Section Data: Pre-Test Summary; Reach (P/PT), Dosage & Fidelity (QPR)

Summarizing Data by Group

Josephine McKelvy

March 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 documentation/explanation that is commented out.

# 0. First time:

* Create a new project with version control (Git) on the top right corner. # 0. Every time:
* Select your existing project. Pull any changes from GitHub, using the Git tab in the Environment panel/window.
* Copy the template .rmd into a project folder and upload the dataset. # 0. Analog prep (first time):
* have a copy of the survey with question numbers and recode values of every closed-ended answer choice–your codebook
* log into oahpmdata.net to see what data you’ll need to report (and how) to OPA

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

getwd()

## [1] "C:/Users/josep/Documents/ayc/OAHPMdata - PPTs & QPRs"

# 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.

rawppt.df <- read.csv("Participant Pre- Post-Test Survey.csv", stringsAsFactors = TRUE)  
# Export .CSV from your survey platform and confirm numeric values, NOT choice text.   
# Make sure filters are turned on or off.

# 3. Clean the analytic dataset.

* I created analytic datasets (“ppt.df” for the pre/post test) 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 <- rawppt.df[2:nrow(rawppt.df),c(12:69,77:115)] #97 variables  
# Confirm the number of obs and variables in the environment with the survey codebooks.  
  
# Rename the variables/columns:  
names(ppt.df) <- c("Group Name", "Participant", "Time",  
 "Alamance County",  
 "Other County (text)",  
 "Birthdate",  
 "Grade",  
 "Race not reported",  
 "White",  
 "Black",  
 "Asian",  
 "Pacific Islander",  
 "Native American",  
 "Other Race (text)",  
 "Hispanic",  
 "Language",  
 "Other Language (text)",  
 "Gender not reported",  
 "Woman",  
 "Man",  
 "Transgender",  
 "Nonbinary/Does not identify",  
 "Other Gender (text)",  
 "Sexual Identity",  
 "Other Sexual Identity (text)",  
 "Oral Sex",  
 "Vaginal Sex",  
 "Anal Sex",  
 "Pregnancy",  
 "Sex using alcohol",  
 "Sex using drugs",  
 "Never had sex",  
 "(Duplicate hidden category)",  
 "Sex with no barrier method",  
 "Sex with no birth control",  
 "EC",  
 "IUD",  
 "Implant",  
 "Shot",  
 "Ring",  
 "Patch",  
 "Pill",  
 "Barrier Method",  
 "Not sure",  
 "Prefer not to answer",  
 "Other Method (text)",  
 "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")  
  
# Lowercase that group name to standardize it  
  
#install.packages("data.table")  
library(data.table)  
ppt.df$`Group Name` <- tolower(ppt.df$`Group Name`)   
  
# Convert birthdate 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") #pre-test only

* Recode missing data:

# Optional: Recode "prefer not to answer" as missing in Pre/Post Test (may drop too many cases in listwise deletion):  
#library(naniar)  
  
#replace\_with\_na\_at(data = ppt.df,  
# .vars = c( "Oral Sex",  
# "Vaginal Sex",  
# "Anal Sex",  
# "Pregnancy",  
# "Sex using alcohol",  
# "Sex using drugs"),  
# condition = ~.x == 5) #"prefer not to answer"  
#replace\_with\_na\_at(data = ppt.df,  
# .vars = c("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"),  
# condition = ~.x == 4) #"not applicable to me"  
#replace\_with\_na\_at(data = ppt.df,  
# .vars = c("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"),  
# condition = ~.x == 5) #"prefer not to answer"  
#replace\_with\_na\_at(data = ppt.df,  
# .vars = c( "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\*"),  
# condition = ~.x == 4) #"prefer not to answer"  
  
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[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.

#Pre/Post Tests  
ppt.df[8:13] <- lapply(ppt.df[8:13], cleanIt) #pre: racial identity options  
ppt.df[18:22] <- lapply(ppt.df[18:22], cleanIt) #pre: gender identity options  
ppt.df[32:45] <- lapply(ppt.df[32:45], cleanIt) #pre: last time sex options  
ppt.df[66:83] <- lapply(ppt.df[66:83], cleanIt) #both: if used correctly Qs

# 3b. Clean the nominal (categorical, ordinal) variables:

* For each multiple choice statement, change the response values to factor, specify sequence & values of 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.)
* Optional: name the sets of labels for each categorical response that you’ll reuse

# P/PT Demographics: pre-test only  
  
ppt.df$Time <- factor(ppt.df$Time,  
 levels = c(1,2),  
 labels = c("Before", "After"))  
ppt.df$`Alamance County` <- factor(ppt.df$`Alamance County`,  
 levels = c(1,0),  
 labels = c("Yes","No/Not Reported"))  
ppt.df$Grade <- factor(ppt.df$Grade,  
 levels = c(1,2,3,4,5,6,7,9,10,11,12),  
 labels = 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"))  
ppt.df$Hispanic <- factor(ppt.df$Hispanic,  
 levels = c(1,2,0),  
 labels = c("Hispanic/Latinx",  
 "Non-Hispanic/Latinx",  
 "Prefer not to answer"))  
ppt.df$Language <- factor(ppt.df$Language,  
 levels = c(1,3,0),  
 labels = c("English",  
 "Spanish",  
 "Prefer not to answer/Other"))  
ppt.df$`Sexual Identity` <- factor(ppt.df$`Sexual Identity`,  
 levels = c(1,2,0),  
 labels = c("LGBQ",  
 "Straight",  
 "Prefer not to answer/Other"))  
  
# Pre-Test Experiences: The labels and 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, as opposed to the sequence in the paper survey/codebook: 1=last 30 days; 2=more than 30 days ago; 3=not sure; 4=never; 5=prefer not to answer (replaced with NA)  
  
labelfreq <- c("Never",  
 "Not sure",  
 "Prefer not to answer",  
 "Within the last 30 days",  
 "Ever (more than 30 days ago")  
  
ppt.df$`Oral Sex` <- factor(ppt.df$`Oral Sex`,  
 levels = c(4,3,5,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$`Vaginal Sex` <- factor(ppt.df$`Vaginal Sex`,  
 levels = c(4,3,5,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$`Anal Sex` <- factor(ppt.df$`Anal Sex`,  
 levels = c(4,3,5,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$Pregnancy <- factor(ppt.df$Pregnancy,  
 levels = c(4,3,5,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$`Sex using alcohol` <- factor(ppt.df$`Sex using alcohol`,  
 levels = c(4,3,5,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
ppt.df$`Sex using drugs` <- factor(ppt.df$`Sex using drugs`,  
 levels = c(4,3,5,1,2),  
 labels = labelfreq,  
 ordered = FALSE)  
  
# P/PT Thoughts codebook: 1=disagree; 2=not sure; 3=agree; 4=not applicable to me (replaced with NA); 5=prefer not to answer  
  
labellikert <- c("Disagree",  
 "Not Sure",  
 "Agree")  
  
ppt.df$`Prepared to say no if not ready` <- factor(ppt.df$`Prepared to say no if not ready`,  
 levels = c(1,2,3),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`Prepared to use a barrier method` <- factor(ppt.df$`Prepared to use a barrier method`,  
 levels = c(1,2,3),  
 labels = labellikert,  
 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(1,2,3),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`Prepared to use birth control` <- factor(ppt.df$`Prepared to use birth control`,  
 levels = c(1,2,3),  
 labels = labellikert,  
 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(1,2,3),  
 labels = labellikert,  
 ordered = FALSE)  
  
# Post-Test Group satisfaction codebook: 1=disagree; 2=not sure; 3=agree; 4=prefer not to answer   
  
ppt.df$`Group leader understood me` <- factor(ppt.df$`Group leader understood me`,  
 levels = c(1,2,3),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`Group mates respect each other` <- factor(ppt.df$`Group mates respect each other`,  
 levels = c(1,2,3),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`I felt judged\*` <- factor(ppt.df$`I felt judged`,  
 levels = c(3,2,1), # 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),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`I understood the material` <- factor(ppt.df$`I understood the material`,  
 levels = c(1,2,3),  
 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),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`I felt free to speak` <- factor(ppt.df$`I felt free to speak`,  
 levels = c(1,2,3),  
 labels = labellikert,  
 ordered = FALSE)  
ppt.df$`I felt uncomfortable participating\*` <- factor(ppt.df$`I felt uncomfortable participating`,  
 levels = c(3,2,1), # reverse-coded  
 labels = labellikert,  
 ordered = FALSE)

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

# Create a variable, called "age" in the individual "ptdemographics.df" dataframes (because this function does not like NAs and you do not want to listwise delete most of your dataset). "Age" is a calculated age in years, based on Q5 (P's DOB), using the age\_calc function in the eeptools library  
  
# Pre/Post Test by group if group name is exact (Update group name below):  
ptdemographics.df <- ppt.df[ppt.df$`Group Name`=="sabgc050121" & ppt.df$Time=="Before",c(1,3,4,6:13,15,16,18:22,24)] #8 Demographic variables   
ptdemographics.df <- na.omit(ptdemographics.df)  
  
#install.packages("eeptools")  
library(eeptools)

## Loading required package: ggplot2

ptdemographics.df$Age <- floor(age\_calc(ptdemographics.df$Birthdate, units = "years")) #make sure your date variable is formatted  
  
# Optional subsets:  
#post.df <- ppt.df[ppt.df$Time=="After", c(1:3,47:60,61:97)]# optional subset of identifiers, satisfaction, contact info to post-test  
#pre.df <- ppt.df[ppt.df$Time=="Before", c(1,3,4,6:13,15,16,18:22,24,26:45)] #optional subset of demographics, experiences  
#pre.df <- na.omit(pre.df)  
  
ptexperlkt.df <- ppt.df[ppt.df$`Group Name`=="sabgc050121" & ppt.df$Time=="Before",c(26:31)] #6 Experience variables (likert)  
ptexpermrd.df <- ppt.df[ppt.df$`Group Name`=="sabgc050121" & ppt.df$Time=="Before",c(32,34:45)] #13 Experience options (MR dummy excluding "Other method" (text))  
#how to make a frequency table of mrd?  
  
ptthoughts.df <- ppt.df[ppt.df$`Group Name`=="sabgc050121",c(1,3,93:97)] #5 Thought variables (likert) for Pre AND Post  
  
ptksa.df <- ppt.df[ppt.df$`Group Name`=="sabgc050121" & ppt.df$Time=="Before" | ppt.df$Time=="After",c(1:3,61:92)] #16 KSA variables with 32 options  
ptsatisfaction.df <- ppt.df[ppt.df$`Group Name`=="sabgc050121" & ppt.df$Time=="After",c(1,47:57)] #11 Group variables on Post

# 5. Compute frequencies and percentages for each multi-response category for tables.

# Convert MR Dummy to categorical variables:  
  
ptdemographics.df$gender\_cat <- apply(ptdemographics.df[14:18], 1, function(x) {ifelse(sum(x) > 1, "Multigender", names(x[x != 0]))})  
gender <- data.frame(table(ptdemographics.df$gender\_cat))  
names(gender) <- c("Demographics","Count")  
  
ptdemographics.df$race\_cat <- apply(ptdemographics.df[6:11], 1, function(x) {ifelse(sum(x) > 1, "Multiracial", names(x[x != 0]))})  
race <- data.frame(table(ptdemographics.df$race\_cat))  
names(race) <- c("Demographics","Count")  
  
age <- data.frame(table(ptdemographics.df$Age))  
names(age) <- c("Demographics","Count")  
  
grade <- data.frame(table(ptdemographics.df$Grade))  
names(grade) <- c("Demographics","Count")  
  
reach <- rbind.data.frame(gender,race,age,grade)  
  
#https://www.youtube.com/watch?v=vQsdJBm0fLw  
#mrdumlastsex <- data.frame(Freq=colSums(ptexpermrd.df[2:14]),  
# Percent=(colSums(ptexpermrd.df[2:14])/nrow(ptexpermrd.df[2:14]))\*100)   
#mrdumlastsex <- mrdumlastsex[order(-mrdumlastsex$Freq),] #sorted, but won't plot in this order  
  
library(sjPlot)

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

tab\_df(reach,  
 title = "Reach for: SABGC050121",  
 file = "SABGC050121 - Demographics Table.doc")

Reach for: SABGC050121

Demographics

Count

Man

29

Black

22

White

6

13

1

14

2

15

7

16

3

17

15

24

1

6th Grade

0

7th Grade

0

8th Grade

3

9th Grade

7

10th Grade

4

11th Grade

14

12th Grade

0

GED Program

0

Technical Training or College

0

Not in school

1

Prefer not to answer

0

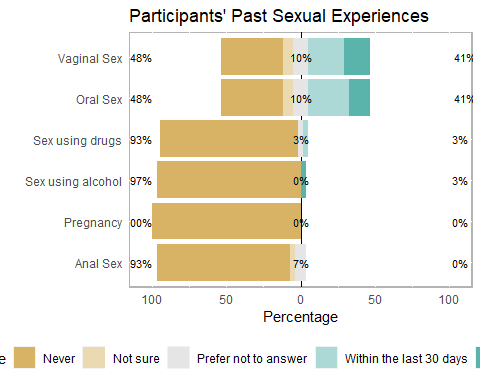
#freq(experlkt.df)  
#freq(thoughts.df)

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

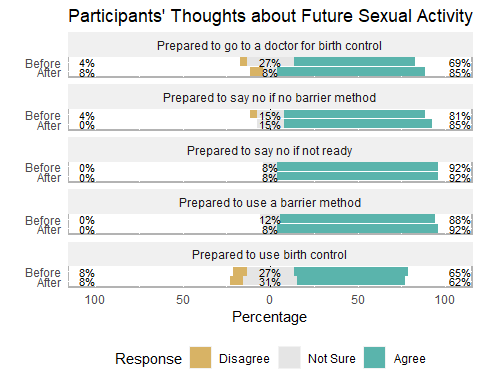
#remove non-answers from ordinal scales  
ptexperlkt.df <- na.omit(ptexperlkt.df) #pre-test  
ptthoughts.df <- na.omit(ptthoughts.df) #pre AND post  
ptsatisfaction.df <- na.omit(ptsatisfaction.df) #post-test  
  
library(likert)

## Loading required package: xtable

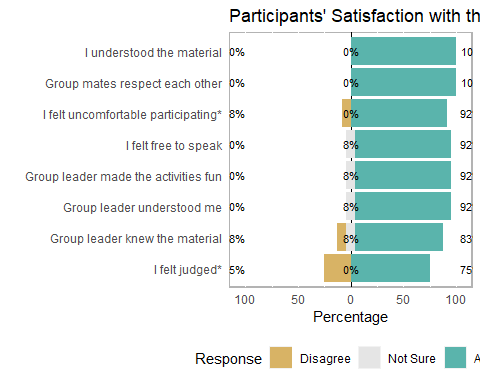
experiencelkt <- likert(items=ptexperlkt.df)  
plot(experiencelkt, positive.order = TRUE) + ggtitle("Participants' Past Sexual Experiences")



ggsave("SABGC050121 - Experiences - Pre.png", width = 9, height = 2.5)  
#ggpubr::ggsummarystats(ptexperlkt.df)  
#experct <- summary(experiencelkt)  
  
#dotchart(as.matrix(mrdumlastsex,labels = row.names(mrdumlastsex)))  
  
#install.packages("ggpubr")  
#library(ggpubr)  
#ggballoonplot(mrdumlastsex, fill = "value") + scale\_fill\_viridis\_c(option = "C")  
  
#ggplot(mrdumlastsex, aes(x=Percent, y=rownames(mrdumlastsex))) +  
# geom\_point(size = 1.5, alpha = .3) +  
# labs(title = "Contraception Use at Last Sexual Activity") +  
# theme\_minimal() +  
# theme(axis.title = element\_blank(),  
# panel.grid.major.x = element\_blank(),  
# panel.grid.minor = element\_blank(),  
# plot.title = element\_text(size = 20, margin = margin(b = 10)),  
# plot.subtitle = element\_text(size = 10, color = "darkslategrey", margin = margin(b = 25)),  
# plot.caption = element\_text(size = 8, margin = margin(t = 10), color = "grey70", hjust = 0))  
#ggsave("Test.png")  
  
thoughtlkt <- likert(ptthoughts.df[,3:7], grouping = ptthoughts.df$Time)  
plot(thoughtlkt, group.order = c("Before","After")) + ggtitle("Participants' Thoughts about Future Sexual Activity")



ggsave("SABGC050121 - Thoughts over time.png", width = 9, height = 4.5)  
  
satisflkt <- likert(ptsatisfaction.df[,2:9])  
plot(satisflkt) + ggtitle("Participants' Satisfaction with the Group")



ggsave("SABCG050121 - Satisfaction.png", width = 9, height = 3.25)