Best Buds

Data Manipulation

```
#Library setup
library(rjson)
library (ggplot2)
library(dplyr)
library(tidyr)
library(tm)

"Start Date: 02/26/2016"
```

```
## [1] "Start Date: 02/26/2016"
```

```
#File Read
result = fromJSON(file =  "/Users/omachowda/Desktop/Projects/Messenger Analysis/message
s/inbox/Highforce_tbMFZx2JWg/message.json")
#Data Head
as.data.frame(result$messages[1])
```

```
##      sender_name timestamp_ms                  content    type
## 1 Shanay Thakkar 1.546214e+12 Light shows and massages Generic
```

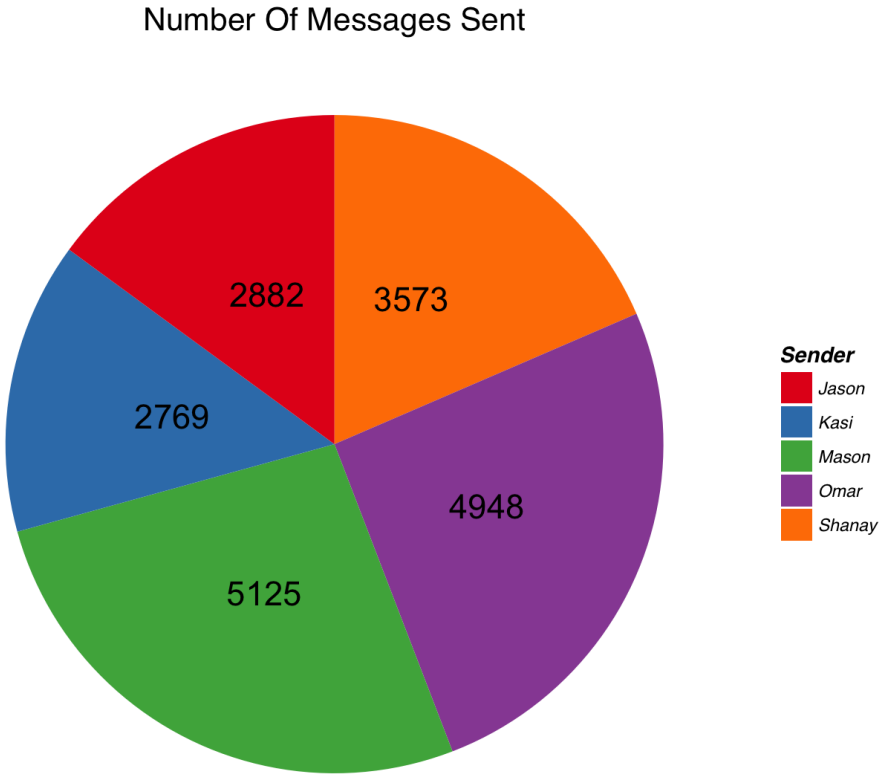
```
#O(n^2) methodology for building the data frame

mydf = as.data.frame(result$messages[1],stringsAsFactors = FALSE)
length(result$messages)
```

```
## [1] 19884
```

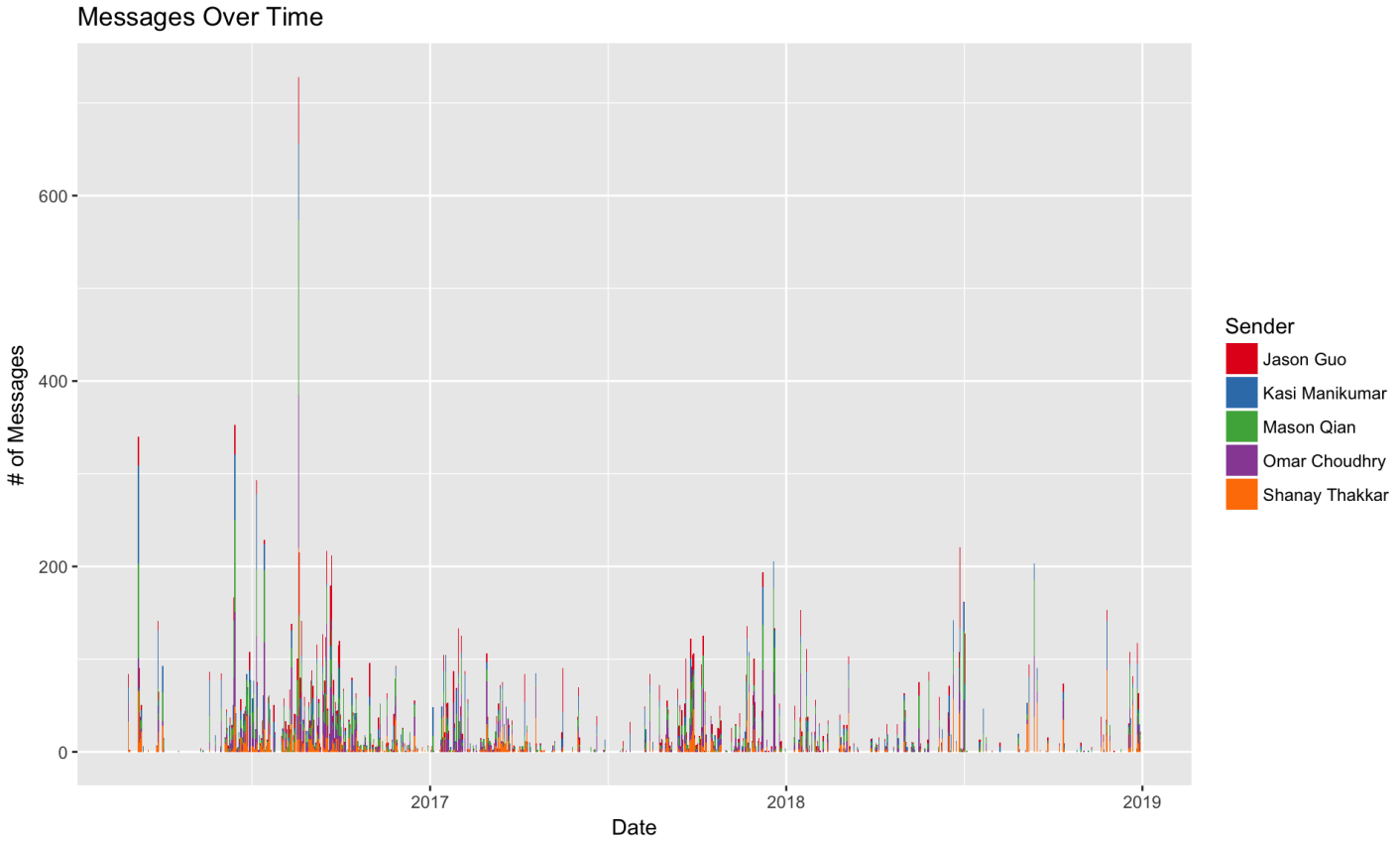
```
#Build Data Frame - get rid of messages with empty content
#O(n) methodology

mydf <- data.frame(sender_name = rep(NA,length(result$messages)),
                   timestamp_ms = rep(NA,length(result$messages)),
                   content = rep(NA,length(result$messages)),
                   type = rep(NA,length(result$messages)),
                   stringsAsFactors = FALSE)

for (i in 1:length(result$messages)){
  if(as.data.frame(result$messages[i])$type == "Generic"
     & is.null(as.data.frame(result$messages[i])$reactions.reaction)
     & is.null(as.data.frame(result$messages[i])$photos.uri)
     & is.null(as.data.frame(result$messages[i])$videos.uri)
     & is.null(as.data.frame(result$messages[i])$files.uri)
     & is.null(as.data.frame(result$messages[i])$uri)
     & is.null(as.data.frame(result$messages[i])$audio_files.uri)
     & !is.null(as.data.frame(result$messages[i])$content)){
    mydf[i,] = as.data.frame(result$messages[i],stringsAsFactors = FALSE)
  }
}

mydf = mydf %>% drop_na()

#Convert To Date Format
# @ms: a numeric vector of milliseconds (big integers of 13 digits)
# @t0: a string of the format "yyyy-mm-dd", specifying the date that
#      corresponds to 0 millisecond
# @timezone: a string specifying a timezone that can be recognized by R
# return: a POSIXct vector representing calendar dates and times

ms_to_date = function(ms, t0="1970-01-01", timezone){
  sec = ms / 1000
  as.POSIXct(sec, origin=t0, tz=timezone)
}

#Convert ms to DateTime
date_in_ms = mydf$timestamp_ms
mydf$datetime = ms_to_date(date_in_ms, timezone="America/Chicago")
mydf$timestamp_ms = NULL

#Add Date Parameter Columns
mydf$weekdays = factor(weekdays(mydf$datetime),levels = c('Monday', 'Tuesday', 'Wednesda
y', 'Thursday',
                                                          'Friday', 'Saturday', 'Sunday'
))
mydf$months = months((mydf$datetime))
mydf$date = as.Date(mydf$datetime)
mydf$hours = format(mydf$datetime, '%H')
mydf$monthyear = format(mydf$datetime, '%m/%Y')
mydf$monthyeartest = format(mydf$datetime, '%y/%m')
mydf$monthday = format(mydf$datetime, '%m/%d/%Y')
```

Plots

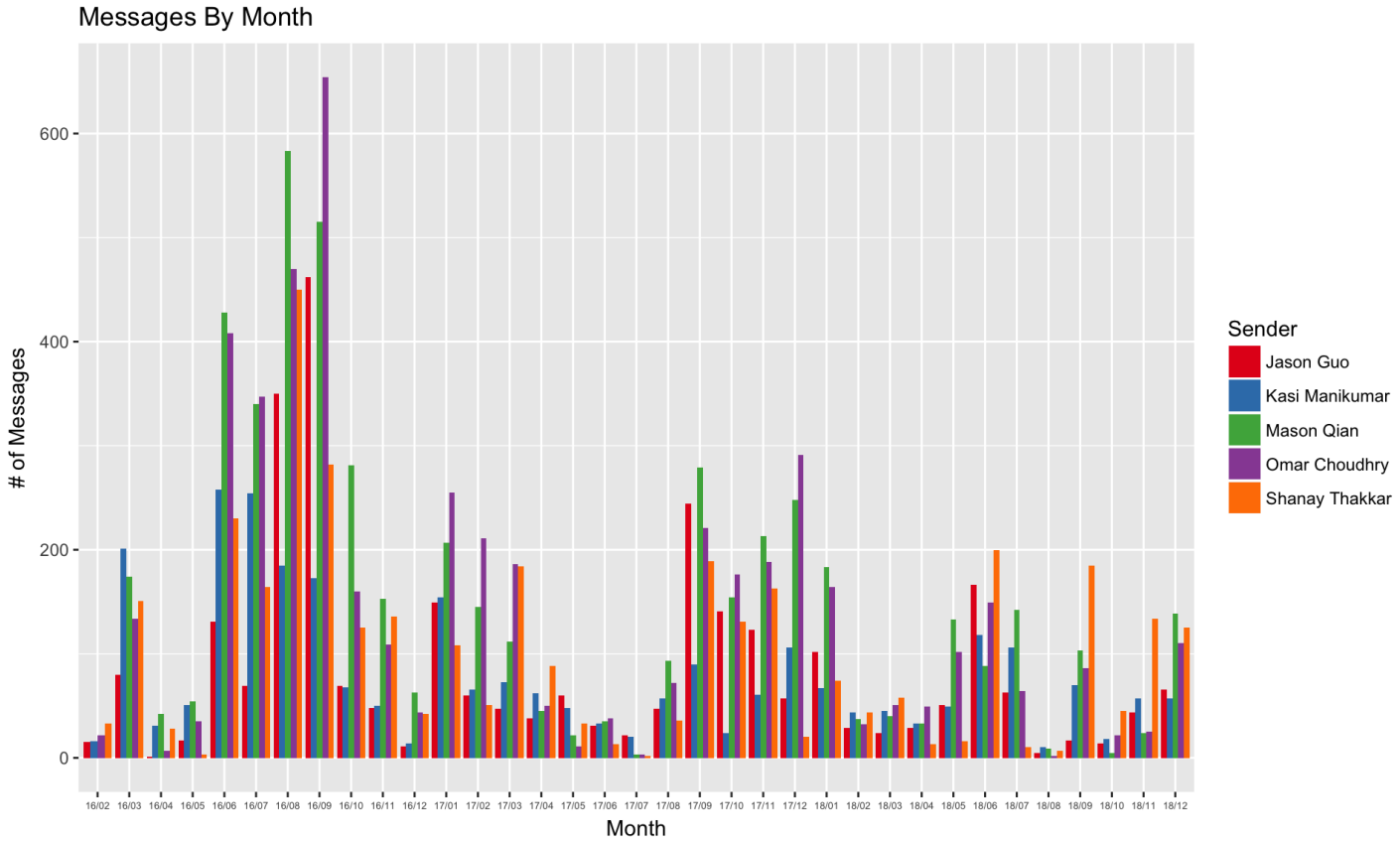
```
#Plot Themes

minimalplottheme <- theme_void() +
  theme(plot.title = element_text(family = 'Helvetica',
                                  colour = 'black',
                                  size = (16)),
        legend.title = element_text(family = 'Helvetica',
                                    colour = 'black',
                                    face = 'bold.italic'),
        legend.text = element_text(family = 'Helvetica',
                                   colour = 'black',
                                   face = 'italic'),
        axis.title = element_blank(),
        axis.text = element_blank())

#Pie Chart
tbl = (table(mydf$sender_name))

df <- data.frame(
  sender = factor(c('Jason', 'Kasi', 'Mason', 'Omar', 'Shanay'),
                  levels = c('Jason', 'Kasi','Mason','Omar','Shanay')),
  value = c(as.numeric(as.character(tbl['Jason Guo'])),
            as.numeric(as.character(tbl['Kasi Manikumar'])),
            as.numeric(as.character(tbl['Mason Qian'])),
            as.numeric(as.character(tbl['Omar Choudhry'])),
            as.numeric(as.character(tbl['Shanay Thakkar']))))

bp <-
  ggplot(df, aes(x='', y=value, fill=sender)) +
  geom_bar(width = 1, stat = 'identity')
bp +
  ggtitle('Number Of Messages Sent') +
  coord_polar('y',start=0) +
  minimalplottheme +
  geom_text(aes(y = rev(c(1500,6000,11000,15000,18000)),
                label = value),
            size=6,
            colour='black') +
  scale_fill_brewer(palette='Set1')+
  labs(fill = "Sender")
```



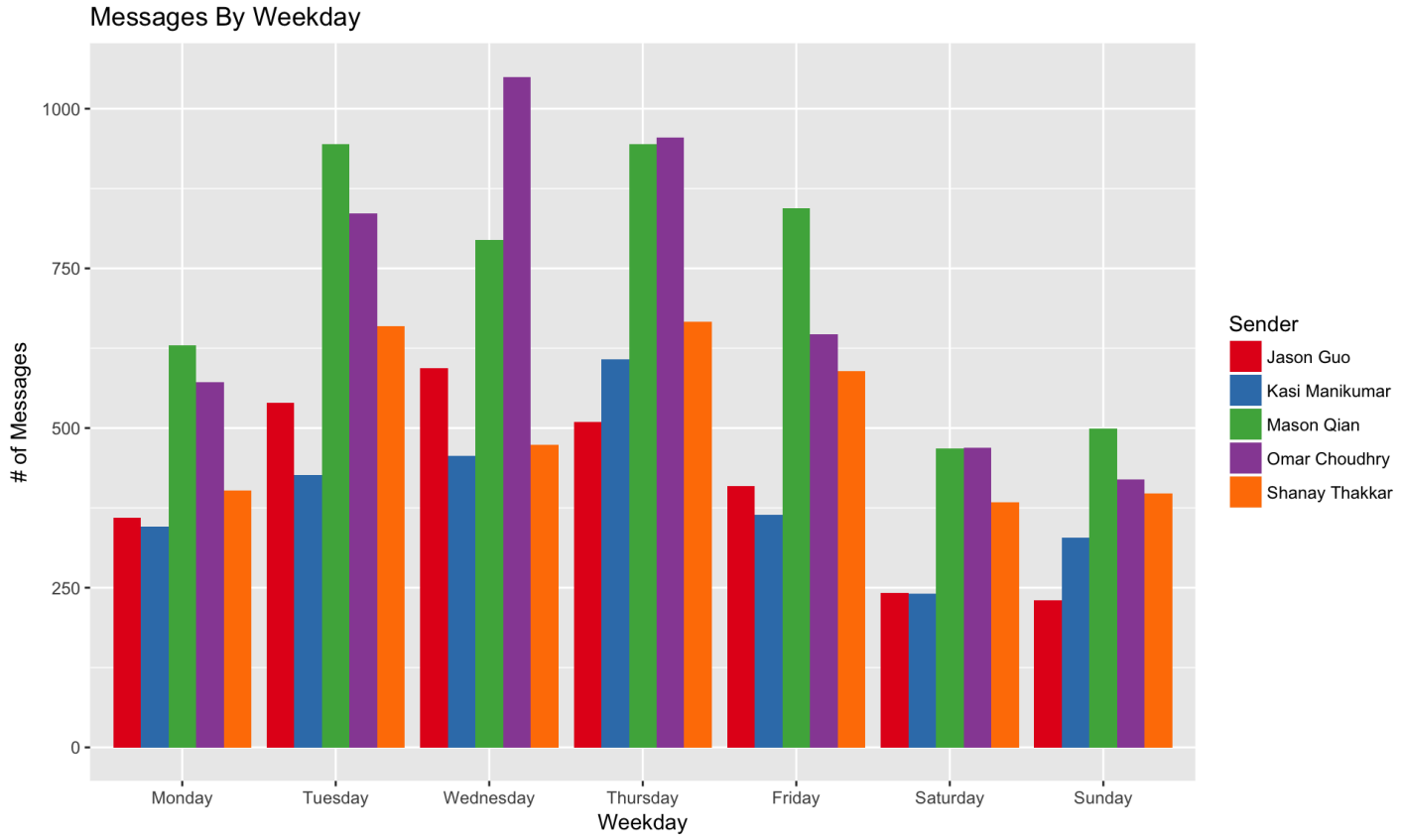

```
#Messages Over Time
ggplot(mydf,
       aes(x = date, fill = sender_name)) +
  ggtitle('Messages Over Time') +
  geom_histogram(binwidth = 1) +
  scale_fill_brewer(palette='Set1') +
  labs(fill = "Sender",x = "Date",y = "# of Messages")
```



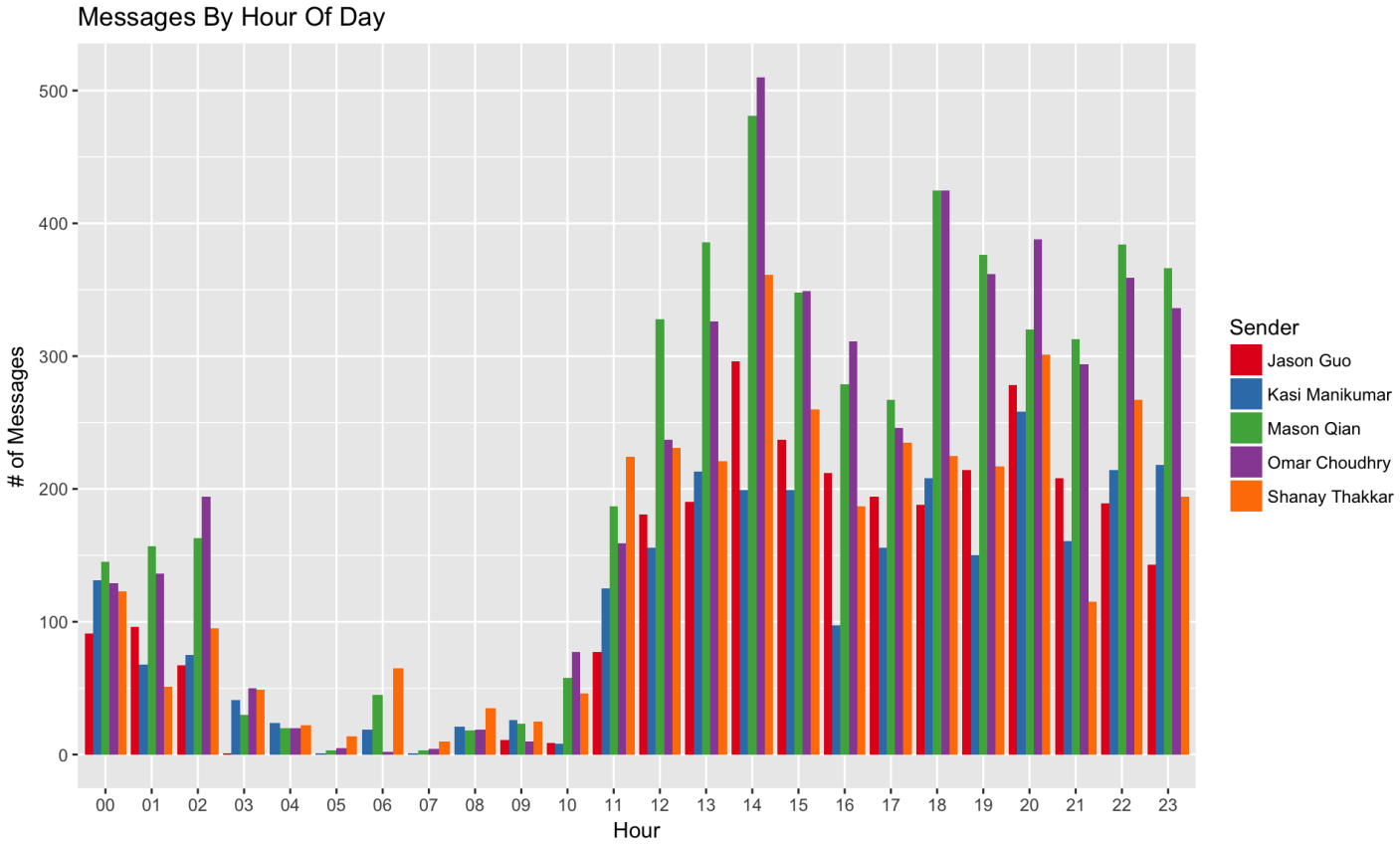

```
#Messages Over Time By Month
ggplot(mydf,
       aes(x = monthyeartest, fill = sender_name)) +
  ggtitle('Messages By Month') +
  geom_histogram(position = 'dodge', stat = 'count') +
  scale_fill_brewer(palette='Set1')+
  labs(fill = "Sender",x = "Month",y = "# of Messages")+
  theme(axis.text.x = element_text(size=5))
```



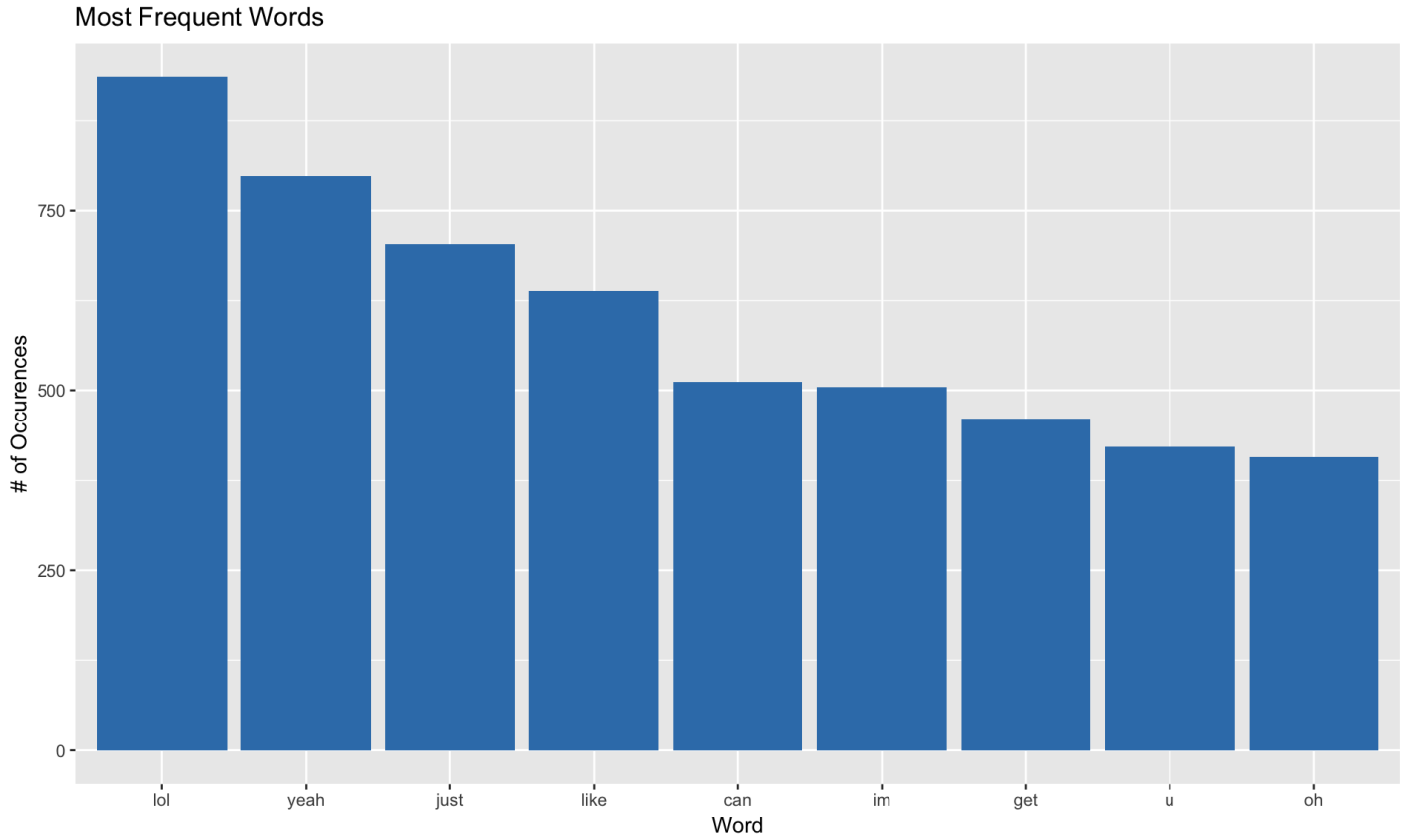

```
#Messages Over Weekday
ggplot(mydf,
       aes(x = weekdays, fill = sender_name)) +
  ggtitle('Messages By Weekday') +
  geom_histogram(binwidth = 7, position = 'dodge', stat = 'count') +
  scale_fill_brewer(palette='Set1')+
  labs(fill = "Sender",x = "Weekday",y = "# of Messages")
```



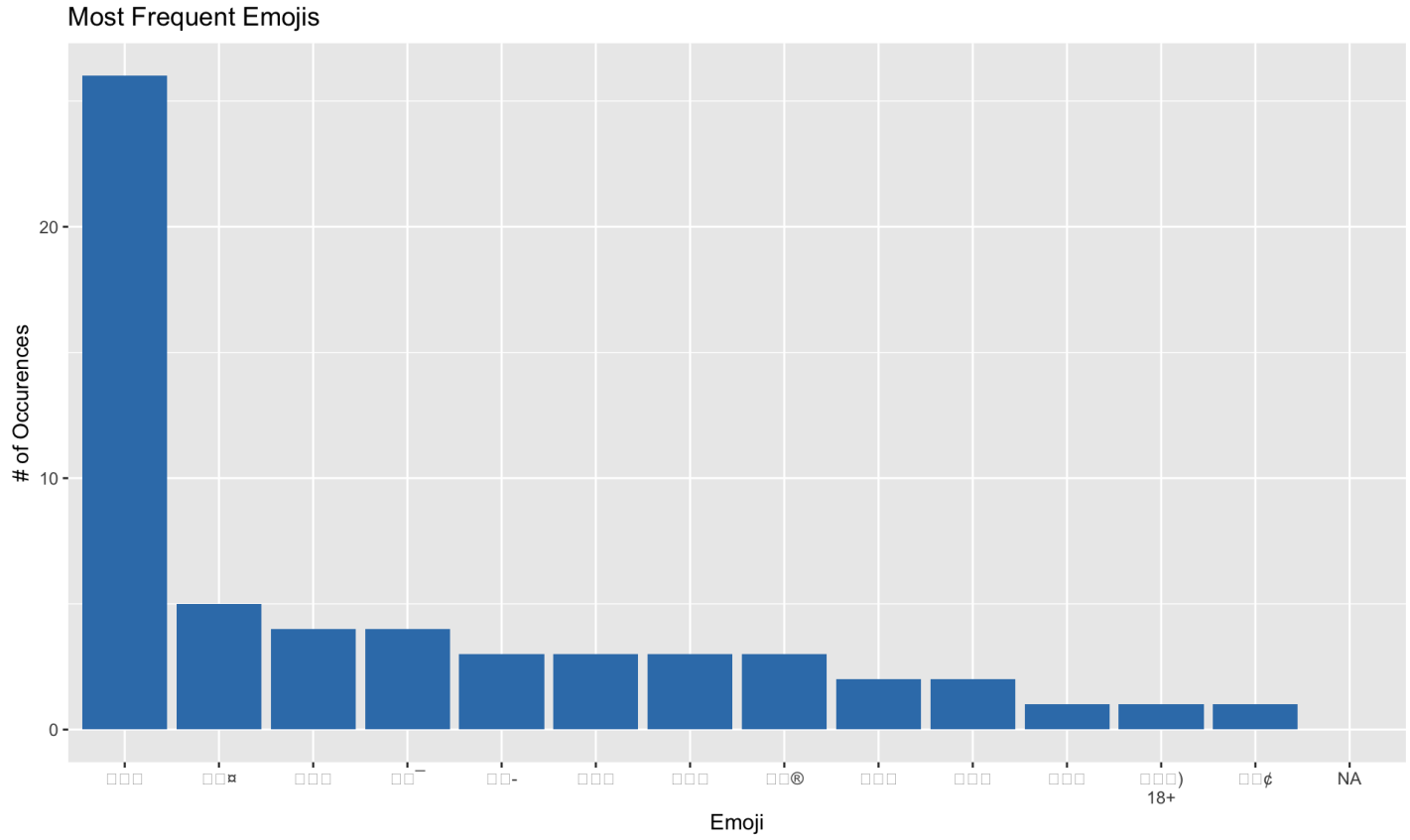

```
#Messages Over Hour Of Day
ggplot(mydf,
       aes(x = hours, fill = sender_name)) +
  ggtitle('Messages By Hour Of Day') +
  geom_histogram(position = 'dodge', stat = 'count') +
  scale_fill_brewer(palette='Set1')+
  labs(fill = "Sender",x = "Hour",y = "# of Messages")
```




```
#Common Words Graph

mytext = mydf$content
y = removeWords(tolower(na.omit(unlist(strsplit(as.character(mytext), " ")))),words = st
opwords("en"))
wordfreq = as.data.frame(sort(table(y), decreasing=T)[2:10])

ggplot(wordfreq, aes(y, Freq)) +
  ggtitle('Most Frequent Words') +
  geom_col(fill=scale_fill_brewer(palette='Set1')$palette(8)[2])+
  labs(x = "Word",y = "# of Occurences")
```




```
#Most Common Emojis

splitted = y[startsWith(x = y,prefix = "\u00f0\u009f\u0098")]
splitted1 = unlist(strsplit(splitted, "ð"))

emojifreq = as.data.frame(sort(table(splitted1), decreasing=T)[2:15])

ggplot(emojifreq, aes(splitted1, Freq)) +
  ggtitle('Most Frequent Emojis') +
  geom_col(fill=scale_fill_brewer(palette='Set1')$palette(8)[2])+
  labs(x = "Emoji",y = "# of Occurences")
```



Interesting Statistics

```
#Number Of Words
length(y)
```

```
## [1] 83692
```

```
#Average Message length
length(y)/length(mytext)
```

```
## [1] 4.337047
```

```
#Most Active Day
active = mydf

options(tibble.print_max = Inf)
activeday = count(mydf, monthday)
activeday[which.max(activeday$n),]
```

```
## # A tibble: 1 x 2
##   monthday       n
##   <chr>      <int>
## 1 08/19/2016   915
```

```
#Messages Per Day
length(result$messages)/length(activeday$monthday)
```

```
## [1] 38.46035
```

```
#Number Of Unique Words
length(sort(table(y), decreasing=T))
```

```
## [1] 9065
```