Cleaning and Imputation

Team 97

10/29/2021

Cleaning Data

Goals:

1) Clean data from everything_merged.csv to produce a smaller file with:

- dates for the weeks
- limited predictor and outcome variables that are available weeks 1-33
  - exception: SCHLHRS - weeks 13-33

2) Impute missing values with mice library

```
library(readr)
library(data.table)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.3     v dplyr   1.0.3
## v tibble  3.0.1     v stringr 1.4.0
## v tidyr   1.0.3     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
library(DataExplorer)
library(mice)
```

```
## Warning: package 'mice' was built under R version 4.0.5
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
require(foreign)
```

```
## Loading required package: foreign
```

```
require(ggplot2)
require(MASS)
```

```
## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.0.5

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
require(Hmisc)
```

```
## Loading required package: Hmisc

## Warning: package 'Hmisc' was built under R version 4.0.5

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
require(reshape2)
```

```
## Loading required package: reshape2
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

Step 1: Data Cleaning

1a: Loading the data

If you're running this on your own machine, you'd need to download the data from our repo:

- https://github.gatech.edu/DVA-group97/our-lovely-repo/blob/master/Merging_and_Cleaning_Files/everything_merged.csv
- https://github.gatech.edu/DVA-group97/our-lovely-repo/blob/master/Merging_and_Cleaning_Files/dates.csv

```
# Update the working directory below with the location of the downloaded files (linked above)
setwd("C:/Users/katri/our-lovely-repo/Merging_and_Cleaning_Files")

# Note: used fread instead of read.csv because this is a large file (3 GB)
df <- fread("everything_merged.csv")
dates <- fread("dates.csv")
```

1b: Preliminary Cleaning:

- Select variables that are present weeks 1-33 (+ SCHLHRS which is weeks 13-33)
  - Note: not all questions were asked every week or sometimes the format changed. We selected variables to work with that were consistent during this time period.
- Add date variable based on the survey week. Dates manually entered in dates.csv
  - We used the start date of the survey period as the "date". Information pulled from the U.S. Census Household Pulse Survey website: https://www.census.gov/programs-surveys/household-pulse-survey/datasets.html

You can find a copy of the clean data up until this point at on our github at: https://github.gatech.edu/DVA-group97/our-lovely-repo/blob/master/Merging_and_Cleaning_Files/clean_data.csv

```
# the predictors and digital divide outcomes below are present through weeks 1-33 or 13-33
predictors <- c("EEDUC","RHISPANIC", "RRACE", "THHLD_NUMKID", "THHLD_NUMPER", "INCOME", "CURFOODSUF", "
outcomes <- c("TEACH1", "TEACH2", "TEACH3", "TEACH4", "TEACH5", "COMPAVAIL", "INTRNTAVAIL", "SCHLHRS")

# selecting the varaibles above for the cleaned dataset
clean <- df %>%
  dplyr::select(WEEK, PWEIGHT, predictors, outcomes) %>%
  filter(WEEK <= 33)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(predictors)` instead of `predictors` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(outcomes)` instead of `outcomes` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
# removing the raw data from the workspace to save memory (it's a big file)
rm(df)

# adding in the dates as a variable
clean <- merge(dates, clean, by = "WEEK")

# For a csv output of the cleaning done until this point, uncomment and run the line below
# write.csv(clean,"clean_data.csv", row.names = FALSE)
```

1c: Preparing Data for Regression Later:

- Filtering for data where we have a response to SCHLHRS (our chosen outcome variable)
- Putting in NA values for cases with no response
- Changing survey responses to factor variables (from numeric) for the multiple choice questions
  - the multiple choice survey responses have numeric codes

```r
#saves clean data into df variable
df <- clean %>%  filter(SCHLHRS > 0) # focusing on data only where we have responses for SCHLHRS (outco

df[df < 0] <- NA  # turns all negative values (skipped questions) into NA

# changing factor variables (represented as int) to factors
# for some reason I was having trouble automating this, so I did a lot by hand
df <- df %>%
  mutate(WEEK = as.factor(as.character(WEEK)) ) %>%
  mutate(EEDUC = as.factor(as.character(EEDUC)) ) %>%
  mutate(RHISPANIC = as.factor(as.character(RHISPANIC)) ) %>%
  mutate(THHLD_NUMKID = as.factor(as.character(THHLD_NUMKID) )) %>%
  mutate(THHLD_NUMPER = as.factor(as.character(THHLD_NUMPER) )) %>%
  mutate(INCOME = as.factor(as.character(INCOME) )) %>%
  mutate(CURFOODSUF = as.factor(as.character(CURFOODSUF)) ) %>%
  mutate(MORTCONF = as.factor(as.character(MORTCONF) ) )%>%
  mutate(EST_ST = as.factor(as.character(EST_ST) )) %>%
  mutate(TEACH1 = as.factor(as.character(TEACH1) )) %>%
  mutate(TEACH2 = as.factor(as.character(TEACH2) )) %>%
  mutate(TEACH3 = as.factor(as.character(TEACH3) )) %>%
  mutate(TEACH4 = as.factor(as.character(TEACH4) )) %>%
  mutate(TEACH5 = as.factor(as.character(TEACH5) )) %>%
  mutate(COMPAVAIL = as.factor(as.character(COMPAVAIL)) ) %>%
  mutate(INTRNTAVAIL = as.factor(as.character(INTRNTAVAIL)) ) %>%
  mutate(SCHLHRS = as.factor(as.character(SCHLHRS) )) %>%
   mutate(RRACE = as.factor(as.character(RRACE) ))

# prints out all the variables with their types
str(df)
```

```
## Classes 'data.table' and 'data.frame':   318989 obs. of  22 variables:
##  $ WEEK        : Factor w/ 21 levels "13","14","15",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ DATE        : chr  "8/19/2020" "8/19/2020" "8/19/2020" "8/19/2020" ...
##  $ PWEIGHT     : num  1149.7 1028.2 193.5 136.7 51.5 ...
##  $ EEDUC       : Factor w/ 7 levels "1","2","3","4",..: 7 7 6 4 4 3 5 4 6 7 ...
##  $ RHISPANIC   : Factor w/ 2 levels "1","2": 1 1 1 2 1 2 1 1 1 1 ...
##  $ RRACE       : Factor w/ 4 levels "1","2","3","4": 1 1 1 4 1 1 1 1 1 1 ...
##  $ THHLD_NUMKID: Factor w/ 6 levels "0","1","2","3",..: 4 3 4 3 2 4 2 3 2 2 ...
##  $ THHLD_NUMPER: Factor w/ 10 levels "1","10","2","3",..: 6 5 6 5 3 6 4 6 4 4 ...
##  $ INCOME      : Factor w/ 8 levels "1","2","3","4",..: 8 4 6 5 2 1 4 3 6 7 ...
##  $ CURFOODSUF  : Factor w/ 4 levels "1","2","3","4": 2 1 1 2 3 2 NA 2 1 1 ...
##  $ TSPNDFOOD   : num  400 200 200 200 100 250 50 150 200 140 ...
##  $ TSPNDPRPD   : num  75 250 20 100 15 80 0 0 50 40 ...
##  $ MORTCONF    : Factor w/ 5 levels "1","2","3","4",..: 4 4 4 3 1 1 4 2 4 4 ...
##  $ EST_ST      : Factor w/ 51 levels "1","10","11",..: 1 1 11 11 49 5 42 33 42 49 ...
##  $ TEACH1      : Factor w/ 1 level "1": NA NA 1 1 NA NA 1 NA NA NA ...
##  $ TEACH2      : Factor w/ 1 level "1": 1 NA NA 1 NA 1 NA 1 NA 1 ...
##  $ TEACH3      : Factor w/ 1 level "1": 1 NA NA NA 1 NA NA NA NA NA ...
##  $ TEACH4      : Factor w/ 1 level "1": NA 1 NA NA NA NA NA NA NA NA ...
##  $ TEACH5      : Factor w/ 1 level "1": NA NA NA NA NA NA NA NA 1 NA ...
##  $ COMPAVAIL   : Factor w/ 5 levels "1","2","3","4",..: 2 1 1 1 2 2 1 3 1 1 ...
##  $ INTRNTAVAIL : Factor w/ 5 levels "1","2","3","4",..: 2 1 3 1 1 3 1 2 1 1 ...
##  $ SCHLHRS     : Factor w/ 4 levels "1","2","3","4": 4 2 1 3 3 4 2 1 1 3 ...
```

```
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "sorted")= chr "WEEK"
```

1d: Combining Race and Ethnicity (Hispanic) Data

We made a variable for both race and ethnicity (Hispanic or not) combined, so we can consider this information together.

This was tough because there's overlap between identifying as Hispanic and all the races, so it's not a clear cut "bucket." To explore this issue, we made a table (see below) exploring how many respondents of each race identified as Hispanic and what percentage of the total that was for each race.

For reference 1 = White, 2 = Black, 3 = Asian, 4 = Other.

```
## # A tibble: 4 x 4
##   RRACE  hisp  total percent_total
## * <fct> <dbl>  <dbl>         <dbl>
## 1 1     29694 253597        0.117
## 2 2      1869  27098        0.0690
## 3 3       959  19731        0.0486
## 4 4      4873  18563        0.263
```

In the table above, it looks like people who identified as Hispanic mostly put White as their race.

For simplicity's sake, we put anyone who identified as Hispanic in the Hispanic category of the new variable. For the new variable, that means for White, Black, Asian, or Other (RRACE codes 1-4 respectively) these would all be people who did not identify as Hispanic.

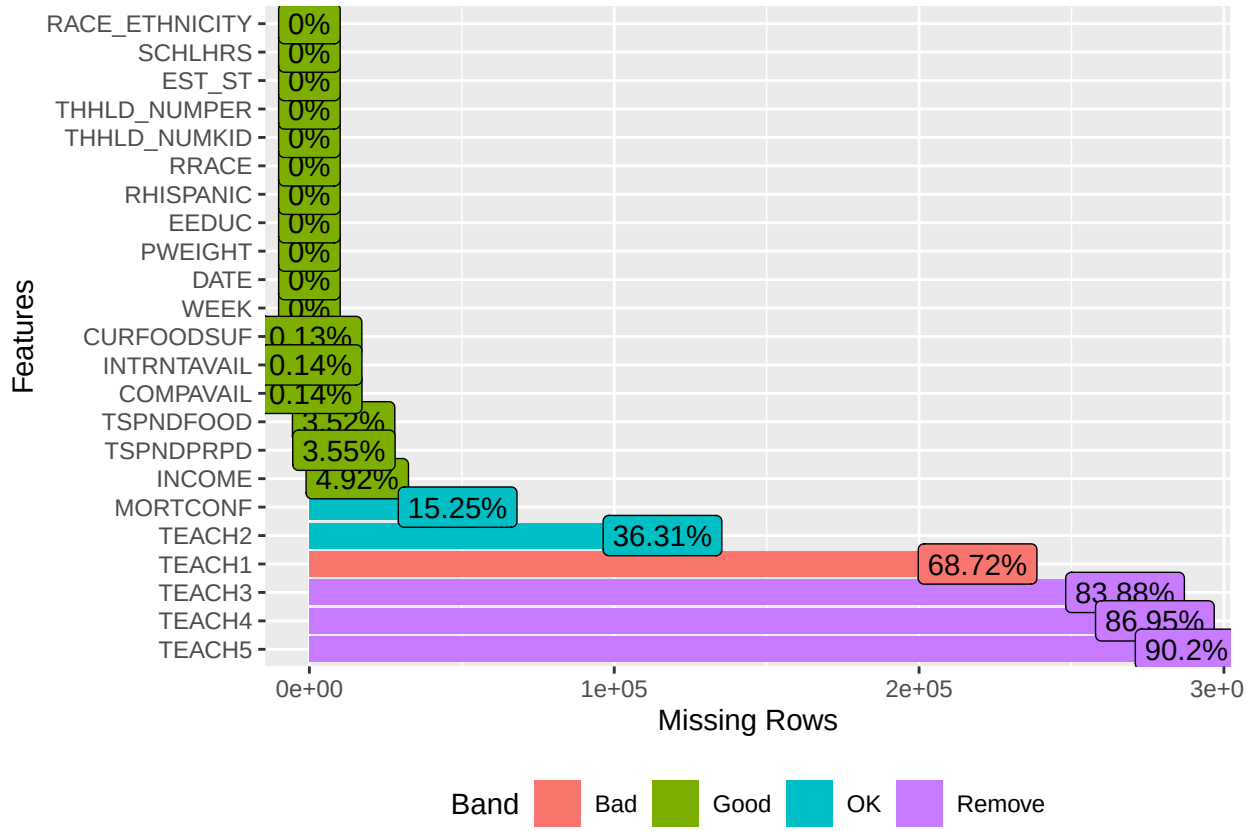
Step 2: Imputing Missing Values

2a: Exploring Missing Values

The following had more than 5% of the data missing:

- TEACH 1-5
- MORTCONF

We removed these variables from our analysis because too much of the data was missing.



2b: Imputation with Mice Library

Then, imputes missing values using Mice. When we imputed the values, we assumed that missing data was "missing at random"

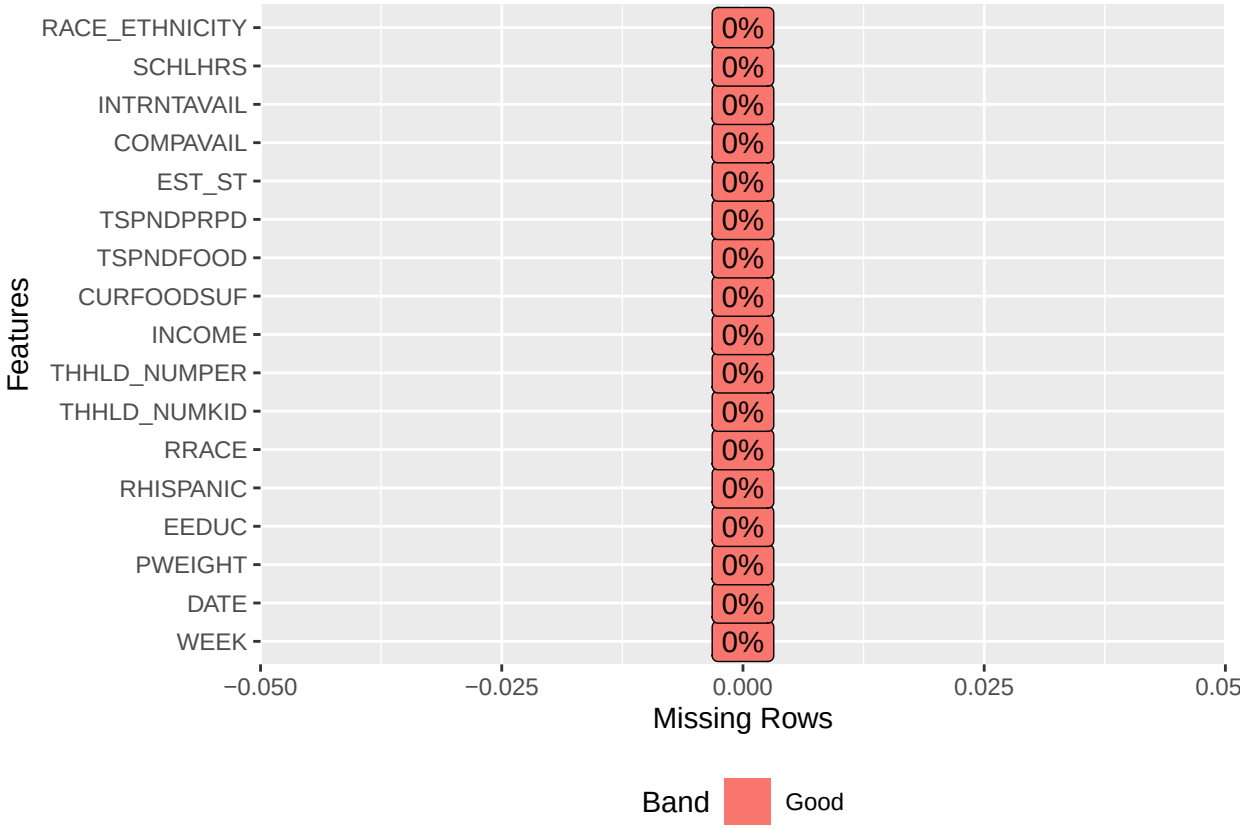
Notes for mice:

- You have to specify a method for imputing each variable. "" skips the variable (no missing values)
- See mice "method" selection info here: https://www.rdocumentation.org/packages/mice/versions/3.13.0/topics/mice
- We used "polr" for ordinal variables and "norm.predict" (linear regression) for continuous variables
- m=1, maxit = 1 does one iteration of imputation to save time – This whole process takes a while because we still have a lot of data
- Used quickpred to limit number of predictors used for imputation for each variable: https://www.rdocumentation.org/packages/mice/versions/3.13.0/topics/quickpred
  - This means it only use variables with correlation of at least .4 to build the models for imputation

```
##
##  iter imp variable
##   1   1  INCOME  CURFOODSUF  TSPNDFOOD  TSPNDPRPD  COMPAVAIL  INTRNTAVAIL
```

The cell below checks to ensure our current dataset has nothing missing.

You can find the cleaned dataset with the imputed values on our github at: https://github.gatech.edu/DVA-group97/our-lovely-repo/blob/master/Merging_and_Cleaning_Files/imputed_data.csv


RACE_ETHNICITY
SCHLHRS
INTRNTAVAIL
COMPAVAIL
EST_ST
TSPNDPRPD
TSPNDFOOD
CURFOODSUF
INCOME
THHLD_NUMPER
THHLD_NUMKID
RRACE
RHISPANIC
EEDUC
PWEIGHT
DATE
WEEK
0%
Features
−0.050
−0.025
0.000
0.025
0.05
Missing Rows
Band
Good