Flu Near You (FNY)

Description

This document provides:

- decriptions of existing variables within the FNY database
- descriptions of recommended derived variables
- descriptions of best practices for cleaning FNY data
- sample R code for creating new variables and cleaning FNY data

Required packages for R code:

- tidyr
- dplyr
- zipcode

Please contact Kristin Baltrusaitis (kristin.balt@gmail.com) with any questions

Participant IDs

Existing variables:

user_id: unique ID for each *user*

user_household_id: unique ID for each *household member* (NA for users)

user_entry_id: unique ID for each *report*

guest: indicator variable (Y) for non-registered users (blank for registered users)

New variables:

participant_id: unique ID for each *participant* (participants are defined as *users* and their *household members*).

This variable is created by concatinating the user_id and the user_household_id with a '.'

ili: an indicator variable representing the incidence of *influenza-like illness*

This variable is equal to 1 when the participant reports either fever and cough OR fever and sorethroat in the same report and 0 otherwise.

Best Practices:

Because some participants may have multiple reports within the same week, we recommend subsetting the dataset such that each participant has only 1 report per week, prioritizing reports of ili. In other words, if one participant has two reports within the same week, where one report is ili=0 and the other report is ili=1, the report with ili=1 is selected.

Sample R Code:

```r
# Create participant ID
fny <- fny %>% replace_na(list(user_household_id=0)) %>%
  mutate(participant_id=paste(user_id, user_household_id, sep = "."))

# Create ILI variable
fny <- fny %>% replace_na(list(fever=0, cough=0, sorethroat=0)) %>%
  mutate(ili = ifelse(fever==1 & cough==1, 1,
                      ifelse(fever==1 & sorethroat==1, 1, 0)))

# Subset to one entry per participant per week
fny2<-fny %>% arrange(participant_id, week_of, desc(ili)) %>%
  distinct(participant_id, week_of, .keep_all = TRUE)
```

Dates

Existing variables:

reg_date: registration date of user (includes date and time)

week_of: date of reporting week (Monday)

entry_date: date symptom report was submitted (includes date and time)

ill_date: start date of illness, self-reported by user (blank if no symptoms are reported)

Best Practices:

There are a few weeks with incorrect week_of dates. We recommend removing these entries from the dataset.

Incorrect week_of dates:

- 2015-01-11
- 2015-11-22
- 2015-12-06
- 2017-01-01
- 2017-12-31

Sample R Code:

```r
# Subset to one entry per participant per week
fny2<-fny2 %>%  filter(week_of !='2015-01-11' & week_of !='2015-11-22' &
                   week_of !='2015-12-06' & week_of !='2017-01-01' &  week_of!='2017-12-31')
```

Participant Characteristics

Existing variables:

zip: user zip code provided at registration

state: character variable with state abbreviation

state_id: numeric state variable

gender: registered user's gender (F=female, M=male)

dob: registered user's date of birth (month/Year)

household_gender: household member's date of birth (NA for users)

household_dob: household member date of birth (NA for users)

New variables:

participant_dob: participant's date of birth

participant_gender: participant's gender

participant_state: participant's state

Best Practices:

Because state is missing for multiple entries, we recommend using the zip code to fill in these missing values.

Sample R Code:

```r
# For users, participant_dob = dob, for household members participant_dob = household_dob
fny2<- fny2 %>% mutate(participant_dob= ifelse(user_household_id==0, as.character(dob),
                 ifelse(user_household_id>0, as.character(household_dob),NA)))

# For users, participant_gender = gender, for household members participant_gender = household_gender
fny2<- fny2 %>% mutate(participant_gender= ifelse(user_household_id==0, as.character(gender),
                 ifelse(user_household_id>0, as.character(household_gender),NA)))

# Clean zipcodes
fny2<- fny2 %>% mutate(clean_zip=clean.zipcodes(zip))

# Import zip code data
data(zipcode)
zips<-zipcode %>%  select(zip, state) %>% rename(new_state=state)

fny3<-merge(fny2, zips, by.x="clean_zip", by.y="zip", all.x=T)

fny3<- fny3 %>% mutate(participant_state=ifelse(!is.na(new_state), new_state, as.character(state)))
```

Symptom Reports

Existing variables:

no_symptoms: user did not report symtpoms for the participant (1= no symptom reported, NA if any symptom reported)

For all variables below, 1= symptom reported, NA if symptom not reported)

fever, diarrhea, chills, nausea, cough, sorethroat, bodyache, headache, fatigue, breath, rash, red_eyes, joint_pain, eye_pain, dark_urine, yellow_eyes, running_nose

fever_f: numeric, recorded only if user reports fever, NA otherwise

medical_attention_no: Y=Did not seek medical attention N=no symptoms OR did seek medical attention

For all variables below, Y=sought medical attention N=did not receive medical attention at the location

medical_attention_doctors_office, medical_attention_urgent, medical_attention_clinic, medical_attention_emergency, medical_attention_hospital, medical_attention_other

New variables:

medical_attention_any: composite of all medical attention variables

Best Practices:

The meaning of medical_attention_no changed at the beginning of 2017, and we recommend using the composite medical_attention_any variable instead.

Sample R Code:

```r
# Create composite medical care variable
fny2 <- fny2 %>% mutate(medical_attention_any=ifelse(medical_attention_doctors_office=='Y'|
                      medical_attention_urgent=='Y'| medical_attention_clinic=='Y'|
                        medical_attention_emergency=='Y'| medical_attention_hospital=='Y'|
                        medical_attention_other=='Y',1,0))
```