AYC TA Log - 2020-Q2: Oct to Dec 2020

Summarizing Counts (MRDum) by Date

Josephine McKelvy

February 08, 2021

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

Click the **+c Insert** button to insert code chunks (or functions). Add documentation outside of the code chunks as reminders of what your code does.

When you click the **Knit** button, it compiles all of the code chunks and generates a document that includes both content as well as the output of any embedded R code chunks within the document. Use the **gear** icon to adjust settings (e.g., “show output only”) for each code chunk.

The shortcut key for “<-” is ALT + [dash] CTRL + ENTER will run the highlighted line(s) of code. The hashtag is code that is commented out.

# 1. Find/Set the working directory, where you’ll import/export files.

getwd()

## [1] "C:/Users/Josephine/Documents/ayc/2021-02-08 TA Log"

# Wherever this code is saved, the file(s) for your dataset(s) should be there too

# 2. Name & import your data frame, or set of information (like a .csv file of survey responses), as factor variables.

* I created a dataframe object (which I named “raw.df”) for each dataset that will read in the respective .csv files and convert categorical variables into factor variables. (Copy and paste the name of the file. There’s a hidden character in the SurveyMonkey exports.)

raw.df <- read.csv("AYC TA Log 2020-2021.csv", stringsAsFactors = TRUE)  
  
# Export .CSV from your survey platform and confirm numeric values, NOT choice text.   
  
# Clean variable names in advance (e.g., insert a row of Q#s). Do NOT start variable names with numbers.

# 3. Clean the analytic dataset.

* I created an analytic dataset (which I named “ta.df”) that will subset the data and remove informational header rows that are not survey responses.

(**1-based index**: R starts with the first row of obs. being “1”; 3:nrow means that your responses/obs. start on row 4 until the Nth row. Check your dataset to see which row the actual observations start.)

ta.df <- raw.df[3:nrow(raw.df),] #17 obs  
# Confirm the number of obs and variables.

* Recode missing data:

ta.df[ta.df==""] <- NA

# 3a. Clean the nominal (categorical, dichotomous/MRdum, ordinal) variables.

* For each multiple choice statement, change the response values to factor, specify levels, and rename the value labels, based on the survey/codebook: (open both the dataset and the codebook/survey with question numbers and recode values to make sure the labels and levels are correct)

labelspec <- c("Lisa Garland",  
 "Caro Welker",  
 "Josephine McKelvy",  
 "Sophia Durant",  
 "Other")  
labeltta <- c("Training",  
 "Technical Assistance")  
labelmode <- c("Phone",  
 "Web-based for one organization",  
 "In-person for one organization",  
 "In-person for multiple organizations",  
 "Web-based for multiple organizations")  
labelorg <- c("Alamance Achieves",  
 "Alamance County Health Dept",  
 "CityGate Dream Center",  
 "CrossRoads",  
 "Alamance County Dept of Social Srvs",  
 "Children's Home Society",  
 "Positive Attitude Youth Center",  
 "Elon University",  
 "Exchange Club",  
 "Salvation Army Boys & Girls Club")  
  
ta.df$Q1 <- factor(ta.df$Q1,  
 levels = c(1,2,3,4,5),  
 labels = labelspec)  
ta.df$Q3 <- factor(ta.df$Q3,  
 levels = c(1,2),  
 labels = labeltta)  
ta.df$Q6 <- factor(ta.df$Q6,  
 levels = c(1,2,3,4,5),  
 labels = labelmode)

* Reverse-coded columns/variables: <https://www.theanalysisfactor.com/easy-reverse-code/>

Convert the factor variable to numeric so that you can subtract it from a value to reverse code that value

# Example:  
# chrsap.df$Q2.1\_3 <- as.numeric(as.character(chrsap.df$Q2.1\_3))  
# chrsap.df$Q2.1\_3 <- 8-chrsap.df$Q2.1\_3 #on a scale of 1 to 7

* Create an object (called “cleanIt”) that performs a function to a vector, which serves as a placeholder for something like a column. In that function, convert the vector to a character variable. You have to convert this factor variable to a character variable first because you may see two types of labels (e.g., “1” and “NA”), but there could be more than 2 hidden levels in your factor variable. Where the variable is “NA” or a blank, replace with zero (0). Then convert the vector to a numeric variable that can be summed.

cleanIt <- function(vec){  
 chars <- as.character(vec)  
 chars[is.na(chars)] <- "0"  
 chars[chars==""] <- "0"  
 return(as.numeric(chars))  
}

* There is a multiple-response item/column with varying numbers of options for each implementing organization (IO): columns 16-25 of the ta.df. Use list apply (lapply) to repeat the “cleanIt” function to those columns, i.e., your vectors.

ta.df[16:25] <- lapply(ta.df[16:25], cleanIt)

* Compute frequencies and percentages for each response category for each question.

mrdumdisc <- data.frame(Freq=colSums(ta.df[16:25]),  
 Pct.of.Answ = (colSums(ta.df[16:25])/sum(ta.df[16:25]))\*100,  
 Pct.of.Case = (colSums(ta.df[16:25])/nrow(ta.df[16:25]))\*100)  
  
# Plot and save bar charts for these percentages of cases/obs.  
# library(ggplot2)

# 3e. Convert continuous variables from factor to numeric.

(Factor variables are stored as integer codes–not numeric value–to create levels (e.g., responses ranging from 10 to 12 are three levels). Convert to character and then numeric.)

ta.df$adult <- as.numeric(as.character(ta.df$Q17))  
sum(ta.df$adult)

## [1] 104

ta.df$minutes <- as.numeric(as.character(ta.df$Q5))  
#Optional: Convert to hours  
ta.df$hours <- ta.df$minutes/60

# 4. Subset data by date ranges (e.g., quarters)

# Convert column/variable to "date" class (%m = 2-digit month; %d = 2-digit day; %[capital]Y = 4-digit year)  
ta.df$Q4 <- as.Date(ta.df$Q4,"%m/%d/%Y")  
  
# Subset data between two dates, inclusive  
# qtr1 <- ta.df[ta.df$Q4 >="2020-07-01" & ta.df$Q4 <="2020-09-30",]  
qtr2 <- ta.df[ta.df$Q4 >="2020-10-01" & ta.df$Q4 <="2020-12-31",]  
qtr3 <- ta.df[ta.df$Q4 >="2021-01-01" & ta.df$Q4 <="2021-03-31",]  
# qtr4 <- ta.df[ta.df$Q4 >="2021-04-01" & ta.df$Q4 <="2021-06-30",]

* How many adults (from IOs, CAG) were trained this quarter (by T vs TA)?

aggregate(x= qtr2$adult,  
 by= list(qtr2$Q3),  
 FUN=sum)

## Group.1 x  
## 1 Training 39  
## 2 Technical Assistance 13

aggregate(x = qtr3$adult,  
 by= list(qtr3$Q3),  
 FUN=sum)

## Group.1 x  
## 1 Training 42  
## 2 Technical Assistance 10

* How many hours of T/TA were provided this quarter?

aggregate(x=qtr2$hours,  
 by= list(qtr2$Q3),  
 FUN=sum)

## Group.1 x  
## 1 Training 6.00  
## 2 Technical Assistance 4.75

aggregate(x=qtr3$hours,  
 by= list(qtr3$Q3),  
 FUN=sum)

## Group.1 x  
## 1 Training 8.5  
## 2 Technical Assistance 3.5