How I Data Mined My Text Message History

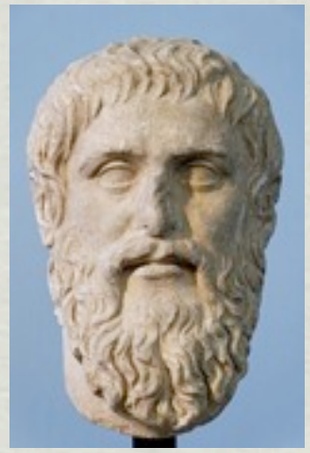
By: Joe Cannatti
Puppy Sound LLC
joe@puppysound.com
@JoeCannatti

How I Data Mined My Text Message History

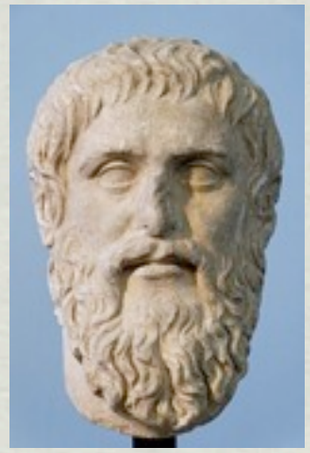
By: Joe Cannatti
Puppy Sound LLC
joe@puppysound.com
@JoeCannatti

Know thyself...

Know thyself...

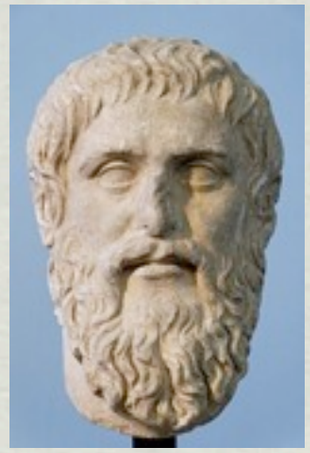


Know thyself...



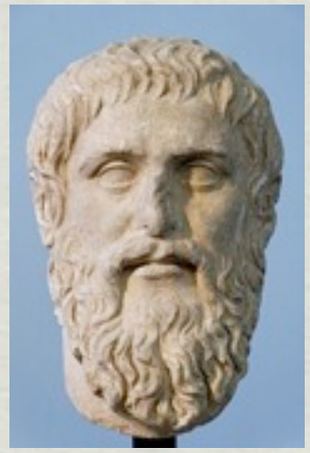
- γνῳθι σεαυτόν

Know thyself...



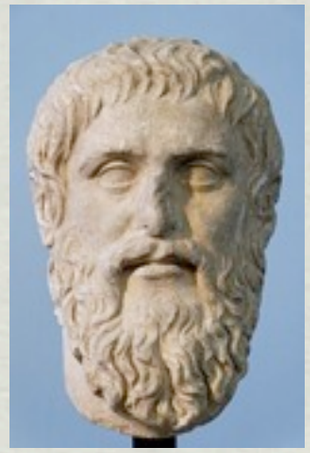
- γνῳθι σεαυτόν
- Temple of Apollo at Delphi

Know thyself...



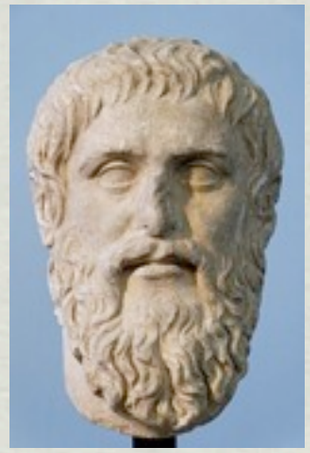
- γνῳθι σεαυτόν
- Temple of Apollo at Delphi
- Plato

Know thyself...



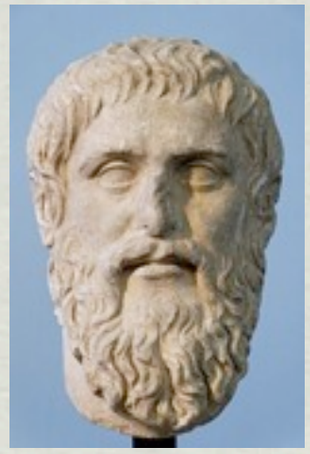
- γνῳθι σεαυτόν
- Temple of Apollo at Delphi
- Plato
- Our memories are poor

Know thyself...



- γνῳθι σεαυτόν
- Temple of Apollo at Delphi
- Plato
- Our memories are poor
- We are not good at summing up our experiences

Know thyself...



- γνῳθι σεαυτόν
- Temple of Apollo at Delphi
- Plato
- Our memories are poor
- We are not good at summing up our experiences
- Enter.....DATA

Text Messages

Text Messages

- Increased use

Text Messages

- Increased use
- Is it a good subset of your overall communication?

Text Messages

- Increased use
- Is it a good subset of your overall communication?
- What can we learn from it?

Text Messages

- Increased use
- Is it a good subset of your overall communication?
- What can we learn from it?
- SCIENCE!!!

Tools Used

Tools Used

- R

Tools Used

- R
- RStudio

Tools Used

- R
- RStudio
- Navicat for Sqlite

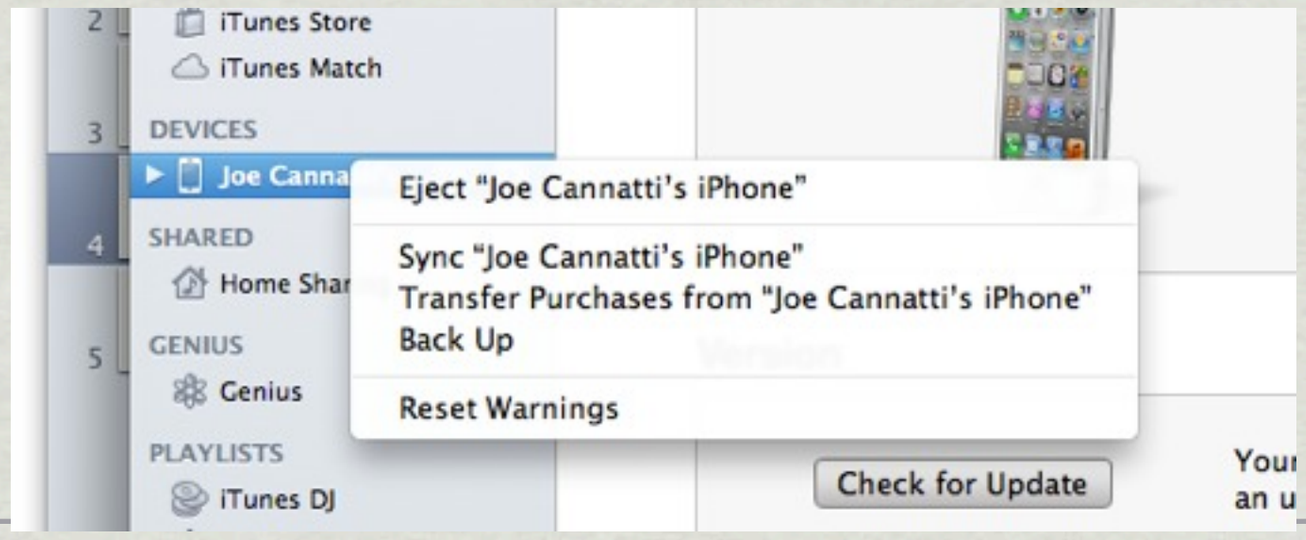
Tools Used

- R
- RStudio
- Navicat for Sqlite
- Bash

So, what are we going to do?

iTunes backup

- Backup your phone to your machine (unencrypted)



Where dem bitz at?

- ~/Library/Application Support/MobileSync/Backup/
- 3d0d7e5fb2ce288813306e4d4636395e047a3d2

Hmmm...what else is in there?

- Bash to the rescue
- find . -name *3d0d7e5fb2ce288813306e4d4636395e047a3d2*
- for i in $(find . -type f); do sqlite3 $i ".databases" &> /dev/null; if [[$? == "0"]]; then echo $i; fi; done
- 190 sqlite DBs

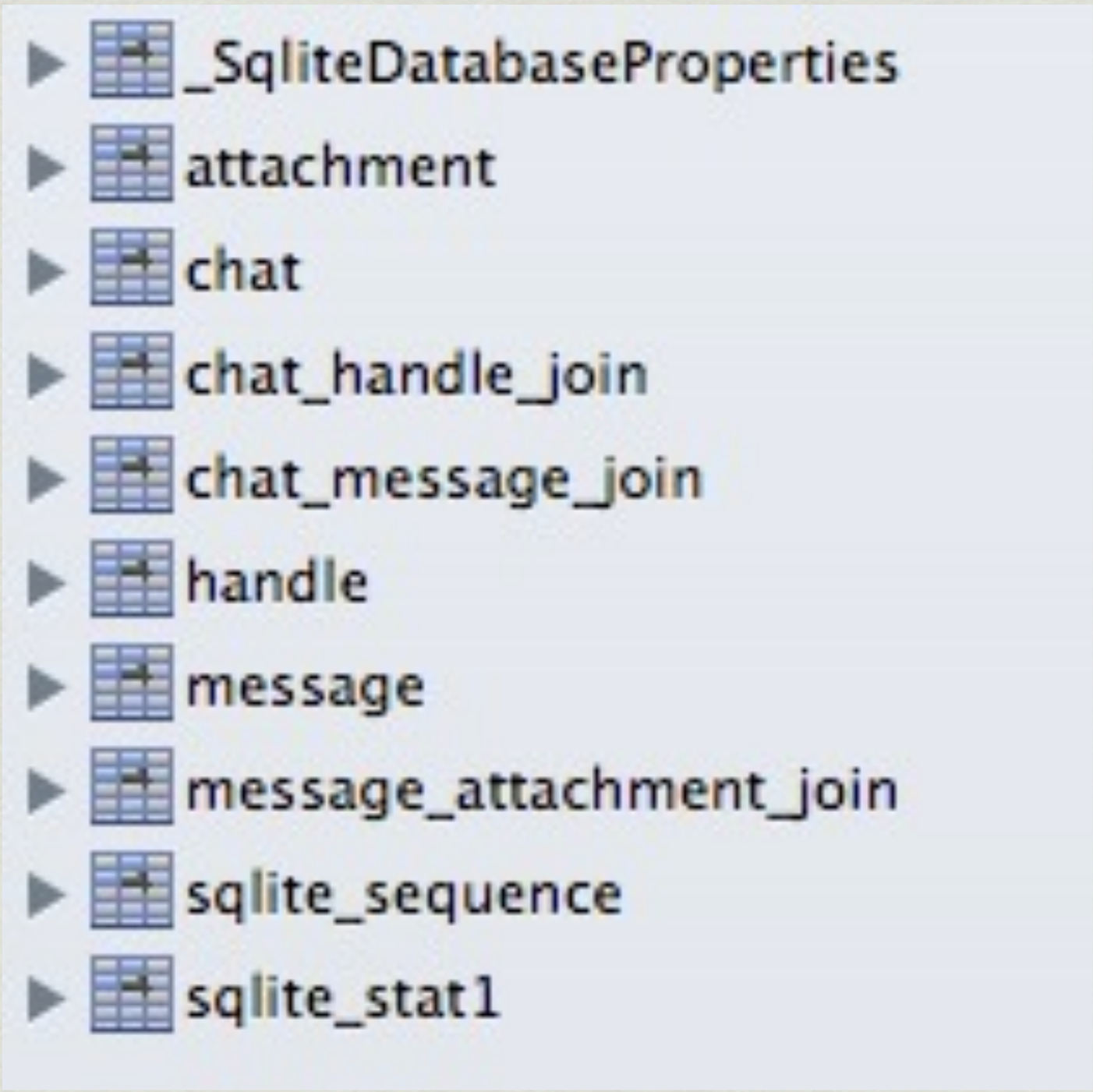
Bash Fu

```
for i in $(find . -type f); do
  sqlite3 $i ".databases" &> /dev/null;
  if [[ $? == "0" ]]; then
    echo $i;
  fi;
done
```

Getting ready to open the project

- Download the RSQLite .tar.gz
- R CMD INSTALL RSQLite-<version>.tar.gz
- install.packages(‘DBI’)

Tables


_SqliteDatabaseProperties
attachment
chat
chat_handle_join
chat_message_join
handle
message
message_attachment_join
sqlite_sequence
sqlite_stat1

Our main view

```
SELECT
    *
FROM
    message m
JOIN chat_message_join j ON j.message_id = m.ROWID
JOIN chat c ON j.chat_id = c.ROWID;
```

Adding Contact data

```
add_to_db <- function(vect,con){
  funct <- function(data){
    rs <- dbSendQuery(con, paste("insert into demographics (num
  }
  sapply(vect, FUN=funct)
}
data <- sort(unique(message_data$chat_identifier))
numbers <- data[substring(data, 1, 1) == "+"]
add_to_db(numbers, con)
```

A note about R

A note about R

Problems with the Language

Problems with the Language

```
> apply(times,1,function(data){print(data$times)})
Error in data$times : $ operator is invalid for atomic vectors
> apply(times,1,function(data){print(data[3])})
```

Problems with the Language

```
> apply(times,1,function(data){print(data$times)})
Error in data$times : $ operator is invalid for atomic vectors
> apply(times,1,function(data){print(data[3])})
```

ANY APL OR J PROGRAMMERS HERE?

My besties

```
 9
10  SELECT
11      count(*), name
12  FROM
13      message m
14  JOIN chat_message_join j ON j.message_id = m.ROWID
15  JOIN chat c ON j.chat_id = c.ROWID
16  JOIN demographics d ON c.chat_identifier =  ('+' || d.number)
17  group by name
18  order by count(*) desc;
```

Result

count(*)	name
1338	Lissa M
795	Corissa Bragg
699	Angie Denicholas
659	Kyllea
553	Erica H
464	Ash Roc
396	Bill Seeholzer
240	[Null]
182	Jon Knapp
177	Dad
166	Shawn Jackson
163	Shultz
156	Mark Mraz
135	Adriana
118	Kevin Solorio
85	Bob Lib
82	Jessica Tag

← **MY BABY MOMA**

Result

count(*)	name
1338	Lissa M
795	Corissa Bragg
699	Angie Denicholas
659	Kyllea
553	Erica H
464	Ash Roc
396	Bill Seeholzer
240	[Null]
182	Jon Knapp
177	Dad
166	Shawn Jackson
163	Shultz
156	Mark Mraz
135	Adriana
118	Kevin Solorio
85	Bob Lib
82	Jessica Tag

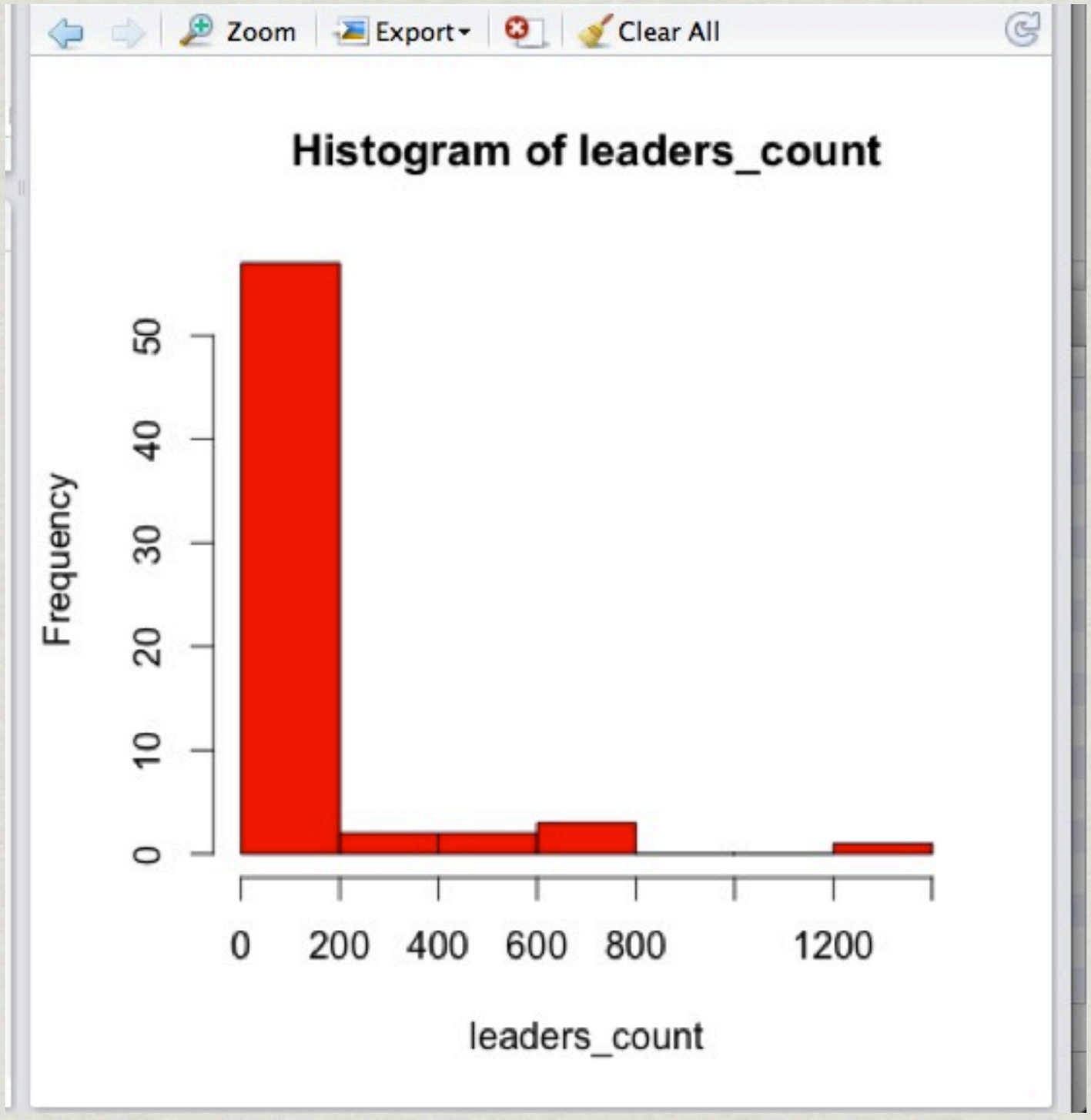
MY BABY MOMA

Result

count(*)	name
1338	Lissa M
795	Corissa Bragg
699	Angie Denicholas
659	Kyllea
553	Erica H
464	Ash Roc
396	Bill Seeholzer
240	[Null]
182	Jon Knapp
177	Dad
166	Shawn Jackson
163	Shultz
156	Mark Mraz
135	Adriana
118	Kevin Solorio
85	Bob Lib
82	Jessica Tag

MARK G'S GIRL

MY BABY MOMA


Zoom
Export
Clear All
Histogram of leaders_count
Frequency
0
10
20
30
40
50
0
200
400
600
800
1200
leaders_count

```
hist(leaders_count, col=c('red'))
hist(leaders_count, breaks=10, col=c('red'))
hist(leaders_count, breaks="FD", col=c('red'))
```

```
hist(leaders_count, col=c('red'))
hist(leaders_count, breaks=10, col=c('red'))
hist(leaders_count, breaks="FD", col=c('red'))
```

HISTORGRAMS CAN BE PRETTY WACK

```
hist(leaders_count, col=c('red'))
hist(leaders_count, breaks=10, col=c('red'))
hist(leaders_count, breaks="FD", col=c('red'))
```

HISTORGRAMS CAN BE PRETTY WACK

THE FREEDMAN-DIACOIS RULE

```r
hist(leaders_count, col=c('red'))
hist(leaders_count, breaks=10, col=c('red'))
hist(leaders_count, breaks="FD", col=c('red'))
```

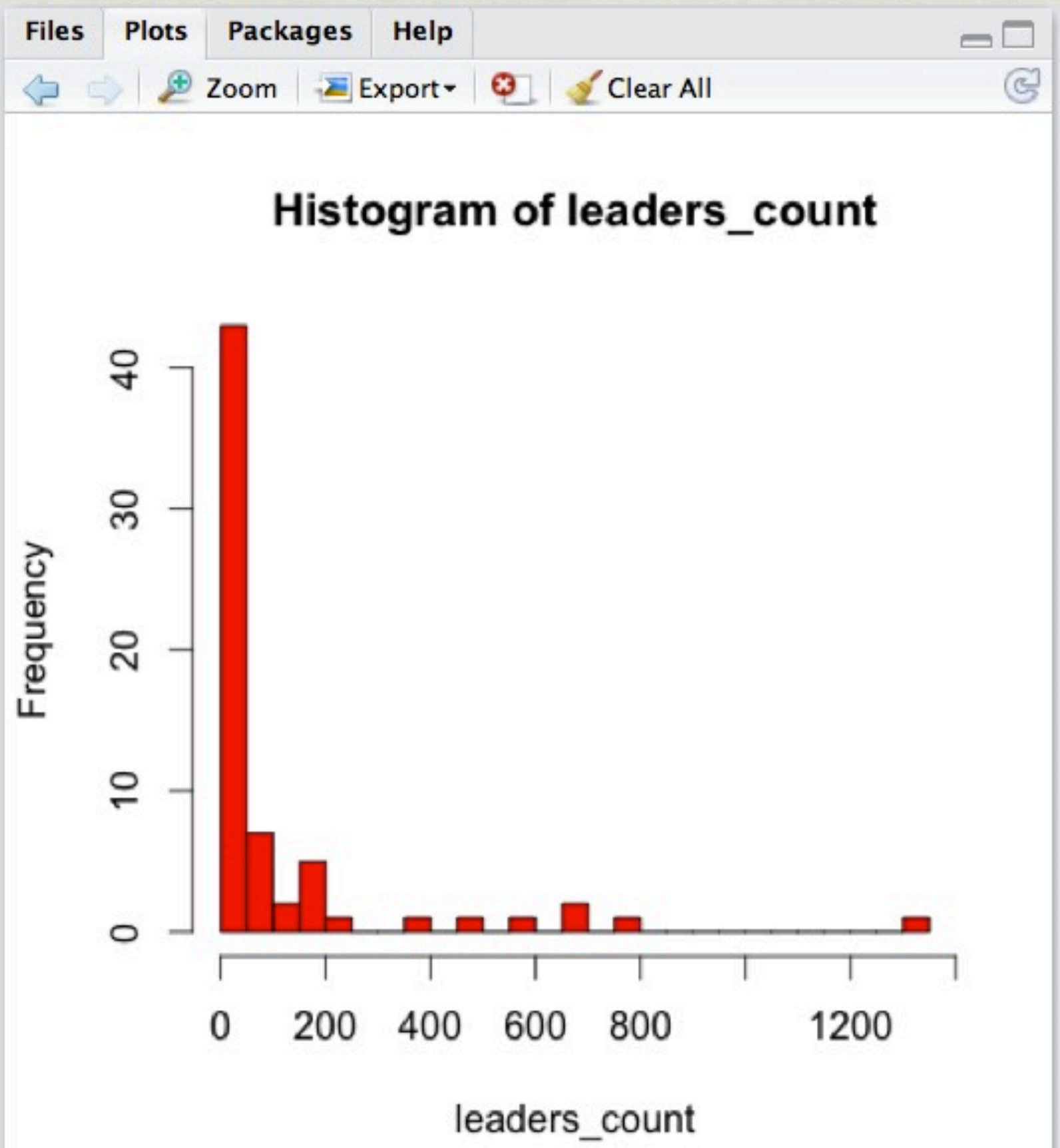
HISTORGRAMS CAN BE PRETTY WACK

THE FREEDMAN-DIACOIS RULE

$$h = 2 * \mathrm{IQR} * n^{-1/3}$$

Files | Plots | Packages | Help

Zoom | Export | Clear All



Demo Frequency

Our main Data.Frame

```
load_message_data <- function(con){
  rs <- dbSendQuery(con, "select * from message m
                          join chat_message_join j
                          on j.message_id = m.ROWID
                          join chat c
                          on j.chat_id = c.ROWID;")
  data <- fetch(rs, n=Inf)
  return(data)
}
```

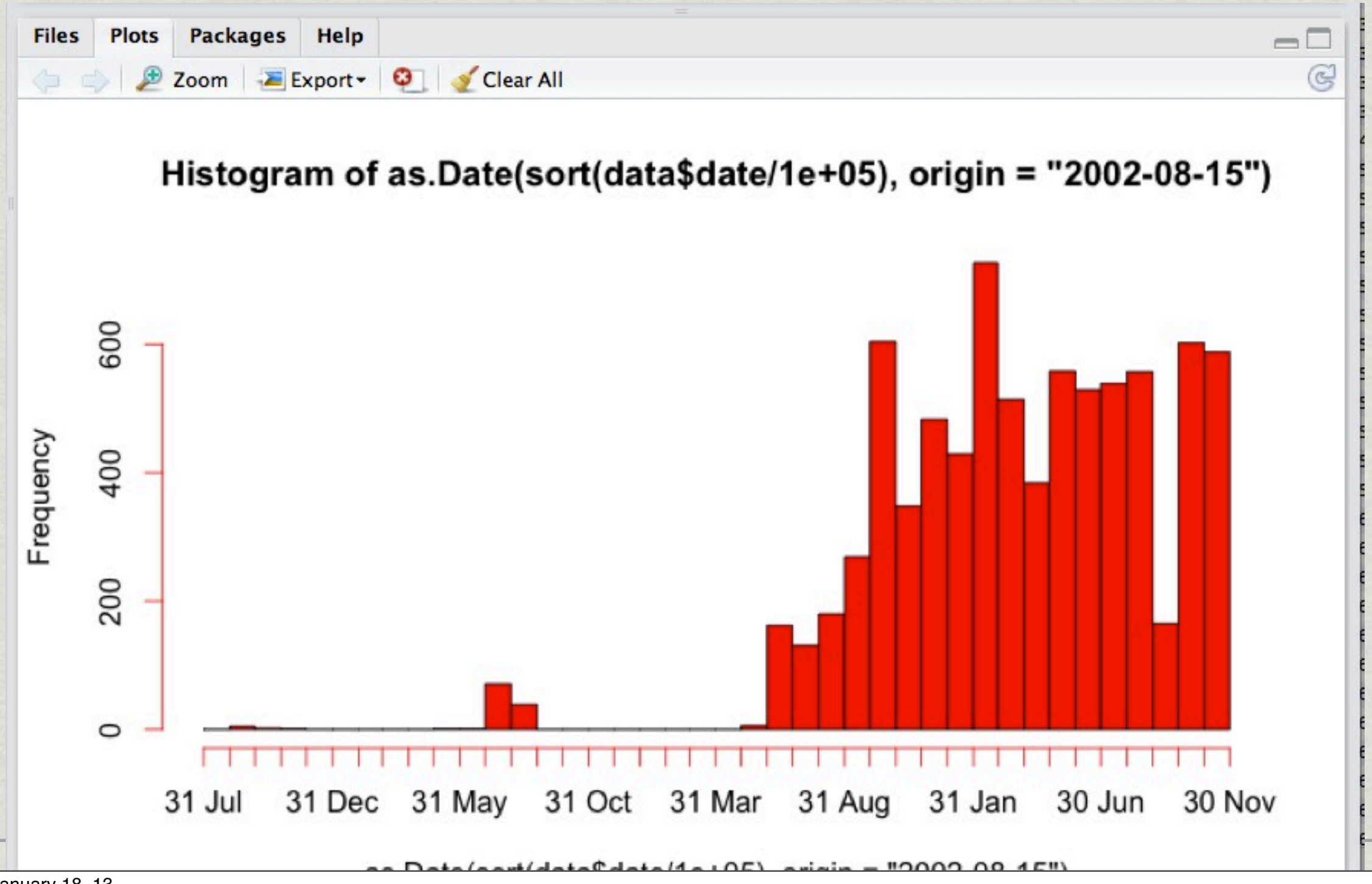
colnames(message_data)

```
> colnames(message_data)
 [1] "ROWID"              "guid"                 "text"
 [4] "replace"            "service_center"       "handle_id"
 [7] "subject"            "country"              "attributedBody"
[10] "version"            "type"                 "service"
[13] "account"            "account_guid"         "error"
[16] "date"               "date_read"            "date_delivered"
[19] "is_delivered"       "is_finished"          "is_emote"
[22] "is_from_me"         "is_empty"             "is_delayed"
[25] "is_auto_reply"      "is_prepared"          "is_read"
[28] "is_system_message"  "is_sent"              "has_dd_results"
[31] "is_service_message" "is_forward"           "was_downgraded"
[34] "is_archive"         "cache_has_attachments" "cache_roomnames"
[37] "was_data_detected"  "was_deduplicated"     "chat_id"
[40] "message_id"         "ROWID"                "guid"
[43] "style"              "state"                "account_id"
[46] "properties"         "chat_identifier"      "service_name"
[49] "room_name"          "account_login"        "is_archived"
[52] "last_addressed_handle"
```

Codez

```
35 text_frequency_by_month <- function(data){
36   hist(as.Date(sort(data$date/100000), origin="2002-08-15"),
37        breaks='months',
38        freq = TRUE,
39        format = "%d %b",
40        col=c('red'))
41 }
42
```

Text Frequency By Month



Demo send/receive

```
load_balance_data <- function(con){
  rs <- dbSendQuery(con, "  select chat_identifier, count(*), sum(is_from_me) from_me,
  count(*) - sum(is_from_me) as to_me, name, gender from message m
  join chat_message_join j
  on j.message_id = m.ROWID
  join chat c
  on j.chat_id = c.ROWID
  JOIN demographics d
  ON c.chat_identifier =  ('+' || d.number)
  group by chat_identifier;")
  data <- fetch(rs, n=Inf)
  return(data)
}
```

Saving to a file

```
png('39.png')
pie_chart_for_text_balance_ratio(balance_data[39,])
dev.off()
```

Let Talk About Sex

Let Talk About Sex

GET YOUR MIND OUT OF THE GUTTER!!!

```sql
select
(select count(*) from demographics where gender = 1) as num_men,
(select count(*) from demographics where gender = 0) as num_women;
```

num_men	num_women
40	26

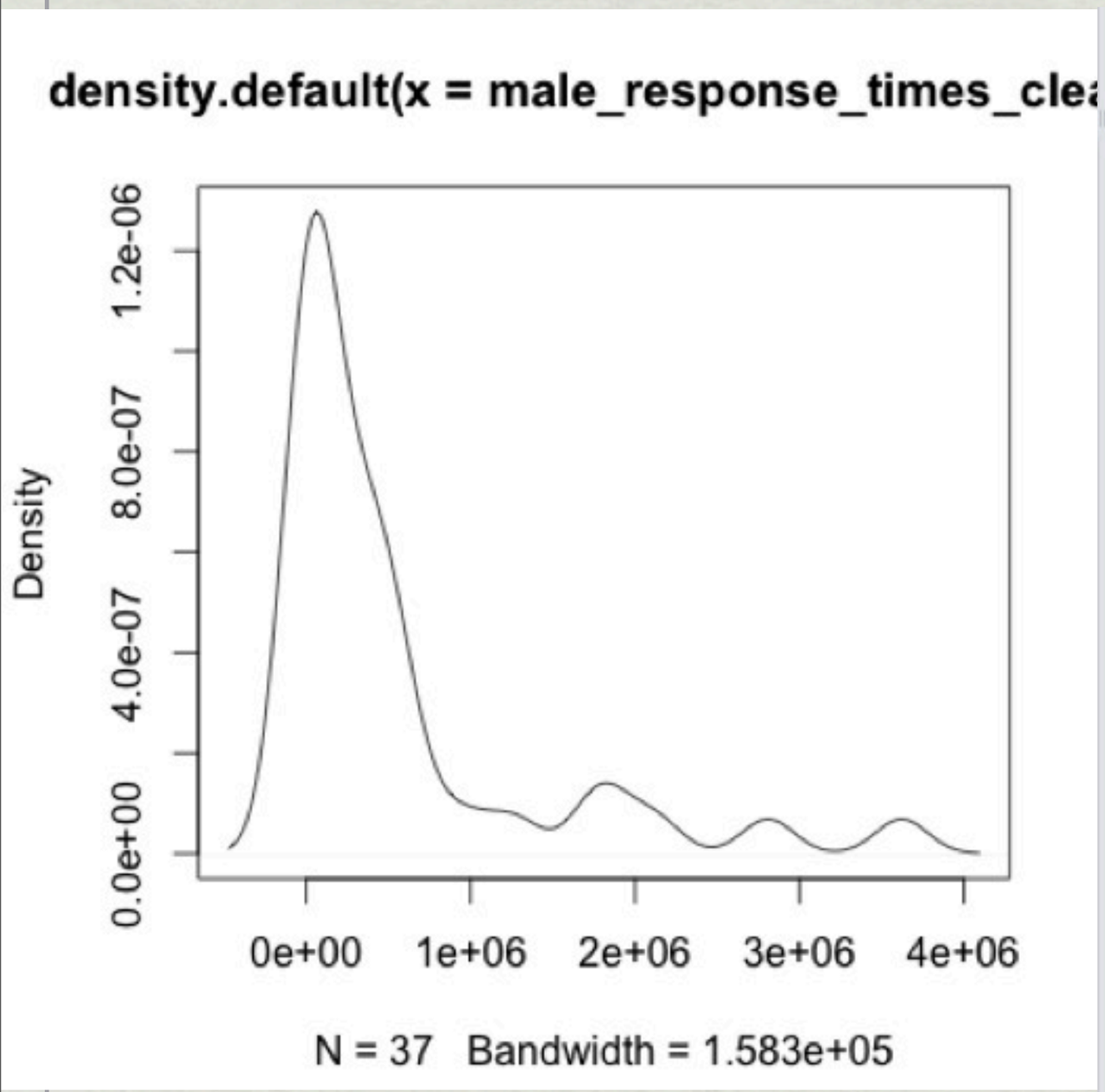
```
27  SELECT
28      count(*), name, group_concat(date)
29  FROM
30      message m
31  JOIN chat_message_join j ON j.message_id = m.ROWID
32  JOIN chat c ON j.chat_id = c.ROWID
33  JOIN demographics d ON c.chat_identifier =  ('+' || d.number)
34  group by name
35  order by count(*) desc;
36
```

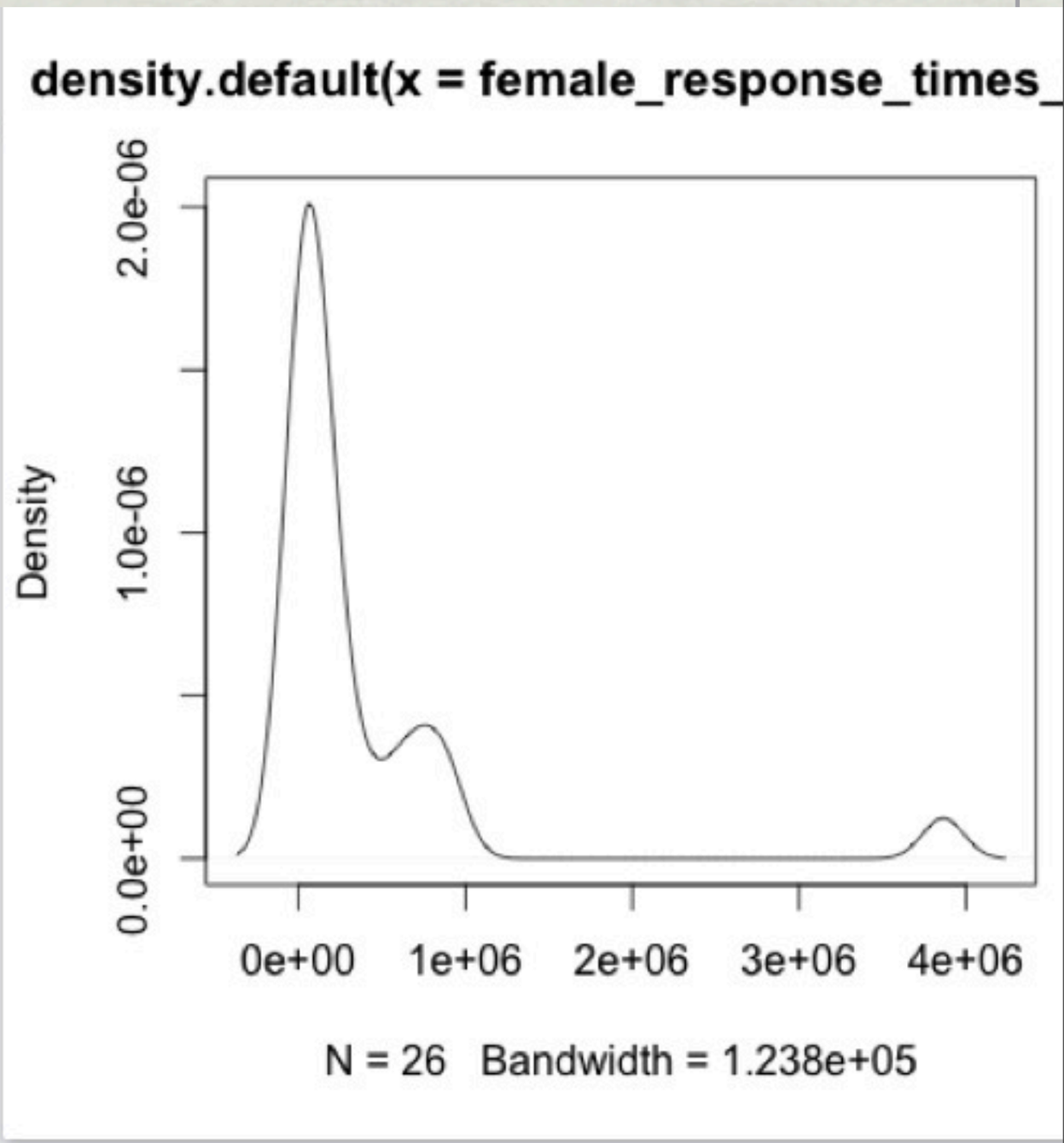
R?

```
127  median_response_times <- rep(0, length(res))
128  count = 0
129  for(r in res){
130    count = count + 1
131    comps <- unlist(strsplit(r, ","))
132    print(length(comps))
133    diffs <- rep(0, length(comps) - 1)
134    i = 0
135    for(c in comps){
136      i = i + 1
137      if(i > 1){
138        diffs[i] <- (as.integer(c) - as.integer(comps[i-1]))
139      }
140    }
141    median_response_times[count] <- mean(diffs)
142  }
```

R

```
times$median_response_time <- median_response_times
male_response_times <- times[times$gender == 1,]
male_response_times_clean <- male_response_times[!is.na(male_response_times$median_response_ti
female_response_times <- times[times$gender == 0,]
female_response_times_clean <- female_response_times[!is.na(female_response_times$median_respo
mean(female_response_times_clean)
mean(male_response_times_clean)
t.test(female_response_times_clean, male_response_times_clean)
male_d <- density(male_response_times_clean)
female_d <- density(female_response_times_clean)
plot(female_d)
plot(male_d)
```


density.default(x = male_response_times_clea
Density
0.0e+00
4.0e-07
8.0e-07
1.2e-06
0e+00
1e+06
2e+06
3e+06
4e+06
N = 37 Bandwidth = 1.583e+05


density.default(x = female_response_times_
Density
0.0e+00
1.0e-06
2.0e-06
0e+00
1e+06
2e+06
3e+06
4e+06
N = 26 Bandwidth = 1.238e+05