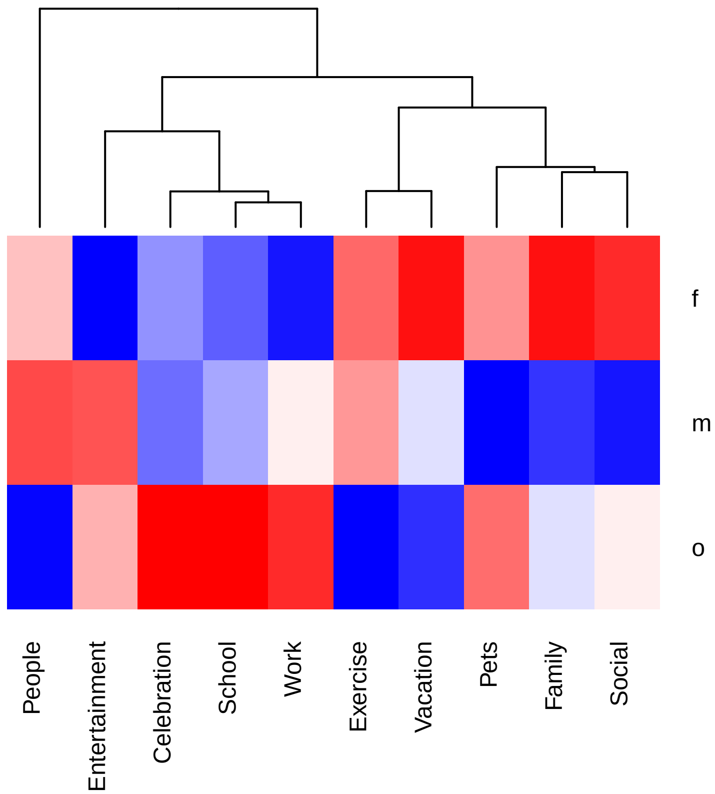
```
word_count_by_topic <- hm_data.df %>%
  unnest_tokens(word, text) %>%
  group_by(ldahash) %>%
  count(word, sort = TRUE)

word_count_by_topic %>%
  slice(1:10) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n, color = ldahash)) + geom_col() +  facet_wrap(~ldahash, scales = "free") +
```



Step 4.3 - Heatmap Analysis

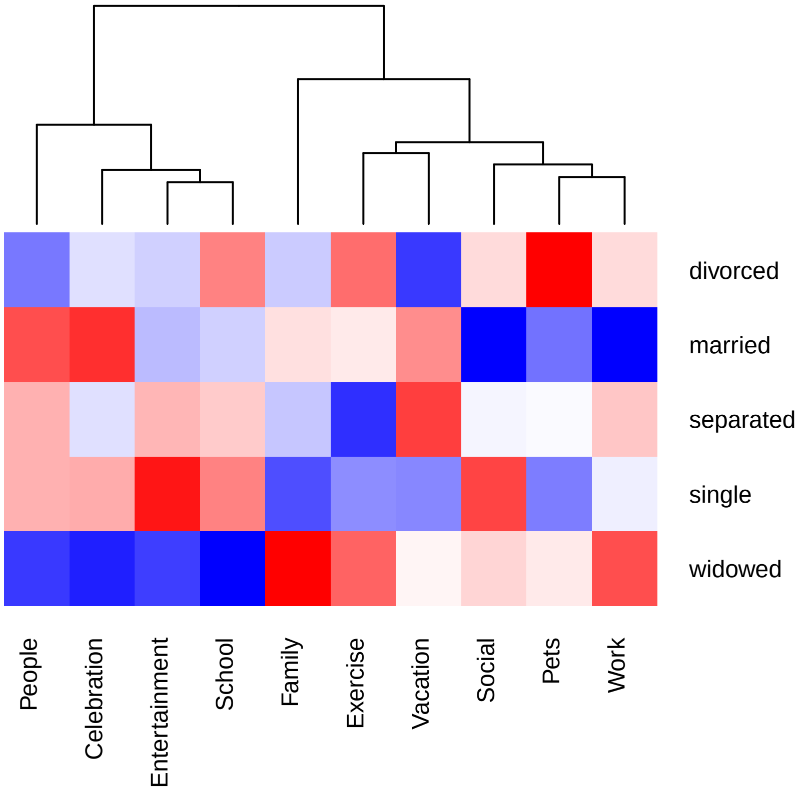
We use heatmap to see the weight allocation of topics grouped by gender. Note that the red color indicates higher weights on that topic. Let's ignore "others" for now, and only focus on male and female. There is a clear trend in between: - Female tend to mention "family", "vacation", and "friends". - Male focus on "people", "entertainment", and "exercise".

```r
par(mar=c(1,1,1,1))
topic.summary1=tbl_df(hm_data.df[!is.na(hm_data.df$gender),])%>%
              select(gender, Work:Exercise)%>%
              group_by(gender)%>%
              summarise_each(funs(mean)) %>%
              as.data.frame()
heatmap.2(as.matrix(topic.summary1[,-1]), Rowv = FALSE,
           scale = "column", key=F, na.rm = T,
           col = bluered(100), labRow = c("f","m","o"),
           cexRow = 0.9, cexCol = 0.9, margins = c(8, 8),
           trace = "none", density.info = "none")
```



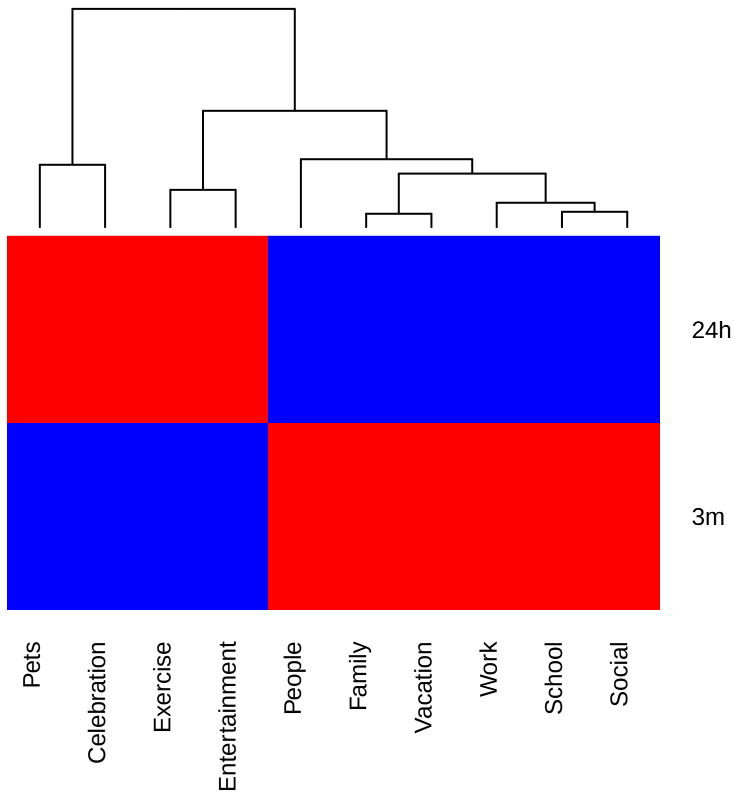
We use heatmap to see the weight allocation of topics grouped by marital status. Different from previous analysis, the information coming from this heatmap may not make perfect sense, probably due to our selection of topics. For example, married people are the only group who focus on "people" and "entertainment" and they don't mention "friends" or "work".

```r
par(mar=c(1,1,1,1))
topic.summary2=tbl_df(hm_data.df[!is.na(hm_data.df$marital),])%>%
              select(marital, Work:Exercise)%>%
              group_by(marital)%>%
              summarise_each(funs(mean)) %>%
              as.data.frame()
heatmap.2(as.matrix(topic.summary2[,-1]), Rowv = FALSE,
           scale = "column", key=F, na.rm = T, col = bluered(100),
          labRow = c("divorced","married","separated","single","widowed"),
           cexRow = 0.9, cexCol = 0.9, margins = c(8, 8),
           trace = "none", density.info = "none")
```



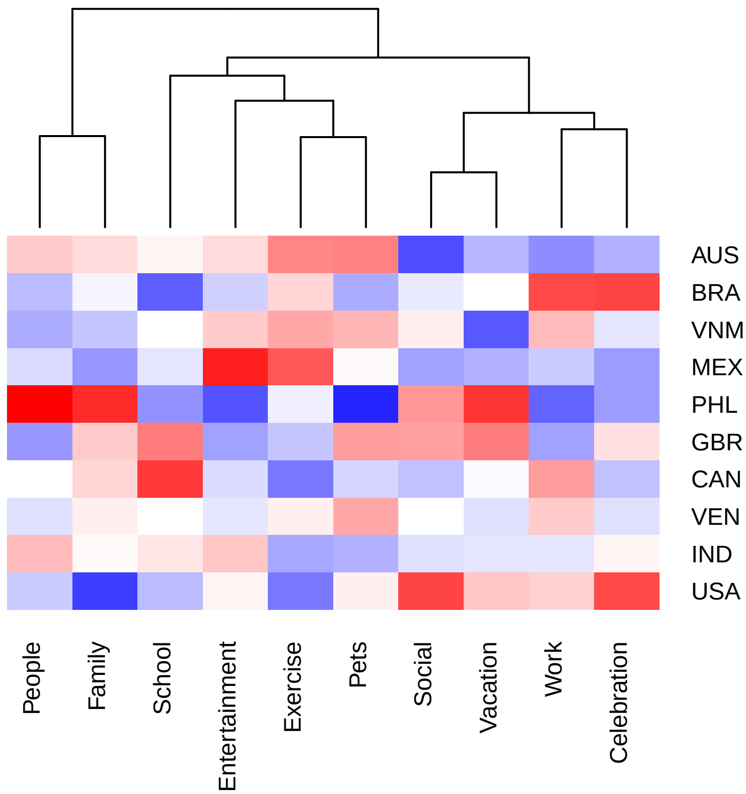
We use heatmap to see the weight allocation of topics grouped by reflection period. This heatmap prefectly illustrates the difference between the two groups. For short-term happy memories, people tend to say things about "pets", "entertainment", "celebration", and "exercise"; while for long-term happy memories, people often mentions things like their "people", "work", or "school".

```
par(mar=c(1,1,1,1))
topic.summary3=tbl_df(hm_data.df[!is.na(hm_data.df$reflection_period),])%>%
              select(reflection_period, Work:Exercise)%>%
              group_by(reflection_period)%>%
              summarise_each(funs(mean)) %>%
              as.data.frame()
heatmap.2(as.matrix(topic.summary3[,-1]), Rowv = FALSE,
           scale = "column", key=F, na.rm = T, col = bluered(100),
          labRow = c("24h","3m"),
           cexRow = 0.9, cexCol = 0.9, margins = c(8, 8),
           trace = "none", density.info = "none")
```



Again, similar to the marital status heatmap, the heatmap data on the top 10 countries with the most data entries does not provide much information either. I was expecting countries like "AUS", "GBR", and "USA" to behave similarly, but they tend to differ a little. I was also expecting asian countries like "PHL", "VNM" and "IND" to have similar colors as well.

```
par(mar=c(1,1,1,1))
topic.summary4=subset(hm_data.df, country %in% country.int) %>%
              tbl_df()%>%
              select(country, Work:Exercise)%>%
              group_by(country)%>%
              summarise_each(funs(mean)) %>%
              as.data.frame()
heatmap.2(as.matrix(topic.summary4[,-1]), Rowv = FALSE,
           scale = "column", key=F, na.rm = T, col = bluered(100),
          labRow = country.int,
           cexRow = 0.9, cexCol = 0.9, margins = c(8, 8),
           trace = "none", density.info = "none")
```



Step 5 - Summary

(1) Few NLP methods were used throughout my project, including sentiment analysis and topic modeling. And several analytics/visualization tools were used, including boxplot, wordcloud, and heatmap.

(2) I first examined the relationship between predicted_cateory (given in the dataset) vs. other parameters. The main takeaways are: -For both male and female, most frequent words are both "friend", "day" and "time". -However, the difference is that for 24h data, words that tend to be memorized in a short term are also very hot, such as "watched", "morning", "dinner"; while for 3m data, words that represent more significant events tend to show up more, such as "job", "home", and "birthday".

(3) Sentiment value was also utilized to determine if different demographic groups behave differently. But the result shows that the difference (among gender, marital status, or age) is quite minor.

(4) By using LDA method, a list of 10 topics was manually entered. This heatmap on reflection_period prefectly illustrates the difference between the two groups. For short-term happy memories, people tend to say things about "pets", "entertainment", "celebration", and "exercise"; while for long-term happy memories, people often mentions things like their "people", "work", or "school".