Cleaning 2022 Magic Data

Brittney Hernandez

2023-04-02

Introduction

This R Markdown file contains the information and code for cleaning the MAGIC Mentor Network data from Year 3 of the program which ran in the 2021-2022 academic year.

Load the data

These data are from the end of the year evaluation survey. They are downloaded using a Qualtrics API.

```
# import data
# qualtrics api key
qualtrics_api_credentials(api_key = "Jc9vraFFrDBsaiJQGjvn32yjtMOrehmT9uYWjkT6",
                          base_url = "uconn.iad1.qualtrics.com", install = TRUE, overwrite = TRUE)
```

```
## Your original .Renviron will be backed up and stored in your R HOME directory if needed.
```

```
## Your Qualtrics key and base URL have been stored in your .Renviron.
## To use now, restart R or run `readRenviron("~/.Renviron")`
```

```
# 2022 survey
surveyid <- "SV_6DLdZQw7P8LpHW6"
data2022 <- fetch_survey(surveyid,
                         force_request = TRUE, # forces to make a new request
                         label = TRUE, convert = TRUE) # export survey responses as choice text
```

```
##   |                                                                              |
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   StartDate = col_datetime(format = ""),
##   EndDate = col_datetime(format = ""),
##   Progress = col_double(),
##   `Duration (in seconds)` = col_double(),
##   Finished = col_logical(),
##   RecordedDate = col_datetime(format = ""),
##   RecipientLastName = col_logical(),
##   RecipientFirstName = col_logical(),
```

```
##   RecipientEmail = col_logical(),
##   ExternalReference = col_logical(),
##   LocationLatitude = col_double(),
##   LocationLongitude = col_double(),
##   gradfac_3_TEXT = col_logical(),
##   mentee_orienthelp_3_TEXT = col_logical(),
##   mentee_filters_3 = col_logical(),
##   mentee_filters_9 = col_logical(),
##   mentee_mq_matching_5_TEXT = col_logical(),
##   mentee_nmentor = col_double(),
##   mentee_nmeetings = col_double(),
##   mentee_mq_topics_7 = col_logical()
##   # ... with 14 more columns
## )
## i Use `spec()` for the full column specifications.
```

```
# convert from tibble to data frame
data2022 <- as.data.frame(data2022)

# save the survey questions and ids
questions2022 <- as.data.frame(survey_questions(surveyid))
```

Remove any test cases

```
# only keep cases where status doesn't equal survey preview
data2022 <- data2022[which(data2022$Status != "Survey Preview"),]
```

Clean name variable

Combine the drop down and free response name columns.

```
# rename and restructure name variable
# set name = the response from name dropdown
names(data2022)[names(data2022) == "intro_name-drop"] <- "name"


data2022$name[which(is.na(data2022$name) == TRUE | data2022$name == "Not listed")] <- {
  data2022$`intro_name-text`[which(is.na(data2022$name) == TRUE | data2022$name == "Not listed")]}
```

Some respondents vary in how they report their names. Sometimes they use accents or nicknames. Therefore, create a new variable called name_clean which strips any accents from respondents names to use for matching across years.

```
# use a pre-made function to clean names
source("functions.R")

data2022$name_clean <- cleanName(data2022$name)
```

Remove any cases where the name is missing or says test.

```r
data2022 <- data2022[which(is.na(data2022$name) == FALSE),]

# only keep cases where test is not true
data2022 <- data2022[which(str_detect(data2022$name, "test") == FALSE),]
data2022 <- data2022[which(str_detect(data2022$name, "Test") == FALSE),]
```

Find duplicates

First remove any exact duplicates.

```r
# remove exact duplicates
data2022 <- data2022[which(duplicated(data2022) == FALSE),]
```

Next, remove any duplicated names. The code below creates a list of matching names and then extracts a list of unique elements.

```r
# get duplicate names
duplicate_names <- data2022$name_clean[which(duplicated(data2022$name_clean) == TRUE)]
duplicate_names <- unique(duplicate_names)
```

For the duplicate names, take the most recent case and try to input any missing information from the prior case.

1. initiate a for loop
2. create a subset of just the duplicate data
3. get the number of rows in the subset (how many times was this name duplicated)
4. create a subset of just the last case (the data is ordered by timestamp so the last case is the most recent)
5. get a list of the missing column names from the last case
6. if there are missing columns (i.e., if it is not character(0))
7. then go through each column
8. replace the missing column with whatever is in the prior case
9. reset row names
10. get the row numbers for the cases with the duplicated name
11. remove any cases with the duplicated name
12. add back just the last case

```r
for (name in duplicate_names) { # 1
  subset <- data2022[which(data2022$name_clean == name),] # 2
  rows <- nrow(subset) # 3
  last_case <- subset[rows, ] # 4

  missing_columns <- names(which(colSums(is.na(last_case)) > 0)) # 5

  if (identical(missing_columns, character(0)) != TRUE) { # 6
    for (column in missing_columns) { # 7
      last_case[1, column] <- subset[rows - 1, column] # 8
    }
  }

  rownames(data2022) <- c() # 9
```

```
  row_index <- as.numeric(rownames(data2022)[which(data2022$name_clean == name)]) # 10
  data2022 <- data2022[-c(row_index), ] # 11
  data2022 <- rbind(data2022, last_case) # 12
}
```

Subset the columns

The data no longer needs the timestamp variable. The columns are also renamed so when they are combined with other data sets, it is clear that this information comes from the 2022 data.

```
# keep end date + the columns after the quatrics metadata
data2022 <- data2022[, c(2, 18:ncol(data2022))]
```

Format the columns

```
# format character
data2022 %>% mutate(across(where(is.factor), as.character)) -> data2022
```

Set the match column

Match should reflect the match status for the upcoming year. First set the match to what their current status is.

```
# assign match 1 = mentor, 0 = mentee
data2022$match <- NA
data2022$match[which(data2022$intro_part == "I participated as a mentor")] <- 1
data2022$match[which(data2022$intro_part == "I participated as a mentee")] <- 0

# if they're signing up, mentor or mentee?
data2022$match[which(data2022$intro_parttype == "Mentor")] <- 1
data2022$match[which(data2022$intro_parttype == "Mentee")] <- 0
```

Create a single continue column, which combines the way the question was phrased for mentors (cont1) and mentees (cont2). The difference in phrasing is because mentees can continue as a mentor if they matriculate through the program.

```
# rename cont1 to cont
names(data2022)[names(data2022) == "common_cont1"] <- "cont"

# if yes is detected in cont2, then update cont
data2022$cont[grep("Yes", data2022$common_cont2, ignore.case = TRUE)] <- "Yes"
data2022$cont[grep("No", data2022$common_cont2, ignore.case = TRUE)] <- "No"
```

```
# if mentor or mentee is detected in cont2, then update match
data2022$match[grep("Mentor", data2022$common_cont2, ignore.case = TRUE)] <- 1
data2022$match[grep("Mentee", data2022$common_cont2, ignore.case = TRUE)] <- 0
```

Recode meeting method

```
data2022$q228 <- paste(data2022$Q228_1, data2022$Q228_2, data2022$Q228_3, data2022$Q228_4,
                       data2022$Q228_5, data2022$Q228_6, data2022$Q228_7, data2022$Q228_7_TEXT,
                       sep = ", ")

# remove combined na's
data2022$q228 <- str_remove_all(data2022$q228, ", NA")
data2022$q228 <- str_remove_all(data2022$q228, "NA, ")
data2022$q228 <- str_remove_all(data2022$q228, "NA")
```

Recode identity categories

Race/ethnicity

```
data2022$black <- NA
data2022$black[which(is.na(data2022$menteei_race_3) == FALSE)] <- 1
data2022$black[grep("black", data2022$menteei_race_7_TEXT, ignore.case = TRUE)] <- 1
data2022$black[grep("african", data2022$menteei_race_7_TEXT, ignore.case = TRUE)] <- 1
```

```
data2022$hispanic <- NA
data2022$hispanic[which(is.na(data2022$menteei_race_6) == FALSE)] <- 1
data2022$hispanic[grep("hispanic", data2022$menteei_race_7_TEXT, ignore.case = TRUE)] <- 1
data2022$hispanic[grep("latin", data2022$menteei_race_7_TEXT, ignore.case = TRUE)] <- 1
data2022$hispanic[grep("spanish", data2022$menteei_race_7_TEXT, ignore.case = TRUE)] <- 1
```

```
# create a combined race variable
data2022$race <- paste(data2022$menteei_race_1, data2022$menteei_race_2, data2022$menteei_race_3,
                       data2022$menteei_race_4, data2022$menteei_race_5, data2022$menteei_race_6,
                       data2022$menteei_race_7, data2022$menteei_race_7_TEXT, sep = ", ")

# remove combined na's
 data2022$race <- str_remove_all( data2022$race, ", NA")
 data2022$race <- str_remove_all( data2022$race, "NA, ")
 data2022$race <- str_remove_all(data2022$race, "NA")
```

First generation college student

```
data2022$firstgen <- NA

# parent/guardian 1
data2022$firstgen[which(data2022$menteei_firstgen_1 == "No college")] <- 1
data2022$firstgen[which(data2022$menteei_firstgen_1 == "Some college")] <- 1

# parent/guardian 2
data2022$firstgen[which(data2022$menteei_firstgen_2 == "No college")] <- 1
data2022$firstgen[which(data2022$menteei_firstgen_2 == "Some college")] <- 1
```

```
# this will overwrite the codes so if mentees had at least 1 parent/guardian
# graduate from college then they are not first gen
data2022$firstgen[which(data2022$menteei_firstgen_1 == "Graduated college")] <- 0
data2022$firstgen[which(data2022$menteei_firstgen_2 == "Graduated college")] <- 0
```

LGBTQIA+

```
data2022$lgbtqia <- NA
data2022$lgbtqia[which(data2022$menteei_gender == "Non-binary")] <- 1
data2022$lgbtqia[which(is.na(data2022$menteei_gender_4_TEXT) == FALSE)] <- 1
```

Disability

```
data2022$disability <- NA
data2022$disability[which(data2022$menteei_disability == "Yes")] <- 1
data2022$disability[which(data2022$menteei_disability == "Prefer to self-describe ")] <- 1
data2022$disability[which(data2022$menteei_disability == "No")] <- 0
```

Combine broad interest data

```
data2022$broad <- paste(data2022$ra_broad_1, data2022$ra_broad_13, data2022$ra_broad_14,
                        data2022$ra_broad_15, data2022$ra_broad_16, data2022$ra_broad_17,
                        data2022$ra_broad_18, data2022$ra_broad_19, data2022$ra_broad_20,
                        data2022$ra_broad_21, data2022$ra_broad_22, data2022$ra_broad_23,
                        data2022$ra_broad_24, data2022$ra_broad_24_TEXT, sep = ", ")

# remove combined na's
data2022$broad <- str_remove_all(data2022$broad, ", NA")
data2022$broad <- str_remove_all(data2022$broad, "NA, ")
data2022$broad <- str_remove_all(data2022$broad, "NA")
```

Subset the data

Mentor data

```
mentor2022 <- data2022[which(data2022$intro_part == "I participated as a mentor"),]

# get columns where the number of NA's is equal to the number of rows
empty_columns <- which(colSums(is.na(mentor2022))==nrow(mentor2022))

# remove empty columns
mentor2022 <- mentor2022[-empty_columns]
```

Mentee data

```
mentee2022 <- data2022[which(data2022$intro_part == "I participated as a mentee"),]

# get columns where the number of NA's is equal to the number of rows
empty_columns <- which(colSums(is.na(mentee2022))==nrow(mentee2022))

# remove empty columns
mentee2022 <- mentee2022[-empty_columns]
```

Interest data

```
mentor_interest2022 <- data2022[which(data2022$intro_parttype == "Mentor"),]

# get columns where the number of NA's is equal to the number of rows
empty_columns <- which(colSums(is.na(mentor_interest2022))==nrow(mentor_interest2022))

# remove empty columns
mentor_interest2022 <- mentor_interest2022[-empty_columns]
```

Mentor

```
mentee_interest2022 <- data2022[which(data2022$intro_parttype == "Mentee"),]

# get columns where the number of NA's is equal to the number of rows
empty_columns <- which(colSums(is.na(mentee_interest2022))==nrow(mentee_interest2022))

# remove empty columns
mentee_interest2022 <- mentee_interest2022[-empty_columns]
```

```
# add _2022 onto all columns
colnames(data2022) <- paste(colnames(data2022), "_2022", sep = "")

# rename so name and name_clean do not have _2022 added
# these will be used to match on in with data from other years.
names(data2022)[names(data2022) == "name_clean_2022"] <- "name_clean"
```

Mentee