Analysis of text message conversations combined from multiple sources

By: Aaron Stearns

This analysis focuses on understanding the text messaging interactions from a 14,000+ message conversation between my former girlfriend and myself over a period of 18 months.

These messages come from two sources: the SMS messenger on our phones, and Facebook Messenger.

To obtain these messages, I used two apps. The first app allows you to download messages from your Android phone as text/PDF/CSV: <https://play.google.com/store/apps/details?id=com.aes.textbackup>

And the second app is a Chrome plugin that allows you to download any Facebook messenger conversation in its entirety: <https://chrome.google.com/webstore/detail/messages-saver-for-facebo/oaimmbbnnhmibhbocchkfcakiankbnlk>

First off, I’ll be installing the following packages:

library(dplyr)  
library(plotrix)  
library(tidyr)  
library(stringr)  
library(stringi)  
library(ggplot2)  
library(lubridate)  
library(sqldf)

### Importing and cleaning SMS messages

For the sake of privacy, I did a little cleanup with a separate script to remove names from the “Received.from.Sent.to” column of the messages downloaded from our phone conversation. Now instead of “Received from (her name)” and “Sent to (her name)”, the values are “Her” and “Me”.

sms <- read.csv("SMS\_Msgs.csv", stringsAsFactors = FALSE)  
str(sms)

## 'data.frame': 2258 obs. of 10 variables:  
## $ Message.ID : int 9545 9076 8967 8592 8383 8382 8357 8356 8355 8354 ...  
## $ Received.from.Sent.to: chr "Her" "Her" "Her" "Her" ...  
## $ Received.Sent.on : chr " on Thu Jun 29 - 2017 08:37 PM" " on Sun Mar 26 - 2017 04:34 PM" " on Fri Feb 17 - 2017 05:40 PM" " on Sat Dec 03 - 2016 05:03 PM" ...  
## $ Message : chr "Message : Final stretch! \_\xd9\xf7\xf7" "Message : Hello!" "Message : Free in 15" "Message : Headed back to the apartment come over whenever you'd like!" ...  
## $ X : chr "Inbox" "Inbox" "Inbox" "Inbox" ...  
## $ X.1 : chr "" "" "" "" ...  
## $ X.2 : chr "" "" "" "" ...  
## $ X.3 : chr "" "" "" "" ...  
## $ X.4 : chr "" "" "" "" ...  
## $ X.5 : chr "" "" "" "" ...

The formatting here is very messy. Some of the messages are split across multiple columns, and the column furthest to the right includes the words “Inbox” and “Sent”, but in some cases, those words are shifted one column to the left and seemingly wrapped to the message. Let’s start by cleaning this up and collapsing the columns to combine the messages.

sms$Message.ID <- as.numeric(sms$Message.ID)  
  
# remove rows where Message.ID is NA  
sms <- sms[complete.cases(sms), ]  
  
# remove first part of 'Received.Sent.on' column "on Tue..."  
sms$Received.Sent.on <- stri\_sub(sms$Received.Sent.on, 8)  
  
# convert "Received.Sent.on" column to date/time using lubridate  
sms$Received.Sent.on <- mdy\_hm(sms$Received.Sent.on)  
  
# remove first 9 characters in every row in Message column "Message: "  
sms$Message <- stri\_sub(sms$Message, 10)  
  
# combine data from Message to X.5, which were oddly separated strings  
sms$Message <- paste(sms$Message, sms$X, sms$X.1, sms$X.2, sms$X.3, sms$X.4, sms$X.5, sep = " ")  
  
# subset sms to remove X:X.5  
sms <- subset(sms[,1:4])  
  
# remove 'Inbox' and 'Sent' from Message: they were inserted when columns were pasted together  
sms$Message <- sapply(str\_replace\_all(sms$Message, "Inbox", " "), function(x) x)  
sms$Message <- sapply(str\_replace\_all(sms$Message, "Sent", " "), function(x) x)  
  
# remove missing symbols and underscores  
sms$Message <- sapply(str\_replace\_all(sms$Message, "�", " "), function(x) x)  
sms$Message <- sapply(str\_replace\_all(sms$Message, "\_", " "), function(x) x)  
  
# create "Source" column where every row is "sms". This will be used to identify message  
# source when the SMS and Facebook data are combined  
sms$Src <- as.character("sms")  
  
# subset df to remove MessageID column  
sms <- subset(sms[,2:5])  
  
# rename cols  
names(sms)[1] <- "User"  
names(sms)[2] <- "Date"  
  
# reorder cols  
sms <- sms %>% select(Date, Src, User, Message)

glimpse(sms)

## Observations: 2,258  
## Variables: 4  
## $ Date <dttm> 2017-06-29 20:37:00, 2017-03-26 16:34:00, 2017-02-17 17…  
## $ Src <chr> "sms", "sms", "sms", "sms", "sms", "sms", "sms", "sms", …  
## $ User <chr> "Her", "Her", "Her", "Her", "Her", "Her", "Her", "Me", "…  
## $ Message <chr> " Final stretch! ", " Hello! ", " Free …

Great! Now the SMS messages are formatted into 4 columns: Date, Src, User, and Message

### Importing and cleaning Facebook messages

I previously converted the names to “Her” and “Me” in a separate script for this data as well.

chats <- read.csv("fb\_msgs.csv", stringsAsFactors = FALSE)  
glimpse(chats)

## Observations: 11,803  
## Variables: 4  
## $ Date <chr> "4/7/16 21:27", "4/7/16 21:38", "4/7/16 21:39", "4/7…  
## $ UserID <dbl> 1.000000e+14, 4.404505e+06, 1.000000e+14, 1.000000e+…  
## $ UserName <chr> "Me", "Her", "Me", "Me", "Her", "Me", "Me", "Her", "…  
## $ MessageBody <chr> ":p", "Ah!! Thank you :-)", "Just going through the …

# split names into first and last name, keep only first name  
clnNames <- sapply(strsplit(chats$UserName, split = " "), function(x) x[1])  
chats$UserName <- clnNames  
  
# format the date column with lubridate: year, month, day, hour, minute  
chats$Date <- mdy\_hm(chats$Date)  
  
# create new column where every row is "fb" to identify fb/sms msgs when data is joined  
chats$Src <- as.character("fb")  
  
# rename cols  
names(chats)[3] <- "User"  
names(chats)[4] <- "Message"  
  
# reorder mdy to ymd to match sms df  
chats$Month <- month(chats$Date)  
chats$Day <- day(chats$Date)  
chats$Year <- year(chats$Date)  
chats$Hour <- hour(chats$Date)  
chats$Minute <- minute(chats$Date)  
  
chats$Date <- paste(chats$Year, chats$Month, chats$Day, chats$Hour, chats$Minute, separate = "-")  
chats$Date <- ymd\_hm(chats$Date)  
  
# re-order and subset cols  
chats <- chats %>% select(Date, Src, User, Message)

### Combining SMS and Facebook messages

# create master df with sms and fb messages  
msgs <- bind\_rows(sms, chats)  
  
# order all msgs chronologically from oldest to newest  
msgs <- msgs %>% arrange(Date)  
  
# make Src and User into factors  
msgs$Src <- as.factor(msgs$Src)  
msgs$User <- as.factor(msgs$User)  
  
# create indexed column of all row nums  
msgs$MsgID <- seq.int(nrow(msgs))  
  
# create separate year, month, and day columns   
msgs$Year <- year(msgs$Date)  
msgs$Month <- month(msgs$Date)  
msgs$Day <- day(msgs$Date)

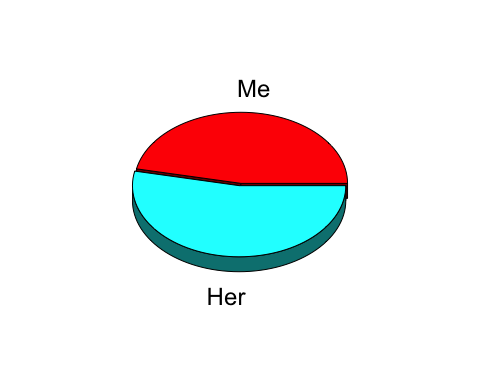
#### Dimensions of combined datasets:

dim(msgs)

## [1] 14061 8

Now that we have our master data frame, let’s see who the more frequent sender is:

# separate messages into two separate data frames, one for each person  
a\_msgs <- msgs %>% filter(User == "Me")   
s\_msgs <- msgs %>% filter(User == "Her")   
  
a\_msgs\_length <- length(a\_msgs$User)  
s\_msgs\_length <- length(s\_msgs$User)  
  
pie1\_msgs <- c(a\_msgs\_length, s\_msgs\_length)  
pie1\_lbls <- c("Me", "Her")  
pie3D(pie1\_msgs,   
 labels = pie1\_lbls,   
 height = 0.07,   
 theta = pi/3,   
 explode = 0.08,   
 shade = 0.5)



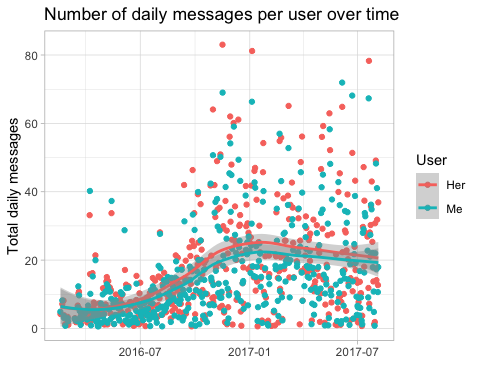
Looks like she sent more messages than I did. Let’s see how many:

msgs\_num <- msgs %>% group\_by(User, MsgID) %>% summarise(number = n())  
ggplot(msgs\_num, aes(User, number, fill = User)) +  
 geom\_bar(stat = "identity", alpha = I(0.7)) +  
 ggtitle("Number of messages sent by each person") +  
 xlab("Names") + ylab("Number of messages") +   
 theme\_bw()



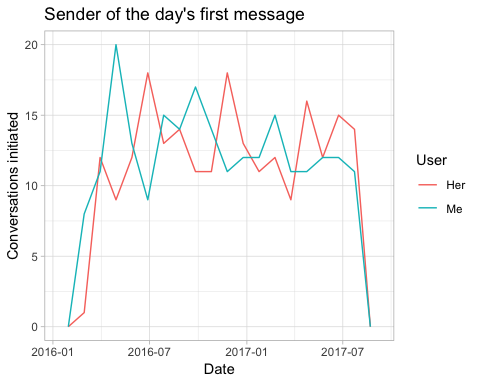
And now let’s see a daily breakdown of message frequency per sender:

# create a new date column that removes hour and minute data  
msgs$YMD <- paste(msgs$Year, msgs$Month, msgs$Day, sep = "-")  
msgs$YMD <- ymd(msgs$YMD)  
msgs\_grouped <- msgs %>% group\_by(User, YMD) %>% summarise(nums = n())  
  
ggplot(msgs\_grouped, aes(YMD, nums, color = User)) +  
 geom\_jitter() +  
 geom\_smooth() +  
 ggtitle("Number of daily messages per user over time") +  
 xlab("") +   
 ylab("Total daily messages") +  
 theme\_light()



Now we’ll take a look to see who initiated the conversation every day

first\_sender <- msgs %>%   
 select(YMD, User) %>%   
 group\_by(YMD) %>%   
 arrange(YMD) %>%   
 mutate(count = row\_number(),   
 count == 1 & User == "Me") %>%   
 filter(count == 1)  
  
ggplot(first\_sender, aes(YMD, color = User)) +  
 geom\_freqpoly(binwidth = 30) +  
 ggtitle("Sender of the day's first message") +  
 xlab("Date") + ylab("Conversations initiated") +  
 theme\_light()

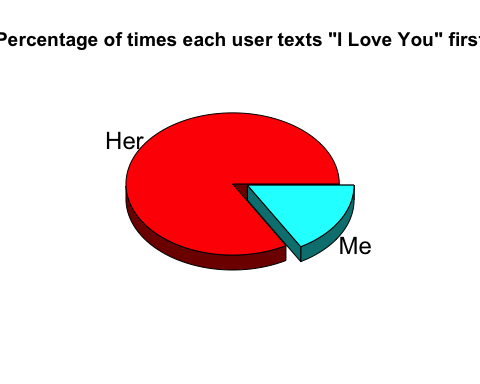


It looks like I was the pursuer at the beginning of the relationship and then initiated fewer conversations as time went on. However, of the 450 days we texted, I initiated 228 of them. Almost exactly half.

Now let’s take a look at the percentage of times each user texts “I love you” first in our conversations.

I’ll be using the sqldf package for this, which allows you to use SQL syntax in your R code:

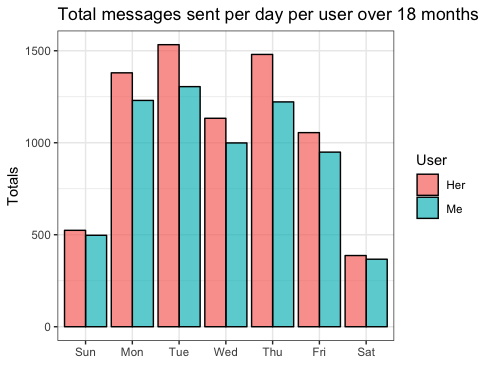
love\_selector <- sqldf("SELECT \* FROM msgs  
 WHERE Message LIKE '%ove Yo%'  
 OR Message LIKE '%ove yo%'  
 OR Message LIKE '%uv you%'  
 OR Message LIKE '%uv u%'  
 OR Message LIKE '%wuv you'  
 OR Message LIKE '%wuv u%' ")  
  
love\_false\_pos\_elim <- sqldf("SELECT \* FROM love\_selector  
 WHERE NOT Message LIKE '%who love%'   
 AND NOT Message LIKE '%they love%'  
 AND NOT Message LIKE '%we love%' ")  
  
love\_delay <- love\_false\_pos\_elim %>%   
 group\_by(YMD) %>%   
 arrange(YMD) %>%   
 mutate(DailyIndex = row\_number())  
  
love\_delay$DailyIndex <- as.integer(love\_delay$DailyIndex)  
  
love\_delay <- love\_delay %>% separate(Date,   
 into = c('mdy', 'hm'),   
 sep = 11)  
  
love\_delay <- love\_delay %>%   
 group\_by(mdy) %>%   
 arrange(mdy) %>%   
 mutate(DailyMsgIndex = n()) %>%  
 filter(DailyMsgIndex >= 2) %>%   
 mutate(lag = row\_number())  
  
love\_delay1 <- love\_delay %>%   
 filter(lag == 1)  
love\_delay2 <- love\_delay %>%   
 filter(lag == 2)  
  
love\_delay3 <- data.frame(love\_delay1, love\_delay2)  
  
love\_delay3$mdy <- ymd(love\_delay3$mdy)  
love\_delay3$hm <- hms(love\_delay3$hm)  
love\_delay3$mdy.1 <- ymd(love\_delay3$mdy.1)  
love\_delay3$hm.1 <- hms(love\_delay3$hm.1)  
  
love\_delay3$lag\_elapsed <- love\_delay3$hm.1 - love\_delay3$hm  
  
love\_delay\_totals <- love\_delay3 %>%   
 group\_by(User) %>%   
 summarise(UserTotal = n()) %>%   
 arrange(desc(UserTotal))  
  
love\_delay\_totals <- data.frame(love\_delay\_totals)  
  
love\_delay\_totals$Perc <- love\_delay\_totals$UserTotal / sum(love\_delay\_totals$UserTotal) \* 100  
  
pie1\_data <- love\_delay\_totals$UserTotal  
pie1\_lbls <- love\_delay\_totals$User  
  
pie3D(pie1\_data, labels = pie1\_lbls,   
 height = 0.07,   
 theta = pi/3,   
 explode = 0.08,   
 shade = 0.5,   
 main = "Percentage of times each user texts \"I Love You\" first")



Yikes! It looks like although we both initiated the same amount of conversations, she said “I love you” first 86% of the time!

Now let’s look into our texting habits. We’ll see how many texts we each sent per the day of the week, over the course of 18 months.

msgs\_group <- msgs %>% group\_by(User)  
  
msgs\_group <- msgs\_group %>%   
 group\_by(User, YMD) %>%   
 mutate(DayIndex = row\_number())  
  
msgs\_group$Day <- wday(msgs\_group$Date, label=TRUE)  
  
grouped\_dailyMsgs <- msgs\_group %>%   
 group\_by(Day, User) %>%   
 summarise(Totals = n())  
  
ggplot(grouped\_dailyMsgs, aes(x = Day,   
 y = Totals,   
 fill = User)) +   
 geom\_bar(position = "dodge",   
 stat = "identity",   
 alpha = I(0.7),   
 col = I("black")) +   
 ggtitle("Total messages sent per day per user over 18 months") +  
 xlab("") +  
 theme\_bw()



Exactly as I suspected: Mondays, Tuesdays, and Thursdays, which were the days that we didn’t usually see each other, we texted more. On Wednesdays and Fridays, we usually saw each other in the evenings, so although we texted throughout the day, the overall number of texts was lower on those days. On the weekends, we were usually together most of the time, so that is why there are so few messages. Everything checks out!