CAG Baseline Survey - Winter 2020

Visualizing Longitudinal Likert Scales

Josephine McKelvy

January 27, 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"

# 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.

raw.df <- read.csv("Systems Change Baseline Assessment - CAG.csv", stringsAsFactors = TRUE)  
  
# Export numeric values, NOT choice text, from your survey platform.   
  
# Clean variable names in advance (e.g., insert a row of Q#s). Do NOT start variable names with numbers.  
  
# Confirm the number of obs and variables.

#3. Clean the dataset. - I created an analytic dataset (which I named “cag.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.)

cag.df <- raw.df[3:nrow(raw.df),] #26 obs

* Recode missing data:

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

#3a. Clean the nominal (categorical, dichotomous) 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)

levelorg <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19)  
labelorg <- c("Community resident",  
 "African American Cultural Arts & History Center",  
 "Alamance Achieves",  
 "ABSS",  
 "Alamance Cares",  
 "ACDSS",  
 "ACHD",  
 "Alamance Pride",  
 "Alcohol Drug Services",  
 "BHA",  
 "Children's Home Society",  
 "CityGate Dream Center",  
 "CrossRoads",  
 "Elon University",  
 "Family Abuse Services",  
 "NC DPS",  
 "PAYC",  
 "Salvation Army Boys and Girls Club",  
 "United Way of Alamance County")  
  
cag.df$Q1 <- factor(cag.df$Q1,  
 levels = levelorg,  
 labels = labelorg)  
  
levelrole <- c(1,2,3,4)  
labelrole <- c("Community resident",  
 "Frontline staff",  
 "Supervisory staff",  
 "Organizational leadership")  
  
cag.df$Q2 <- factor(cag.df$Q2,  
 levels = levelrole,  
 labels = labelrole)  
  
cag.df$Q5 <- factor(cag.df$Q5,  
 levels = c(1,2),  
 labels = c("Before Trainings","After Trainings"))

* 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

#3. Clean the ordinal (Likert-scale) variables. - For each Likert-style 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.)

levelksa <- c(1,2,3,4,5)  
labelksa <- c("Novice","Advanced Beginner", "Competent","Proficient","Expert")  
  
cag.df$Q6 <- factor(cag.df$Q6,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)  
  
cag.df$Q7 <- factor(cag.df$Q7,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)  
  
cag.df$Q8 <- factor(cag.df$Q8,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)  
  
cag.df$Q9 <- factor(cag.df$Q9,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)  
  
cag.df$Q10 <- factor(cag.df$Q10,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)  
  
cag.df$Q11 <- factor(cag.df$Q11,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)  
  
cag.df$Q12 <- factor(cag.df$Q12,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)  
  
cag.df$Q13 <- factor(cag.df$Q13,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)  
  
cag.df$Q14 <- factor(cag.df$Q14,  
 levels = levelksa,  
 labels = labelksa,  
 ordered = TRUE)

* Create an object (called “ksa.df”, etc.) that is a subset of the time variable and Likert statements (e.g., columns 14 to 23). Name the columns/variables for each subset.

ksa.df <- cag.df[,14:23]  
names(ksa.df) <- c("Time",  
 "Incorporating trauma-informed strategies",  
 "Understanding historical impacts",  
 "Incorporating root cause approaches",  
 "Interviewing community members",  
 "Having difficult conversations",  
 "Exploring feedback loops across systems",  
 "Aligning personal values to my work",  
 "Translating my perceptions into actions",  
 "Sharing what I've learned")  
  
# Optional for Pre/Post: Make sure both have the same column names. Add a new column for the timepoint variable (e.g., 1 for before and 2 for after an intervention.   
   
# benefits1 <- data.frame(chrsap.df[,14:24])  
# names(benefits1) <- c("Teaching nursing students",  
# "Integrating nursing students",  
# "Increasing my knowledge base",  
# "Keeping current and stimulated",  
# "Influencing change on my work place",  
# "Gaining personal satisfaction",  
# "Being recognized as a role model",  
# "Improving my teaching skills",  
# "Sharing my knowledge",  
# "Learning from nursing students",  
# "Contributing to my profession",  
# "Increasing my involvement in the organization",  
# "Improving my organizational skills",  
# "Improving my chances for promotion")  
# benefits1$time = 1  
  
# benefits2 <- data.frame(chrsap.df[,32:45])  
# names(benefits2) <- c("Teaching nursing students",  
# "Integrating nursing students",  
# "Increasing my knowledge base",  
# "Keeping current and stimulated",  
# "Influencing change on my work place",  
# "Gaining personal satisfaction",  
# "Being recognized as a role model",  
# "Improving my teaching skills",  
# "Sharing my knowledge",  
# "Learning from nursing students",  
# "Contributing to my profession",  
# "Increasing my involvement in the organization",  
# "Improving my organizational skills",  
# "Improving my chances for promotion")   
# benefits2$time = 2  
  
# Optional for Pre/Post: Append the Pre/Post subsets as dataframes, using rbind.   
# benefits <- rbind(benefits1,benefits2)  
  
# Optional for Pre/Post: Factor the time variable you created  
# benefits$time <- factor(benefits$time,  
# levels = c(1,2),  
# labels = c("Before","After"),  
# ordered = TRUE)  
  
# if there are problems with NA in these likerts:  
ksa.df <- na.omit(ksa.df)

* Create a title for each subset of Likert data:

ksat <- "CAG Members' Reported Knowledge and Skills"

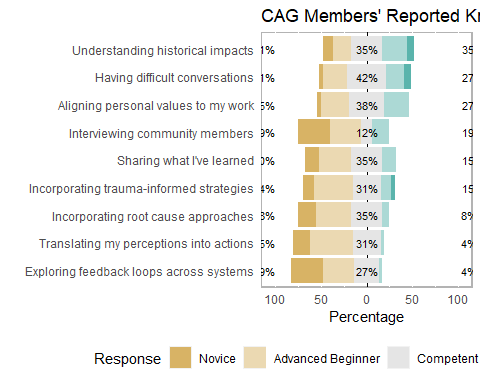
#4c. Plot and save diverging bar charts for each set of Likert statements.

# install.packages("likert")  
# update.packages("ggplot2")  
library(likert)

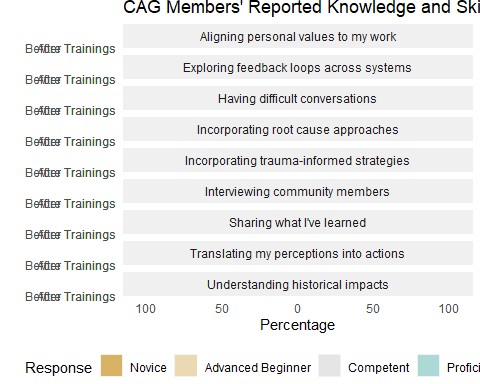
## Loading required package: ggplot2

## Loading required package: xtable

lktksac <- likert(ksa.df[,2:10])  
plot(lktksac) + ggtitle(ksat)



ggsave("diverging-ksas-crossectional.png", width = 10, height = 4, units = "in")  
  
lktksal <- likert(ksa.df[,2:10], grouping = ksa.df$Time)  
plot(lktksal, group.order = c("Before Trainings","After Trainings")) + ggtitle(ksat)



ggsave("diverging-ksas-longitudinal.png", width = 8, height = 12, units = "in")