Cleaning General TGP Data

# TGP Data Quality goals

* Increase data quality
* Increase data use to answer questions
* Encourage data-driven decision making throughout the organization

# Using R to maximize ETO data

The following document outlines how we can use R to improve our ETO data so that we can generate more accurate insights from the information we collect from our members.

**Process**

1. Import data

2. Select relevant variables

3. Manipulate data

4. Join data

5. Explore data

6. Summarize data

# Importing data into R

To make it easier for me to import data I used RStudio which allows you to directly import data into the system as well as select only the variables that you would like to work with.

You go to the *Environment* pane and then click on the *Import Dataset* icon and then select your file.

Another option is to use the readxl package in r to import your data. An example is provided below

Library (readxl)

Demographics\_Original <- read\_excel("~/R/Demographics\_Original.xlsx")

* filepath is the file name
* sheet = is the sheet number within the excel spreadsheet
* col\_names set to TRUE allows you to import the colomn names
* col\_types is set to NULL allowing R to select the column types
* skip is set to 0 because we don't want to skip any rows. If you want to skip a certain number of rows you can set a number.

I imported four files because I wanted to look at various data points but for our initial clean-up efforts you should use the data that you are most interested in at the time

**Files I imported for this exercise**

* Visits Per Member
* Member Demographics
* Member Employment
* Member Marital Status

**Selecting Variables**

For tgp\_race I wanted to make sure I only had the race variable along with member unique id, first name, and last name. The numbers represent the columns removed from original dataset

tgp\_race <- Demographics\_Original[, c(-5,-6, -8)]

# Manipulating the Data

library(tidyr)

library(dplyr)

**Editing column names with ones that are easier to read**

tgp\_race$`THE GATHERING PLACE RACE`[tgp\_race$`THE GATHERING PLACE RACE` == "American Indian/Alaska Native/ Indo Americano"] <- "Native American"

tgp\_race$`THE GATHERING PLACE RACE`[tgp\_race$`THE GATHERING PLACE RACE` == "White/Blanco"] <- "White"

tgp\_race$`THE GATHERING PLACE RACE`[tgp\_race$`THE GATHERING PLACE RACE` == "Other/Otro"] <- "Other"

tgp\_race$`THE GATHERING PLACE RACE`[tgp\_race$`THE GATHERING PLACE RACE` == "Black/African American/Africano Americano"] <- "Black/African American"

tgp\_race$`THE GATHERING PLACE RACE`[tgp\_race$`THE GATHERING PLACE RACE` == "Asian/Asiatico"] <- "Asian"

tgp\_race$`THE GATHERING PLACE RACE`[is.na(tgp\_race$`THE GATHERING PLACE RACE`)] <- "Unknown"

**Removing duplicates**

Depending on how the data was compiled it is important that any duplicate data is removed.

tgp\_race <- tgp\_race[!duplicated(tgp\_race), ]

**Create a value column**

In order to be able to identify the members with multiple selection on a variable such as race we need to associate each selection with a value. The following steps provide an overview of that process.

**Create race value**

tgp\_race <- tgp\_race %>%

mutate(Race\_Value = ifelse(`THE GATHERING PLACE RACE` == "Unknown", "0", "1"))

**Convert the variable to a numeric value**

tgp\_race$Race\_Value <- as.numeric(tgp\_race$Race\_Value)

**Inspect data**

str(tgp\_race)

**Sum the race values for each member**

tgp\_race\_mr <- tgp\_race %>% group\_by(`PARTICIPANT UNIQUE IDENTIFIER`) %>%

summarise(multipleR = sum(`Race\_Value`)) %>%

ungroup()

**Join sum with the rest of the data**

tgp\_race\_joined <- left\_join(tgp\_race, tgp\_race\_mr, by = "PARTICIPANT UNIQUE IDENTIFIER")

**Remove any additional duplicates generated through calculations**

tgp\_race\_joined <- tgp\_race\_joined[!duplicated(tgp\_race\_joined), ]

**Generate a mixed race value for the Race column**

tgp\_race\_joined[tgp\_race\_joined$multipleR > 1, "THE GATHERING PLACE RACE"] <- "Mixed Race"

**Remove race value since it is no longer needed**

tgp\_race\_joined <- tgp\_race\_joined[, -6]

# **Repeat these steps for health issues**

tgp\_health <- Demographics\_Original[ , c(-5, -6, -7)] #select relevant columns

tgp\_health <- tgp\_health[!duplicated(tgp\_health), ] #remove duplicates

**Rename NAs**

tgp\_health$`ANY HEALTH ISSUES`[is.na(tgp\_health$`ANY HEALTH ISSUES`)] <- "Unknown"

**Create a value variable for health issues**

tgp\_health <- tgp\_health %>% mutate(health\_value =

ifelse(`ANY HEALTH ISSUES` %in% c("Unknown", "none"), "0", "1"))

**Convert health values to numerics because they are currently text**

tgp\_health$health\_value <- as.numeric(tgp\_health$health\_value)

**Sum values for each person**

tgp\_health\_mh <- tgp\_health %>%

group\_by(`PARTICIPANT UNIQUE IDENTIFIER`) %>%

summarize(multipleH = sum(`health\_value`)) %>%

ungroup()

**Join data**

tgp\_health\_joined <- tgp\_health %>%

left\_join(tgp\_health\_mh, by = "PARTICIPANT UNIQUE IDENTIFIER")

**Change value for all those with more than 1 health issue**

tgp\_health\_joined[tgp\_health\_joined$multipleH > 1, "ANY HEALTH ISSUES"] <- "Multiple Health Issues"

**Remove duplicates now that we have identified those with multiple health issues**

tgp\_health\_joined <- tgp\_health\_joined[!duplicated(tgp\_health\_joined), ]

**Remove Health Value column**

tgp\_health\_joined <- tgp\_health\_joined[, -6]

**Other**

Pull out Hispanic and Homeless Variable from Demographics Data

Hm\_Hs\_Var <- Demographics\_Original[, c(1:6)]

Hm\_Hs\_Var <-Hm\_Hs\_Var[!duplicated(Hm\_Hs\_Var), ]

# Combine all data into one dataset

all\_tgp\_data1 <- merge(Compiled\_tgp\_data, Hm\_Hs\_Var)

all\_tgp\_data2 <- merge(Marital\_Status, Monthly\_Income\_Employment)

all\_tgp\_data3 <- merge(all\_tgp\_data1, all\_tgp\_data2)

all\_tgp\_data <- merge(all\_tgp\_data3, Visits)

# Start clean-up efforts

Review each columns and identify all the NAs

Clean-up compiled data to get rid of NAs

all\_tgp\_data$`TGP MARITAL STATUS`[is.na(all\_tgp\_data$`TGP MARITAL STATUS`)] <- "Unknown"

all\_tgp\_data$`MY FAMILY CONSISTS OF`[is.na(all\_tgp\_data$`MY FAMILY CONSISTS OF`)] <- "Unknown"

all\_tgp\_data$EMPLOYMENT[is.na(all\_tgp\_data$EMPLOYMENT)] <- "Unknown"

all\_tgp\_data$`MONTHLY INCOME`[is.na(all\_tgp\_data$`MONTHLY INCOME`)] <- "0"

**Create an age column**

library(eeptools)

**Change the date of birth column to a date**

all\_tgp\_data$`DATE OF BIRTH` <- as.Date(all\_tgp\_data$`DATE OF BIRTH`)

**Create an age column**

all\_tgp\_data <- all\_tgp\_data %>%

mutate(AGE = round(age\_calc(`DATE OF BIRTH`,

enddate = as.Date('2017-10-01'), units = "years")))

**Rename all columns so it’s easier to use data**

fix(all\_tgp\_data)

str(all\_tgp\_data)

**Change monthly income to numerics**

all\_tgp\_data$MaxIncome[all\_tgp\_data$MaxIncome == "less than $100"] <- "100"

all\_tgp\_data$MaxIncome[all\_tgp\_data$MaxIncome == "less than $500"] <- "500"

all\_tgp\_data$MaxIncome[all\_tgp\_data$MaxIncome == "less than $800"] <- "800"

all\_tgp\_data$MaxIncome[all\_tgp\_data$MaxIncome == "less than $1,000"] <- "1000"

all\_tgp\_data$MaxIncome[all\_tgp\_data$MaxIncome == "less than $1,500"] <- "1500"

all\_tgp\_data$MaxIncome[all\_tgp\_data$MaxIncome == "less than $2,000"] <- "2000"

all\_tgp\_data$MaxIncome[all\_tgp\_data$MaxIncome == "0"] <- "0"

all\_tgp\_data$MaxIncome[all\_tgp\_data$MaxIncome == "more than $2,000"] <- "4000"

all\_tgp\_data$MaxIncome <- as.numeric(all\_tgp\_data$MaxIncome)