# Stat GR5243 Project 1

Yicheng Guo
StudentID: yg2956

HappyDB is a corpus of 100,000 crowd-sourced happy moments via Amazon's Mechanical Turk. You can read more about it on https://arxiv.org/abs/1801.07746

In this R notebook, we process the raw textual data for our data analysis.

### Step 0 - Load all the required libraries

From the packages' descriptions:

- `tm` is a framework for text mining applications within R;
- `tidyverse` is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures;
- `tidytext` allows text mining using 'dplyr', 'ggplot2', and other tidy tools;
- `DT` provides an R interface to the JavaScript library DataTables.

```
library(tm)
library(knitr)
library(tidytext)
library(tidyverse)
library(DT)
library(wordcloud2)
library(scales)
library(gridExtra)
library(ngram)
```

### Step 1 - Load the data to be cleaned and processed

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

### Step 2 - Preliminary cleaning of text

We clean the text by converting all the letters to the lower case, and removing punctuation, numbers, empty words and extra white space.

```
corpus <- VCorpus(VectorSource(hm_data$cleaned_hm))%>%
  tm_map(content_transformer(tolower))%>%
  tm_map(removePunctuation)%>%
  tm_map(removeNumbers)%>%
  tm_map(removeWords, character(0))%>%
  tm_map(stripWhitespace)
```

### Step 3 - Stemming words and converting tm object to tidy object

Stemming reduces a word to its word *stem*. We stem the words here and then convert the "tm" object to a "tidy" object for much faster processing.

```
stemmed <- tm_map(corpus, stemDocument) %>%
  tidy() %>%
  select(text)
```

### Step 4 - Creating tidy format of the dictionary to be used for completing stems

We also need a dictionary to look up the words corresponding to the stems.

```
dict <- tidy(corpus) %>%
  select(text) %>%
  unnest_tokens(dictionary, text)
```

```
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
```

### Step 5 - Removing stopwords that don't hold any significant information for our data set

We remove stopwords provided by the "tidytext" package and also add custom stopwords in context of our data.

```
data("stop_words")

word <- c("happy","ago","yesterday","lot","today","months","month",
                 "happier","happiest","last","week","past")

stop_words <- stop_words %>%
  bind_rows(mutate(tibble(word), lexicon = "updated"))
```

### Step 6 - Combining stems and dictionary into the same tibble

Here we combine the stems and the dictionary into the same "tidy" object.

```
completed <- stemmed %>%
  mutate(id = row_number()) %>%
  unnest_tokens(stems, text) %>%
  bind_cols(dict) %>%
  anti_join(stop_words, by = c("dictionary" = "word"))
```

```
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
```

### Step 7 - Stem completion

Lastly, we complete the stems by picking the corresponding word with the highest frequency.

```
completed <- completed %>%
  group_by(stems) %>%
  count(dictionary) %>%
  mutate(word = dictionary[which.max(n)]) %>%
  ungroup() %>%
  select(stems, word) %>%
  distinct() %>%
  right_join(completed) %>%
  select(-stems)
```

### Step 8 - Pasting stem completed individual words into their respective happy moments

We want our processed words to resemble the structure of the original happy moments. So we paste the words together to form happy moments.

```
completed <- completed %>%
  group_by(id) %>%
  summarise(text = str_c(word, collapse = " ")) %>%
  ungroup()
```

### Step 9 - Keeping a track of the happy moments with their own ID

```
hm_data <- hm_data %>%
  mutate(id = row_number()) %>%
  inner_join(completed)
```

### Exporting the processed text data into a CSV file

```
write_csv(hm_data, "processed_moments.csv")
```

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

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

### Combine both the data sets and keep the required columns for analysis

We select a subset of the data that satisfies specific row conditions.

```
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"))
```

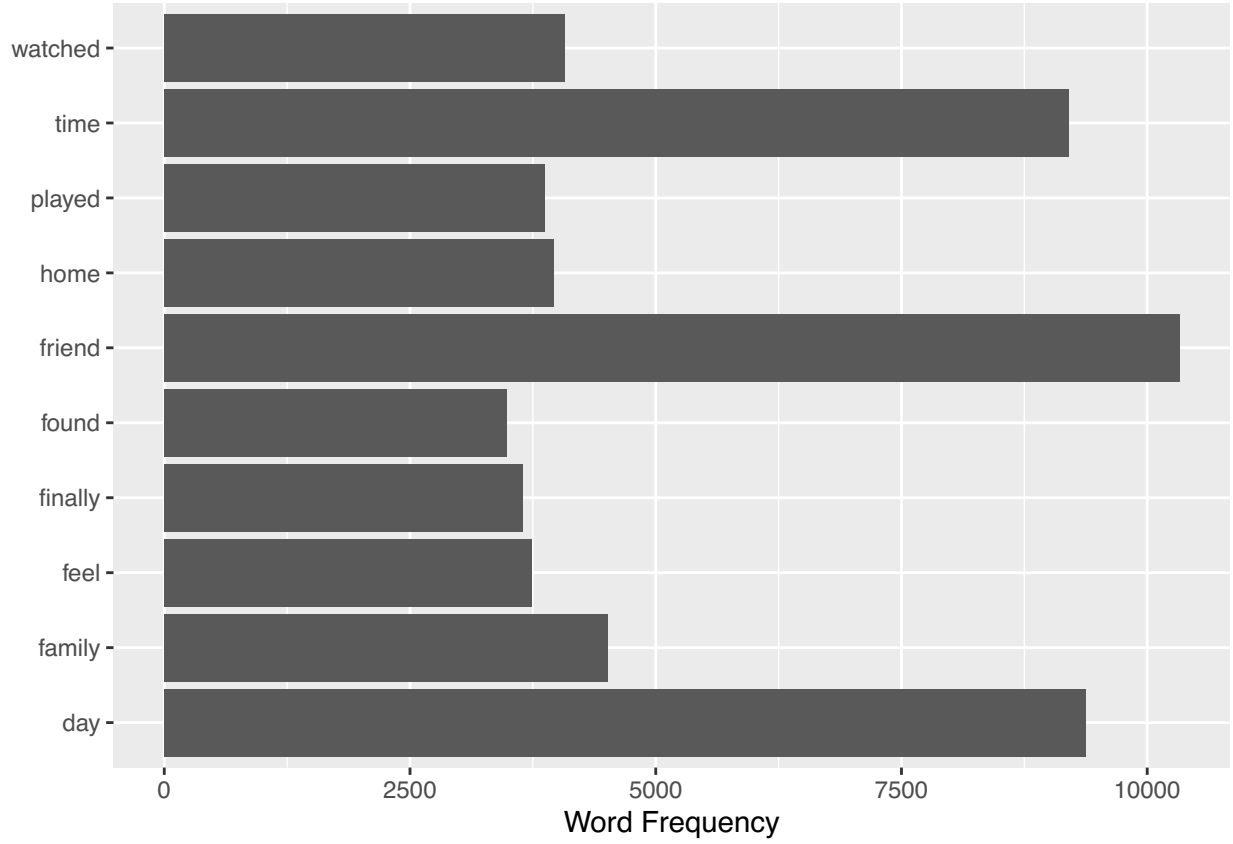
**Create a bag of words using the text data**

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

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

**Create bigrams using the text data**

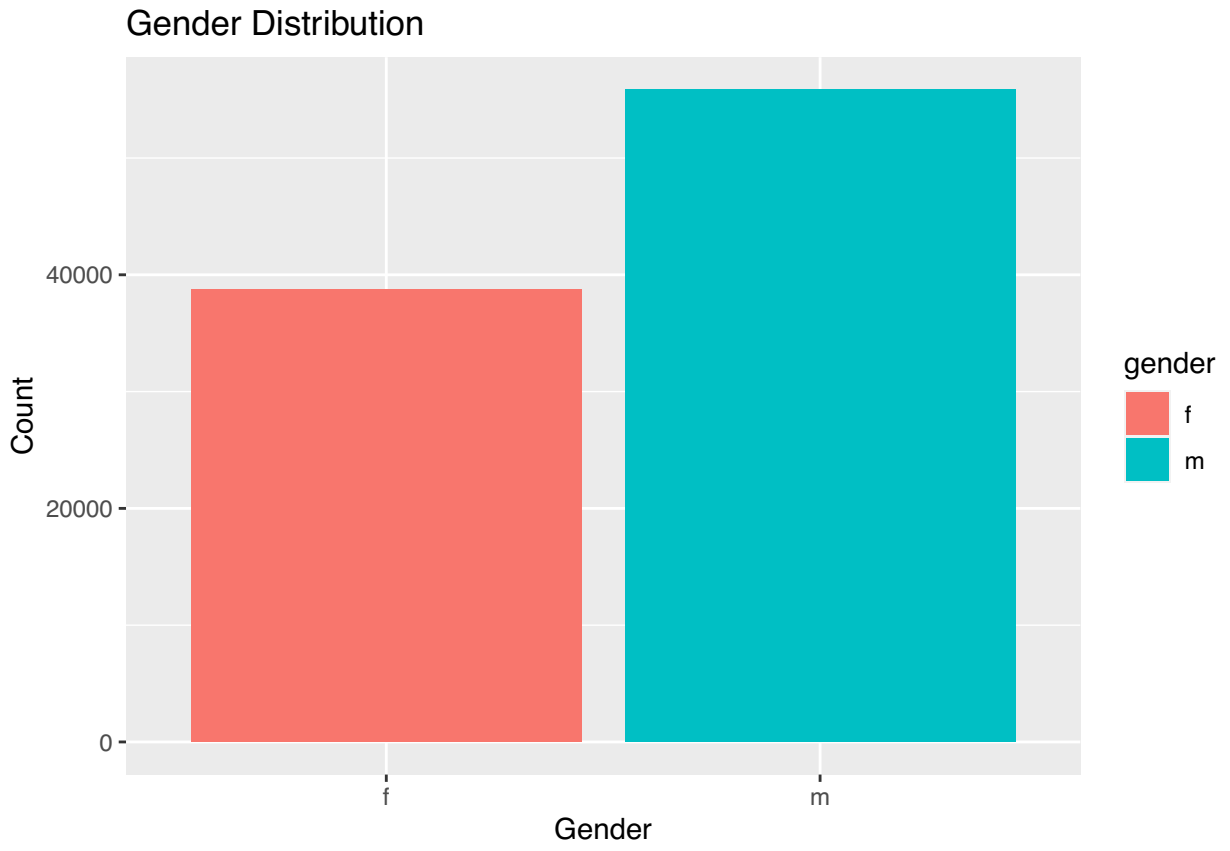
```
hm_bigrams <- hm_data %>%
  filter(count != 1) %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)

bigram_counts <- hm_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  count(word1, word2, sort = TRUE)
```

```
# Word Frequency Bar Chart
word_count_plot <- word_count %>%
  slice(1:10) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  ylab("Word Frequency") +
  coord_flip()

# Display the plot
print(word_count_plot)
```



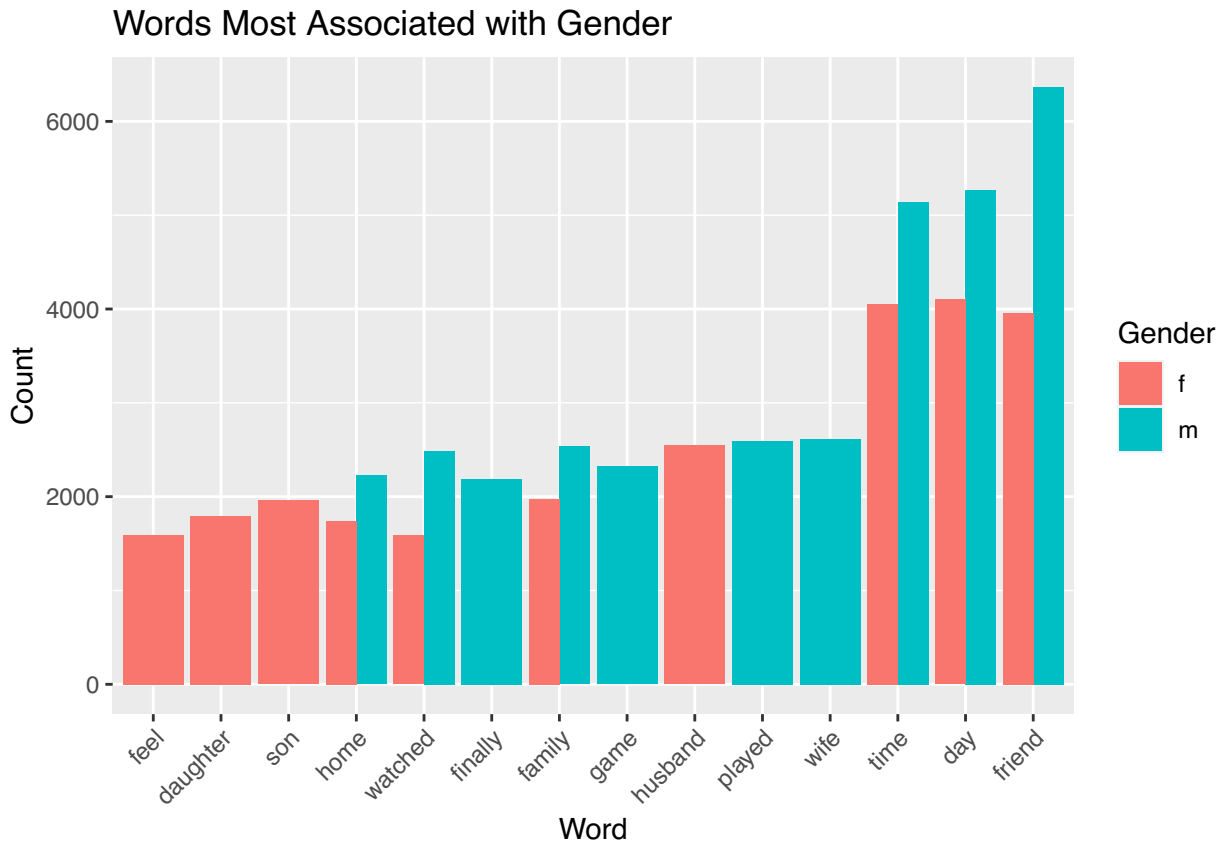

The bar chart above shows the top ten most frequently occurring words in the processes moments dataset. The word "friend" appeared the most times with over 10,000 times. The second most frequently occuring word was "day" while the third is "time". Other words also appearing in the top ten include words related to activities or doing something such as "watched", "played", "found", and "feel". The word "family" is also in the top ten.

```
# Bar Chart of Gender Distribution
gender_distribution_plot <- hm_data %>%
  ggplot(aes(x = gender, fill = gender)) +
  geom_bar() +
  labs(title = "Gender Distribution", x = "Gender", y = "Count")

# Display the plot
print(gender_distribution_plot)
```



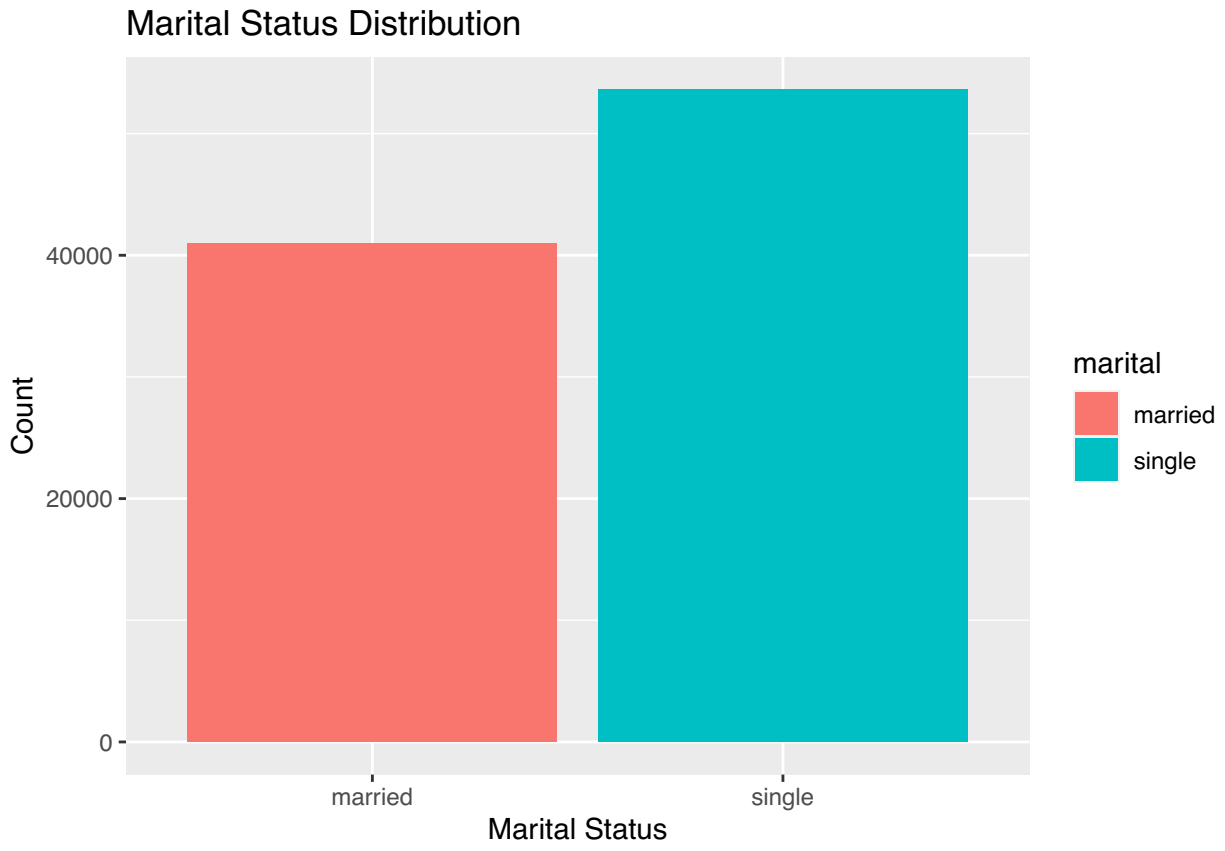

The bar chart above shows the gender distribution of the respondents. There are more males than females.

```
# Words Most Associated with Gender
gender_word_association_plot <- hm_data %>%
  unnest_tokens(word, text) %>%
  filter(gender %in% c("m", "f")) %>%
  count(gender, word, sort = TRUE) %>%
  group_by(gender) %>%
  top_n(10, n) %>%
  ungroup() %>%
  ggplot(aes(x = reorder(word, n), y = n, fill = gender)) +
  geom_col(position = "dodge") +
  labs(title = "Words Most Associated with Gender",
       x = "Word", y = "Count",
       fill = "Gender") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Display the plot
print(gender_word_association_plot)
```



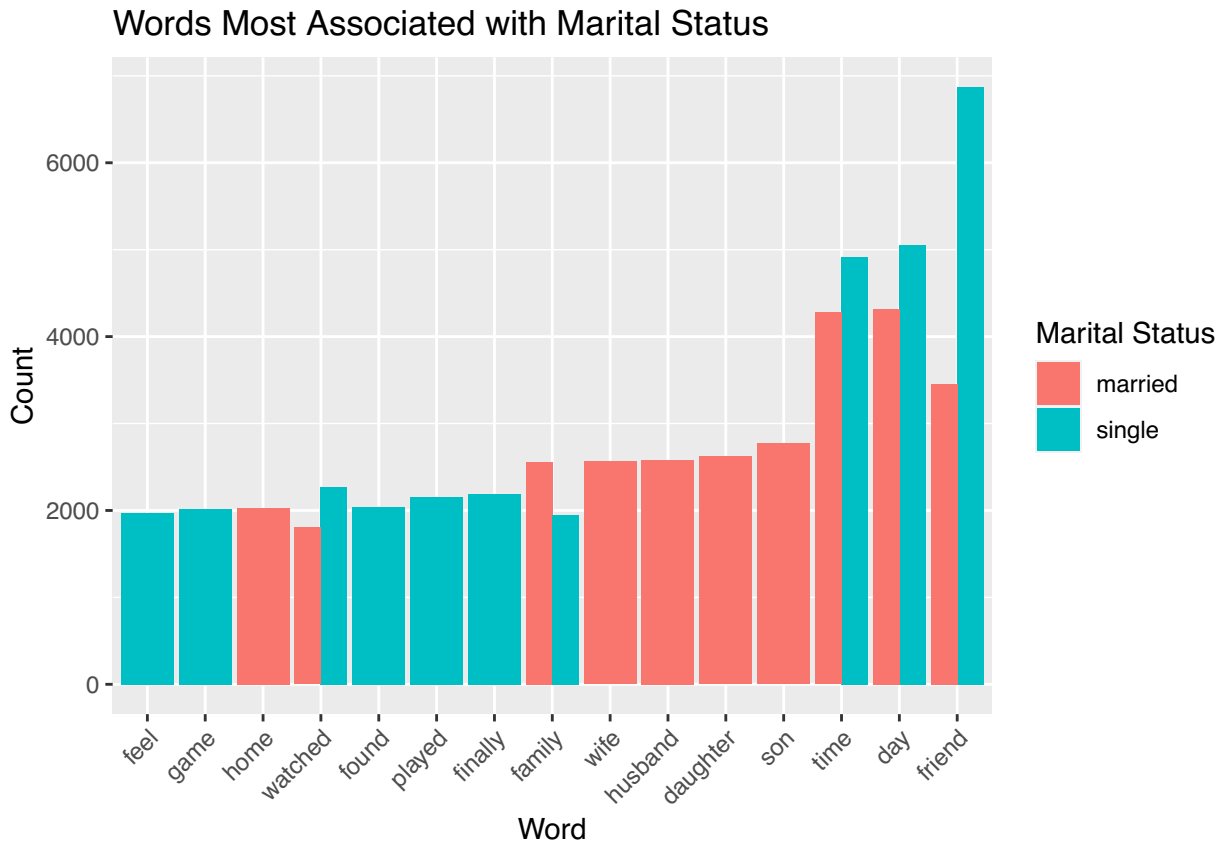

The clustered bar chart above shows the words most associated per gender. The words feel, daughter, son, and husband were exclusive to the females while the words finally, game, played, and wife were exclusive to the males. Words shared by both genders were home, watched, family, time, day, and friend, with the latter three being the top three most frequently occurring words for both gender. This implies that both males and females associate the terms time, day, and friend to happiness.

```
# Bar Chart of Marital Status Distribution
marital_distribution_plot <- hm_data %>%
  ggplot(aes(x = marital, fill = marital)) +
  geom_bar() +
  labs(title = "Marital Status Distribution", x = "Marital Status", y = "Count")

# Display the plot
print(marital_distribution_plot)
```



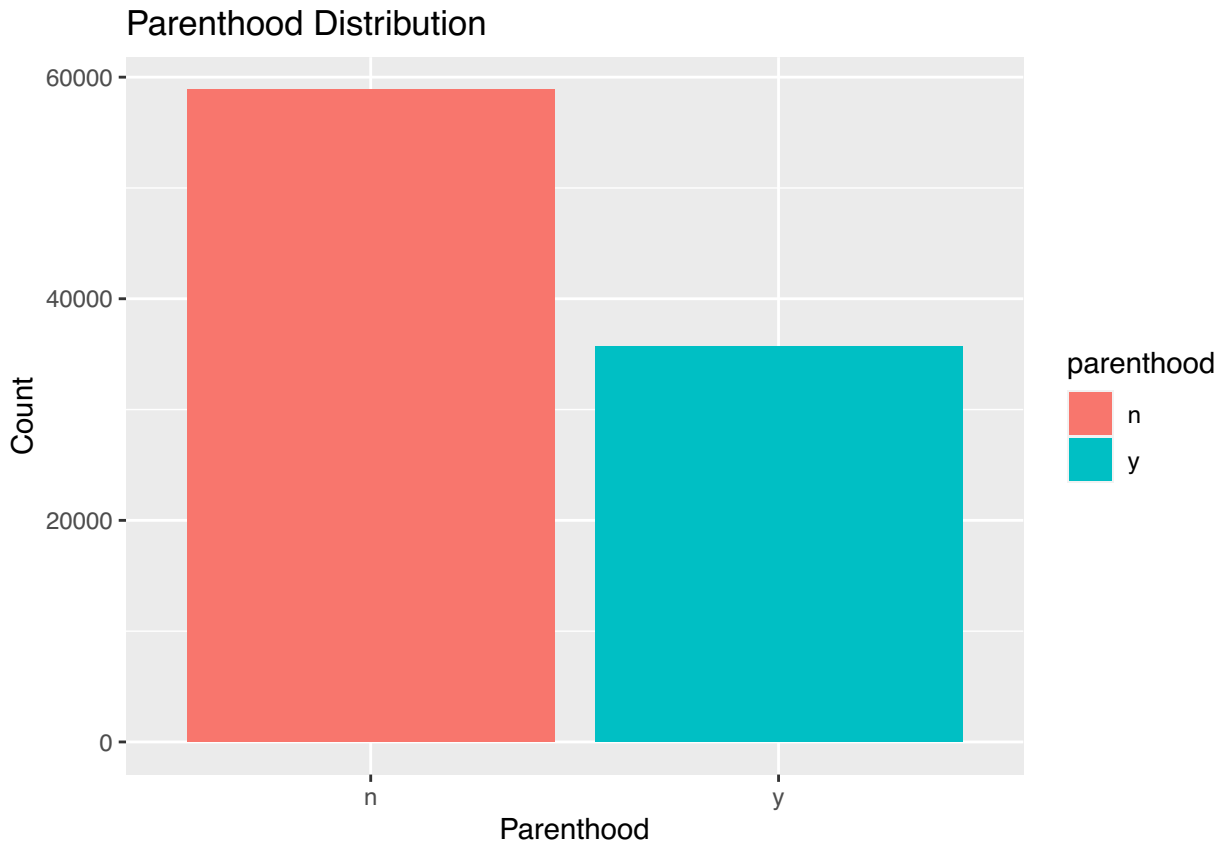

The bar chart above shows the marital status distribution of the respondents. There are more singles than married respondents.

```
# Words Most Associated with Marital Status
marital_word_association_plot <- hm_data %>%
  unnest_tokens(word, text) %>%
  filter(marital %in% c("single", "married")) %>%
  count(marital, word, sort = TRUE) %>%
  group_by(marital) %>%
  top_n(10, n) %>%
  ungroup() %>%
  ggplot(aes(x = reorder(word, n), y = n, fill = marital)) +
  geom_col(position = "dodge") +
  labs(title = "Words Most Associated with Marital Status",
       x = "Word", y = "Count",
       fill = "Marital Status") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Display the plot
print(marital_word_association_plot)
```



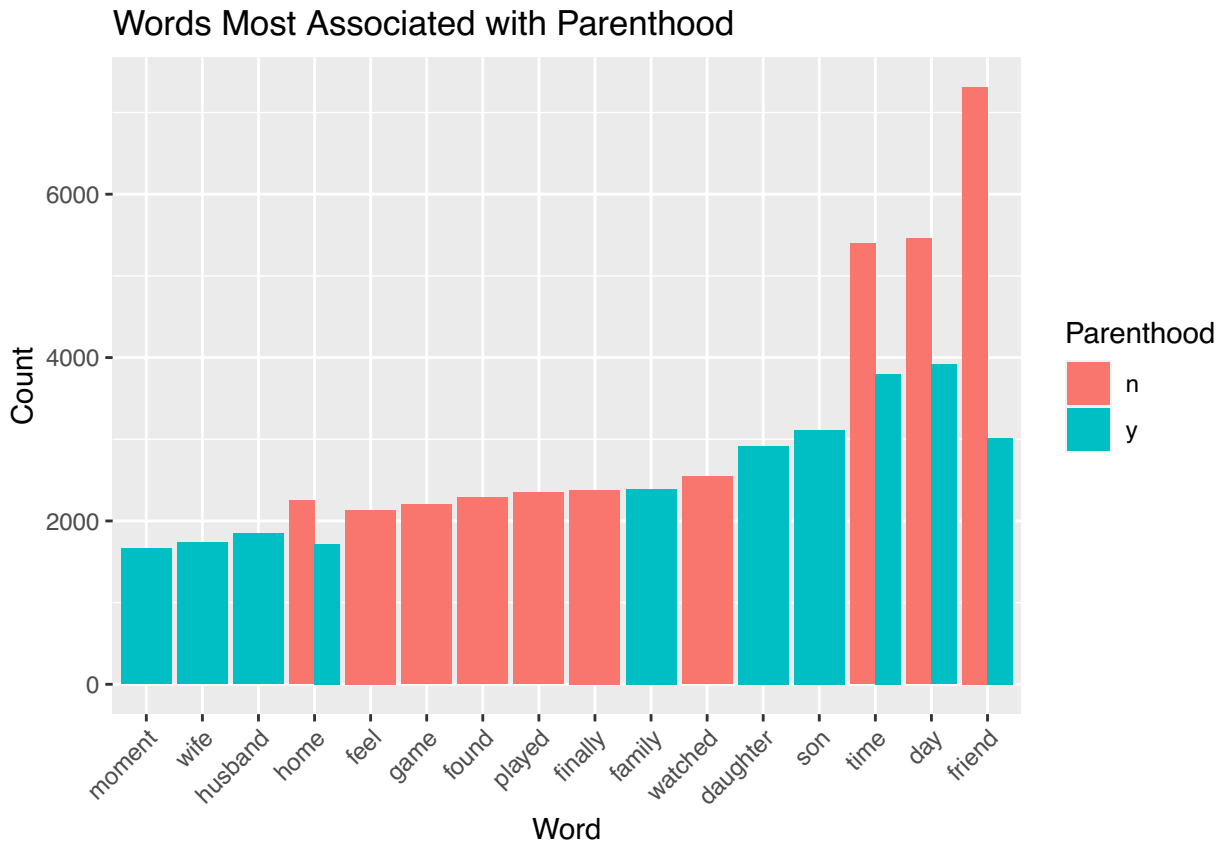

The clustered bar chart above shows the words most associated per marital status. Not surprisingly, the words home, wife, husband, daughter, and son were the words exclusive to married individuals. These terms are the ones that married individuals associate with happiness. On the other hand, the terms feel, game, found, played, and finally are the terms associated with single individuals. These terms are somehow related to activities. Words shared by both married and single respondents were watched, time, day, and friend, with the latter three being the top three most frequently occurring words for both singles and married individuals. This implies that both singles and married respondents associate the terms time, day, and friend to happiness.

```
# Bar Chart of Parenthood Distribution
parenthood_distribution_plot <- hm_data %>%
  ggplot(aes(x = parenthood, fill = parenthood)) +
  geom_bar() +
  labs(title = "Parenthood Distribution", x = "Parenthood", y = "Count")

# Display the plot
print(parenthood_distribution_plot)
```




The bar chart above shows the parenthood distribution of the respondents. There are more non-parents than parents.

```
# Words Most Associated with Parenthood
parenthood_word_association_plot <- hm_data %>%
  unnest_tokens(word, text) %>%
  filter(parenthood %in% c("n", "y")) %>%
  count(parenthood, word, sort = TRUE) %>%
  group_by(parenthood) %>%
  top_n(10, n) %>%
  ungroup() %>%
  ggplot(aes(x = reorder(word, n), y = n, fill = parenthood)) +
  geom_col(position = "dodge") +
  labs(title = "Words Most Associated with Parenthood",
       x = "Word", y = "Count",
       fill = "Parenthood") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Display the plot
print(parenthood_word_association_plot)
```




The clustered bar chart above shows the words most associated per parenthood status. Not surprisingly, the words wife, husband, daughter, and son were the words exclusive to parents. The terms family and moment are also exclusive to parents. These terms are the ones that parents associate the most with happiness. On the other hand, the terms feel, game, found, played, finally, and watched are exclusive to non-parents. These terms pertain to activities done such as watching and playing. Four terms are associated with both parents and non-parents: home, time, day, and friend, with the latter three being the most frequent.

Overall, it would seem like people, regardless of gender, marital status, or parenthood status most associate the termm happeiness with time, day, and friend, with friend being the top one in all cases. It would seem like people are always happy when they spend their time with their friends.

References: Akari Asai, Sara Evensen, Behzad Golshan, Alon Halevy, Vivian Li, Andrei Lopatenko, Daniela Stepanov, Yoshihiko Suhara, Wang-Chiew Tan, Yinzhan Xu, “HappyDB: A Corpus of 100,000 Crowdsourced Happy Moments’ ’, LREC ’18, May 2018. (to appear)