Section: Reach (Pre-Test Demographics)

Summarizing Pre-Tests by Group

Josephine McKelvy

February 17, 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/2020-02-17 PAYC PPT"

# 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("Participant Pre- Post-Test Survey.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 “ppt.df”) that will select/subset the data columns and remove informational header rows that are not survey responses.

(**1-based index**: R starts with the first row of obs. being “1”; 2:nrow means that your responses/obs. start on row 3 until the Nth row. Check your dataset to see which row the actual observations start.)

ppt.df <- raw.df[2:nrow(raw.df),c(12:67,75:113)] #11 obs, 95 variables  
# Confirm the number of obs and variables.  
  
names(ppt.df) <- c("Time",  
 "Group Name",  
 "Birthdate",  
 "Live in County",  
 "Home County",  
 "Grade",  
 "Race not reported",  
 "White",  
 "Black",  
 "Asian",  
 "Native American",  
 "Other Race",  
 "Hispanic",  
 "Language",  
 "Other Language",  
 "Gender not reported",  
 "Woman",  
 "Man",  
 "Transgender",  
 "Nonbinary",  
 "Other Gender",  
 "Sexual Identity",  
 "Other Sexual Identity",  
 "Oral Sex",  
 "Vaginal Sex",  
 "Anal Sex",  
 "Pregnancy",  
 "Sex using alcohol",  
 "Sex using drugs",  
 "Never had sex",  
 "Ever had sex",  
 "No barrier method",  
 "No birth control",  
 "EC",  
 "IUD",  
 "Implant",  
 "Shot",  
 "Ring",  
 "Patch",  
 "Pill",  
 "Barrier Method",  
 "Not sure",  
 "Not reported",  
 "Other Method",  
 "Group leader understood me",  
 "Group mates respect each other",  
 "I felt judged\*",  
 "Group leader knew the material",  
 "I understood the material",  
 "Group leader made the activities fun",  
 "I felt free to speak",  
 "I felt uncomfortable participating\*",  
 "Topics I will remember",  
 "Group did this well",  
 "Improvements",  
 "First name",  
 "Email address",  
 "Phone number",  
 "Future",  
 "Healthy Relationship",  
 "Dating Abuse",  
 "Consent",  
 "PrEP",  
 "abstinence-neither",  
 "abstinence-pregnancy",  
 "abstinence-STI",  
 "birth control-neither",  
 "birth control-pregnancy",  
 "birth control-STI",  
 "external condom-neither",  
 "external condom-pregnancy",  
 "external condom-STI",  
 "lambskin condom-neither",  
 "lambskin condom-pregnancy",  
 "lambskin condom-STI",  
 "internal condom-neither",  
 "internal condom-pregnancy",  
 "internal condom-STI",  
 "withdrawal-neither",  
 "withdrawal-pregnancy",  
 "withdrawal-STI",  
 "Store the condom",  
 "Open the package",  
 "Pinch the tip",  
 "Unroll the condom",  
 "Hold the base",  
 "STI infection",  
 "Water-based lubricants",  
 "EC up to 5 days",  
 "EC without prescription",  
 "Prepared to say no if not ready",  
 "Prepared to use a barrier method",  
 "Prepared to say no if no barrier method",  
 "Prepared to use birth control",  
 "Prepared to go to a doctor for birth control")

* Recode missing data:

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

#3a. Clean the dichotomous (CATA/MRdum) variables: - 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.

# SurveyMonkey Quirk: Even though each check in "Check all that apply" (multiple response dummies) is dichotomous or binary (i.e., zeroes and ones), selecting the "none of the above" or "prefer not to answer" option/column is marked as "zero". Also, the 4th option/column, if checked, is marked as "four". With dummy variables, zero = "no" and one = "yes", which helps with summing each option. The function below converts the CATA/MR into dummy variables. But sometimes, zero means yes. So convert those "yeses" to one. Then convert remaining blanks to zero  
  
#cleanIt <- function(vec){  
# chars <- as.character(vec)  
# chars[is.na(chars)] <- "0"  
# chars[chars==""] <- "0"  
# chars[chars!="0"] <- "1"  
# return(as.numeric(chars))  
#}  
  
cleanIt <- function(vec){  
 chars <- as.character(vec)  
 chars[chars!=""] <- "1" #if not blank (including zeroes), replace with "one"  
 chars[is.na(chars)] <- "0" #if missing, replace with zero  
 chars[chars==""] <- "0" #if blank, replace with zero  
 return(as.numeric(chars)) #convert values to numbers  
}

* Use list apply (lapply) to repeat the “cleanIt” function to those columns, i.e., your vectors.

ppt.df[7:11] <- lapply(ppt.df[7:11], cleanIt) #racial identity options  
ppt.df[16:20] <- lapply(ppt.df[16:20], cleanIt) #gender identity options  
ppt.df[30:43] <- lapply(ppt.df[30:43], cleanIt) #last time sex options  
  
ppt.df[64:81] <- lapply(ppt.df[64:81], cleanIt) #if used correctly Qs

# 3b. Clean the nominal (categorical, 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)

labelgrade <- c("6th Grade",  
 "7th Grade",  
 "8th Grade",  
 "9th Grade",  
 "10th Grade",  
 "11th Grade",  
 "12th Grade",  
 "GED Program",  
 "Technical Training or College",  
 "Not in school",  
 "Prefer not to answer")  
labelfreq <- c("Prefer not to answer",  
 "Never",  
 "Not sure",  
 "Within the last 30 days",  
 "Ever (more than 30 days ago")  
labelnalikert <- c("Not applicable",  
 "Disagree",  
 "Not Sure",  
 "Agree",  
 "Prefer not to answer")  
labellikert <- c("Disagree",  
 "Not Sure",  
 "Agree",  
 "Prefer not to answer")  
  
#demographics  
  
# Convert column/variable to "date" class (%m = 2-digit month; %d = 2-digit day; %[capital]Y = 4-digit year)(https://www.statmethods.net/input/dates.html; https://www.statology.org/subset-by-date-range-in-r/)  
ppt.df$Birthdate <- as.Date(ppt.df$Birthdate, "%m/%d/%Y")  
  
# create a variable, called "age" in the dataframe "ppt.df" that is a calculated age in years, based on Q5 (P's DOB), using the age\_calc function in the eeptools library  
# install.packages("eeptools")  
library(eeptools)

## Loading required package: ggplot2

ppt.df$age <- floor(age\_calc(ppt.df$Birthdate, units = "years")) #make sure your date variable is formatted  
  
ppt.df$Grade <- factor(ppt.df$Grade,  
 levels = c(1,2,3,4,5,6,7,8,9,10,11),  
 labels = labelgrade)  
ppt.df$Time <- factor(ppt.df$Time,  
 levels = c(1,2),  
 labels = c("Before", "After"))  
ppt.df$Hispanic <- factor(ppt.df$Hispanic,  
 levels = c(0,1,2),  
 labels = c("Prefer not to answer",  
 "Yes",  
 "No"))  
ppt.df$Language <- factor(ppt.df$Language,  
 levels = c(0,1,2),  
 labels = c("Prefer not to answer",  
 "English",  
 "Spanish"))  
ppt.df$`Sexual Identity` <- factor(ppt.df$`Sexual Identity`,  
 levels = c(0,1,2),  
 labels = c("Prefer not to answer",  
 "LGBQ",  
 "Straight"))  
#experiences  
  
#the levels are re-ordered to how I want them to appear in the legend with "prefer not to answer" on one end or the other of the Likert scale  
ppt.df$`Oral Sex` <- factor(ppt.df$`Oral Sex`,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$`Vaginal Sex` <- factor(ppt.df$`Vaginal Sex`,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$`Anal Sex` <- factor(ppt.df$`Anal Sex`,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$Pregnancy <- factor(ppt.df$Pregnancy,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$`Sex using alcohol` <- factor(ppt.df$`Sex using alcohol`,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$`Sex using drugs` <- factor(ppt.df$`Sex using drugs`,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
#thoughts  
  
ppt.df$`Prepared to say no if not ready` <- factor(ppt.df$`Prepared to say no if not ready`,  
 levels = c(4,1,2,3,5),  
 labels = labelnalikert,  
 ordered = FALSE)  
ppt.df$`Prepared to use a barrier method` <- factor(ppt.df$`Prepared to use a barrier method`,  
 levels = c(4,1,2,3,5),  
 labels = labelnalikert,  
 ordered = FALSE)  
ppt.df$`Prepared to say no if no barrier method` <- factor(ppt.df$`Prepared to say no if no barrier method`,  
 levels = c(4,1,2,3,5),  
 labels = labelnalikert,  
 ordered = FALSE)  
ppt.df$`Prepared to use birth control` <- factor(ppt.df$`Prepared to use birth control`,  
 levels = c(4,1,2,3,5),  
 labels = labelnalikert,  
 ordered = FALSE)  
ppt.df$`Prepared to go to a doctor for birth control` <- factor(ppt.df$`Prepared to go to a doctor for birth control`,  
 levels = c(4,1,2,3,5),  
 labels = labelnalikert,  
 ordered = FALSE)  
  
#group satisfaction  
  
ppt.df$`Group leader understood me` <- factor(ppt.df$`Group leader understood me`,  
 levels = c(1,2,3,4),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`Group mates respect each other` <- factor(ppt.df$`Group mates respect each other`,  
 levels = c(1,2,3,4),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`I felt judged\*` <- factor(ppt.df$`I felt judged`,  
 levels = c(3,2,1,4), # reverse-coded  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`Group leader knew the material` <- factor(ppt.df$`Group leader knew the material`,  
 levels = c(1,2,3,4),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`I understood the material` <- factor(ppt.df$`I understood the material`,  
 levels = c(1,2,3,4),  
 labels = labellikert,  
 ordered = FALSE )  
ppt.df$`Group leader made the activities fun` <- factor(ppt.df$`Group leader made the activities fun`,  
 levels = c(1,2,3,4),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`I felt free to speak` <- factor(ppt.df$`I felt free to speak`,  
 levels = c(1,2,3,4),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`I felt uncomfortable participating\*` <- factor(ppt.df$`I felt uncomfortable participating`,  
 levels = c(3,2,1,4), # reverse-coded  
 labels = labellikert,  
 ordered = FALSE)

# 3c. 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). (<https://stackoverflow.com/questions/6328771/changing-values-when-converting-column-type-to-numeric>; <https://www.geeksforgeeks.org/convert-factor-to-numeric-and-numeric-to-factor-in-r-programming/>) Convert the variable from factor to character (e.g., a ten) and then to numeric.)

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

# 3d. 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:  
# ppt.df$`I felt judged` <- as.numeric(as.character(ppt.df$`I felt judged`))  
# ppt.df$`I felt judged` <- 4-ppt.df$`I felt judged` # on an ordered scale of 1 to 3

# 4. Select/subset specific variables/columns or observations/rows.

# Optional: select the 1st through 3rd columns/variables and start them with Q and separate them with a space  
# myvars <- paste("Q", 1:3, sep="")  
# newdata <- mydata[myvars]  
# or only columns/variables 1 and 5 through 10  
# newdata <- mydata[c(1,5:10)]  
  
# subset all rows/observations that have a column/variable value of greater than or equal to 20 OR less than 10, keeping the column/variables that are named.  
# newdata <- subset(mydata, age >= 20 | age < 10,  
# select=c(ID, Weight))  
  
# subset all men over the age of 25, keeping columns/variables weight, income and all columns/variables between them).  
# newdata <- subset(mydata, sex=="m" & age > 25,  
# select=weight:income)  
  
ppt.df$`Group Name` <- tolower(ppt.df$`Group Name`) #lowercase that group name to standardize it  
  
demographics.df <- ppt.df [ppt.df$`Group Name`=="payc" & ppt.df$Time=="Before",c(1:44,91:96)] #demographics/experiences & thoughts from pre-test for a specific group  
satisfaction.df <- ppt.df [ppt.df$`Group Name`=="payc" & ppt.df$Time=="After",c(45:58)] #post-test only for a specific group  
ksa.df <- ppt.df [ppt.df$`Group Name`=="payc" & ppt.df$Time=="Before" | ppt.df$Time=="After",c(1,59:90)] #KSAs before and after EBP for a specific group with identifier   
thoughts.df <- ppt.df[ppt.df$`Group Name`=="payc" & ppt.df$Time=="Before" | ppt.df$Time=="After",c(1, 91:95)] #thoughts before and after EBP for a specific group with identifier  
  
#remove non-answers from ordinal scales  
experience <- demographics.df[,24:29]  
experience[experience=="Prefer not to answer"] <- NA   
experience <- na.omit(experience)  
  
#thoughts <- demographics.df[,45:49]  
thoughts.df[thoughts.df=="Prefer not to answer"] <- NA  
thoughts.df[thoughts.df=="Not applicable"] <- NA  
thoughts.df <- na.omit(thoughts.df)

# 5. Compute frequencies and percentages for each multi-response category; categorical variable.

age <- data.frame(table(demographics.df$age))  
names(age) <- c("Demographics","Freq")  
grade <- data.frame(table(demographics.df$Grade))  
names(grade) <- c("Demographics","Freq")  
  
mrdumgender <- data.frame(Freq=colSums(demographics.df[16:20]))  
mrdumrace <- data.frame(Freq=colSums(demographics.df[7:11]))  
mrdumlastsex <- data.frame(Freq=colSums(demographics.df[30:43]))  
  
# how to combine with different number of columns? will need to manually create tables for MRDUMs  
  
gender <- data.frame(Demographics=c("Male","Female","Transgender","Does not identify","Gender not reported"),Freq=c(6,4,0,0,0))  
race <- data.frame(Demographics=c("White","Black","Asian","Alaska Native", "Native Hawaiian","Multiracial", "Race not Reported"),Freq=c(0,9,0,0,0,1,0))  
  
payc021621 <- rbind.data.frame(gender,race,age,grade)  
  
library(sjPlot)

## Install package "strengejacke" from GitHub (`devtools::install\_github("strengejacke/strengejacke")`) to load all sj-packages at once!

tab\_df(payc021621,  
 title = "Reach for Group PAYC021621",  
 file = "PAYC021621 Demographics Table.doc")

Reach for Group PAYC021621

Demographics

Freq

Male

6

Female

4

Transgender

0

Does not identify

0

Gender not reported

0

White

0

Black

9

Asian

0

Alaska Native

0

Native Hawaiian

0

Multiracial

1

Race not Reported

0

12

5

14

5

6th Grade

3

7th Grade

2

8th Grade

4

9th Grade

1

10th Grade

0

11th Grade

0

12th Grade

0

GED Program

0

Technical Training or College

0

Not in school

0

Prefer not to answer

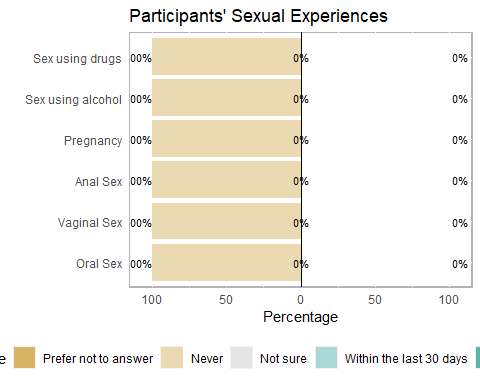
0

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

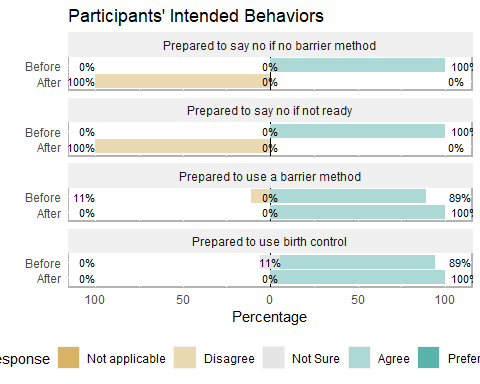
library(likert)

## Loading required package: xtable

experiencelkt <- likert(items=experience)  
plot(experiencelkt, positive.order = TRUE) + ggtitle("Participants' Sexual Experiences")



ggsave("PAYC021621 Experiences.png", width = 9, height = 2.5)  
  
thoughtslkt <- likert(thoughts.df[,2:5], grouping = thoughts.df$Time)  
plot(thoughtslkt, group.order = c("Before","After")) + ggtitle("Participants' Intended Behaviors")



ggsave("PAYC021621 Thoughts.png", width = 9, height = 6)