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 PrePostTest 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 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.)

ppt.df <- raw.df[3:nrow(raw.df),] #11 obs  
# Confirm the number of obs and variables.

* 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.

cleanIt <- function(vec){  
 chars <- as.character(vec)  
 chars[is.na(chars)] <- "0"  
 chars[chars==""] <- "0"  
 chars[chars!="0"] <- "1"  
 return(as.numeric(chars))  
}

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

ppt.df[18:23] <- lapply(ppt.df[18:23], cleanIt) #racial identity options  
ppt.df[27:32] <- lapply(ppt.df[27:32], cleanIt) #gender identity options  
ppt.df[42:55] <- lapply(ppt.df[42:55], cleanIt) #last time sex options

# 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")  
labellikert <- c("Not applicable",  
 "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$Q5 <- as.Date(ppt.df$Q5, "%m/%d/%Y")  
# install.packages("eeptools")  
library(eeptools)

## Loading required package: ggplot2

ppt.df$age <- floor(age\_calc(ppt.df$Q5, units = "years"))  
  
ppt.df$Q8 <- factor(ppt.df$Q8,  
 levels = c(1,2,3,4,5,6,7,8,9,10,11),  
 labels = labelgrade)  
ppt.df$Q15 <- factor(ppt.df$Q15,  
 levels = c(0,1,2),  
 labels = c("Prefer not to answer",  
 "Yes",  
 "No"))  
ppt.df$Q16 <- factor(ppt.df$Q16,  
 levels = c(0,1,2),  
 labels = c("Prefer not to answer",  
 "English",  
 "Spanish"))  
ppt.df$Q24 <- factor(ppt.df$Q24,  
 levels = c(0,1,2),  
 labels = c("Prefer not to answer",  
 "LGBQ",  
 "Straight"))  
#experiences  
ppt.df$Q26 <- factor(ppt.df$Q26,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$Q27 <- factor(ppt.df$Q27,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$Q28 <- factor(ppt.df$Q28,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$Q29 <- factor(ppt.df$Q29,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$Q30 <- factor(ppt.df$Q30,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$Q31 <- factor(ppt.df$Q31,  
 levels = c(5,4,3,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
#thoughts  
ppt.df$Q100 <- factor(ppt.df$Q100,  
 levels = c(4,1,2,3,5),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$Q101 <- factor(ppt.df$Q101,  
 levels = c(4,1,2,3,5),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$Q102 <- factor(ppt.df$Q102,  
 levels = c(4,1,2,3,5),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$Q103 <- factor(ppt.df$Q103,  
 levels = c(4,1,2,3,5),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$Q104 <- factor(ppt.df$Q104,  
 levels = c(4,1,2,3,5),  
 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 to character and then 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:  
# 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

# 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$Q4 <- tolower(ppt.df$Q4) #lowercase that group name  
  
paycpre.df <- ppt.df [ppt.df$Q4=="payc" & ppt.df$Q3==1,c(12:55,109:114)]#pre-test only  
paycpost.df <- ppt.df [ppt.df$Q4=="payc" & ppt.df$Q3==2,] #posttest only  
paycppt.df <- ppt.df [ppt.df$Q4=="payc" & ppt.df$Q3==1 | ppt.df$Q3==2,] #all tests for a group  
  
names(paycpre.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",  
 "Experiences",  
 "Oral Sex",  
 "Vaginal Sex",  
 "Anal Sex",  
 "Pregnancy",  
 "Sex while intoxicated",  
 "Sex using drugs",  
 "Last time",  
 "Ever had sex",  
 "No barrier method",  
 "No birth control",  
 "EC",  
 "IUD",  
 "Implant",  
 "Shot",  
 "Ring",  
 "Patch",  
 "Pill",  
 "Barrier Method",  
 "Not sure",  
 "Not reported",  
 "Other Method",  
 "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",  
 "Age")  
  
#remove non-answers from ordinal scales  
experience <- paycpre.df[,24:29]  
experience[experience=="Prefer not to answer"] <- NA   
experience <- na.omit(experience)  
  
thoughts <- paycpre.df[,45:49]  
thoughts[thoughts=="Prefer not to answer"] <- NA  
thoughts[thoughts=="Not applicable to me"] <- NA  
thoughts <- na.omit(thoughts)

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

age <- data.frame(table(paycpre.df$Age))  
names(age) <- c("Reach","Freq")  
grade <- data.frame(table(paycpre.df$Grade))  
names(grade) <- c("Reach","Freq")  
  
mrdumgender <- data.frame(Freq=colSums(paycpre.df[16:21]))  
mrdumrace <- data.frame(Freq=colSums(paycpre.df[7:12]))  
mrdumlastsex <- data.frame(Freq=colSums(paycpre.df[32:43]))  
  
# how to combine with different number of columns? will need to manually create tables for MRDUMs  
  
gender <- data.frame(Reach=c("Male","Female","Transgender","Does not identify","Gender not reported"),Freq=c(6,4,0,0,0))  
race <- data.frame(Reach=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)

## Learn more about sjPlot with 'browseVignettes("sjPlot")'.

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

Demographics for Group PAYC021621

Reach

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

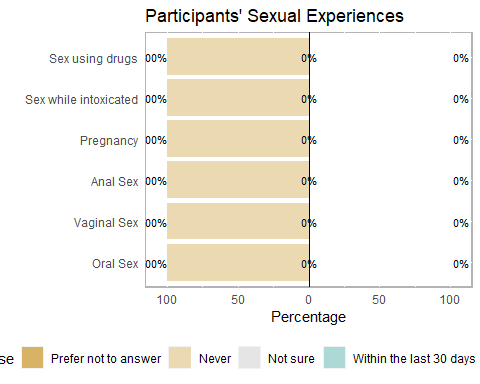
0

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

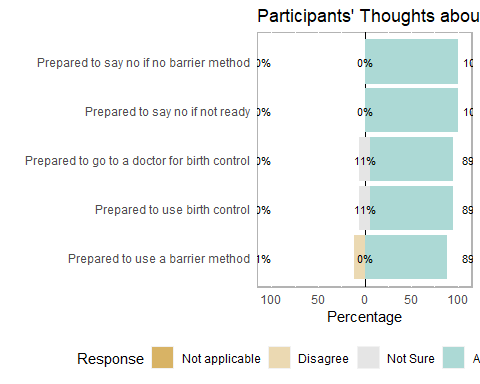
library(likert)

## 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(items = thoughts)  
plot(thoughtslkt, positive.order = TRUE) + ggtitle("Participants' Thoughts about Future Actions")



ggsave("PAYC021621 Thoughts.png", width = 9, height = 2.5)