Global Affairs Capstone 2017

Code for Data Cleaning

```
knitr::opts_chunk$set(eval = FALSE, tidy.opts=list(width.cutoff=50), tidy=TRUE)
```

Input dirty merged file

```
setwd("/Users/Katie/Desktop/capstone")
m <- read.csv("/Users/Katie/Desktop/capstone/merged/newest_merge.csv",
    as.is = TRUE)
```

Get rid of "X.1500" and other column errors

```
new_m <- m[, -which(names(m) %in% c("X.1500", "column1"))]
new_m <- new_m[, -grep("^column", colnames(new_m))]
names <- names(new_m)
```

STANDARDIZE *Missing*

```
new_m[] <- lapply(new_m, as.character)
new_m <- as.data.frame(lapply(new_m, function(x) {
    x <- replace(x, x %in% c("NA", "N/A", ".", "na",
        "-", "8__unknown", "dont_know", "5__unknown",
        "Do_not_know", "No Response", "no responde",
        "No sabe", "Don't know", "no sabe", "no_responde",
        "Not specified.", "", "no answer", "No answer",
        "sin respuesta", "Sin respuesta", "Sin Informacion",
        "Unknown", "UNKNOWN", "unknown", "Unkown",
        "unkown", "Not Specified", "Desconocido", "desconocido",
        "6__unknown", "7__no_answer", "NULL", "Sin respuesta, Por que?",
        "NO ANSWER"), NA)
}))
```

STANDARDIZE *Yes*

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^(yes|si|true)$", replacement = "Yes",
    ignore.case = TRUE)
```

STANDARDIZE *No*

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^false$|^[3_nf]+o\\?*$", replacement = "No",
    ignore.case = TRUE)
```

STANDARDIZE *None*

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[35_n]+[oi]n(|guno)*$", replacement = "No",
    ignore.case = TRUE)
```

STANDARDIZE *Other*

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^(other|otro)$", replacement = "Other",
    ignore.case = TRUE)
```

STANDARDIZE *Distance*

There are two different types of responses. To prevent loss of data, keep both types. If one needs to do analysis across both types, "On-site (< 3km)" can be switched to On-site/Off-site, or to "Less/More_3km"

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^.*(<|less|menos).*2.*km.*", replacement = "Less_2km",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^.*(<|less|menos).*5.*km.*", replacement = "Less_5km",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^.*(<|less|menos).*10.*k+m.*",
    replacement = "Less_10km", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^.*(<|less|menos).*1[^0]*km.*",
    replacement = "Less_1km", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^.*(>|more|mas).*2.*km.*", replacement = "More_2km",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^.*(>|more|mas).*5.*km.*", replacement = "More_5km",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^.*(>|more|mas).*10.*k+m.*", replacement = "More_10km",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[_1o]+n.*3.*$", replacement = "On-site (< 3km)",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[_2o]+n.*3.*$", replacement = "On-site (> 3km)",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[_3o]+f.*3.*$", replacement = "Off-site (< 3km)",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[_4o]+f.*3.*$", replacement = "Off-site (> 3km)",
    ignore.case = TRUE)
```

STANDARDIZE *On-site, Off-site*

Certain variables, such as 'edu_access_f', have categorical rather than binary responses. Code snippet below the following has code to run if you want to keep in binary. Otherwise, don't run.

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(^Ons$|^[EO]N.*SIT.*)", replacement = "On_site",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "1__onsite|(^[YS]*[EI].*[OE]N.*SITE*[IO]*)$",
    replacement = "On_site", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^Y*[ES][SI]+.*F+.*site*[io]*$",
    replacement = "Off_site", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "2__offsite|2__yes_offsite|(^[o|]+ff+.*site+.*)|(^fuera.*sitio.*)",
    replacement = "Off_site", ignore.case = TRUE)
```

Optional, specific variable fixes

```
new_m$edu_access_f <- sub(new_m$edu_access_f, pattern = "Yes",
    replacement = "On_site", ignore.case = TRUE)
new_m$edu_access_f <- sub(new_m$edu_access_f, pattern = "None",
    replacement = "No", ignore.case = TRUE)
new_m$job_farm <- sub(new_m$job_farm, pattern = "^[^NY].*",
    replacement = NA)
```

STANDARDIZE *Gender*

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^F.*(emale|emenino)$", replacement = "Female",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^M.*(ale|asculino)$", replacement = "Male",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^males$", replacement = "Men")
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^females$", replacement = "Women")
```

STANDARDIZE *Intervals*

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "Diaria|(^[1_E]+veryday$)", replacement = "Everyday",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(^([_3O]+nce|Una).*(semana|WE+K)$)",
    replacement = "Once a week", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^((Una|Once).*M[OE][SN].*$)", replacement = "Once a month",
```

```
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "4__every_2_weeks|(^E.*2.*WE+K$)|(^(Cada|Dos).*semana.*$)",
    replacement = "Every 2 weeks", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[_6Ii]+rregular$", replacement = "Irregular",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(^[_7N]+.*(unca|ever))$", replacement = "Never",
    ignore.case = TRUE)
```

STANDARDIZE *Health*

There are certain discrepencies I chose to not consolidate (flu-like illnesses, colds) because I am not sure whether they represent the same category of illness.

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(CH*OLERA)", replacement = "Cholera",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(DIAR+H*EA)", replacement = "Diarrhea",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(Malnutri[tc]ion)", replacement = "Malnutrition",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "Conj[uo][n]*ctivit[e|i][s]*", replacement = "Conjuctivitis",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^(Common\\s)*cold[s]*$", replacement = "Colds",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(infecciones de la piel)|(^Skin\\s(infec[ct]ions|disease)$)",
    replacement = "Skin infections", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^Measles$", replacement = "Measles",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^Dysent[e]*ry$", replacement = "Dysentery",
    ignore.case = TRUE)
```

Mobile Clinic

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[6_M]+.*clinic$", replacement = "Mobile clinic",
    ignore.case = TRUE)
```

Fix Health Variable Errors

```
new_m$health_medicine <- sub(new_m$health_medicine,
    pattern = "^[^NY].*", replacement = NA, ignore.case = TRUE)
```

STANDARDIZE ***Health Providers***

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "Office[/.]*$", replacement = "office",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "He[al]+th", replacement = "health",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "[THER]+e*d+\\sc[ro]+[se]+[/.]*",
    replacement = "Red Cross", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "Burea[u]*", replacement = "bureau",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "Center", replacement = "center",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(^Woreda\\s+health\\s(office|center|Post)$)|(^By Woreda$)",
    replacement = "Woreda Health Office", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "(Gobierno|^goven*ment[.]*$)|(^gover[ne]*ment[.]*.*(RHB|DPPO)*$)",
    replacement = "Government", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[2_iINnO]+[GN][OGN]S*$", replacement = "NGO",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^ONG Int.$", replacement = "International NGO",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "Centro de salud local|^[7_L]+OCAL.*clinic",
    replacement = "Local clinic", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "clinic", replacement = "clinic",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "organi[zs]ation", replacement = "organization",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "Local clinic/med.*practitioners.*$",
    replacement = "Local clinic, medical practitioners",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^[4_O]+ther$|Otro, especificar",
    replacement = "Other", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "^church$", replacement = "Church",
    ignore.case = TRUE)
```

STANDARDIZE ***Non Profit Groups***

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "UNICEF", replacement = "UNICEF",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    sub, pattern = "save.*children.*", replacement = "Save the Children",
    ignore.case = TRUE)
```

STANDARDIZE ***Formatting***

Fix spacing between backslashes, commas and & signs. Standardize separator character to ‘,’

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "(\\s*&\\s)|(\\s*,\\s*)", replacement = ",",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "(\\s*/\\s)", replacement = "/",
    ignore.case = TRUE)
```

STANDARDIZE ***Waste Disposal Mechanisms***

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^No.*sist.*residuos$|^No System$|^[1N_]+o.*Waste.*Disposal.*$",
    replacement = "No waste disposal system", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Quema de residuos|^[3_B]+urning$",
    replacement = "Burning", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Fosa de residuos|^[2G_]+arbage.*pit$",
    replacement = "Garbage pit", ignore.case = TRUE)
```

STANDARDIZE ***Levels and Hygiene***

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^medium", replacement = "Medium",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^high", replacement = "High",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^very.*high", replacement = "Very high",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^low$", replacement = "Low", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^V.*low$", replacement = "Very low",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^[1G_]+ood$", replacement = "Good",
```

```
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^Buenas(\\s\\(Higienica\\))$|^[1G_]+ood(\\s[\\(]*Hygienic[\\)]*)*$",
    replacement = "Good", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Regular\\s\\(No\\smuy\\shigienicas\\)$|not_so_good|^Not\\sso\\sgood(\\s\\(Not\\shy
    replacement = "Not so good", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "[3_N]+o[n_]+usable$", replacement = "Non_usable",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Malas\\s\\(No\\shay\\shigiene\\)",
    replacement = "Bad", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Latrines|toilets$", replacement = "latrines",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "No Sabe|No_answer|^unknown$",
    replacement = NA)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "^No hay$", replacement = "No")
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "HWS", replacement = " Hand Washing Station",
    ignore.case = TRUE)
```

STANDARDIZE ***Info Source***

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Autoridades", replacement = "Authorities",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Families,friends", replacement = "Families/friends",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Social Media", replacement = "Social media",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "mobile phone", replacement = "Mobile phone",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "boca a boca", replacement = "Word of mouth",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Family member/relative$", replacement = "Family member",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "En persona", replacement = "In person",
    ignore.case = TRUE)
```

STANDARDIZE *Occupations*

```
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Agricultur[ea]|^Farming$", replacement = "Agriculture",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Agro-pastoralism|^Agriculture/Livestock$",
    replacement = "Agro-Pastoralism", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "trabajador.*diario|^daily.*laborer$",
    replacement = "Daily Laborer", ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Craftsm[ea]n", replacement = "Craftsman",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Comercio minorista|^Trade$", replacement = "Petty Trade",
    ignore.case = TRUE)
new_m[, 1:ncol(new_m)] <- lapply(new_m[, 1:ncol(new_m)],
    gsub, pattern = "Pesca", replacement = "Fishing",
    ignore.case = TRUE)
```

STANDARDIZE *Dates*

Function that takes in differently formatted dates and outputs them as (DD/MM/YY) Should run through once more after

```
library(stringi)
new_m <- new_m[order(new_m$country), ]
new_m <- new_m[order(new_m$round), ]
round <- ""
prev_date <- c(0, 0)

# For each date variable, standardize the dates in
# the column
for (j in grep(pattern = "(^.*date$)|(^date.*$)", x = colnames(new_m),
    value = TRUE)) {
    for (col in which(colnames(new_m) == j)) {
        for (i in 1:(nrow(new_m) - 1)) {
            if (!is.na(new_m[i, col])) {
                # save each date as a vector of its parts
                date <- as.character(new_m[i, col])
                date <- unlist(strsplit(date, "\\/|-"))

                # rewrite written months as integer values
                date[2][1] <- sub("Jan", "01", date[2][1])
                date[2][1] <- sub("Feb", "02", date[2][1])
                date[2][1] <- sub("Mar", "03", date[2][1])
                date[2][1] <- sub("Apr", "04", date[2][1])
                date[2][1] <- sub("May", "05", date[2][1])
                date[2][1] <- sub("Jun", "06", date[2][1])
                date[2][1] <- sub("Jul", "07", date[2][1])
                date[2][1] <- sub("Aug", "08", date[2][1])
                date[2][1] <- sub("Sep", "09", date[2][1])
```

```
                date[2][1] <- sub("Oct", "10", date[2][1])
                date[2][1] <- sub("Nov", "11", date[2][1])
                date[2][1] <- sub("Dec", "12", date[2][1])

                # If year is in the first slot, swap first and
                # third place values
                if (nchar(date[1][1]) > 3) {
                  temp <- date[1]
                  date[1] <- date[3]
                  date[3] <- date[1]
                }

                # pad/cut the three places so each has length 2
                if (nchar(date[2][1]) < 2)
                  stri_sub(date[2][1], 1, 0) <- 0  # padding
                if (nchar(date[1][1]) < 2)
                  stri_sub(date[1][1], 1, 0) <- 0  # padding
                if (nchar(date[3][1]) > 2)
                  date[3][1] <- stri_sub(date[3][1],
                    3, 4)

                # check if dd/mm/yy or mm/dd/yy
                if (new_m$round[i] == round) {
                  dm <- as.integer(date[1][1]) - as.integer(prev_date[1])
                  md <- as.integer(date[2][1]) - as.integer(prev_date[2])
                  if (abs(dm) < abs(md)) {
                    temp <- date[1]
                    date[1] <- date[2]
                    date[2] <- temp
                  }
                }

                # check if dd/mm/yy or mm/dd/yy
                if (as.integer(date[2][1]) > 12) {
                  temp <- date[1]
                  date[1] <- date[2]
                  date[2] <- temp
                }
                round <- new_m$round[i]
                prev_date[1] <- date[1][1]
                prev_date[2] <- date[2][1]
                date_new <- paste(date, collapse = "/")
                print(date_new)

                # save dates back to file
                new_m[i, col] <- date_new
            }
        }
    }
}
```

Write out cleaned CSV

```
write.csv(new_m, "csv/clean/newest_clean.csv", row.names = FALSE,
    fileEncoding = "UTF-8")
```