Project 1

Mingyu Yang

9/12/2018

What are the things that make single parent happy? Are they any different than these for married ones? What about comparing people who are single to people who are married.

According to data published by the United States' Census Bureau, around 30% of children in the United Staes are raised by single parent. There are many debates on whether single parents are as "good" as married ones. Those debates surround on topis including finanticial stability, psychological effect on childrens, social mobility and etc. These topic typically focus on the state of children instead of the parent herself/himself. An interesting question come in mind is what are the things that make single parent happy? Are they any different than those for married ones? How about just comparing people who are sinlge and who are married?

We will use the dataset HappyDB for this instresting analysis. HappyDB is a corpus of 100,000 happy moments gatherd by the Amazon Mechinical Turk in 2017. We will study the entire dataset and then compare data provided by single parent and married parent respectively. We will then compare study single people and married ones and see if they exsit similar trend with people with children.

The current R version I'm using and the necessary packages used are listed below. The data are imported and stored in the dataframe "hm_data". We did some cleaning of the data. We removed punctuations, extra whitesapce, empty wors and numbers and converted upper case to lower case. We reduce words to ehri stem and remove meaningless stop words. We further combine the demographic data with the original data set. We then imported the demographic data to our cleaned data. We will use this cleaned data for further analysis.

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

R.Version()$version.string
```

```
## [1] "R version 3.5.1 (2018-07-02)"
```

```
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)
```

```
hm_data <- hm_data %>%
  inner_join(demo_data, by = "wid") %>%
```

```
  select(wid,
         original_hm,
         gender,
         marital,
         parenthood,
         reflection_period,
         age,
         country,
         ground_truth_category,
         text) %>%
  mutate(count = sapply(hm_data$text, wordcount)) %>%
  filter(gender %in% c("m", "f")) %>%
  filter(marital %in% c("single", "married")) %>%
  filter(parenthood %in% c("n", "y")) %>%
  filter(reflection_period %in% c("24h", "3m")) %>%
  mutate(reflection_period = fct_recode(reflection_period,
                                        months_3 = "3m", hours_24 = "24h"))
```

```
hm_data.sp <- hm_data[hm_data$marital == "single" & hm_data$parenthood == "y",]
hm_data.mp <- hm_data[hm_data$marital == "married" & hm_data$parenthood == "y",]
hm_data.cp <- rbind(hm_data.sp,hm_data.mp)


hm_data.s <- hm_data[hm_data$marital == "single" & hm_data$parenthood == "n",]
hm_data.m <- hm_data[hm_data$marital == "married" & hm_data$parenthood == "n",]
hm_data.c <- rbind(hm_data.s,hm_data.m)

dim(hm_data.sp)
```

```
## [1] 5036   11
```

```
dim(hm_data.mp)
```

```
## [1] 30679    11
```

```
dim(hm_data.cp )
```

```
## [1] 35715    11
```

```
dim(hm_data.s)
```

```
## [1] 48564    11
```

```
dim(hm_data.m)
```

```
## [1] 10295    11
```

```
dim(hm_data.c )
```

```
## [1] 58859    11
```

General Analysis

Length of Sentences

The length of sentences are stored in the count column of our cleaned dataset and we visualized this through a scatterplot, where each color represent a group within the plot. Married parents wrote more words in descriping their experiences.

```
library(shiny)
```

```
##
## Attaching package: 'shiny'
```

```
## The following objects are masked from 'package:DT':
##
##     dataTableOutput, renderDataTable
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

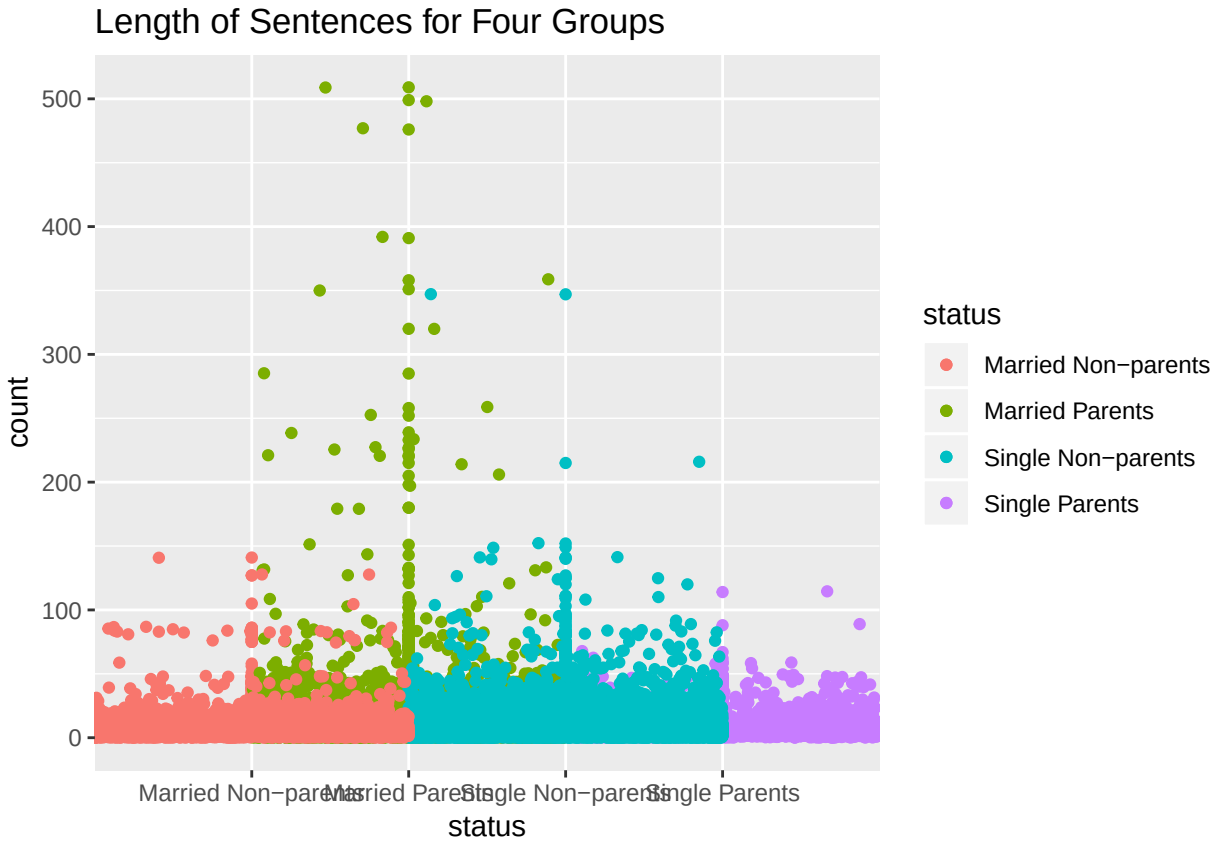
```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
hm_data.cp$status <- ifelse(hm_data.cp$marital == "single", "Single Parents", "Married Parents")
hm_data.c$status <-ifelse(hm_data.c$marital == "single", "Single Non-parents", "Married Non-parents")
alldata <- rbind(hm_data.cp,hm_data.c)


ggplot(alldata, aes(status, count, color = status)) +
  geom_point() +
  geom_jitter(width = 1, height = 1) +
  ggtitle("Length of Sentences for Four Groups")
```



Word Frequency with Wordcloud.

We are going to examine the frequencies of words for each group and visualize this the with wordcloud.

For single parents, the top three most frequent appearance words are “day”, “time” and “friends”. The too three words are “day”, “time” and “son” for married parents. The word “friend” drops to fifth place for married parents. This could indicates friends may be a more important aspect of life for single parent.

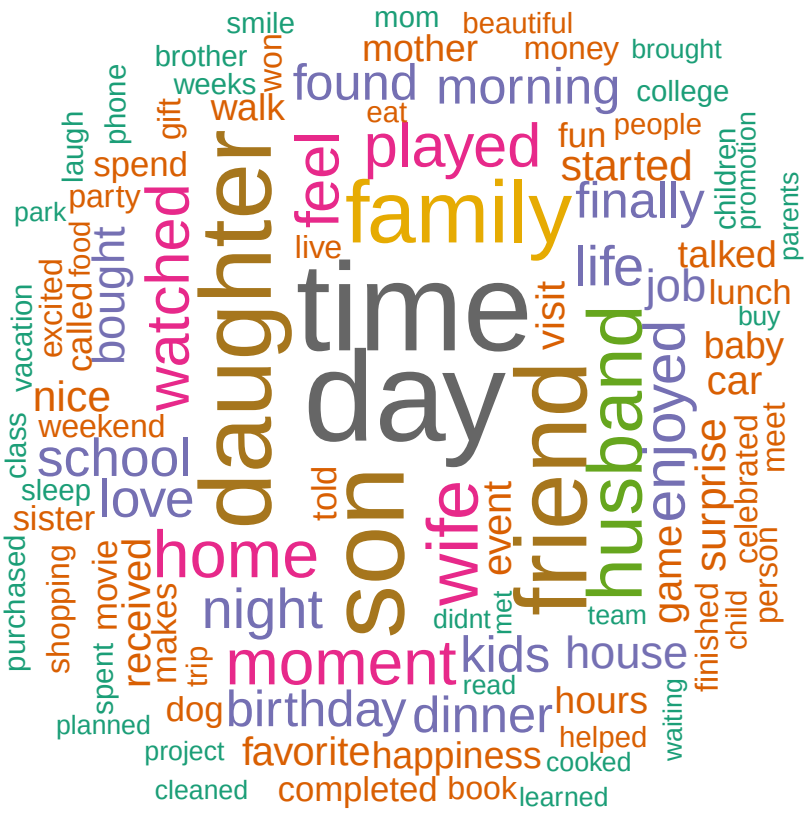
Let’s take a look at the comparison between single non-parents and married non-parents. The top three most frequent words are the same for both groups. They are “date”, “time” and “friends”. Besides the obvious finding that non-parents would not have the term “son” or “daughtor” on their list, the words “family”, “husband”, and “wife” do not make it to the top three. For people who do not have chidren, even if they are married, they seems to put more emphasize on friends instead of families.

```
#wordcloud for single parents.
docs.sp <- Corpus(VectorSource(hm_data.sp$text))
dtm.sp <- TermDocumentMatrix(docs.sp)
m.sp <- as.matrix(dtm.sp)
v.sp <- sort(rowSums(m.sp),decreasing=TRUE)
d.sp <- data.frame(word = names(v.sp),freq=v.sp)

set.seed(123)
wordcloud(words = d.sp$word, freq = d.sp$freq, min.freq = 1,
          max.words=100, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```


```
#wordcloud for married parents
docs.mp <- Corpus(VectorSource(hm_data.mp$text))
dtm.mp <- TermDocumentMatrix(docs.mp)
m.mp <- as.matrix(dtm.mp)
v.mp <- sort(rowSums(m.mp),decreasing=TRUE)
d.mp <- data.frame(word = names(v.mp),freq=v.mp)

set.seed(123)
wordcloud(words = d.mp$word, freq = d.mp$freq, min.freq = 1,
          max.words=100, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```




```
#wordcould for single non-parents.
docs.s <- Corpus(VectorSource(hm_data.s$text))
dtm.s <- TermDocumentMatrix(docs.s)
m.s <- as.matrix(dtm.s)
v.s <- sort(rowSums(m.s),decreasing=TRUE)
d.s <- data.frame(word = names(v.s),freq=v.s)
head(d.s, 10)
```

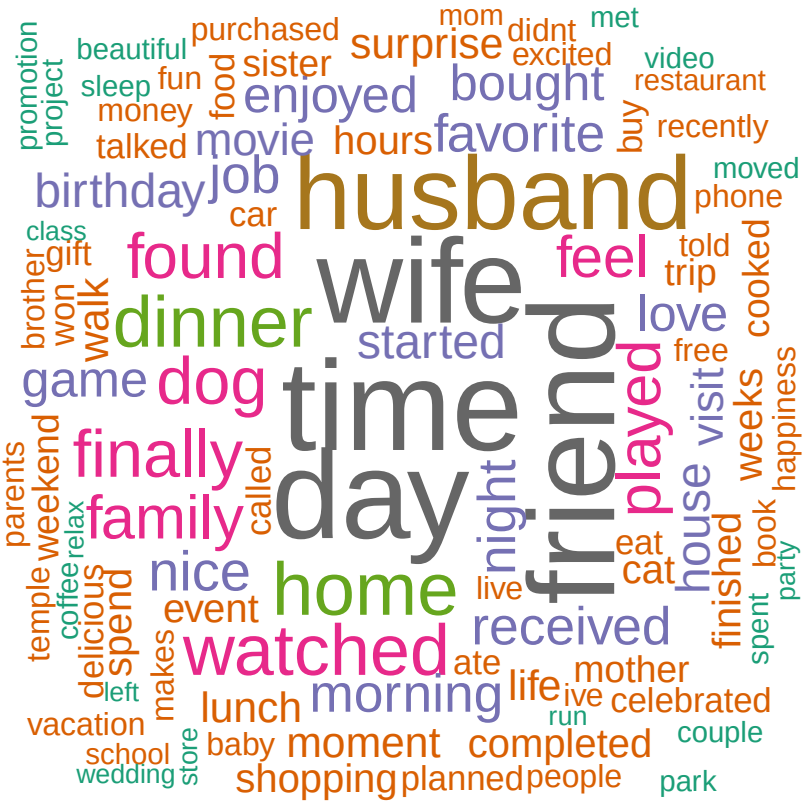
```
##            word freq
## friend   friend 6409
## day         day 4517
## time       time 4439
## watched watched 2070
## played   played 1958
## finally finally 1953
## game       game 1910
## found     found 1869
## feel       feel 1770
## family   family 1741
```

```
set.seed(123)
wordcloud(words = d.s$word, freq = d.s$freq, min.freq = 1,
          max.words=100, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```


```
#wordcould for married non-parents.
docs.m <- Corpus(VectorSource(hm_data.m$text))
dtm.m <- TermDocumentMatrix(docs.m)
m.m <- as.matrix(dtm.m)
v.m <- sort(rowSums(m.m),decreasing=TRUE)
d.m <- data.frame(word = names(v.m),freq=v.m)
head(d.m, 10)
```

```
##            word freq
## time       time  960
## day         day  940
## friend   friend  903
## wife       wife  883
## husband husband  754
## home       home  511
## dinner   dinner  490
## watched watched  476
## dog         dog  432
## finally finally  427
```

```
set.seed(123)
wordcloud(words = d.m$word, freq = d.m$freq, min.freq = 1,
          max.words=100, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```



Sentiment Analysis

We will measure the sentiments of these happy moments. Although they should all exibihit positive and joyful sentiment as inherient by the study, the intensities of the sentments may vary between groups.

We first compare the single parents and married ones. There isn't a significant diffrence by looking at the boxplot. There are a lot more people who identified themselves as married parents than single ones. Lookng at the mean of the sentiment score, both groups have higher sentiment score comparing to the entire samples. Married couple has a slightly higher score than the single ones. We neglect other factors such as docioeconomic status which may also paly a role this analysis.

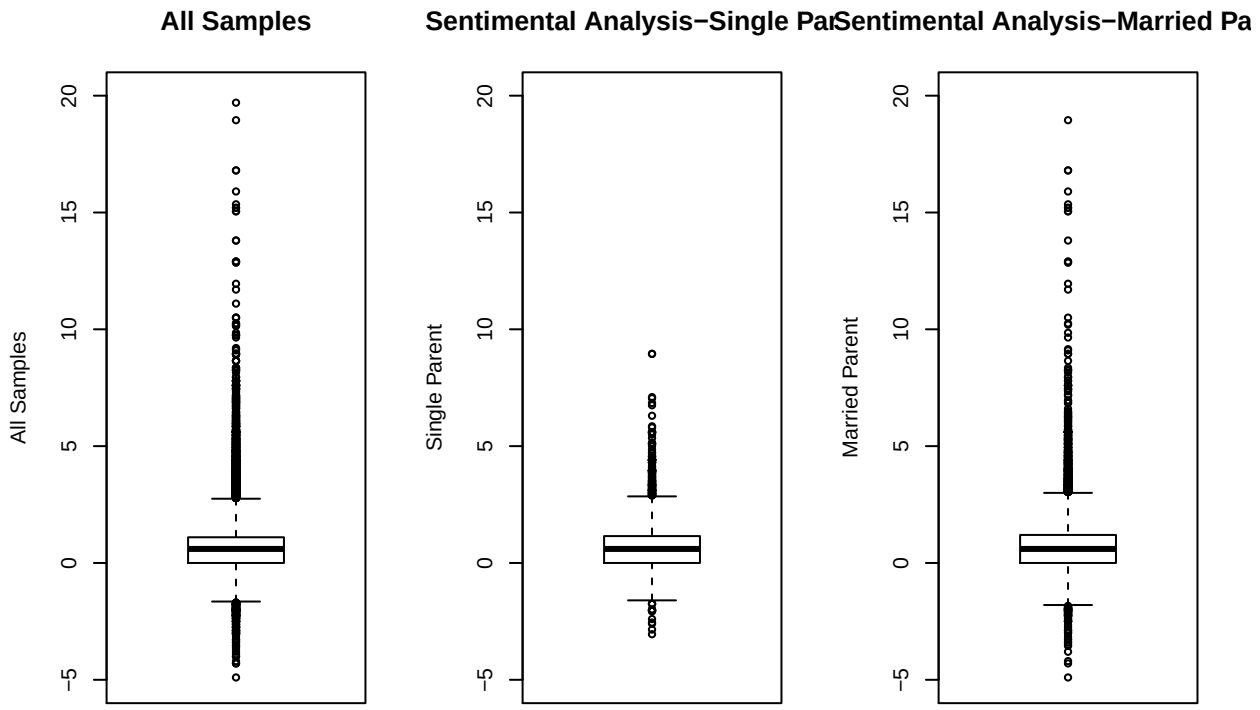
We then take a look at the sentiment scores of non-parents individuals. An interesting finding is that non-parents individuals has lower scores the entire sample. Married individuals still has high score than single ones but the diffirence is really small.

Let's look at the happoest moment for each group. Not suprising, the happiest moments for parents, married or not, include mensioning family members.The happiest moments for non-parents both discribe specific experiences.

```
# all samples vs. single parents vs. married parents
par(mfrow = c(1,3))
sentiment.all <- get_sentiment(hm_data$text, method = "syuzhet")
boxplot(sentiment.all, ylim=c(-5,20), main = "All Samples", ylab = "All Samples")

sentiment.sp <- get_sentiment(hm_data.sp$text, method = "syuzhet")
boxplot(sentiment.sp, ylim=c(-5,20), main = "Sentimental Analysis-Single Parent", ylab = "Single Parent
```

```
sentiment.mp <- get_sentiment(hm_data.mp$text, method = "syuzhet")
boxplot(sentiment.mp, ylim=c(-5,20), main = "Sentimental Analysis-Married Parent", ylab = "Married Pare
```




```
mean(sentiment.all)
```

```
## [1] 0.6965313
```

```
mean(sentiment.sp)
```

```
## [1] 0.7169579
```

```
mean(sentiment.mp)
```

```
## [1] 0.7579452
```

```
#find the happiest sentences
hm_data[which.max(sentiment.sp),]$original_hm
```

```
## [1] "I had a very enjoyable lunch with my children and mother-in-law."
```

```
hm_data[which.max(sentiment.mp),]$original_hm
```

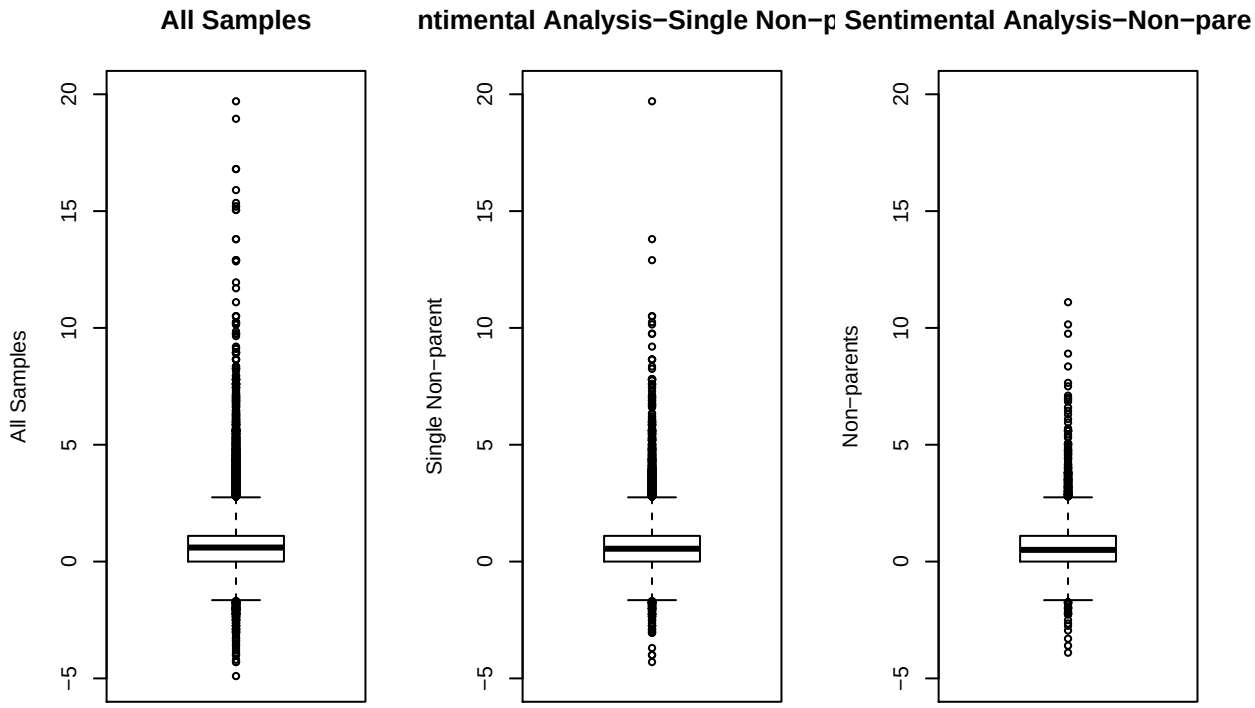
```
## [1] "When my wife and I \"facetime\" with my daughter and we get to see our grandson, our daughter, a
```

```
par(mfrow = c(1,3))

boxplot(sentiment.all, ylim=c(-5,20), main = "All Samples", ylab = "All Samples")

sentiment.s <- get_sentiment(hm_data.s$text, method = "syuzhet")
boxplot(sentiment.s, ylim=c(-5,20), main = "Sentimental Analysis-Single Non-parents", ylab = "Single Non


sentiment.m <- get_sentiment(hm_data.m$text, method = "syuzhet")
boxplot(sentiment.m, ylim=c(-5,20), main = "Sentimental Analysis-Non-parents", ylab = "Non-parents")
```




```
mean(sentiment.all)
```

```
## [1] 0.6965313
```

```
mean(sentiment.s)
```

```
## [1] 0.6614921
```

```
mean(sentiment.m)
```

```
## [1] 0.668815
```

```
#find the happiest sentences
hm_data[which.max(sentiment.s),]$original_hm
```

```
## [1] "Today I signed the paper and took the parcel. My hands were itching to open the packet as my cu
```

```
hm_data[which.max(sentiment.m),]$original_hm
```

```
## [1] "I watch a funny reality show in the TV."
```

Topic Modeling

We fit the LDA model. Since we have to set the number of topics manually, we will try setting three to six numbers. It turns out that setting three topics number make the most sense. We manully tag each topic as "families", "friend" and "Experience" because of word that appear most frequenct in those topics.

```
order.data.cp <- hm_data.cp[order(hm_data.cp$wid),]
order.data.cp$status <- ifelse(order.data.cp$marital == "single", 1, 2)
order.data.c <- hm_data.c[order(hm_data.c$wid),]
order.data.c$status <- ifelse(order.data.c$marital == "single", 3, 4)

combine <- rbind(order.data.cp,order.data.c)
```

```
combine.data <- combine  %>%
                group_by(wid) %>%
                summarise(text = paste(text, collapse = " "), status = mean(status))

combine.docs <- Corpus(VectorSource(combine.data$text))

dtm.combine <- DocumentTermMatrix(combine.docs)
```

```
lda <- LDA(dtm.combine, k = 3, control = list(seed = 1234))
lda
```

```
## A LDA_VEM topic model with 3 topics.
```

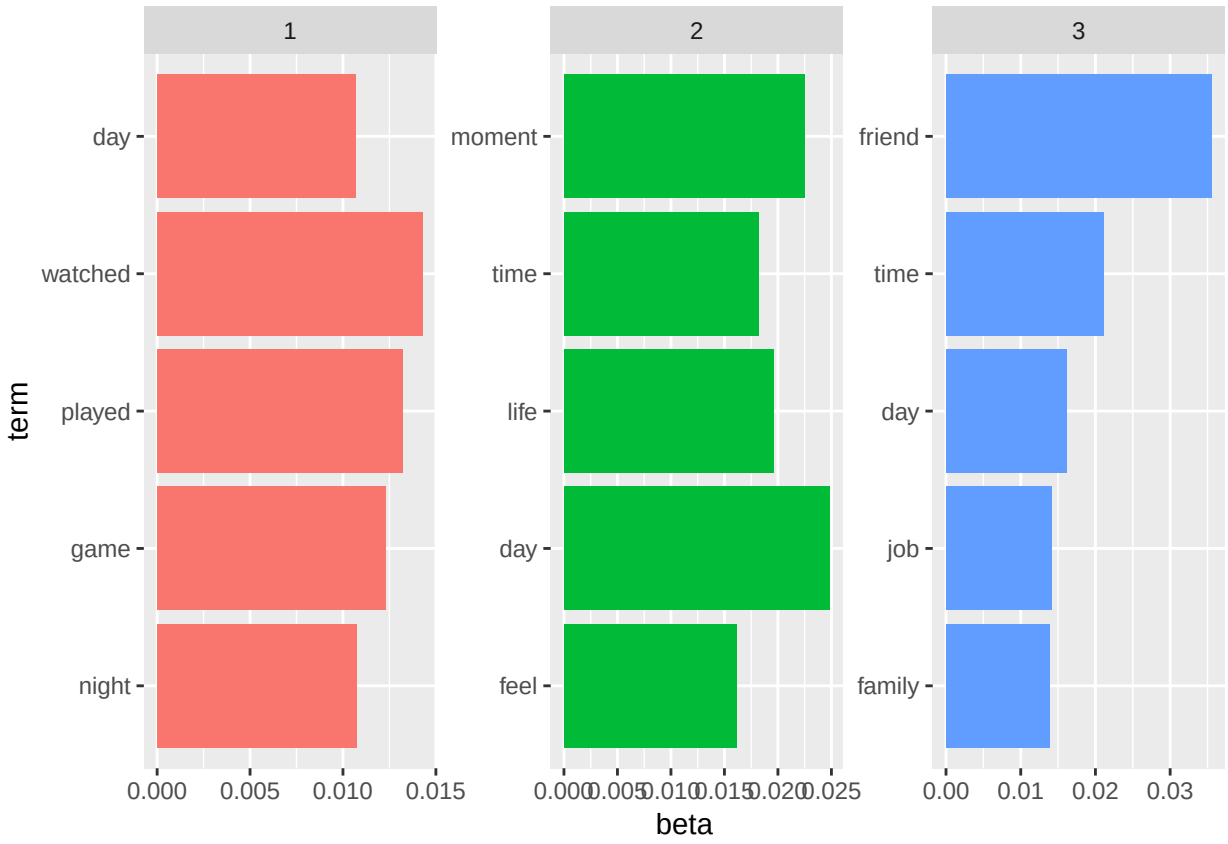
```
topics <- tidy(lda, matrix = c("beta", "gamma"))
topics
```

```
## # A tibble: 55,683 x 3
##    topic term          beta
##    <int> <chr>        <dbl>
##  1     1 accepted  7.52e- 5
##  2     2 accepted  5.05e- 4
##  3     3 accepted  2.33e- 3
##  4     1 afford    1.59e- 4
##  5     2 afford    1.14e- 5
##  6     3 afford    5.15e- 4
##  7     1 allergies 7.19e- 5
##  8     2 allergies 3.30e-11
##  9     3 allergies 3.27e- 5
## 10     1 amount    5.51e- 4
## # ... with 55,673 more rows
```

Find the top ten words in each topic.

```
top.words <- topics %>%
  group_by(topic) %>%
  top_n(5, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top.words %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```



From the heat map, it seems like single parents single and married parents mentions more about families where non-parents mentions more about friends and experiences.

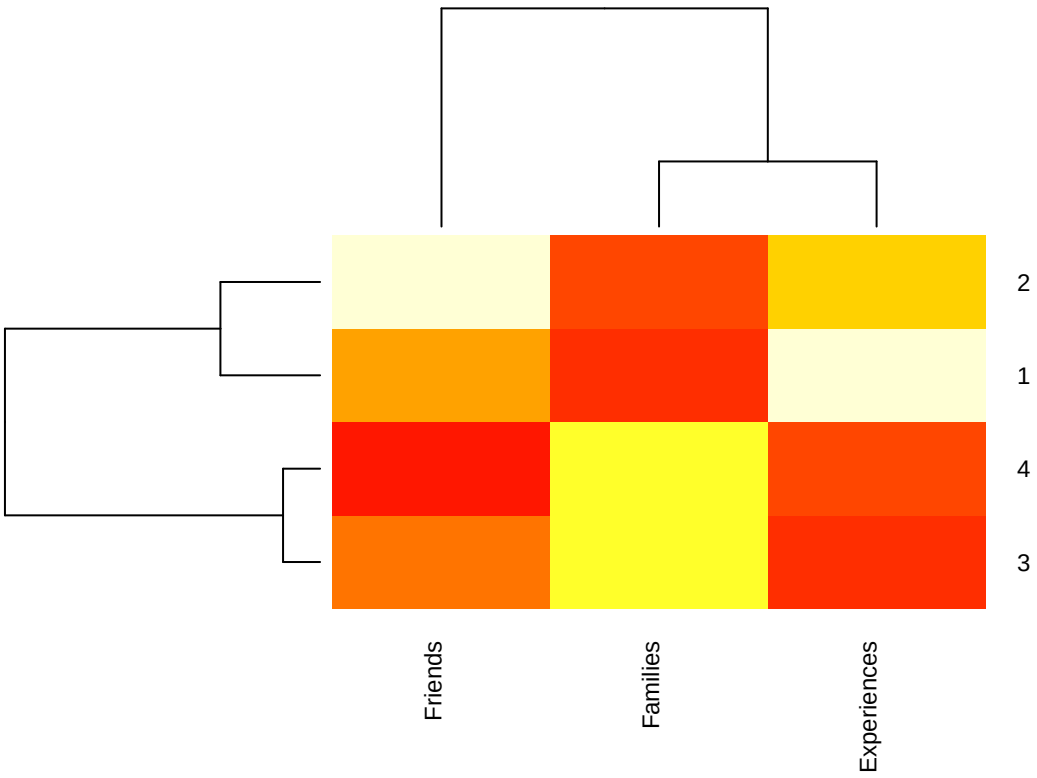
```
topic.prob <- as.data.frame(lda@gamma)
topics.tag <- c("Families", "Friends", "Experiences")
lda.topics <- as.matrix(topics(lda))
combine.data$topic <- as.vector(lda.topics)
combine.data$ldatag <- topics.tag[lda.topics]
colnames(topic.prob) <- topics.tag

combine.data.corpus <- cbind(combine.data,topic.prob)
```

```
topic.status<-tbl_df(combine.data.corpus)%>%
              select(status, Families:Experiences)%>%
              group_by(status)%>%
              summarise_all(funs(mean))


topic.status=as.data.frame(topic.status)

rownames(topic.status)<-topic.status$status
heatmap.2(as.matrix(topic.status[,-1]),
          scale = "column", key=F,
          col = "heat.colors",
          cexRow = 0.9, cexCol = 0.9, margins = c(8, 8),
          trace = "none", density.info = "none")
```

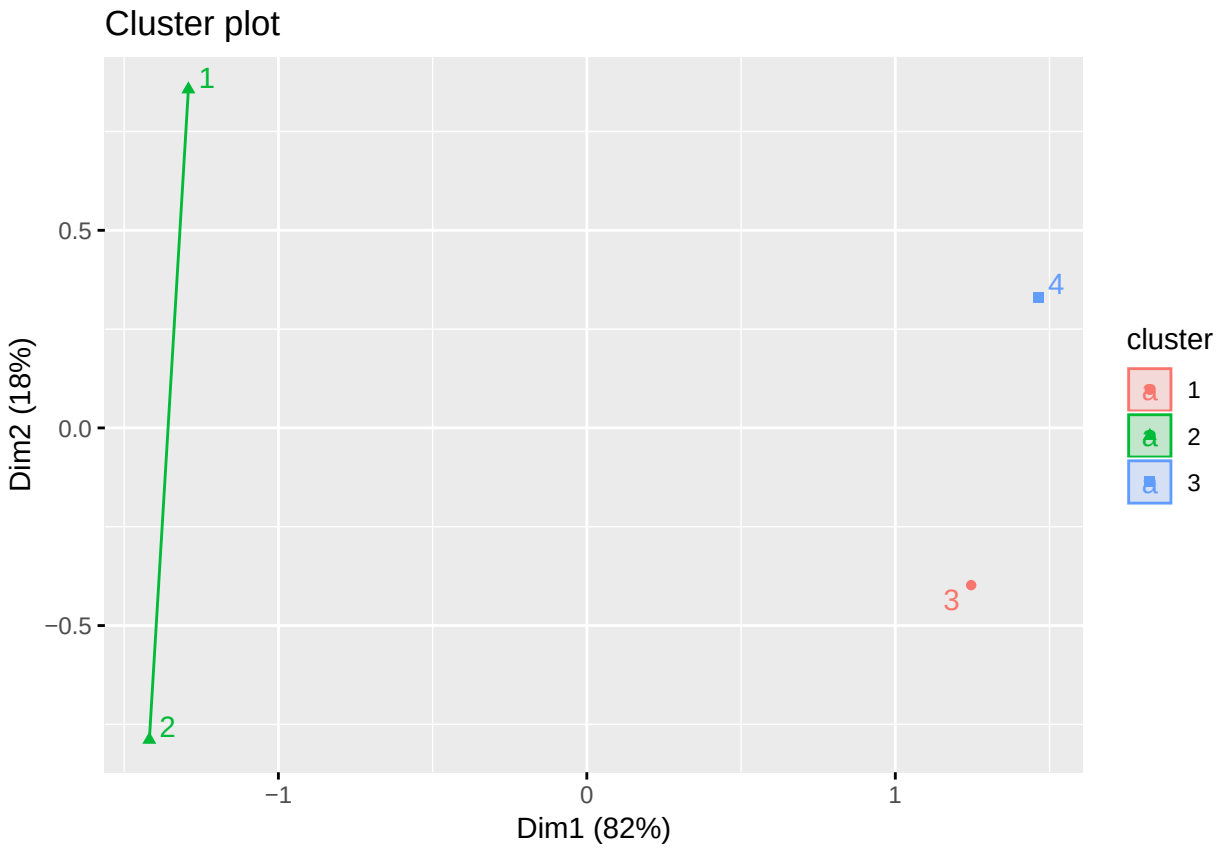


clustering

```
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
km.res=kmeans(scale(topic.status[,-1]), iter.max=200, centers=3)
fviz_cluster(km.res, stand=T, repel= TRUE, data = topic.status[,-1], show.clust.cent=FALSE)
```



sources used:

http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know

https://www.tidytextmining.com/topicmodeling.html

https://medium.freecodecamp.org/a-data-scientists-guide-to-happiness-findings-from-the-happy-experiences-of-10-000-humans