Jennifer Mead

What makes a match? Who gets a date?

# Business question

According to the US Census Bureau there are 110.6 million single or unmarried people over 18 in the US ([figures from 2016](https://www.census.gov/newsroom/facts-for-features/2017/single-americans-week.html)). That’s 45.2% of adults, with 53.2% women and 48.6% men. The dating industry is a profitable one. Nasdaq.com says that online dating sites and apps (such as OkCupid, Match.com, and Tinder) are part of an industry in the US that “generates approximately  [$2 billion in revenue each year](http://www.ibisworld.com/industry/default.aspx?indid=1723) and expanded at an annual rate of 5% between 2010 and 2015.” ([article from 2016](https://www.nasdaq.com/article/of-love-and-money-the-rise-of-the-online-dating-industry-cm579616)).

People invest a lot of time and effort into dating. There are a lot of options for meeting people: apps, sites, matchmakers, bars, activities, and being introduced by friends, for example. **There isn’t a sure-fire way to get a date anytime you want one, but are there things you could do that might help you get what you want?**

In a broader sense, this data deals with preferences, choices, and user ratings. There’s a lot of data in the world that falls in that category. Techniques we use here with speed daters may also be used in those contexts.

# Context of analysis

Researchers at Columbia University ran rounds of speed dating and had the student-participant-daters answer surveys before, during, and after the event. They collected data on many aspects of the daters, including what interests and lifestyles they had, how they expected the event to go, how the daters rated each other, whether or not each couple was a “match” (both saying they wanted to meet again), and how many went on a date after the event.

The data has been released publicly on Kaggle.com, and **the most common analysis is to determine which factors are most likely to lead to a *match*.** **In my analysis, I’m also going to look at which factors are most likely to lead to a *date*.** (Every date started as a match, but every match does not lead to a date. People sometimes change their mind or decide not to follow up.)

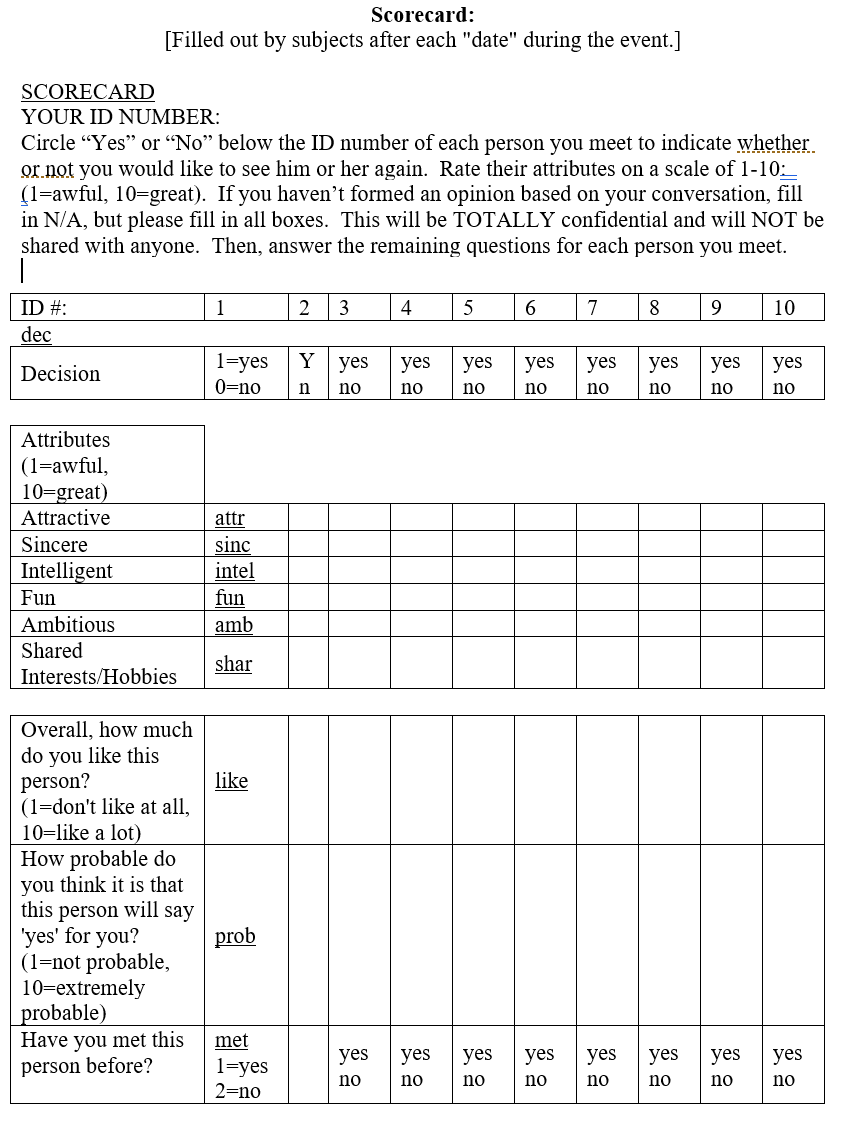
# Background information

If you are unfamiliar with speed dating, here are the basics:

* There are both heterosexual and homosexual speed dating events, but the one at Columbia was just for heterosexual couples.
* Each man and woman are numbered and paired up for a short (3-4 minute) ***speed date***.
* At the end of the date, each person writes down notes and a rating for their partner.
* A bell rings, and the men move to the next woman.
* The cycle repeats until all of the men and women have met.
* At the end of the night, the organizer collects the rating cards.
* If both people chose each other, it’s a ***match***, and the organizer gives them both information to contact each other to set up a ***date*** *after the event*.

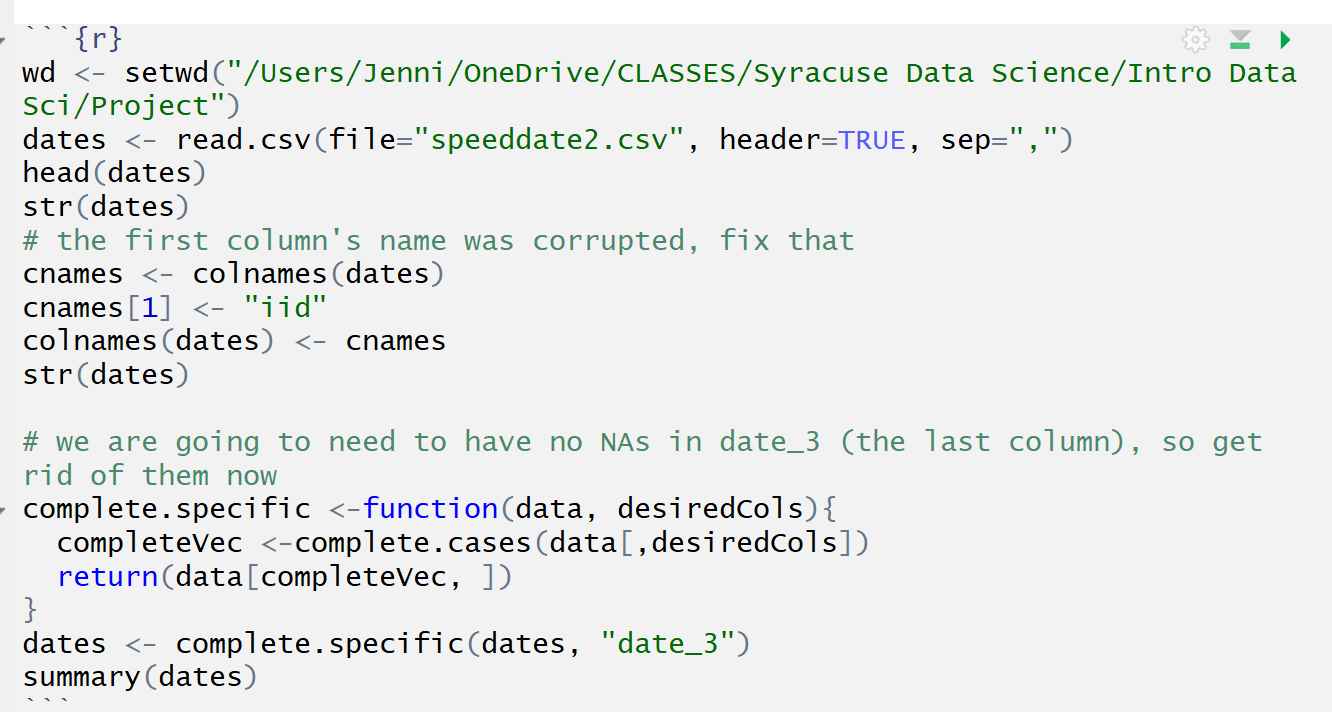
Here’s a [more in-depth description](https://www.theatlantic.com/health/archive/2014/07/speed-dating-in-the-time-of-tinder/372425/) if you need more context.

The scorecard that the Columbia participants filled out:



# Importing data, cleaning data, developing a data dictionary

I got the data from [Kaggle.com](https://www.kaggle.com/annavictoria/speed-dating-experiment/data) in Excel format.



I would have loved to have kept all of the variables, but unfortunately, some columns had a huge number of blanks. Out of about 8000 rows, one column had about 6500 blanks. If I were to use na.omit on every field with blanks when I imported the data to R, I’d lose most of the data. If I replaced that many values with the mean or the median, I’d be pretty skeptical of my results. Instead, I figured out how many blanks were in each column and used that as part of my decision on what to keep. See the color-coded Excel excerpt below (green is few blanks, orange is many). The **definition** of each field is on the right.

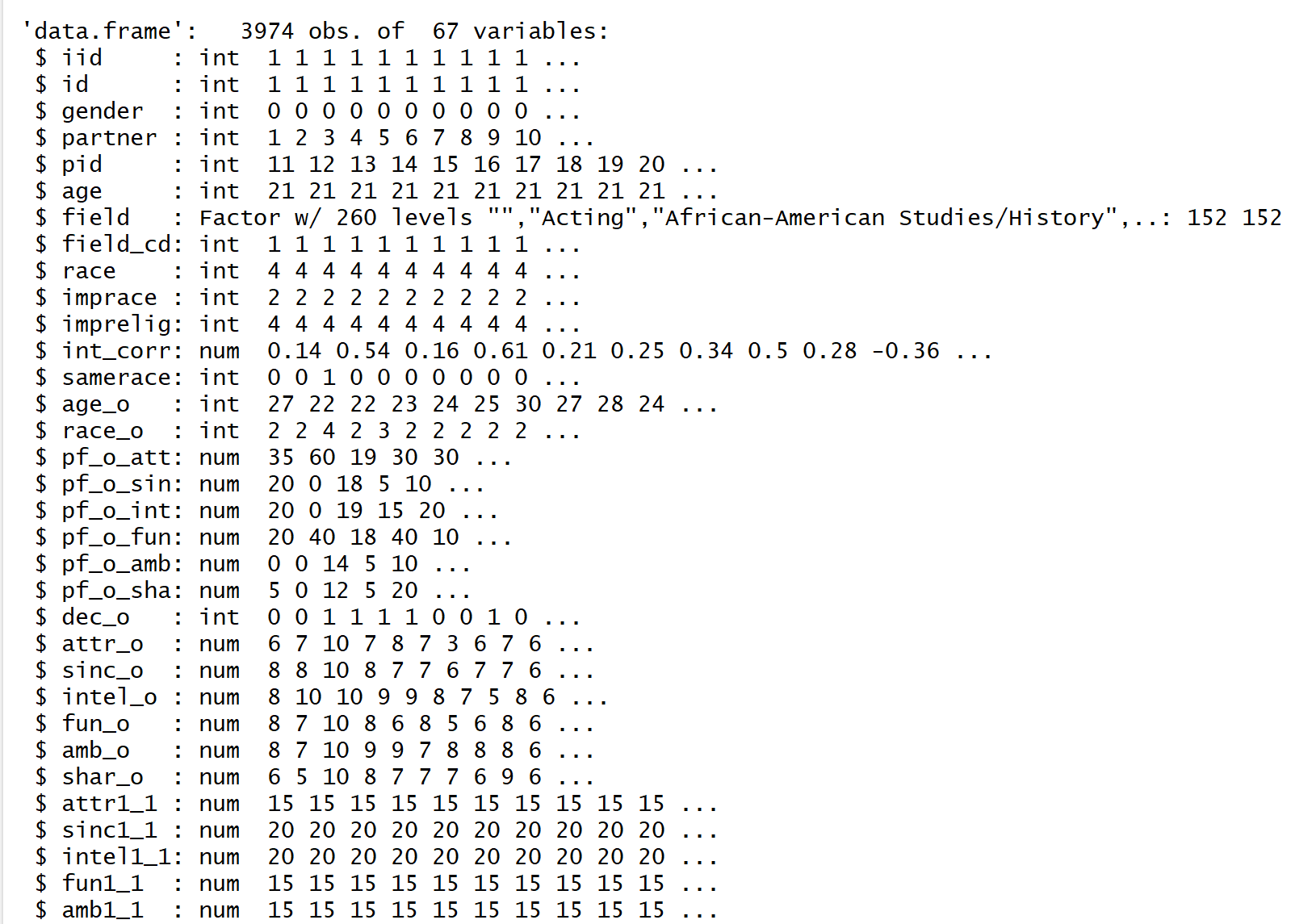
Because you’ll see it a lot in the variable names, let me point out that there are **six main qualities that people are asked to rate**: attractiveness (att), sincerity (sinc or \_sin), intelligence (intel or int), fun (fun), ambition (amb), and shared interests (shar or sha). So, for example, you could rate someone as very attractive and fun but not very smart. Or you could say the opposite gender looks for partners that are ambitious with shared interests. Or you could give yourself top ratings across the board, as one participant actually did.

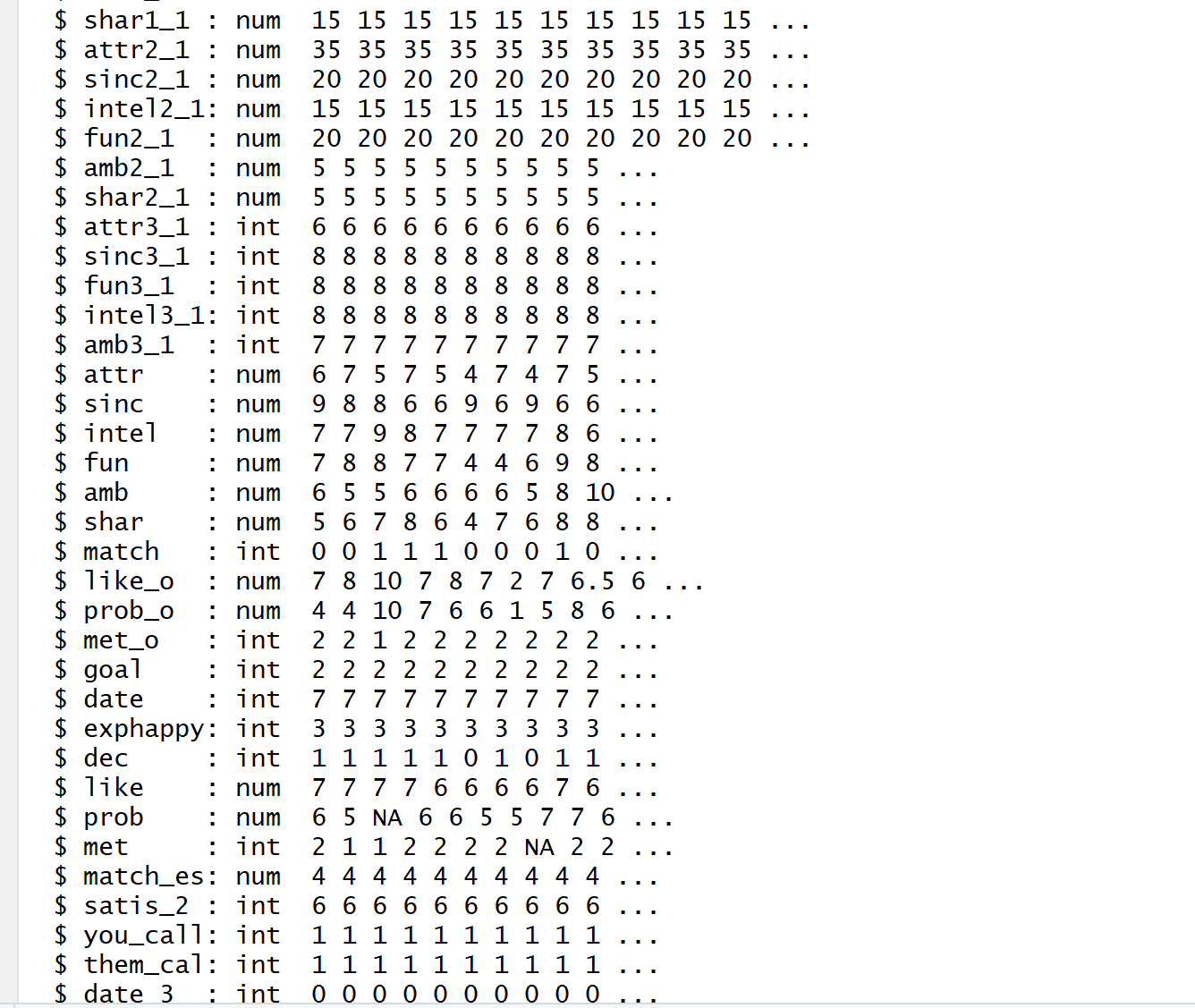
|  |  |  |  |
| --- | --- | --- | --- |
| Col Num in Original Data | Variable Name | Number of Blanks | Variable Definition |
| 1 | iid | 0 | unique id |
| 2 | id | 1 | id in wave |
| 3 | gender | 0 | 0 fem, 1 male |
| 11 | partner | 0 | partner id in wave |
| 12 | pid | 10 | partner unique id |
| 13 | match | 0 | 0 no, 1 yes |
| 14 | int\_corr | 158 | correlation between interests |
| 15 | samerace | 0 | 0 no, 1 yes |
| 16 | age\_o | 104 | age of partner |
| 17 | race\_o | 73 | race of partner |
| 18 | pf\_o\_att | 89 | partner's preference for attractiveness |
| 19 | pf\_o\_sin | 89 | partner's preference for sincerity |
| 20 | pf\_o\_int | 89 | partner's preference for intelligence |
| 21 | pf\_o\_fun | 98 | partner's preference for fun |
| 22 | pf\_o\_amb | 107 | partner's preference for ambition |
| 23 | pf\_o\_sha | 129 | partner's preference for shared interests |
| 24 | dec\_o | 0 | partner's decision |
| 25 | attr\_o | 212 | rating given by partner for attractiveness |
| 26 | sinc\_o | 287 | rating given by partner for sincerity |
| 27 | intel\_o | 306 | rating given by partner for intelligence |
| 28 | fun\_o | 360 | rating given by partner for fun |
| 29 | amb\_o | 722 | rating given by partner for ambition |
| 30 | shar\_o | 1076 | rating given by partner for shared interest |
| 31 | like\_o | 250 | how much do you like this person |
| 32 | prob\_o | 318 | how probable is that that this person will say yes for you |
| 33 | met\_o | 385 | have you met this person before |
| 40 | race | 63 | race of participant |
| 41 | imprace | 79 | how important to date same race |
| 42 | imprelig | 79 | how important to date same religion |
| 68 | exphappy | 101 | how happy do you expect to be with the people you meet |
| 70 | attr1\_1 | 79 | how important is attractiveness to you |
| 71 | sinc1\_1 | 79 | how imp is sincerity to you |
| 72 | intel1\_1 | 79 | how imp is intelligence to you |
| 73 | fun1\_1 | 89 | how imp is fun to you |
| 74 | amb1\_1 | 99 | how imp is ambition to you |
| 75 | shar1\_1 | 121 | how imp is shared interested to you |
| 82 | attr2\_1 | 79 | how important is attractiveness to (other sex) |
| 83 | sinc2\_1 | 79 | how imp is sincerity to (other sex) |
| 84 | intel2\_1 | 79 | how imp is intelligence to (other sex) |
| 85 | fun2\_1 | 79 | how imp is fun to (other sex) |
| 86 | amb2\_1 | 89 | how imp is ambition to (other sex) |
| 87 | shar2\_1 | 89 | how imp is shared interested to (other sex) |
| 88 | attr3\_1 | 105 | how do you measure up on attractiveness |
| 89 | sinc3\_1 | 105 | how do you measure up on sincerity |
| 90 | fun3\_1 | 105 | how do you measure up on fun |
| 91 | intel3\_1 | 105 | how do you measure up on intelligence |
| 92 | amb3\_1 | 105 | how do you measure up on ambition |
| 98 | dec | 0 | decision: 0 no, 1 yes |
| 99 | attr | 202 | rating of partner attractiveness |
| 100 | sinc | 277 | rating of partner sincerity |
| 101 | intel | 296 | rating of partner intelligence |
| 102 | fun | 350 | rating of partner fun |
| 103 | amb | 712 | rating of partner ambition |
| 104 | shar | 1067 | rating of partner shared interests |
| 105 | like | 240 | how much do you like this person |
| 106 | prob | 309 | probability that this person will choose you |
| 107 | met | 375 | have you met this person before |
| 108 | match\_es | 1173 | how many matches do you estimate you'll get (both say yes) |
| 157 | you\_call | 4404 | how many matches did you call to set up a date |
| 158 | them\_cal | 4404 | how many matches called you to set up a date |
| 159 | date\_3 | 4404 | have you been on a date with any of your matches, 0 no 1 yes |

In general, I kept columns with information about:

* ID numbers
* Interests, race, religion, college major
* Preferences in the other sex
* Ratings from partners
* Partner’s preferences in the other sex
* Self-ratings (ex: “How do you measure up in intelligence?”)
* Whether or not the pair was a match (both said yes to meeting after the event)
* Whether or not the pair went on a date after the event

In an unfortunate twist, the variable that I planned to use to determine if the couple had been on a date was one of the variables with many blanks. (See variable 159, “date\_3”, with 4404 blanks.) I decided that for this one column, I was willing to take the hit. If I had only analyzed matches instead of dates, I would have had twice as much data to work with.

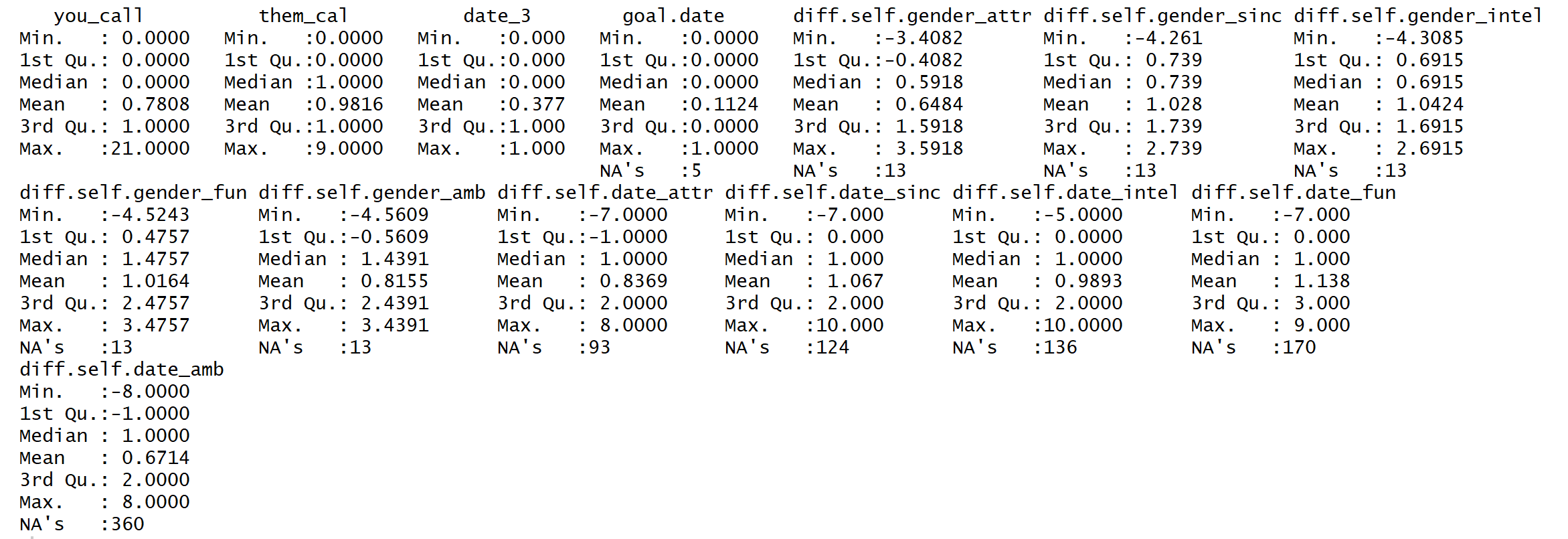




# New features

This data set already had a ton of features, but for the heck of it, I added a few.

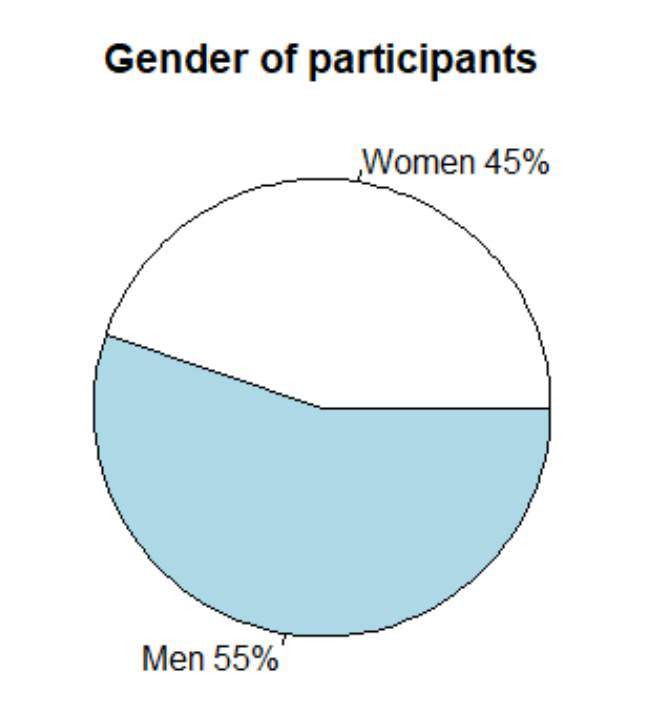
* Does someone’s rating of themselves versus the rating their data gave them relate to results? For example, maybe if a dater is more self-aware, that might translate into more dates. (*diff.self.date\_<quality>)*
* Does someone’s rating versus the average rating that the opposite sex gives relate to results? For example, if a man is hotter than the average man, that might lead to more dates for him. (*diff.self.gender\_<quality>)*
* Does it matter if a person’s goal was to get a date or not? Goals are divided into six categories. I grouped 3 and 4 (“date” and “serious relationship”) into the *goal.date* category as a 1, and all other goals are a 0.



# Getting to know the data: descriptive statistics and questions to inform our models

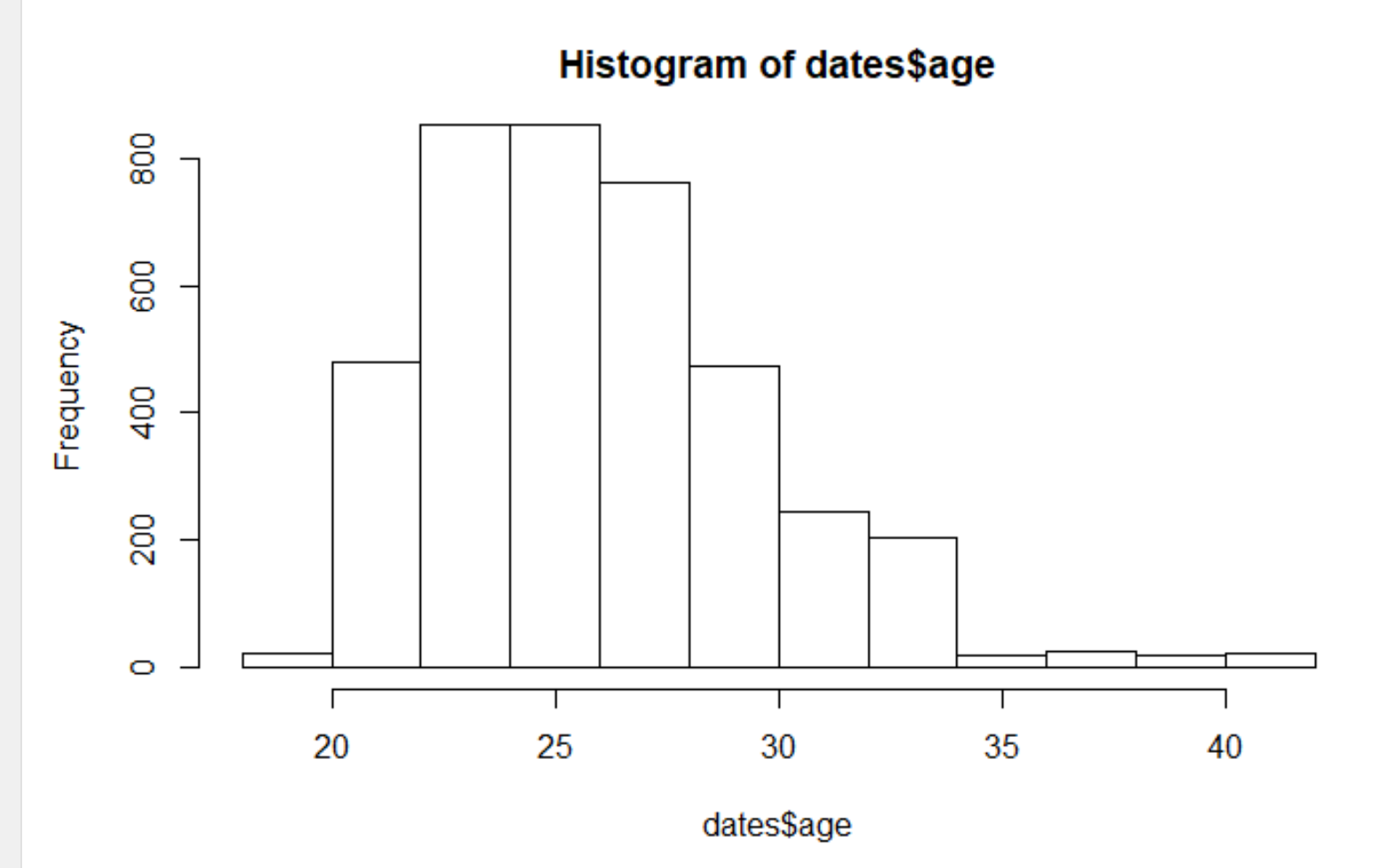
## How many men? How many women?

They are about half women and half men. The full dataset had exactly 50% of each, but remember that we removed NAs in the date\_3 column.



## How old are they?

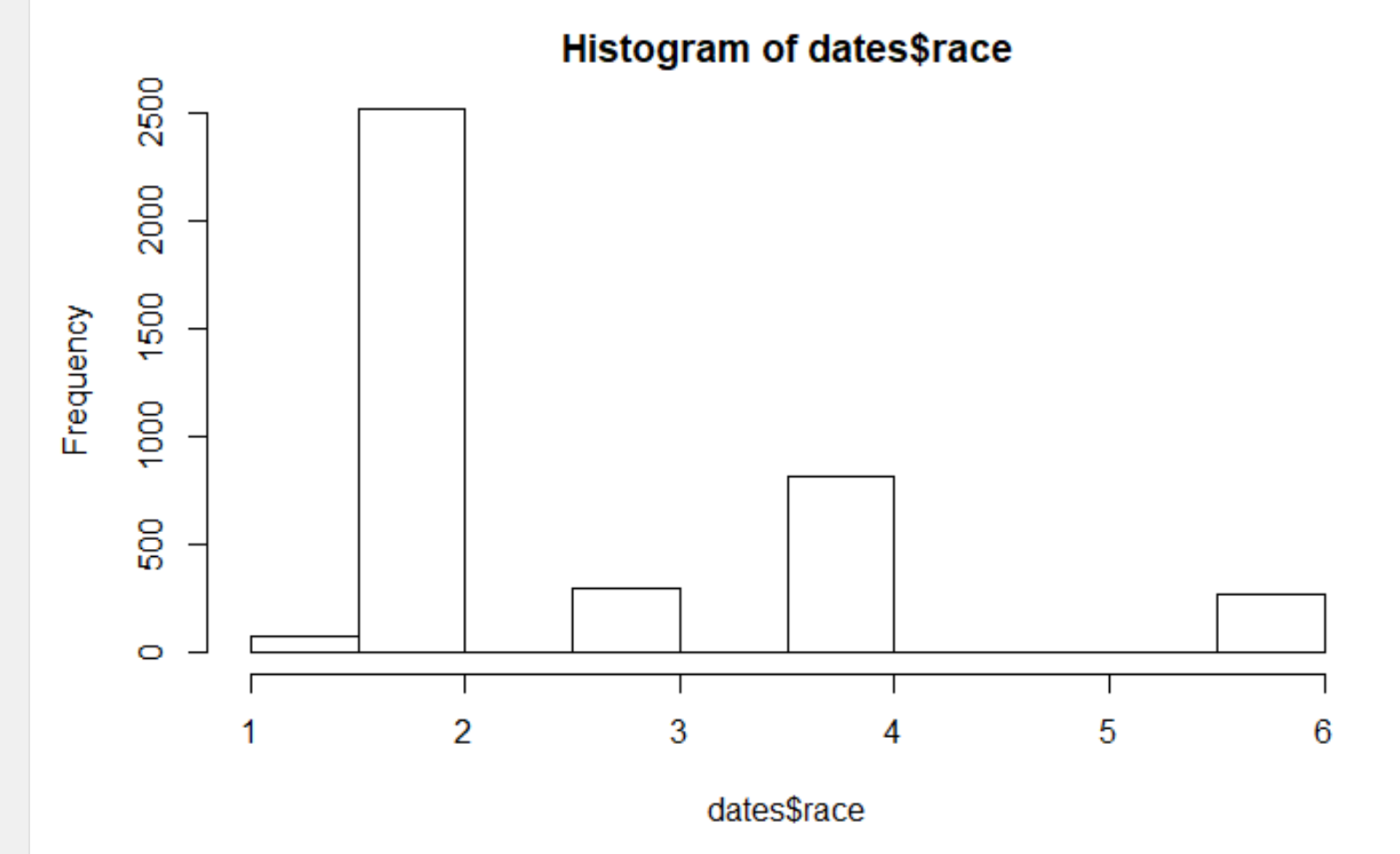
 Most of them are between 20 and 30, which makes sense because they are Columbia University students.



## What race/ethnicity?

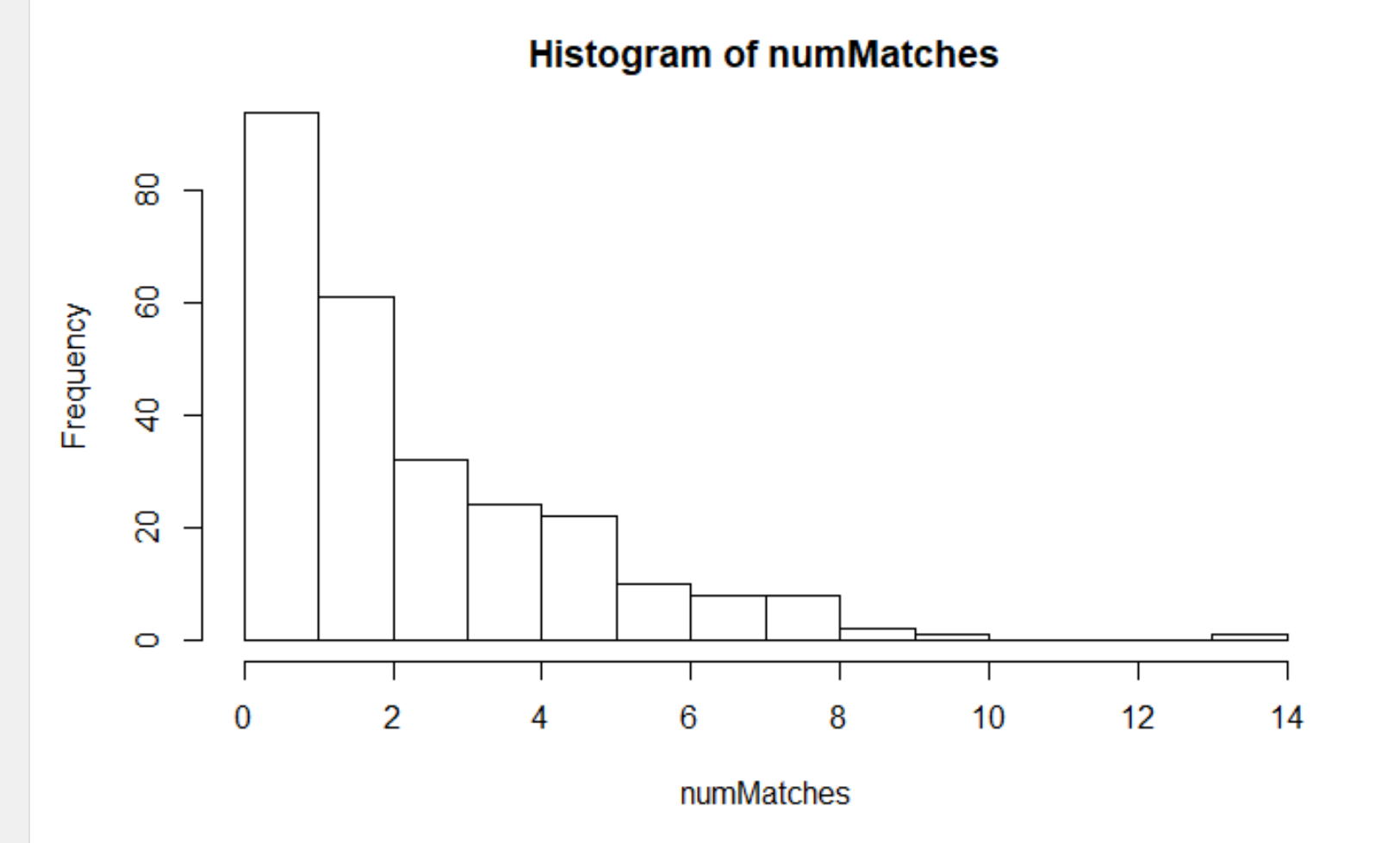
Most of them are what the dataset calls “European/Caucasian-Amercian”. The code numbers from the dataset:

* Black/African American=1;
* European/Caucasian-American=2;
* Latino/Hispanic American=3;
* Asian/Pacific Islander/Asian-American=4;
* Native American=5;
* Other=6



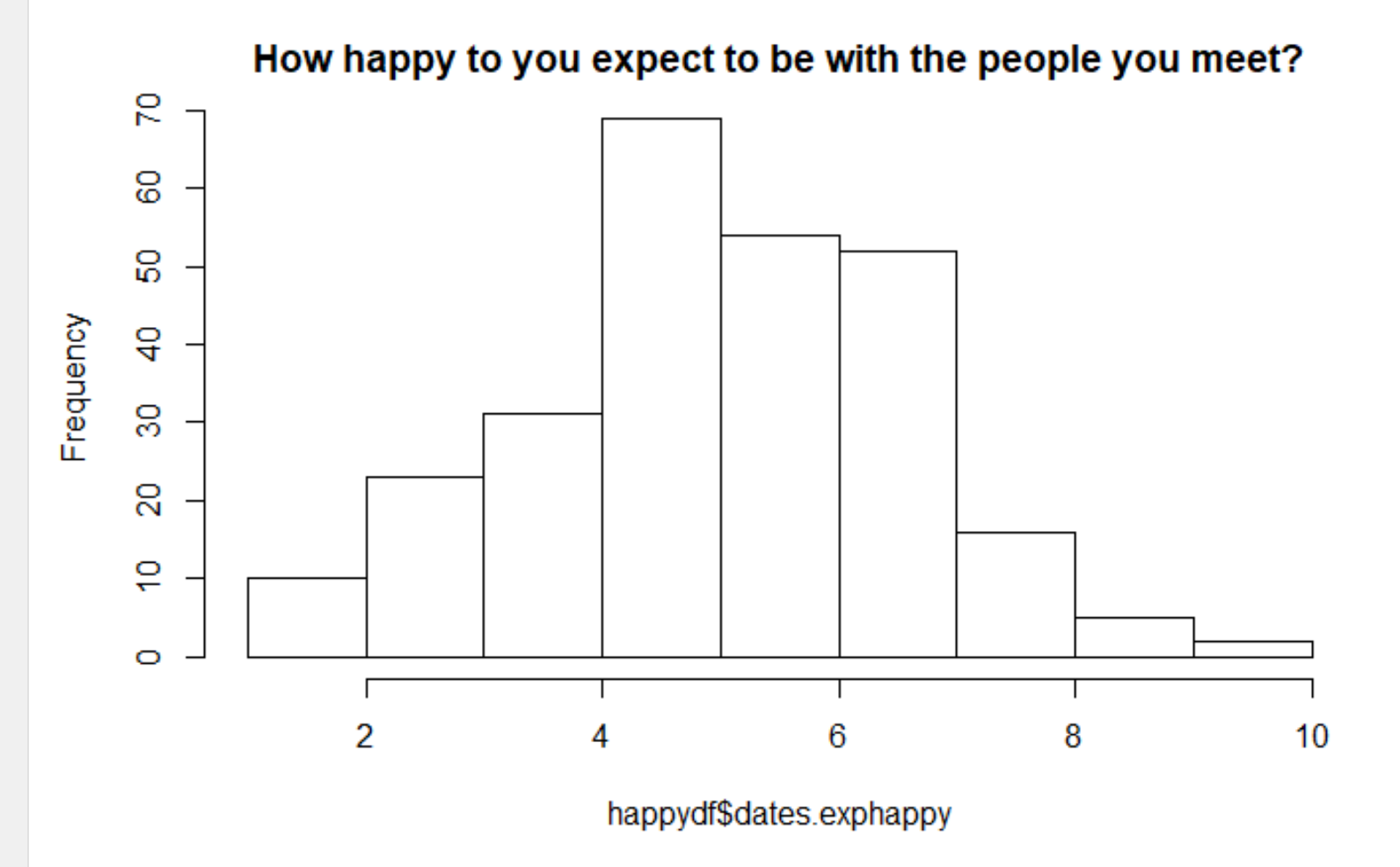
## How many matches did they get?

The most common number of matches is zero. Many people had 1-5 matches. More than 8 matches is rare.



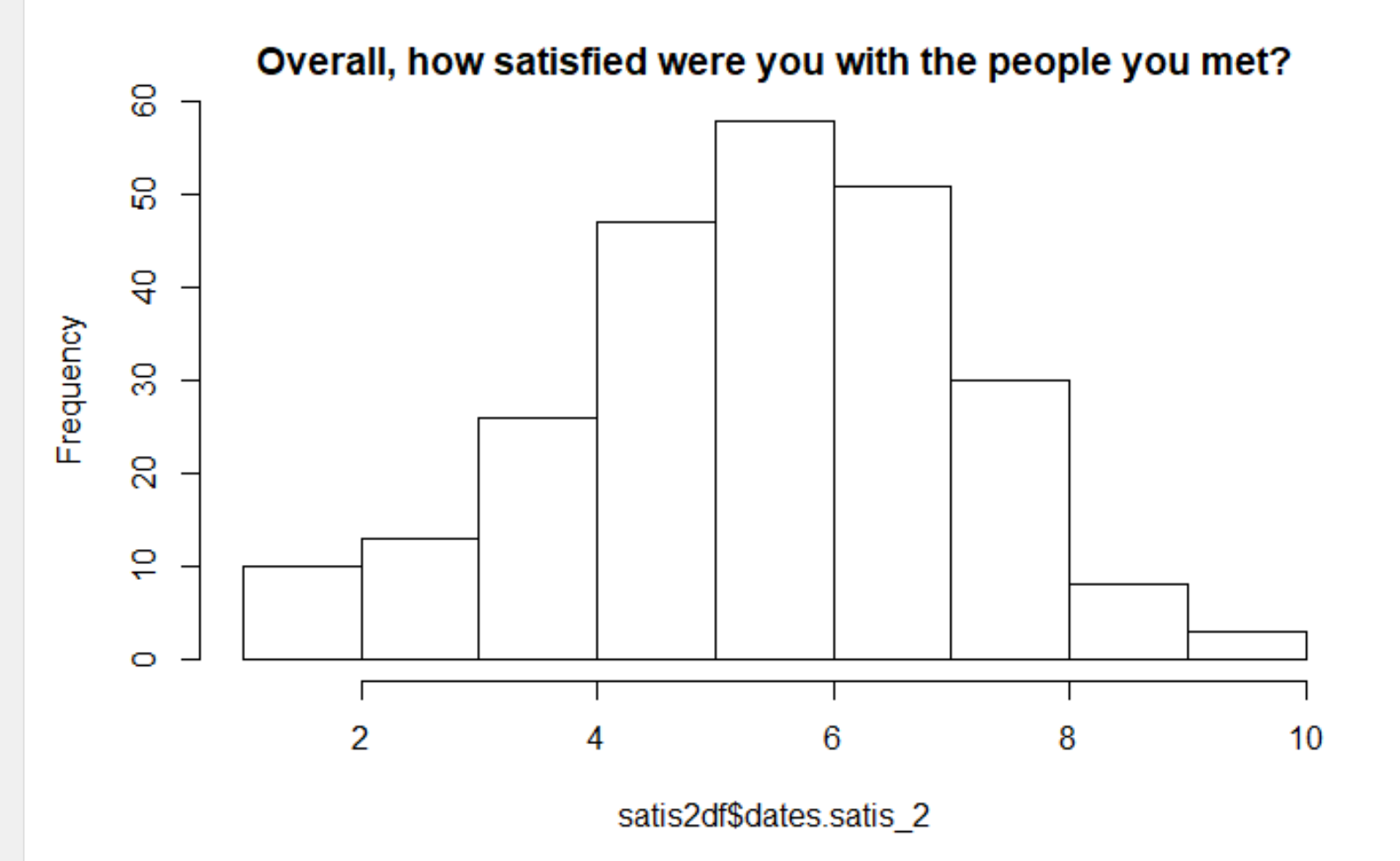
## Before showing up to the speed dating event, how happy did our daters expect to be with the people they would meet?

The most common rating was between a 4 and 7 on a scale of 1-10.



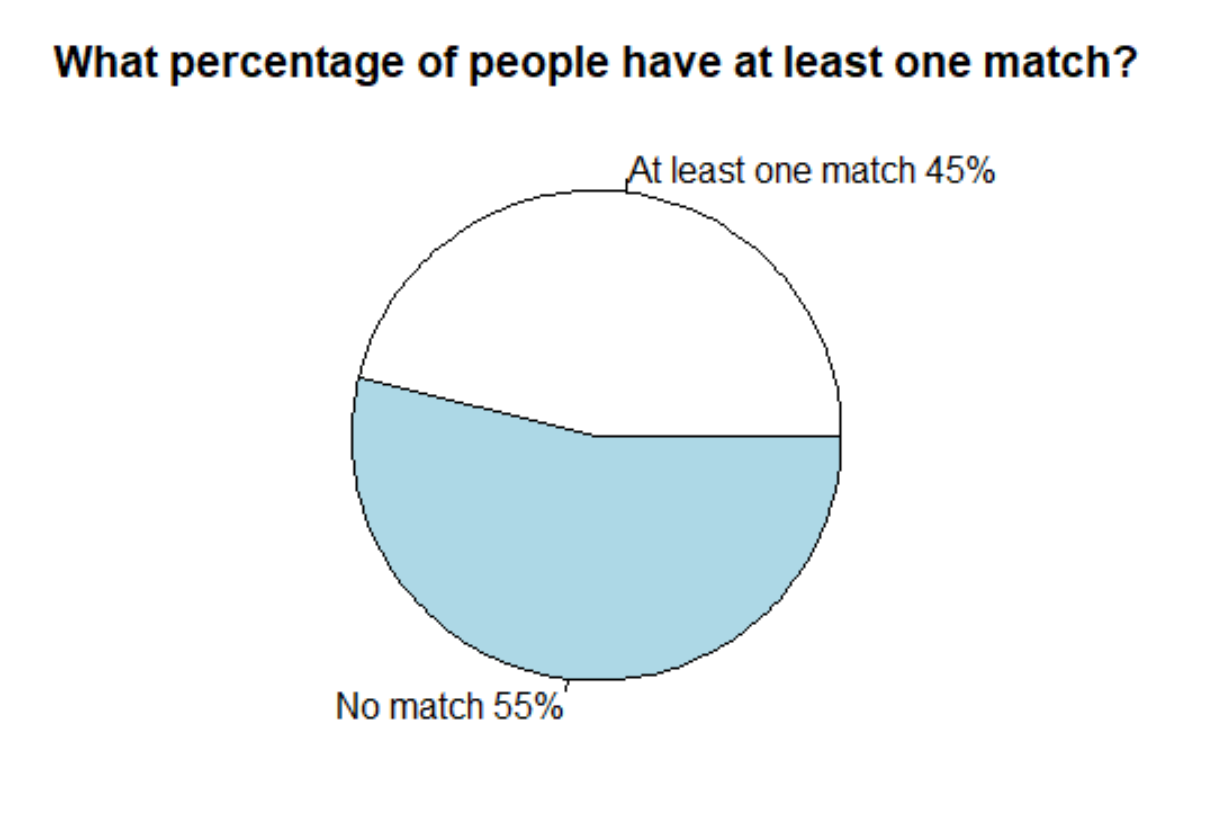
## After the event, how happy were they were the people they met?

Ratings were about the same, but there is a slightly more positive trend.



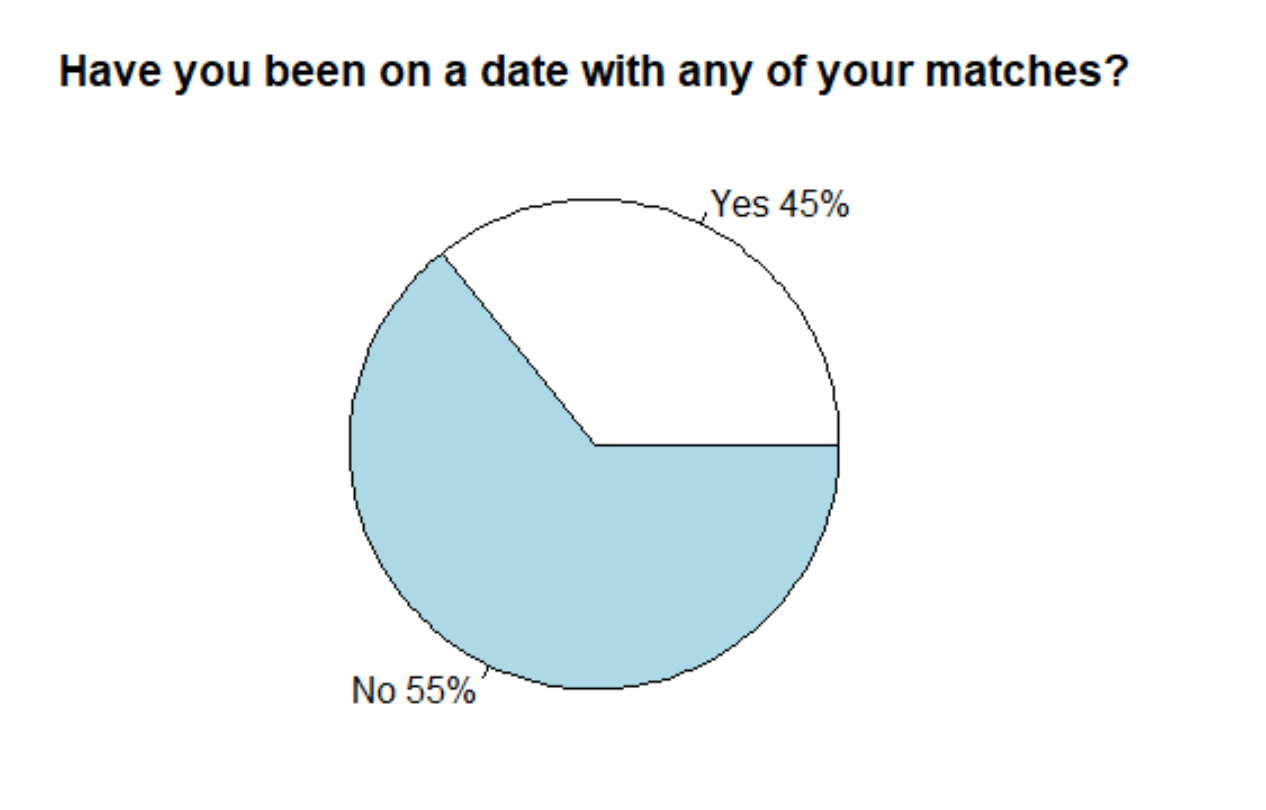
## How many people had at least one match? At least one date?

45% had a match, and 55% had none. That’s fewer people with matches than I expected, but not unreasonable.



Surely, then. The ones with matches went out on dates afterwards.

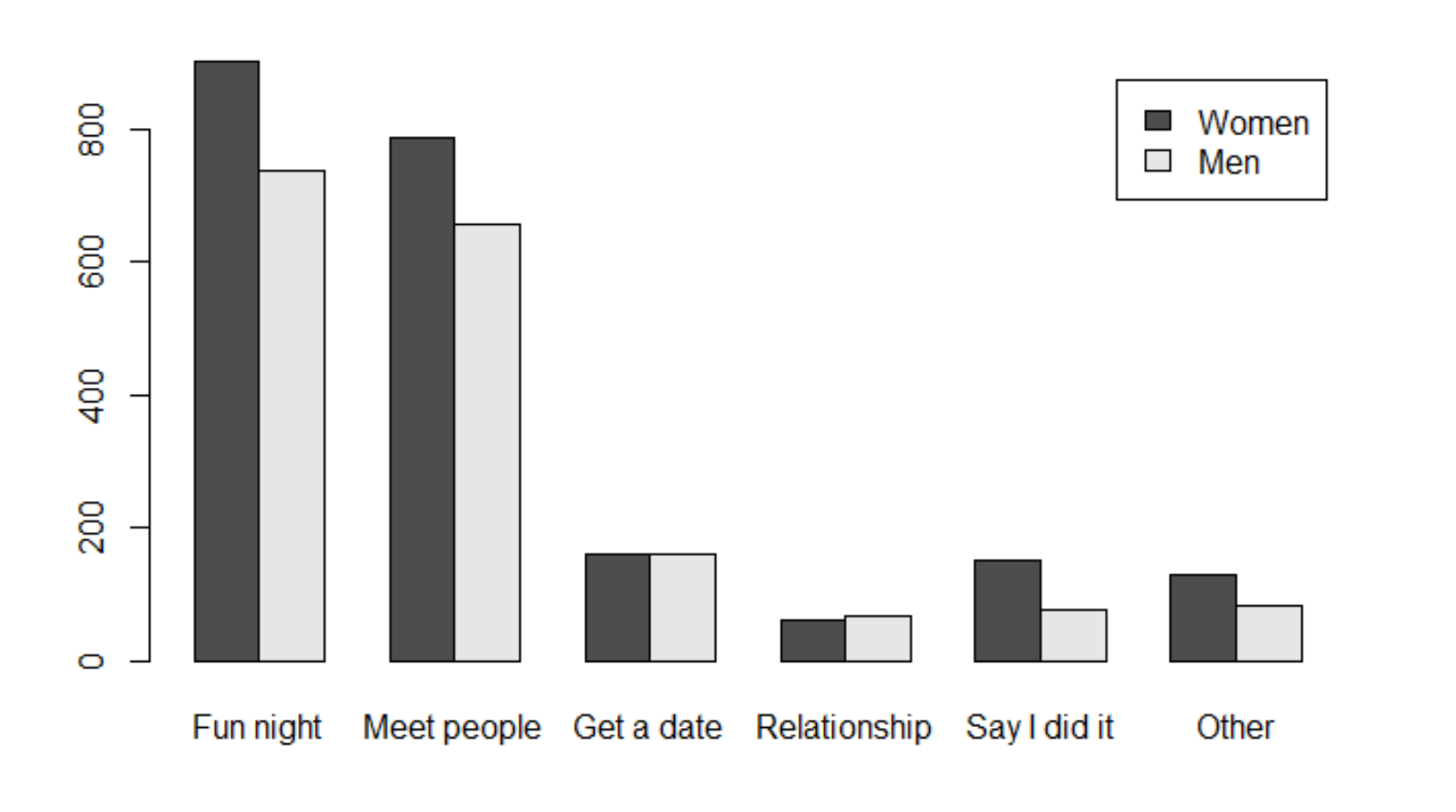
Not really.



That’s odd. Well, maybe people didn’t intend to go on dates.

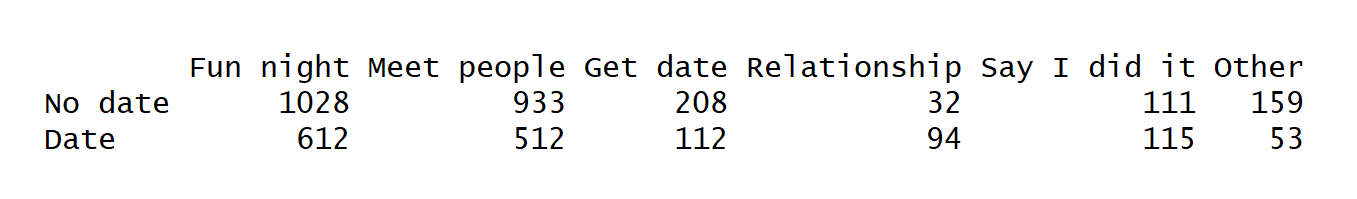
## What were their goals?

“Seems like a fun night” and “Meet people” are the big winners. “Get a date” and “Serious relationship” have about the same number of men and women, but both are low.



That’s not looking good for the “goal.date” feature I made, but we’ll see.

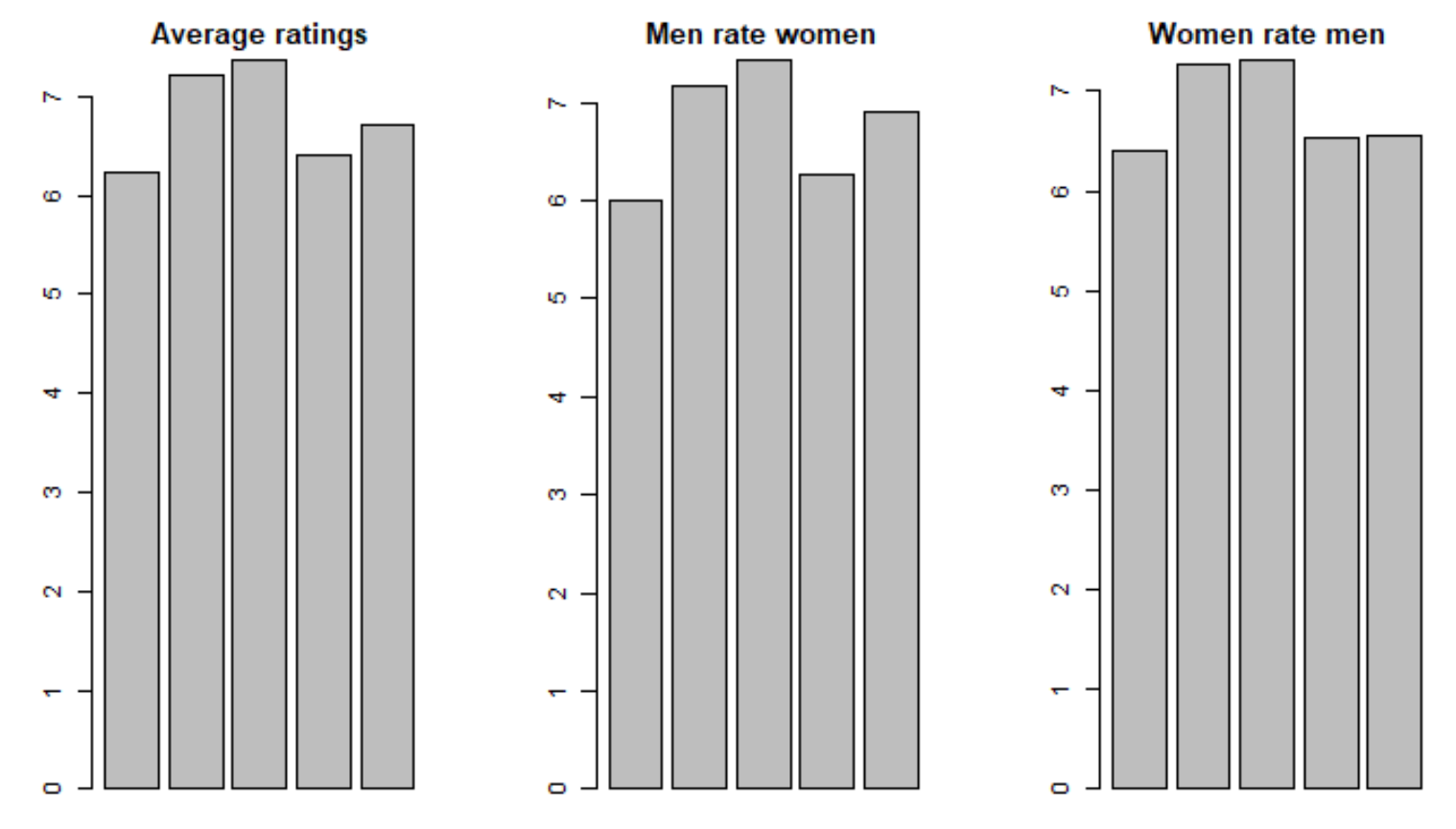
How about looking at how many people with each goal went on a date later? Maybe the date-goers tend towards certain goals.



Nope. “Serious relationship” and “Say I did it” have more people with dates than without, but the rest of the categories seem to represent the something similar to the 55/45 split we saw overall.

## What kind of ratings did men and women give each other on average?

Men seem to think that women are a little less attractive and more ambitious than women find men. Women find men as ambitious as they are fun. But overall, this is looking pretty consistent.



# Use of modeling techniques

I used random forests to figure out which variables had the most influence. As a reminder:

* a *match* means that both people said “yes, I want to see them again” during speed dating
* a *date* means that those people met after speed dating for a date in the real world

I wanted to compare the influencing variables for matches and dates, but when I got initial results, it looks as if having the match data included might be swaying the results for dates. I separated them to compare the results.

I ran three models:

* whether or not a participant had a **match**
* whether or not a participant had a **date**, *excluding* the data that was collected per speed date (i.e. the data used for the match model)
* whether or not a participant had a **date**, *including* the data that was collected per speed date (i.e. the data used for the match model)

## Match model

```{r}

# random forest on matches (matches happen at the end of a speed date, but before dates in the real world)

# create measurement and label variables

colnames(dates)

remove.match.vars <-c(67, 66, 65, 59, 52, 22, 7,5,4,2,1)

# IDs are not predictive: iid = 1, id = 2, partner=4, pid=5,

# Their major throws an error for too many factor levels: field=7,

# A match comes from two yes decisions: dec\_o=22, dec=59,

# We are trying to predict match: match=52,

# These things happen AFTER what we want to predict: you\_call=65, them\_cal=66, date\_3=67

target.match.measurement <- dates[,-remove.match.vars] #remove these fields

target.match.measurement[is.na(target.match.measurement)] <- 0

colnames(target.match.measurement)

target.match.label <- as.factor(dates[,52]) #predict this field

# run random forest

dates.match.rf <-randomForest(target.match.measurement, target.match.label, prox=TRUE)

dates.match.rf

dates.match.rf.importance <- importance(dates.match.rf)

str(dates.match.rf.importance)

dates.match.rf.importance

plot(dates.match.rf.importance)

hist(dates.match.rf.importance)

# looking just at the low values, in order

# keep plot sort, change where cut off is for MDG

imp.match.df <- data.frame(rownames(dates.match.rf.importance), dates.match.rf.importance)

sorted.imp.match <- imp.match.df[order(-imp.match.df$MeanDecreaseGini) ,]

sorted.imp.match

plot(sort(imp.match.df$MeanDecreaseGini))

```

## Date-excluding-match-data model

# run random forest on dates$date\_3: what leads to a date?

# remove dates$match (because we know 100% of dates were also a match)

```{r}

# create measurement and label variables

colnames(dates)

# IDs are not predictive: iid = 1, id = 2, partner=4, pid=5,

# Their major throws an error for too many factor levels: field=7,

# A match comes from two yes decisions: dec\_o=22, match=52, dec=59,

# A date happens after one of you calls: you\_call=65, them\_cal=66,

# We are trying to predict date\_3: date\_3=67

remove.vars <-c(67, 66, 65, 59, 52, 22, 7,5,4,2,1)

target.date.measurement <- dates[,-remove.vars] #remove these fields

target.date.measurement[is.na(target.date.measurement)] <- 0

colnames(target.date.measurement)

target.date.label <- as.factor(dates[,67])

# run random forest

dates.date.rf <-randomForest(target.date.measurement, target.date.label, prox=TRUE)

dates.date.rf

dates.date.rf.importance <- importance(dates.date.rf)

str(dates.date.rf.importance)

dates.date.rf.importance

plot(dates.date.rf.importance)

imp.date.df <- data.frame(rownames(dates.date.rf.importance), dates.date.rf.importance)

sorted.imp.date <- imp.date.df[order(-imp.date.df$MeanDecreaseGini) ,]

sorted.imp.date

plot(sort(imp.date.df$MeanDecreaseGini))

```

## Date-including-match-data model

```{r}

names(dates)

with.matches <- subset(dates, match==1)

with.matches.test.dates <- with.matches[, c(1, 3, 6,8,9,10,11,29:39, 51,56,57,58,63:78)]

with.matches.test.dates

names(dates)

names(with.matches.test.dates)

```

```{r}

#random forest on people with matches to see what predicts a date

# these qualities are per person, not per pair, no partner ratings included

# create measurement and label variables

colnames(with.matches.test.dates)

remove.matches.test.dates.vars <-c(27,26,25,1)

# remove ID, you\_call, them\_cal, date\_3

target.matches.test.dates.measurement <- with.matches.test.dates[,-remove.matches.test.dates.vars] #remove these fields

target.matches.test.dates.measurement[is.na(target.matches.test.dates.measurement)] <- 0

colnames(target.matches.test.dates.measurement)

target.matches.test.dates.label <- as.factor(with.matches.test.dates[,27]) #predict this field

# run random forest

dates.matches.test.dates.rf <-randomForest(target.matches.test.dates.measurement, target.matches.test.dates.label, prox=TRUE)

dates.matches.test.dates.rf

dates.matches.test.dates.rf.importance <- importance(dates.matches.test.dates.rf)

str(dates.matches.test.dates.rf.importance)

dates.matches.test.dates.rf.importance

plot(dates.matches.test.dates.rf.importance)

hist(dates.matches.test.dates.rf.importance)

# sort table of MeanDecreaseGini and plot importance

imp.matches.test.dates.df <- data.frame(rownames(dates.matches.test.dates.rf.importance), dates.matches.test.dates.rf.importance)

sorted.m.t.d <- imp.matches.test.dates.df[order(-imp.matches.test.dates.df$MeanDecreaseGini) ,]

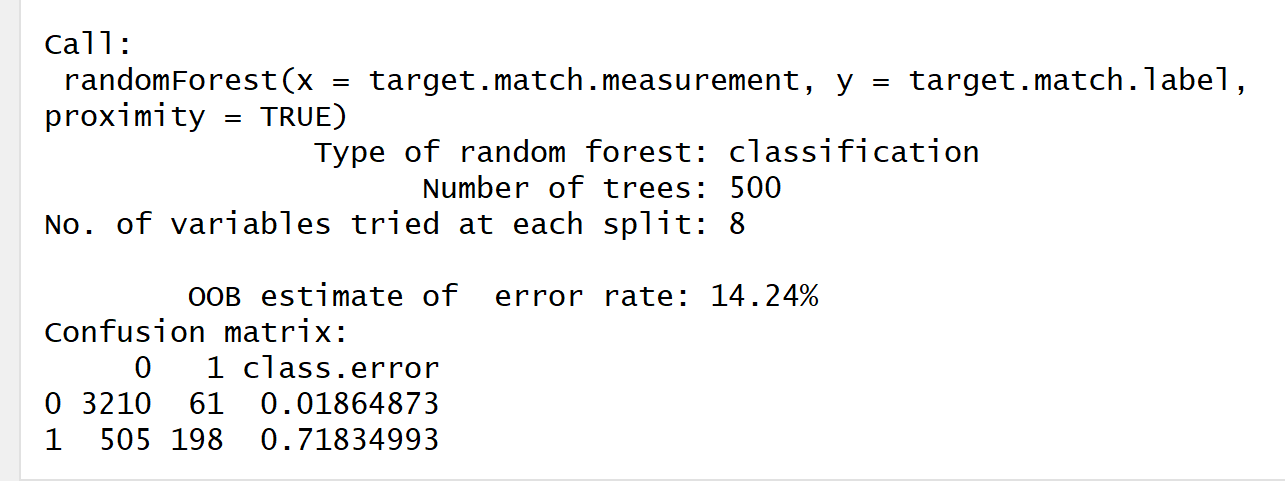
sorted.m.t.d

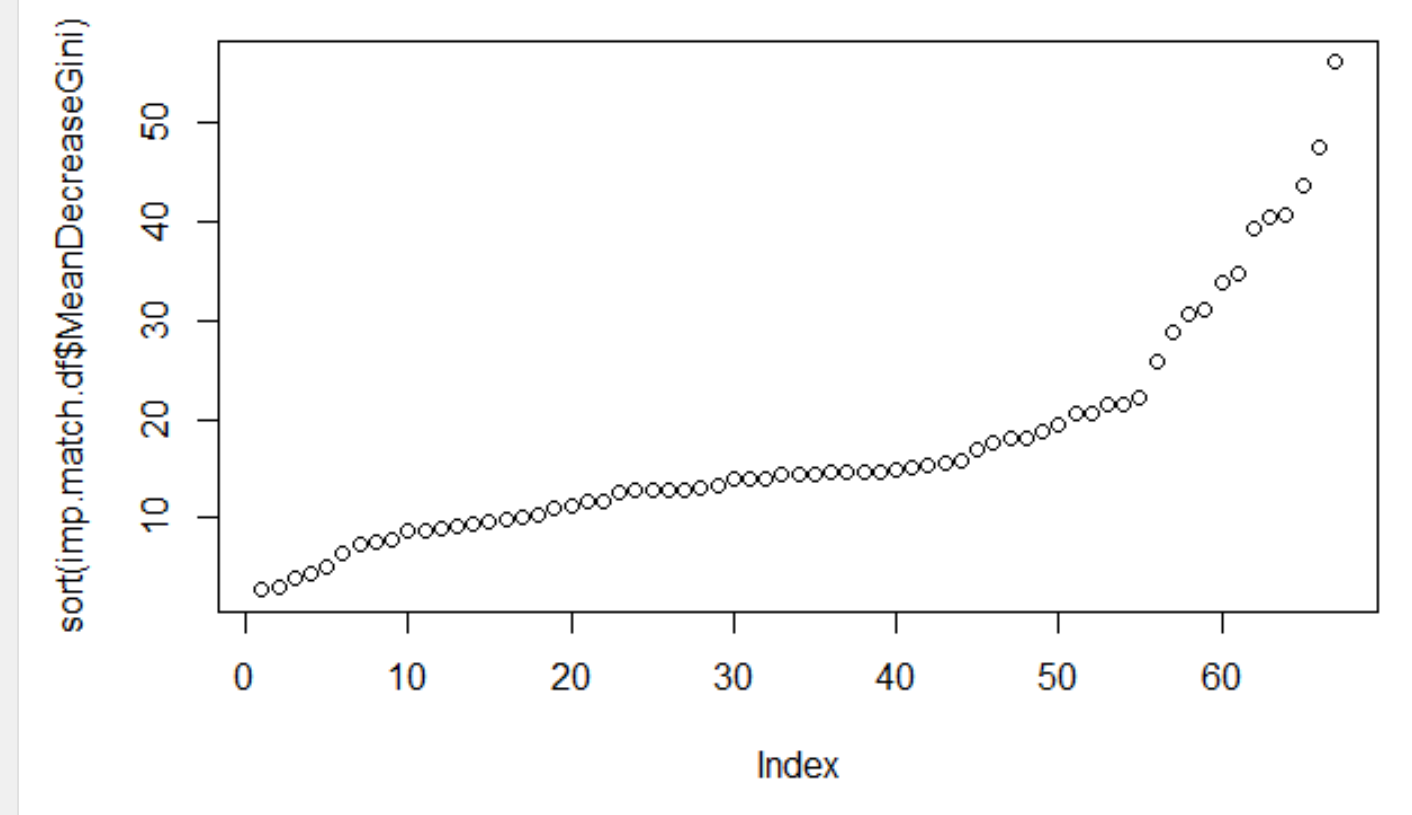
plot(sort(imp.matches.test.dates.df$MeanDecreaseGini))

```

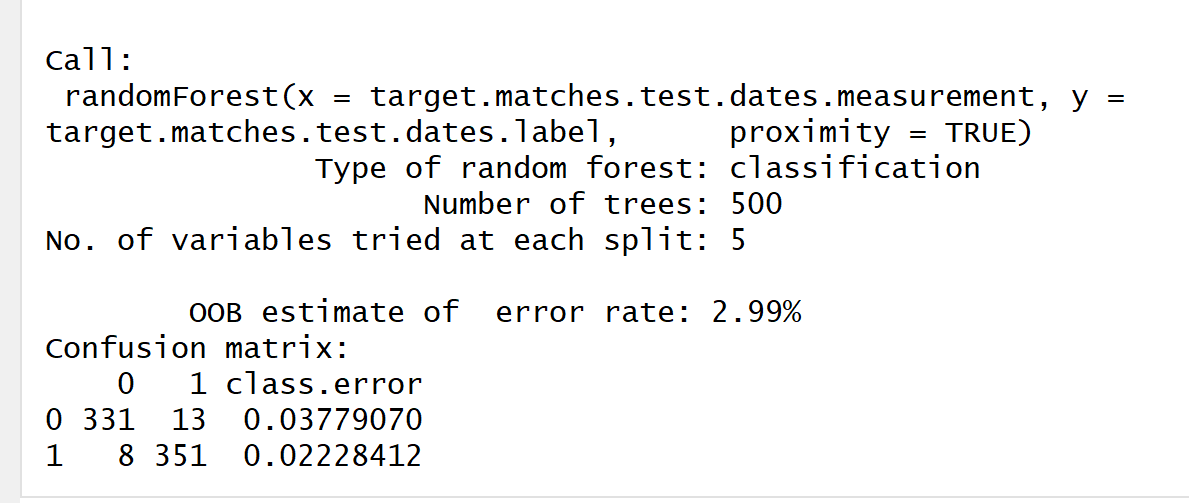
## Error rates

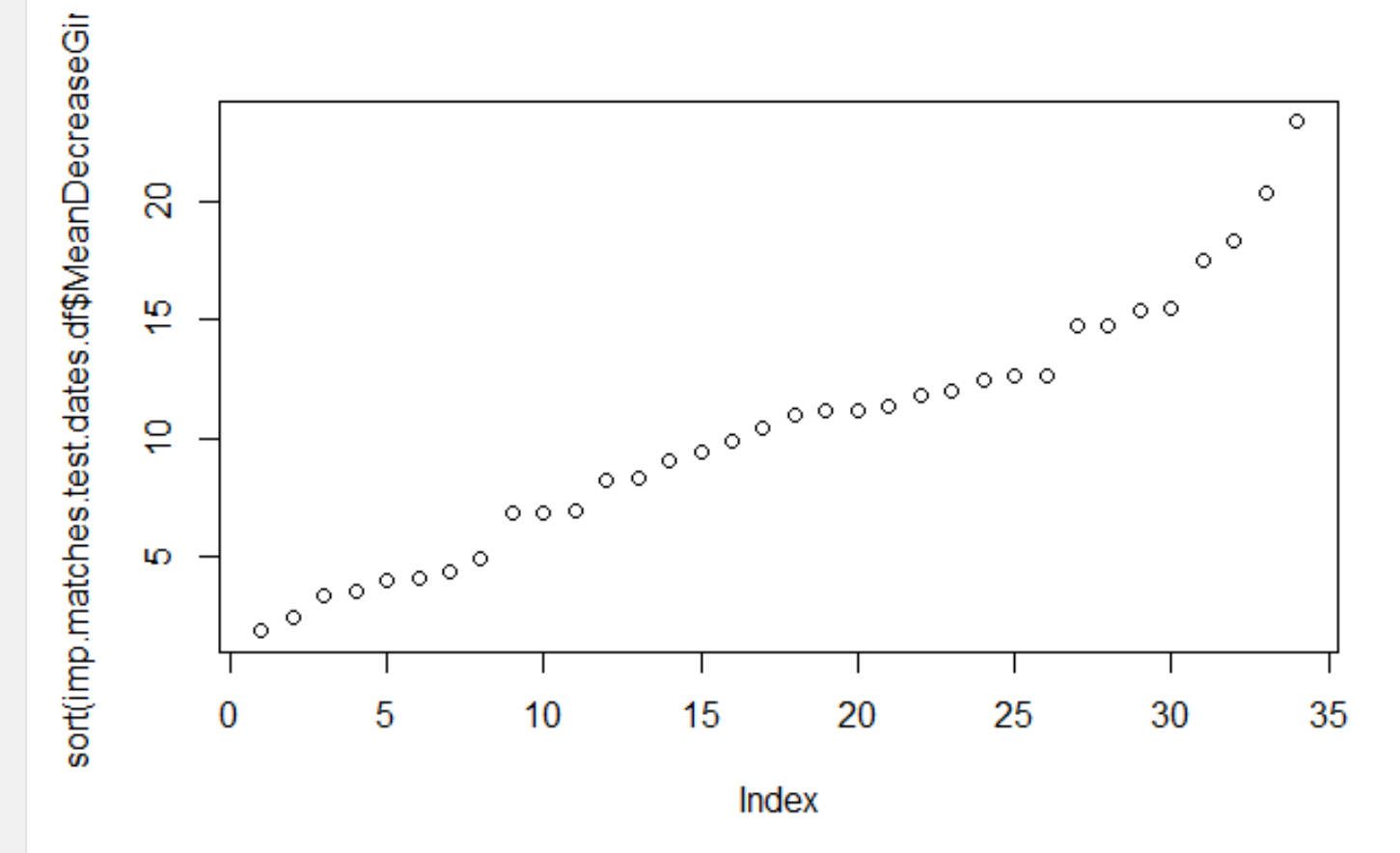
**Matches**



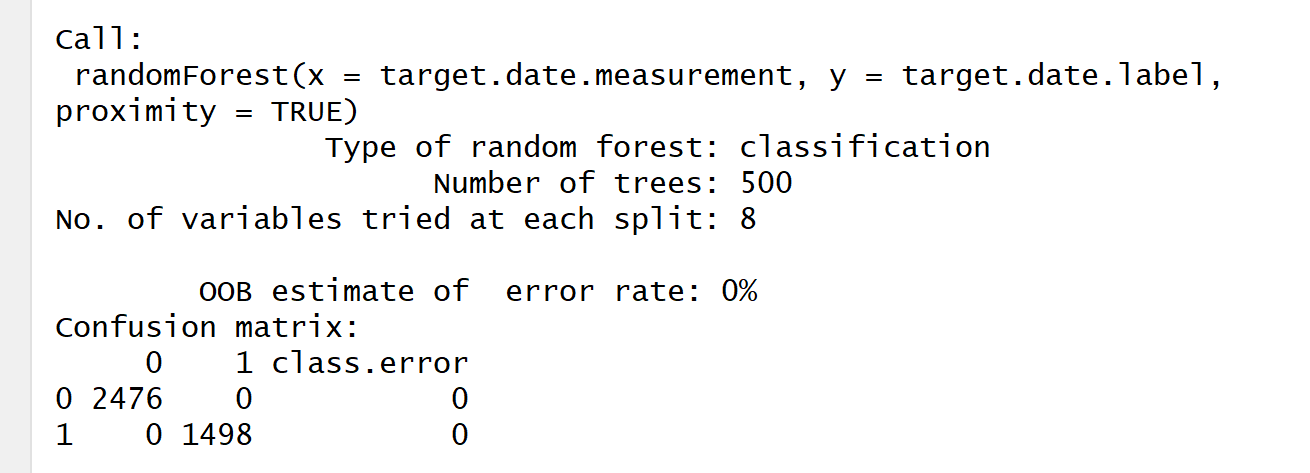


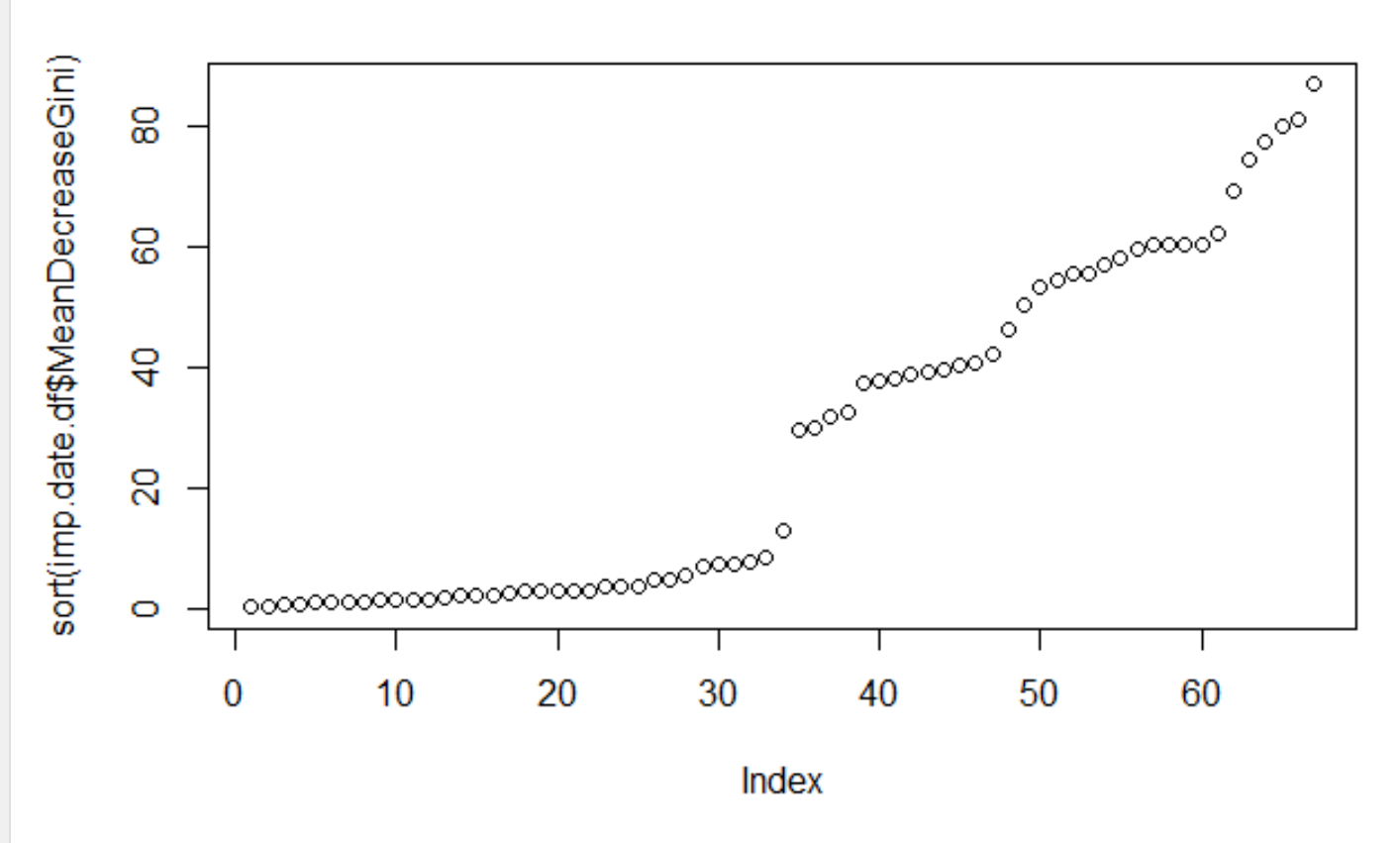
**Dates, excluding match data**





**Dates, including match data**





# Analyzing the results of the models

Comparing error rates

* The match model: 14.24%
* The date-excluding-match model: 2.99%
* The date-including-match model: 0.00%

So we can conclude that having the match data in the date model does influence the results, but doesn’t make a huge difference. The model was already pretty accurate without it.

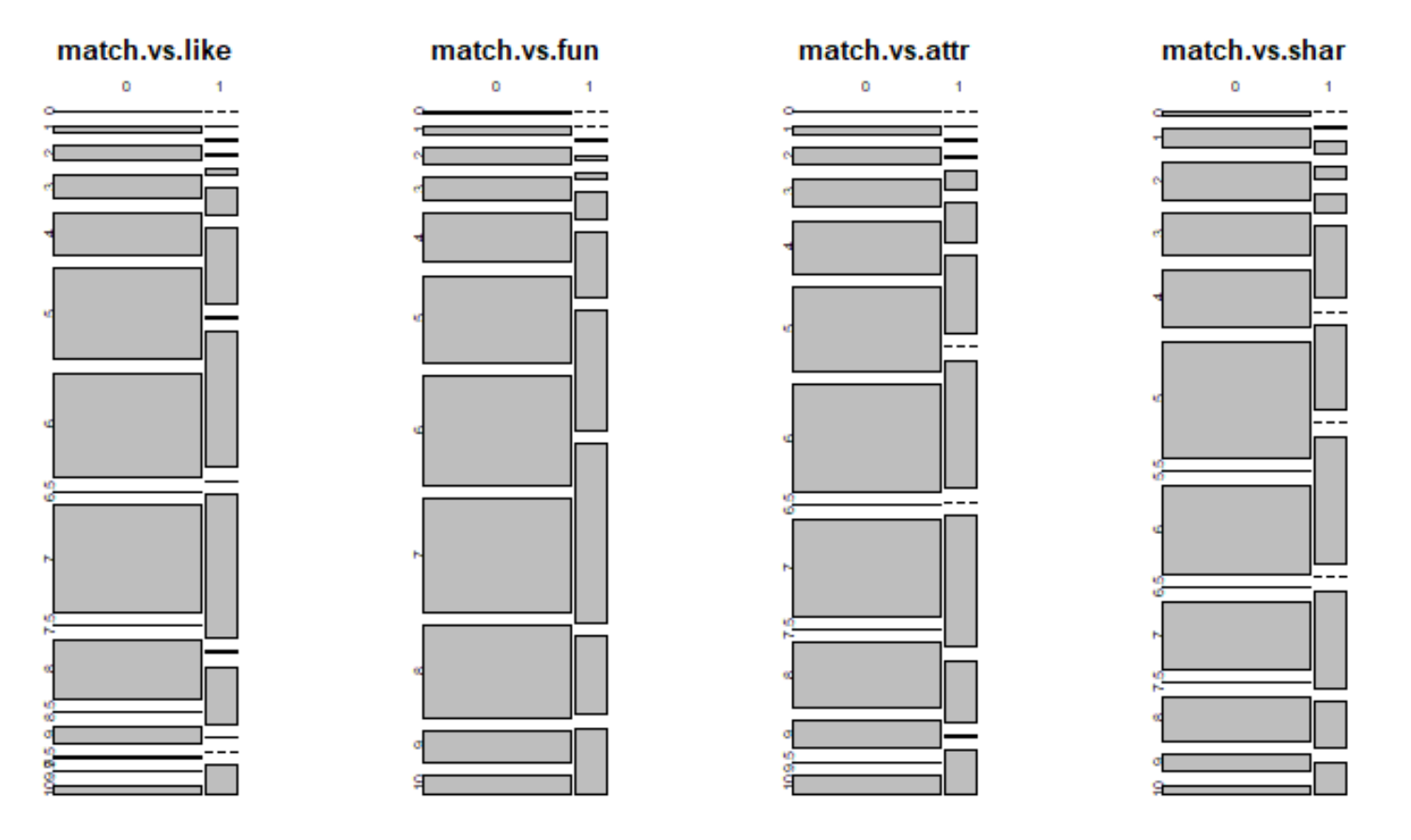
## What has the biggest influence on the match model?

Here’s the top of the list (there are many more variables in the full results):

|  |  |
| --- | --- |
| like | 56.2301 |
| like\_o | 47.5344 |
| fun | 43.6122 |
| attr\_o | 40.6127 |
| attr | 40.5161 |
| prob\_o | 39.2772 |
| shar | 34.7644 |
| prob | 33.7481 |
| fun\_o | 31.0535 |
| shar\_o | 30.6895 |
| int\_corr | 28.8682 |
| diff.self.date\_attr | 25.9002 |

In plain English, you match with someone when you like each other, can tell that the other person likes you, and think your partner is fun, attractive, and have shared interests. As a bonus, it helps if you think you are about as attractive as your date thinks you are.

Here’s a mosaic plot on a few of the match influencers. Look on the right side of each plot, where the “yes, match” people are, and notice that the top of that column has skinny boxes. That means that among matches, there were fewer low ratings for like, fun, attr, and shar.



## Advice to speed daters for getting matches

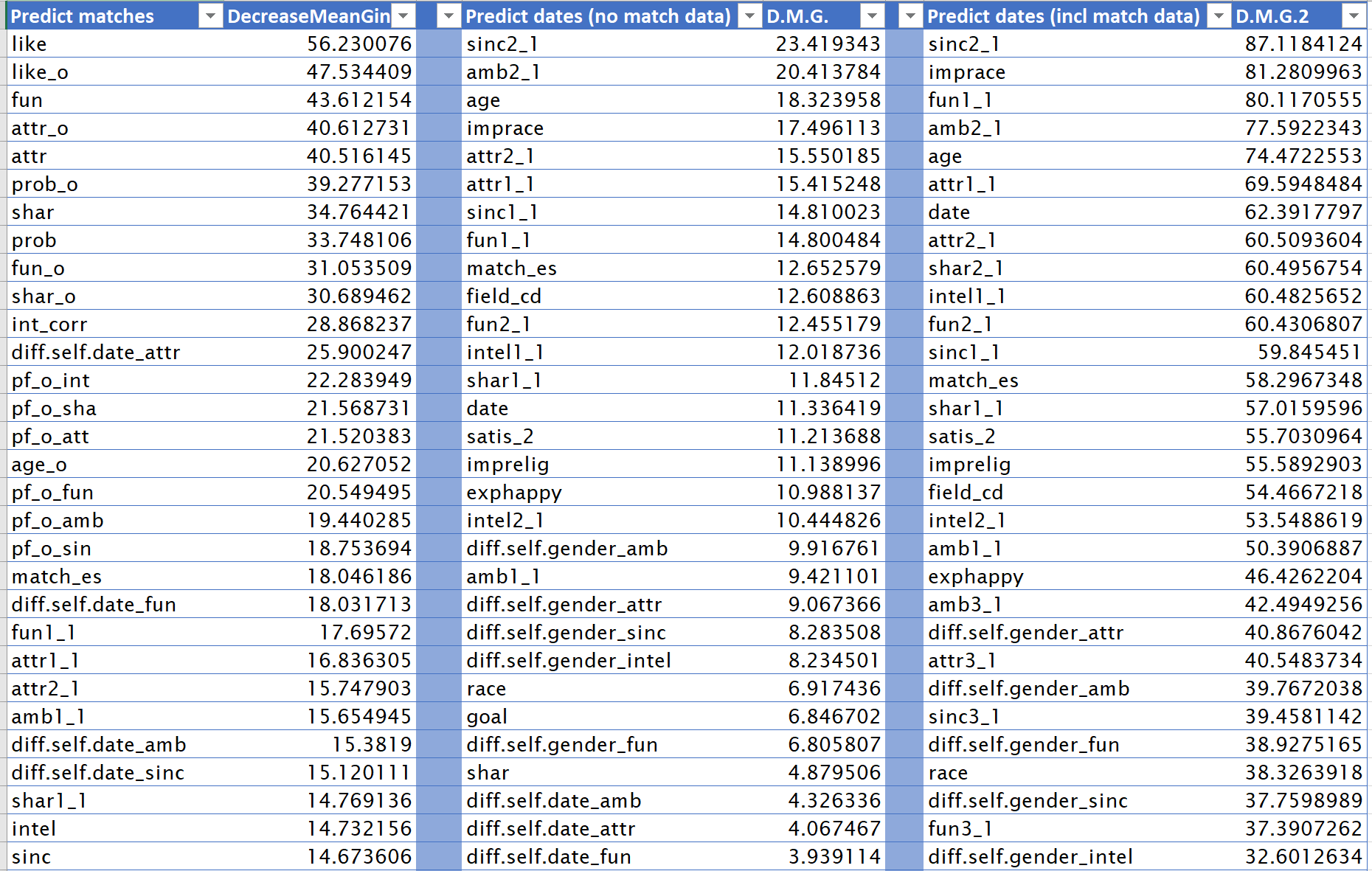
If you wanted advice before going to a speed dating event, and had the goal of getting matches, I’d tell you that it wouldn’t hurt to:

* Dress and groom yourself to show off how attractive you are
* Talk about shared interests on your speed date
* Be fun on purpose
* If you like someone, make sure they can tell you like them

## What are the biggest influences on the date models?

Dates clearly have different influencers than matches.

I’ve put the top influencers for matches and dates in this table below. The two columns on the right are the two date models. Notice that the match column has completely different variables than the two date columns. Notice also that the two date columns have the same 15 variables at the top, but in slightly different orders. Whether or not we include the match data in the date model doesn’t have a huge impact on the results.



It’s clear why people pick their matches. Those results are easy to understand. Dates are trickier.

Whether or not you go on a date is influenced by:

* Age
* how important you think race is, how important you think religion is
* how often you date, how many matches you expect to get
* how important fun, attractiveness, intelligence, and sincerity are to you
* how important you think sincerity, ambition, attractiveness, shared interests, and fun are to the opposite sex
* your major in college

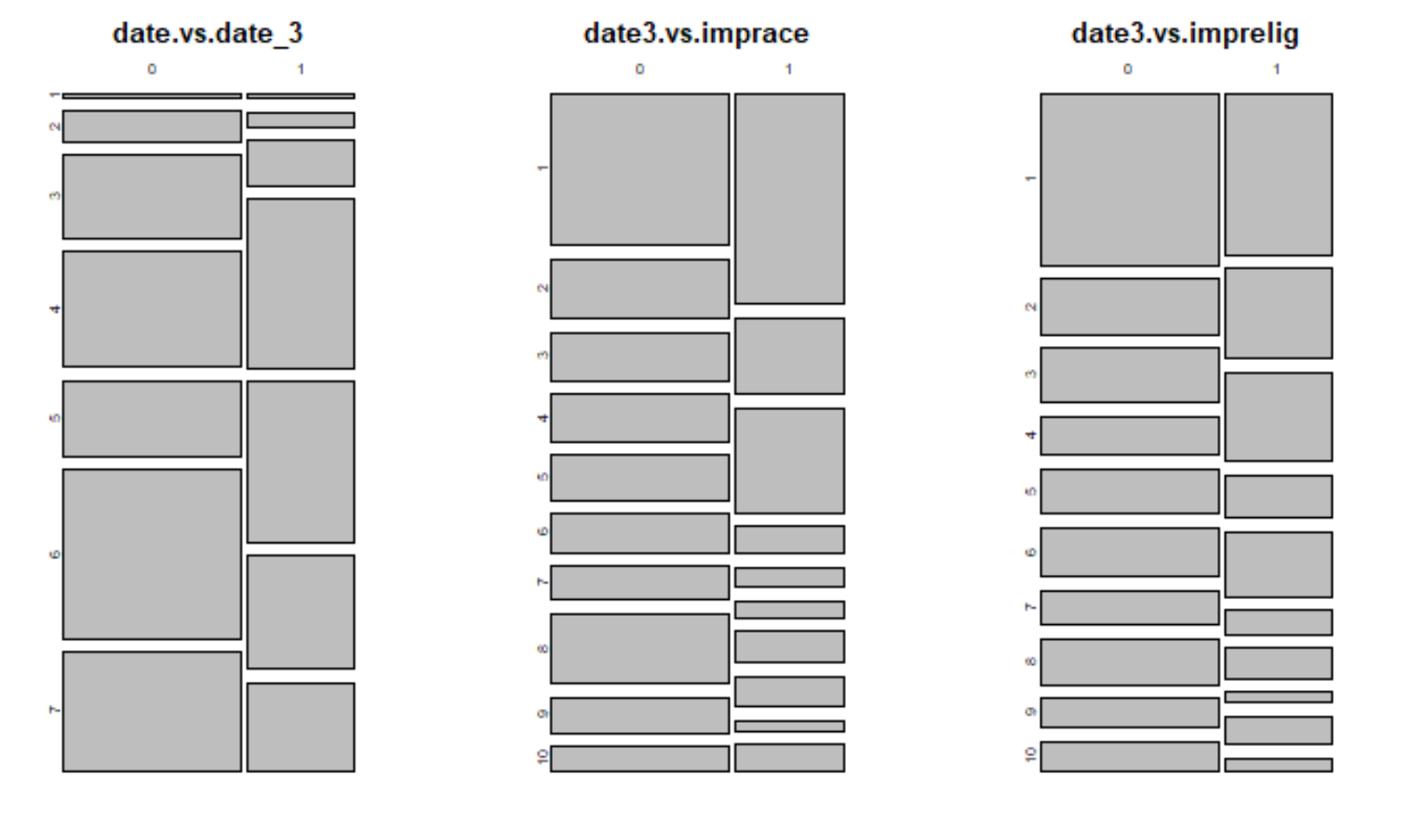
## Advice to speed daters who want dates

If you want a date, the advice is different from matches. In fact, one of the few things I can tell you for sure is that it isn’t necessary to go into speed dating with the goal of getting a date at all. Goal.date, one of the features I created, was ranked dead last as a predictor for matches.

And you don’t need to be someone who goes on dates constantly, either. Among people who went on dates after speed dating, the most common answers to “how frequently do you go on dates” were “twice a month”, “once a month”, and “several times a year”. “Several times a week” was the least popular choice, and “almost never” was the fourth most popular choice.

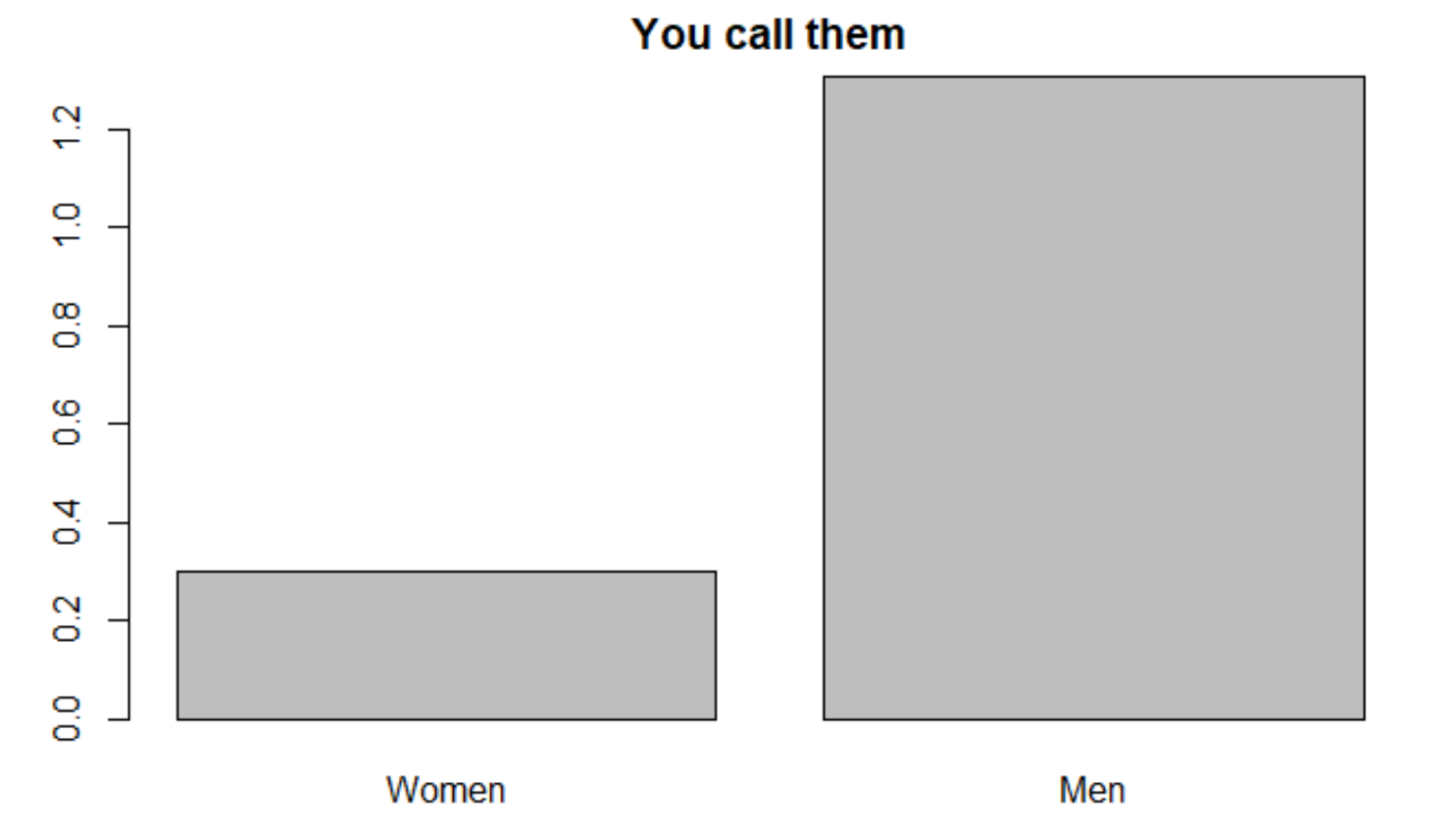
“ In general, how frequently do you go on dates? “

* + Several times a week=1
  + Twice a week=2
  + Once a week=3
  + Twice a month=4
  + Once a month=5
  + Several times a year=6
  + Almost never=7



However, if you think that race and religion aren’t that important to you in deciding whether to date someone, you may have an advantage, possibly because you have more people to consider.

One last bit of advice: if you want a date after speed dating, *call*. I didn’t include the variables you\_call and them\_cal in my models because they were too closely correlated with date\_3. These are the variables that counted how often you called a match to set up a date. Yes, men called more than women did, but both genders called. If you really want the date, you can make it happen. Remember: 100% of dates included someone who called.



# Further analysis

Although goal.date was not a useful feature, the other features I made got interesting results.

* The most overlap between matches and dates-excluding-match data were the variables comparing your rating of yourself and your partner’s rating for you; ex: diff.self.date\_attr or “do you and your partner both think you are hot?”.
* Meanwhile, variables that compare your self-ratings with the average for your gender were moderate influencers for both of the date models; ex: diff.self.gender\_attr or “are you hotter than average?”.

There are more comparisons that can be made with the current data, such as:

* Your rating of yourself versus the average your partner gave their partners
* The ratings you gave versus the ratings your partner got from other partners
* Your age versus your partner’s age plus or minus three years
* Your age versus your partner’s age plus or minus five years
* The average rating you gave yourself versus the average rating your partners gave you
* The range between the highest rating your partner gave you and the lowest

There are also things that could be made if the study were repeated. This study had a lot of data on matches because you couldn’t get your matches without filling out the questionnaire, but there was a huge drop-off on responses after dates. This is to be expected. But you could:

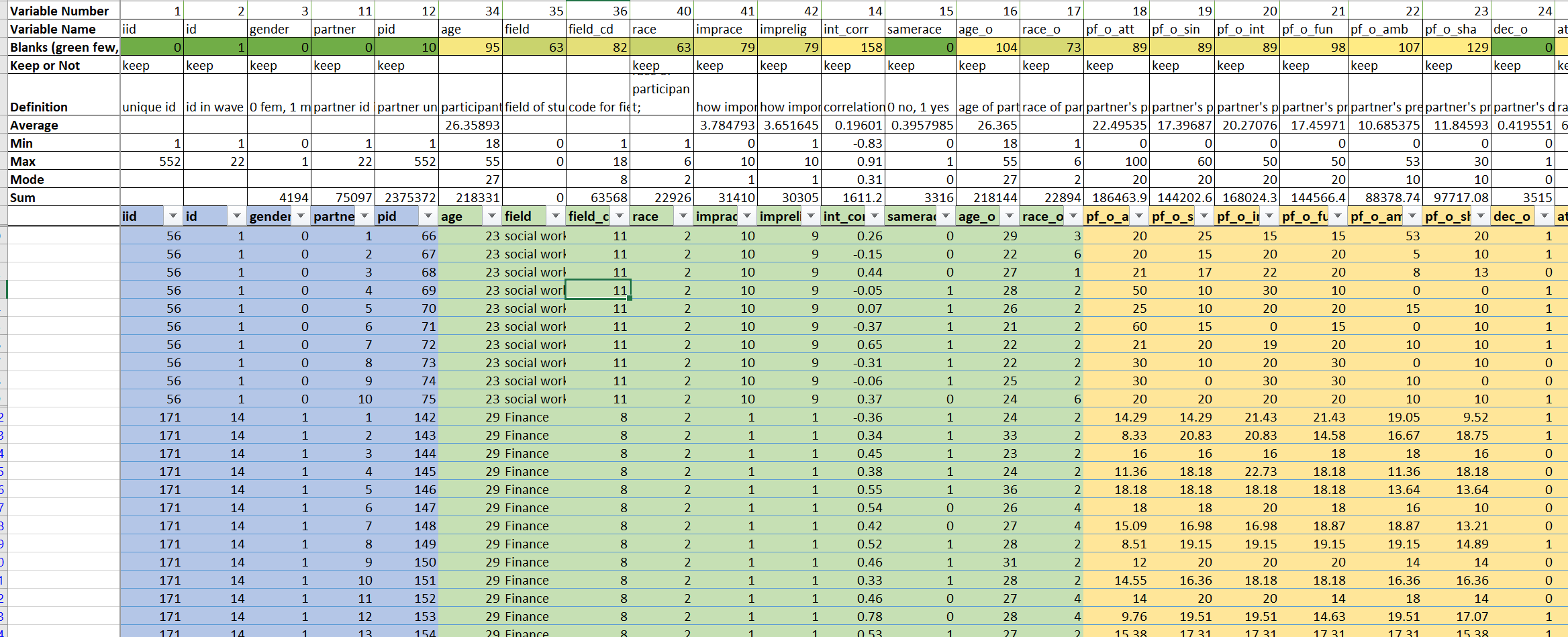
* Change the rating form to include a photo of the person and some basic information (a few of their top interests, their major). This might help the participants remember each other better, leading to a better connection.
* Automatically create email or text messages for each pair so that they could communicate after being matched up. All they would have to do is reply. This removes the stress of making a call, which might lead to more dates.
* Add questions like “why did you choose to go on a date with this person?” There may be new factors that we haven’t collected any data on yet.

Appendix

# Excel for EDA

I found it useful to use an Excel spreadsheet to learn about the data before I imported it into R. For each column I counted blanks, recorded the field definition, and did some basic statistics like min, max, average, and count. Based on that, I chose a subset of fields and imported those into R.

I recommend this step. It saved me a lot of time when I had questions about the definition for a field, whether or not people entered ratings in decimal, and how many blanks a field had.



# R code

---

title: "speeddatesnb"

output: html\_notebook

---

#----

# Import the data from csv file

# reference: http://rprogramming.net/set-working-directory-in-r/

# reference: http://rprogramming.net/read-csv-in-r/

# reference: https://stackoverflow.com/questions/11254524/omit-rows-containing-specific-column-of-na

<!-- completeFun <- function(data, desiredCols) { -->

<!-- completeVec <- complete.cases(data[, desiredCols]) -->

<!-- return(data[completeVec, ]) -->

<!-- } -->

<!-- completeFun(DF, "y") -->

#---

```{r}

wd <- setwd("/Users/Jenni/OneDrive/CLASSES/Syracuse Data Science/Intro Data Sci/Project")

dates <- read.csv(file="speeddate2.csv", header=TRUE, sep=",")

head(dates)

str(dates)

# the first column's name was corrupted, fix that

cnames <- colnames(dates)

cnames[1] <- "iid"

colnames(dates) <- cnames

str(dates)

# we are going to need to have no NAs in date\_3 (the last column), so get rid of them now

complete.specific <-function(data, desiredCols){

completeVec <-complete.cases(data[,desiredCols])

return(data[completeVec, ])

}

dates <- complete.specific(dates, "date\_3")

summary(dates)

```

#---

# look at basics of dataset

#---

```{r}

str(dates)

```

#---

# Do EDA on some columns. Make a few graphs. How many people had a match? How many had a date?

#---

# - Gender of participants

# Note that the numbers were originally even, but because I removed rows that didn't have a value in date\_3 we are seeing a response bias by gender

```{r}

numDaters <- length(dates$iid)

numDaters

numWomen <- sum(dates$gender== 1)

numMen <- numDaters - numWomen

numWomen

numMen

genderOfDaters <- c(numDaters, numWomen, numMen)

names(genderOfDaters) <- c("All participants", "Number of women", "Number of men")

genderOfDaters

```

# - Pie chart of genders

# Reference: https://www.statmethods.net/graphs/pie.html

```{r}

slices <- c(numWomen, numMen)

lbls <- c("Women", "Men")

pct <- round(slices/sum(slices)\*100)

lbls <- paste(lbls, pct) # add percents to labels

lbls <- paste(lbls,"%",sep="") # add % to labels

pie(slices, labels = lbls, main="Gender of participants") #col=rainbow(length(lbls)),

```

```{r}

# what are ages of participants

hist(dates$age)

```

```{r}

# what are races of participants

hist(dates$race)

```

```{r}

hist(dates$samerace)

```

# histogram showing number of matches for women versus men

```{r}

numMatches <- tapply(dates$match, dates$iid, sum)

numMatches[1:100]

```

```{r}

# how to split a dataframe into two sets based on a factor

# reference: https://www.statmethods.net/management/subset.html

# reference: https://stats.stackexchange.com/questions/6759/removing-duplicated-rows-data-frame-in-r

womenBool <- dates$gender==1

womenIids <- unique(dates$iid[womenBool])

womenIids[1:50]

menBool <-dates$gender==0

menIids <-unique(dates$iid[menBool])

menIids[1:50]

```

```{r}

happydf <- data.frame(dates$iid, dates$exphappy)

happydf <-unique(happydf)

hist(happydf$dates.exphappy, main = "How happy to you expect to be with the people you meet?")

```

```{r}

satis2df <- data.frame(dates$iid, dates$satis\_2)

satis2df <- unique(satis2df)

hist(satis2df$dates.satis\_2, main="Overall, how satisfied were you with the people you met?")

```

```{r}

date3df <- data.frame(dates$iid, dates$date\_3)

date3df <- unique(date3df)

date3.slices <- c(sum(date3df$dates.date\_3==1), sum(date3df$dates.date\_3==0))

date3.lbls <- c("Yes", "No")

date3.pct <- round(slices/sum(slices)\*100)

date3.lbls <- paste(date3.lbls, date3.pct)

date3.lbls <- paste(date3.lbls, "%", sep="")

pie(date3.slices, labels = date3.lbls, main="Have you been on a date with any of your matches?")

```

# What percentage of people had at least one match?

```{r}

# date3df <- data.frame(dates$iid, dates$date\_3)

# date3df <- unique(date3df)

# hist(date3df$dates.date\_3, main="Have you been on a date with any of your matches?")

#

# date3.slices <- c(sum(date3df$dates.date\_3==0), sum(date3df$dates.date\_3==1))

# date3.lbls <- c("No", "Yes")

# date3.pct <- round(slices/sum(slices)\*100)

# date3.lbls <- paste(date3.lbls, date3.pct)

# date3.lbls <- paste(date3.lbls, "%", sep="")

# pie(date3.slices, labels = date3.lbls, main="Have you been on a date with any of your matches?")

havematchdf <- data.frame(dates$iid, dates$match)

havematchdf <- unique(havematchdf)

hist(havematchdf$dates.match)

havematchdf.slices <- c(sum(havematchdf$dates.match==1), sum(havematchdf$dates.match==0))

havematchdf.lbls <- c("At least one match", "No match")

havematchdf.pct <- round(slices/sum(slices)\*100)

havematchdf.lbls <- paste(havematchdf.lbls, havematchdf.pct)

havematchdf.lbls <- paste(havematchdf.lbls, "%", sep="")

pie(havematchdf.slices, labels = havematchdf.lbls, main="What percentage of people have at least one match?")

```

# what is their goal?

# reference: https://www.statmethods.net/graphs/bar.html

```{r}

goaldf <- table(dates$gender, dates$goal)

rownames(goaldf) <- c("Women", "Men")

colnames(goaldf) <- c("Fun night", "Meet people", "Get a date", "Relationship", "Say I did it", "Other")

goaldf

barplot(goaldf, legend = c("Women", "Men"), beside = TRUE) #col=c("red", "blue"),

```

#----

# Add columns: the correlation between how the particpant rated themself and how their partner rated them on each of five criteria (attractiveness, sincerity, intelligence, fun, and ambition)

# ```{r}

# dates$corr\_self.prate\_att <- cor(dates$attr3\_1, dates$attr\_o, use = "everything")

# dates$corr\_self.prate\_sinc <- cor(dates$sinc3\_1, dates$sinc\_o, use = "pairwise.complete.obs")

# #dates$corr\_self.prate\_intel <- cor(dates$intel3\_1, dates$intel\_o, use = "all.obs") #error bc NA in data

# dates$corr\_self.prate\_fun <- cor(dates$fun3\_1, dates$fun\_o, use = "complete.obs")

# dates$corr\_self.prate\_amb <- cor(dates$amb3\_1, dates$amb\_o, use = "na.or.complete")

#

#

# dates$corr\_self.prate\_att[1:55]

# dates$corr\_self.prate\_sinc[1:55]

# dates$corr\_self.prate\_intel[1:55]

# dates$corr\_self.prate\_fun[1:55]

# dates$corr\_self.prate\_amb[1:55]

# ```

# Add column: goaldate/nodate

# reference: https://stackoverflow.com/questions/14170778/interpreting-condition-has-length-1-warning-from-if-function; ifelse(a > 0, a/sum(a), 1)

```{r}

dates$goal.date <- ifelse ((dates$goal==3 | dates$goal==4), 1, 0)

dates$goal.date[1:100]

```

# graph of goal date / not goal date

```{r}

goaldate.tab <- table(dates$date\_3, dates$goal.date)

rownames(goaldate.tab) <- c("Didn't go on a date", "Went on a date")

colnames(goaldate.tab) <- c("Didn't want a date", "Wanted a date")

goaldate.tab

barplot(goaldate.tab, col=c("darkgreen", "orange"), legend = c("Didn't go on a date", "Went on a date"), beside = TRUE)

allgoals.tab <- table(dates$date\_3, dates$goal)

allgoals.tab

rownames(allgoals.tab) <- c("No date", "Date")

colnames(allgoals.tab) <- c("Fun night", "Meet people", "Get date", "Relationship", "Say I did it", "Other")

allgoals.tab

barplot(allgoals.tab)

```

```{r}

mean.attr\_o <- mean(dates$attr\_o, na.rm=TRUE)

mean.sinc\_o <- mean(dates$sinc\_o, na.rm=TRUE)

mean.intel\_o <- mean(dates$intel\_o, na.rm=TRUE)

mean.fun\_o <- mean(dates$fun\_o, na.rm=TRUE)

mean.amb\_o <- mean(dates$amb\_o, na.rm=TRUE)

mean.ratingsFromPartner <- c(mean.attr\_o, mean.sinc\_o, mean.intel\_o, mean.fun\_o, mean.amb\_o)

mean.ratingsFromPartner

mean.MrateW.attr\_o <- mean(dates$attr\_o[womenBool], na.rm=TRUE)

mean.MrateW.sinc\_o <- mean(dates$sinc\_o[womenBool], na.rm=TRUE)

mean.MrateW.intel\_o <- mean(dates$intel\_o[womenBool], na.rm=TRUE)

mean.MrateW.fun\_o <- mean(dates$fun\_o[womenBool], na.rm=TRUE)

mean.MrateW.amb\_o <- mean(dates$amb\_o[womenBool], na.rm=TRUE)

mean.menRateWomen <- c(mean.MrateW.attr\_o, mean.MrateW.sinc\_o, mean.MrateW.intel\_o,mean.MrateW.fun\_o, mean.MrateW.amb\_o)

mean.menRateWomen

mean.WrateM.attr\_o <- mean(dates$attr\_o[menBool], na.rm=TRUE)

mean.WrateM.sinc\_o <- mean(dates$sinc\_o[menBool], na.rm=TRUE)

mean.WrateM.intel\_o <- mean(dates$intel\_o[menBool], na.rm=TRUE)

mean.WrateM.fun\_o <- mean(dates$fun\_o[menBool], na.rm=TRUE)

mean.WrateM.amb\_o <- mean(dates$amb\_o[menBool], na.rm=TRUE)

mean.womenRateMen <- c(mean.WrateM.attr\_o, mean.WrateM.sinc\_o, mean.WrateM.intel\_o, mean.WrateM.fun\_o, mean.WrateM.amb\_o)

mean.womenRateMen

par(mfrow= c(1,3))

barplot(mean.ratingsFromPartner, main = "Average ratings")

barplot(mean.menRateWomen, main = "Men rate women")

barplot(mean.womenRateMen, main = "Women rate men")

```

#---

# Add column, difference between self score and mean given to gender

# Add column, difference between self score and rating from your date

#---

```{r}

dates$diff.self.gender\_attr <- ifelse(dates$gender==0 | dates$gender==1, (dates$attr3\_1 - mean.WrateM.attr\_o), (dates$attr3\_1 - mean.MrateW.attr\_o))

str(dates$diff.self.gender\_attr)

dates$diff.self.gender\_sinc <- ifelse(dates$gender==0 | dates$gender==1, (dates$sinc3\_1 - mean.WrateM.sinc\_o), (dates$sinc3\_1 - mean.MrateW.sinc\_o))

str(dates$diff.self.gender\_sinc)

dates$diff.self.gender\_intel <- ifelse(dates$gender==0 | dates$gender==1, (dates$intel3\_1 - mean.WrateM.intel\_o), (dates$intel3\_1 - mean.MrateW.intel\_o))

str(dates$diff.self.gender\_intel)

dates$diff.self.gender\_fun <- ifelse(dates$gender==0 | dates$gender==1, (dates$fun3\_1 - mean.WrateM.fun\_o), (dates$fun3\_1 - mean.MrateW.fun\_o))

str(dates$diff.self.gender\_fun)

dates$diff.self.gender\_amb <- ifelse(dates$gender==0 | dates$gender==1, (dates$amb3\_1 - mean.WrateM.amb\_o), (dates$amb3\_1 - mean.MrateW.amb\_o))

str(dates$diff.self.gender\_amb)

dates$diff.self.date\_attr <- dates$attr3\_1 - dates$attr\_o

str(dates$diff.self.date\_attr)

dates$diff.self.date\_sinc <- dates$sinc3\_1 - dates$sinc\_o

str(dates$diff.self.date\_sinc)

dates$diff.self.date\_intel <- dates$intel3\_1 - dates$intel\_o

str(dates$diff.self.date\_intel)

dates$diff.self.date\_fun <- dates$fun3\_1 - dates$fun\_o

str(dates$diff.self.date\_fun)

dates$diff.self.date\_amb <- dates$amb3\_1 - dates$amb\_o

str(dates$diff.self.date\_amb)

str(dates)

summary(dates[50:78])

```

```{r}

install.packages("randomForest")

library(randomForest)

```

# run random forest on dates$date\_3: what leads to a date?

# remove dates$match (because we know 100% of dates were also a match)

```{r}

# reference: https://github.com/johnsanterre/DataScience/commit/853dc62ea357432de8570e11e221d754412a5328

# example from class

# +#library(randomForest)

# +#M = iris[,-5] # messurment ddata

# +#L = iris[,5] # Species categories, discrete labels Levels == setosa versicolor virginica

# +#iris.rf <- randomForest(M, L, prox=TRUE)

# +#hist(iris[,1]+iris[,2])

# +#hist(iris[,1]-iris[,2], breaks=30)

# +#

# +#predict(iris.rf, iris[1,-5])

# +#plot

# create measurement and label variables

colnames(dates)

# IDs are not predictive: iid = 1, id = 2, partner=4, pid=5,

# Their major throws an error for too many factor levels: field=7,

# A match comes from two yes decisions: dec\_o=22, match=52, dec=59,

# A date happens after one of you calls: you\_call=65, them\_cal=66,

# We are trying to predict date\_3: date\_3=67

remove.vars <-c(67, 66, 65, 59, 52, 22, 7,5,4,2,1)

target.date.measurement <- dates[,-remove.vars] #remove these fields

target.date.measurement[is.na(target.date.measurement)] <- 0

colnames(target.date.measurement)

target.date.label <- as.factor(dates[,67])

# run random forest

dates.date.rf <-randomForest(target.date.measurement, target.date.label, prox=TRUE)

dates.date.rf

dates.date.rf.importance <- importance(dates.date.rf)

str(dates.date.rf.importance)

dates.date.rf.importance

plot(dates.date.rf.importance)

imp.date.df <- data.frame(rownames(dates.date.rf.importance), dates.date.rf.importance)

sorted.imp.date <- imp.date.df[order(-imp.date.df$MeanDecreaseGini) ,]

sorted.imp.date

plot(sort(imp.date.df$MeanDecreaseGini))

```

# random forest on matches (matches happen at the end of a speed date, but before dates in the real world)

```{r}

# random forest on matches (matches happen at the end of a speed date, but before dates in the real world)

# create measurement and label variables

colnames(dates)

remove.match.vars <-c(67, 66, 65, 59, 52, 22, 7,5,4,2,1)

# IDs are not predictive: iid = 1, id = 2, partner=4, pid=5,

# Their major throws an error for too many factor levels: field=7,

# A match comes from two yes decisions: dec\_o=22, dec=59,

# We are trying to predict match: match=52,

# These things happen AFTER what we want to predict: you\_call=65, them\_cal=66, date\_3=67

target.match.measurement <- dates[,-remove.match.vars] #remove these fields

target.match.measurement[is.na(target.match.measurement)] <- 0

colnames(target.match.measurement)

target.match.label <- as.factor(dates[,52]) #predict this field

# run random forest

dates.match.rf <-randomForest(target.match.measurement, target.match.label, prox=TRUE)

dates.match.rf

dates.match.rf.importance <- importance(dates.match.rf)

str(dates.match.rf.importance)

dates.match.rf.importance

plot(dates.match.rf.importance)

hist(dates.match.rf.importance)

# looking just at the low values, in order

# keep plot sort, change where cut off is for MDG

imp.match.df <- data.frame(rownames(dates.match.rf.importance), dates.match.rf.importance)

sorted.imp.match <- imp.match.df[order(-imp.match.df$MeanDecreaseGini) ,]

sorted.imp.match

plot(sort(imp.match.df$MeanDecreaseGini))

```

```{r}

names(dates)

with.matches <- subset(dates, match==1)

with.matches.test.dates <- with.matches[, c(1, 3, 6,8,9,10,11,29:39, 51,56,57,58,63:78)]

with.matches.test.dates

names(dates)

names(with.matches.test.dates)

```

```{r}

#random forest on people with matches to see what predicts a date

# these qualities are per person, not per pair, no partner ratings included

# create measurement and label variables

colnames(with.matches.test.dates)

remove.matches.test.dates.vars <-c(27,26,25,1)

# remove ID, you\_call, them\_cal, date\_3

target.matches.test.dates.measurement <- with.matches.test.dates[,-remove.matches.test.dates.vars] #remove these fields

target.matches.test.dates.measurement[is.na(target.matches.test.dates.measurement)] <- 0

colnames(target.matches.test.dates.measurement)

target.matches.test.dates.label <- as.factor(with.matches.test.dates[,27]) #predict this field

# run random forest

dates.matches.test.dates.rf <-randomForest(target.matches.test.dates.measurement, target.matches.test.dates.label, prox=TRUE)

dates.matches.test.dates.rf

dates.matches.test.dates.rf.importance <- importance(dates.matches.test.dates.rf)

str(dates.matches.test.dates.rf.importance)

dates.matches.test.dates.rf.importance

plot(dates.matches.test.dates.rf.importance)

hist(dates.matches.test.dates.rf.importance)

# sort table of MeanDecreaseGini and plot importance

imp.matches.test.dates.df <- data.frame(rownames(dates.matches.test.dates.rf.importance), dates.matches.test.dates.rf.importance)

sorted.m.t.d <- imp.matches.test.dates.df[order(-imp.matches.test.dates.df$MeanDecreaseGini) ,]

sorted.m.t.d

plot(sort(imp.matches.test.dates.df$MeanDecreaseGini))

```

# graphs to look at influential variables

```{r}

print("Match vs. Like"); match.vs.like <- table(dates$match, dates$like); match.vs.like

print("Match vs. Fun"); match.vs.fun <- table(dates$match, dates$fun); match.vs.fun

print("Match vs. Attr"); match.vs.attr <- table(dates$match, dates$attr); match.vs.attr

print("Match vs. Shar"); match.vs.shar <- table(dates$match, dates$shar); match.vs.shar

#Less interesting

# print("Match vs. prob"); match.vs.prob <- table(dates$match, dates$prob); match.vs.prob; mosaicplot(match.vs.prob)

par(mfrow=c(1,4))

mosaicplot(match.vs.like)

mosaicplot(match.vs.fun)

mosaicplot(match.vs.attr)

mosaicplot(match.vs.shar)

```

```{r}

# the plots of the importance results, looking to see if there is a dropoff that shows some variables are better influencers than others

plot(sort(imp.match.df$MeanDecreaseGini))

plot(sort(imp.matches.test.dates.df$MeanDecreaseGini))

plot(sort(imp.date.df$MeanDecreaseGini))

```

#Looking at some of the variables that influence date\_3

```{r}

#The levels for date are: 1 high 7 low

print("Date (how often you date) vs. date\_3"); date.vs.date\_3 <- table(dates$date\_3, dates$date)

date.vs.date\_3

print("Date\_3 vs. imprace"); date3.vs.imprace <- table(dates$date\_3, dates$imprace)

date3.vs.imprace

print("Date\_3 vs. imprelig"); date3.vs.imprelig <- table(dates$date\_3, dates$imprelig)

date3.vs.imprelig

par(mfrow=c(1,3))

mosaicplot(date.vs.date\_3)

mosaicplot(date3.vs.imprace)

mosaicplot(date3.vs.imprelig)

#These weren't very interesting

# print("Date\_3 vs. age"); date3.vs.age <- table(dates$date\_3, dates$age)

# date3.vs.age; mosaicplot(date3.vs.age)

# print("Date\_3 vs. match\_es"); date3.vs.match\_es <- table(dates$date\_3, dates$match\_es)

# date3.vs.match\_es; mosaicplot(date3.vs.match\_es)

# print("Date\_3 vs. exphappy"); date3.vs.exphappy <- table(dates$date\_3, dates$exphappy)

# date3.vs.exphappy; mosaicplot(date3.vs.exphappy)

# print("Date\_3 vs. sinc2\_1"); date3.vs.sinc21 <- table(dates$date\_3, dates$sinc2\_1); date3.vs.sinc21; mosaicplot(date3.vs.sinc21)

# print("Date\_3 vs. amb2\_1"); date3.vs.amb21 <- table(dates$date\_3, dates$amb2\_1); date3.vs.amb21; mosaicplot(date3.vs.amb21)

# print("Date\_3 vs. attr2\_1"); date3.vs.attr21 <- table(dates$date\_3, dates$attr2\_1); date3.vs.attr21; mosaicplot(date3.vs.attr21)

# print("Date\_3 vs. age"); date3.vs.age <- table(dates$date\_3, dates$age); date3.vs.age; mosaicplot(date3.vs.age)

```

```{r}

calls <- data.frame(dates$iid, dates$gender, dates$you\_call, dates$them\_cal)

calls.pp <-unique(calls)

calls.pp[1:20 ,]

avg.youcall.gender <- tapply(calls.pp$dates.you\_call, calls.pp$dates.gender, mean)

avg.themcal.gender <- tapply(calls.pp$dates.them\_cal, calls.pp$dates.gender, mean)

rownames(avg.youcall.gender) <- c("Women", "Men")

rownames(avg.themcal.gender) <- c("Women", "Men")

barplot(avg.youcall.gender, main="You call them")

barplot(avg.themcal.gender, main="They call you")

```