Code for McClanahan et al. GPIR Submission

April 30, 2018

## How to Use this Document:

This document contains the code that we used to conduct our analyses (for both the main text and the supplemental materials). It contains both chunks of R code and annotation which explains what the coding is doing. In addition to the explanation between the code chunks, there is line-by-line annotation (denoted with a #) is embedded within the code. For example:

sample <- "this is an example line of R code"  
#this is a comment

In addition to containing the code used, this document also contains all of the *results* that are generated by running this code in R. For example, see the expression below:

#when we type this:  
3\*4

## [1] 12

#the result (12) is also generated.

If you’d like to run these analyses in R yourself, you can also open the .Rmd version of this file in R. This will enable you to run all of the code in this document (please note that some file directories will need to be changed and package installation may need to be altered depending on what is already installed on your computer).

The layout of this document is roughly equivalent to the layout of the results in the paper. However, as with most projects, our data required some cleaning before it was ready for analysis. All data wrangling and preliminary analyses for this paper were conducted in R and are included in this document. Thus, for each study, there are sections of code that are used for cleaning/compiling, etc. Beyond the code for cleaning the data, we have divided the core analyses of the paper into 16 “chunks” for organizational purposes. Each chunk represents an essential analysis in the paper. For example, each of the three studies has four chunks: demographics, scales, manipulation check, and 3-way interaction (as this follows that layout of the paper; Study 1 is chunks 1-4; Study 2 is chunks 4-8; Study 3 is chunk 9-12). There are two chunks for the combined data: demographics and 3-way interaction (chunk 12-13). Finally, there is a chunk of code to produce the table of descriptive statistics and a chunk of code to analyze our post-test for (chunks 15 and 16, respectively).

Following the code for the main text is all the code used to produce the supplemental materials. The code for this section follows the organization of the supplemental materials.

## Preparing for Analyses:

In addition to base R, we used several different packages to help us easily perform specific functions within R. These packages are installed and loaded below. The function that each package serves in the paper is listed in the in-line comments below:

#installing necessary packages  
install.packages('dplyr') #this package makes it easier to sort, filter, and manipulate data  
install.packages('psych') #this packages makes it easy to calculate reliabilties (alphas)  
install.packages('lsr') #this package allows us to calculate the partial eta-squared on our ANOVA  
install.packages('broom') #this package allows us to easily turn regression results into a data frame  
install.packages('Hmisc') #I use this package for my correlation tables, but I do'nt library it until I need it because it changes some defaults in a way that I don't like  
install.packages("devtools") #this allows us to install packages from github (below)  
devtools::install\_github("jeff-hughes/reghelper") #this makes it easy to get the simple slopes from an interaction

After installing these packages, we load them into the workspace:

library(lsr)  
library(dplyr)  
library(psych)  
library(broom)  
library(reghelper)

Now we’re ready to look at the data, which I laid out study by study:

# Study 1 Analyses

Before doing anything with the Study 1 data, we need to import the data into R. We start with the .csv files we downloaded from Qualtrics. These .csv files come from Qualtrics with a variable labels row; this causes problems for R, so I removed this row from the file prior to importing. There are also 3 columns which were removed from the dataset. These columns represented write-in questions that probed for participants’ suspicion (given that our study relied on manipulation). I have removed these questions because they caused problems during upload (.csv files use commas for separating cells and the comments frequently contained commas, creating confusion).

Study 1 was actually comprised of Study 1a and Study 1b so there will be two files to upload here. If running these analyses on a different computer, the file directory below will need to be changed to reflect the location of the .csv files.

#Study 1a  
study1a <- read.csv(file="study 1a no labels.csv")  
  
#Study 1b  
study1b <- read.csv(file="study 1b no labels.csv")

We now have two data frames, study1a and study1b. These data frames comprise everyone who took our Study 1 survey. However, we are only interested in investigating participants who completed the study and consented to letting us use our data:

#Study 1a  
nrow(study1a) # we start with 379 observations in Study 1a, some of which are incomplete

## [1] 379

study1a <- study1a %>%  
 filter(Q65 == 1) #this filters out people who didn't complete the study or didn't let us use data  
  
nrow(study1a) #there are now 327 people in Study 1a

## [1] 327

#Study 1b  
nrow(study1b) #451 cases in Study 1b to start with

## [1] 451

study1b <- study1b %>%  
 filter(Q65 == 1) #filtering out people who didn't complete study or didn't consent to letting us use their data  
  
nrow(study1b) #there are now 402 people in Study 1b

## [1] 402

We now have two data frames with all the complete observations.

## Cleaning Studies 1a and 1b

There are a number of things we need to do get this raw data into a form where we can run analyses on it. We will do most of this after combining Study 1a and Study 1b to be one dataset, but there is one key difference between these datasets that needs to be changed before the merge: their condition variables.

For Study 1a, which has three condition, we will create a numeric condition variable which is coded as follows:  
Reputable condition = 3  
Neutral condition = 2  
Disreputable condition = 1

Here is the code:

#creating the condition variable from qualtric's randomization variable; high numbers = better reputation target  
study1a$condition[study1a$DO.BR.FL\_15 == "target does something good"] <- 3  
study1a$condition[study1a$DO.BR.FL\_15 == "target does something neutral"] <- 2  
study1a$condition[study1a$DO.BR.FL\_15 == "target does something bad"] <- 1  
  
#checking to make sure this worked:  
#each condition should have approximately the same N in each, and status should be highest in cond 3:  
study1a %>%   
 group\_by(condition) %>%  
 summarise(N = n(), "Mean Status" = mean(status)) #this worked as planned

## # A tibble: 3 x 3  
## condition N `Mean Status`  
## <dbl> <int> <dbl>  
## 1 1.00 106 3.70  
## 2 2.00 108 5.13  
## 3 3.00 113 5.73

Now we do a similar thing for Study 1b, but there are only 2 conditions here, so we used contrast coding (high status = 1, low status = -1):

study1b$cond[study1b$DO.BR.FL\_15 == "target does something good"] <- 1  
study1b$cond[study1b$DO.BR.FL\_15 == "target does something bad"] <- -1  
  
#checking to make sure that conditions are approximately equal in size, and status is high in condition 3  
study1b %>%  
 group\_by(cond) %>%  
 summarise(N = n(), "Mean Status" = mean(status))

## # A tibble: 2 x 3  
## cond N `Mean Status`  
## <dbl> <int> <dbl>  
## 1 -1.00 197 3.52  
## 2 1.00 205 5.80

## Merging Studies 1a and 1b

Now that the condition variables are created, we can merge Study 1a and Study 1b. Before merging, we will create a new study1a dataframe that only has the reputable and disreputable condition, which need to be contrast coded. The study1a dataset will now only have two conditions, but we will make a dataset called study1afull that retains the neutral condition (this will be used for analyses in the supplemental materials).

#filtering out the neutral condition  
study1afull <- study1a  
study1a <- study1a %>%  
 filter(condition == 3 | condition == 1) #new dataframe with just cond 1 and 3  
  
#creating a "cond" variable in study1a which is bad/good condition but contrast code (to match study 1b)  
study1a$cond = study1a$cond - 2 #subtracting 2 to make it a contrast code (from 1 and 3)  
  
#making the cond variables factors in each dataframe  
study1a$cond = as.factor(study1a$cond)   
study1b$cond = as.factor(study1b$cond)  
  
#checking to make sure it worked  
summary(study1a$cond) #condition N's are the same as the original variable

## -1 1   
## 106 113

summary(study1b$cond) #condition N's are the same as the original variable

## -1 1   
## 197 205

The age variable in Study 1a has some bizarre entries that will throw an error during the merge if not corrected. Additionally, R has somewhat stringent rules about variables of different types are merged, so the age variable needs to be changed from a factor to an integer. (You need to make it a character variable first to make this work correctly.)

#manually recode errors  
study1a$demo\_age[study1a$demo\_age == "OH"] <- NA  
study1a$demo\_age[study1a$demo\_age == 224] <- NA  
  
#factor -> character -> integer  
study1a$demo\_age <- as.integer(as.character(study1a$demo\_age))   
class(study1a$demo\_age) #class is now integer

## [1] "integer"

All of the relevant variables are ready to merge, so let’s make smaller dataframes that only contain the variables we will need for our analyses:

#selecting variables for the merge from study 1a  
study1amerge <- study1a %>%  
 select(V1,demo\_age, demo\_gend, starts\_with("ethnic"), attn, starts\_with("SDO"), status,   
 attn2, starts\_with("hypo"), cond) #study1a is ready to go  
  
#selecting variables for the merge from study 1b  
study1bmerge <- study1b %>%  
 rename(hypo\_1 = Q78, hypo\_2 = Q80) %>% #these variables aren't named in this study for some reason  
 select(V1,demo\_age, demo\_gend, starts\_with("ethnic"), attn, starts\_with("SDO"), status,   
 attn2, starts\_with("hypo"), cond)

Now we are ready to merge! We will ask R to create a new variable, source1aor1b in our new dataframe, which will tell us which study (1a or 1b) the observation came from.

#merging  
study1 <- bind\_rows("study1a" = study1amerge, "study1b" = study1bmerge, .id = "source1aor1b")  
class(study1$cond)

## [1] "factor"

#checking the merge... are the N's correct?  
nrow(study1) #621 rows, that's equal to 106 + 113 + 197 +205

## [1] 621

summary(study1$cond) #checking the conditions--they look correct (303 bad, 318 good)

## -1 1   
## 303 318

study1 %>%  
 group\_by(cond) %>%  
 summarise(mean(status)) #status is still higher in the reputable condition, everything looks good

## # A tibble: 2 x 2  
## cond `mean(status)`  
## <fct> <dbl>  
## 1 -1 3.58  
## 2 1 5.78

## Cleaning Study 1 Data

Now that we have a single dataset, we can go about prepping it for analyses. Specifically, we will need to create some scales (SDO, ethnic identification, formativeness and categorization). We also need to change ethnicity and gender into factors.

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#SDO  
#reverse code items:  
study1$SDO7\_5 <- 8-study1$SDO7\_5  
study1$SDO7\_6 <- 8-study1$SDO7\_6  
study1$SDO7\_7 <- 8-study1$SDO7\_7  
study1$SDO7\_8 <- 8-study1$SDO7\_8  
study1$SDO7\_13 <- 8-study1$SDO7\_13  
study1$SDO7\_14 <- 8-study1$SDO7\_14  
study1$SDO7\_15 <- 8-study1$SDO7\_15  
study1$SDO7\_16 <- 8-study1$SDO7\_16  
  
#create the scale  
study1$sdo <- study1 %>%  
 select(starts\_with("SDO7\_")) %>%  
 rowMeans(na.rm = TRUE)   
  
#checking for NAs (empty cases)--there shouldn't really be any because everyone should have at least some SDO values in this   
anyNA(study1$sdo) #false means no NAs

## [1] FALSE

#\_\_\_\_\_\_\_\_\_\_\_\_\_   
#ETHNIC ID   
#we want to use ethnic\_id2 & ethnic\_id3; those are the only ones in all studies.  
study1$ethnic <- rowMeans(subset(study1, select = c(ethnic\_id2, ethnic\_id3)), na.rm = TRUE) #making scale  
summary(study1$ethnic) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.500 4.000 3.896 5.500 7.000

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#FORMATIVENESS & CATEGORIZATION  
#hypo 1 and 2 are categorization; hypo 4-5, 11-12 are formativeness (but 11 and 12 are not included in Study 1a so we are not looking at them in this study)  
study1$form <- rowMeans(subset(study1, select = c(hypo\_4, hypo\_5), na.rm = TRUE))  
study1$cat <- rowMeans(subset(study1, select = c(hypo\_1, hypo\_2), na.rm = TRUE))  
summary(study1$form) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 3.500 4.000 3.714 4.000 7.000

summary(study1$cat) #1 NA, that's fine

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 1.000 3.000 4.000 3.532 4.000 7.000 1

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#MAKING ETHNIC ID & GENDER FACTORS  
study1$ethnic\_id <- factor(study1$ethnic\_id, labels = c("Black/African American", "Asian/Asian American/Pacific Islander", "White", "Latino/Hispanic American", "Middle Eastern/Arab American","Native American", "Biracial/Mixed race (please specify)", "Other (please specify)"))  
  
study1$demo\_gend2 <- factor(study1$demo\_gend, labels = c("male", "female"))

Great! The data is now ready for analyses!

## Analysis Chunk 1: Study 1 Demographics

First, let’s filter the dataset until we have only the participants we want, then look at the *N* and the demographics:

#Checking the N from each study  
  
#first, how many were in study 1a (all conditions?)  
nrow(study1afull)

## [1] 327

#and in just the reputable condition?  
study1 %>%  
 group\_by(source1aor1b) %>%  
 summarise(n()) #219 in study 1a, 402 in study 1b:

## # A tibble: 2 x 2  
## source1aor1b `n()`  
## <chr> <int>  
## 1 study1a 219  
## 2 study1b 402

nrow(study1) #making a total of 621 people in study 1

## [1] 621

#how many white participants?  
study1 %>%   
 filter(ethnic\_id == "White") %>%  
 nrow() #470 White participants

## [1] 470

#selecting white participants who completed the attention checks; final sample  
study1f <- study1 %>%  
 filter(ethnic\_id == "White" & attn ==5 & attn2 == 3) #making FINAL study 1  
nrow(study1f) #final sample is 459

## [1] 459

470-459 #11 people didn't pass attention checks

## [1] 11

#\_\_DEMOGRAPHICS\_\_  
#ages  
#checking for outliers  
study1f$demo\_age[study1f$demo\_age > 100 | study1f$demo\_age < 18] #there is one outlier (335 years old?)

## [1] 335

#removing outlier  
study1f$demo\_age[study1f$demo\_age > 100 | study1f$demo\_age < 18] <- NA   
  
#calulating mean age  
describe(study1f$demo\_age, na.rm = TRUE) #mean age is 35.90; sd = 12.08

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 458 35.9 12.08 33 34.58 10.38 18 70 52 0.91 0.1  
## se  
## X1 0.56

# calculating gender distribution  
summary(study1f$demo\_gend2) #207 males, 252 females;

## male female   
## 207 252

252/(252+207) #54.9% females

## [1] 0.5490196

## Analysis Chunk 2: Study 1 Scales

Now we’ll check to make sure the scales are robust:

#ethnic id correlation  
cor.test(study1f$ethnic\_id2, study1f$ethnic\_id3) #r = .71 p < .001

##   
## Pearson's product-moment correlation  
##   
## data: study1f$ethnic\_id2 and study1f$ethnic\_id3  
## t = 21.423, df = 457, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.6590215 0.7507403  
## sample estimates:  
## cor   
## 0.707852

#sdo reliability  
psych::alpha(select(study1f, starts\_with("SDO7\_"))) #alpha is .95

##   
## Reliability analysis   
## Call: psych::alpha(x = select(study1f, starts\_with("SDO7\_")))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.94 0.94 0.95 0.49 16 0.0042 2.6 1.2  
##   
## lower alpha upper 95% confidence boundaries  
## 0.93 0.94 0.95   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## SDO7\_1 0.94 0.94 0.94 0.50 15 0.0044  
## SDO7\_2 0.93 0.93 0.94 0.49 14 0.0046  
## SDO7\_3 0.93 0.93 0.94 0.49 14 0.0045  
## SDO7\_4 0.93 0.94 0.94 0.49 15 0.0044  
## SDO7\_5 0.93 0.94 0.94 0.49 15 0.0044  
## SDO7\_6 0.94 0.94 0.94 0.50 15 0.0044  
## SDO7\_7 0.94 0.94 0.95 0.52 16 0.0040  
## SDO7\_8 0.94 0.94 0.95 0.51 15 0.0042  
## SDO7\_9 0.93 0.93 0.94 0.49 14 0.0046  
## SDO7\_10 0.93 0.94 0.94 0.49 15 0.0044  
## SDO7\_11 0.93 0.94 0.94 0.49 15 0.0045  
## SDO7\_12 0.94 0.94 0.94 0.50 15 0.0043  
## SDO7\_13 0.93 0.94 0.94 0.49 14 0.0045  
## SDO7\_14 0.93 0.93 0.94 0.48 14 0.0046  
## SDO7\_15 0.93 0.94 0.94 0.49 14 0.0045  
## SDO7\_16 0.93 0.93 0.94 0.48 14 0.0046  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## SDO7\_1 458 0.70 0.70 0.69 0.65 2.1 1.5  
## SDO7\_2 458 0.79 0.79 0.79 0.76 2.5 1.6  
## SDO7\_3 459 0.79 0.79 0.78 0.75 2.7 1.6  
## SDO7\_4 459 0.72 0.72 0.71 0.68 2.2 1.5  
## SDO7\_5 458 0.72 0.72 0.70 0.67 2.5 1.5  
## SDO7\_6 458 0.70 0.71 0.68 0.66 2.2 1.4  
## SDO7\_7 457 0.46 0.46 0.40 0.39 2.5 1.7  
## SDO7\_8 457 0.61 0.61 0.57 0.55 2.8 1.7  
## SDO7\_9 458 0.79 0.79 0.78 0.76 2.5 1.7  
## SDO7\_10 458 0.74 0.73 0.71 0.69 3.0 1.9  
## SDO7\_11 458 0.74 0.74 0.72 0.70 2.7 1.7  
## SDO7\_12 459 0.68 0.67 0.64 0.62 3.0 1.8  
## SDO7\_13 458 0.75 0.76 0.74 0.71 2.2 1.4  
## SDO7\_14 459 0.84 0.84 0.83 0.81 2.7 1.6  
## SDO7\_15 458 0.75 0.76 0.74 0.71 2.5 1.5  
## SDO7\_16 457 0.81 0.81 0.80 0.78 2.6 1.7  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## SDO7\_1 0.50 0.21 0.10 0.08 0.09 0.02 0.00 0  
## SDO7\_2 0.39 0.21 0.12 0.13 0.10 0.05 0.01 0  
## SDO7\_3 0.33 0.20 0.15 0.15 0.12 0.04 0.02 0  
## SDO7\_4 0.47 0.20 0.08 0.12 0.08 0.03 0.00 0  
## SDO7\_5 0.36 0.21 0.17 0.17 0.05 0.02 0.02 0  
## SDO7\_6 0.40 0.26 0.15 0.10 0.05 0.01 0.02 0  
## SDO7\_7 0.37 0.23 0.17 0.09 0.06 0.04 0.05 0  
## SDO7\_8 0.28 0.25 0.14 0.16 0.08 0.03 0.04 0  
## SDO7\_9 0.38 0.23 0.12 0.13 0.07 0.04 0.03 0  
## SDO7\_10 0.32 0.19 0.11 0.14 0.12 0.07 0.05 0  
## SDO7\_11 0.33 0.22 0.14 0.14 0.09 0.03 0.04 0  
## SDO7\_12 0.27 0.20 0.16 0.15 0.11 0.05 0.06 0  
## SDO7\_13 0.40 0.26 0.16 0.11 0.04 0.02 0.01 0  
## SDO7\_14 0.31 0.21 0.19 0.14 0.08 0.03 0.02 0  
## SDO7\_15 0.35 0.23 0.20 0.09 0.07 0.04 0.02 0  
## SDO7\_16 0.35 0.21 0.15 0.14 0.07 0.05 0.03 0

#formativeness correlation  
cor.test(study1f$hypo\_4, study1f$hypo\_5) #r = .66 p < .001

##   
## Pearson's product-moment correlation  
##   
## data: study1f$hypo\_4 and study1f$hypo\_5  
## t = 18.675, df = 457, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.6026624 0.7068625  
## sample estimates:  
## cor   
## 0.6578997

## Analysis Chunk 3: Study 1 Manipulation Check

The next step is to run a manipulation check—Did participants in the reputable condition actually perceive the target as being more reputable? We could run a t-test but an ANOVA produces equivalent results and allows us to look at the partial eta squared.

s1manip <- aov(with(study1f, status ~ cond)) #create the model  
summary(s1manip) #see results from the model

## Df Sum Sq Mean Sq F value Pr(>F)   
## cond 1 592.5 592.5 300.9 <2e-16 \*\*\*  
## Residuals 457 900.0 2.0   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

etaSquared(s1manip) #calculates the effect size; this is where the lsr package comes into play!

## eta.sq eta.sq.part  
## cond 0.397009 0.397009

#getting the mean/sd for each condition  
study1f %>%  
 group\_by(cond) %>%  
 summarise(mean = mean(status), sd = sd(status))

## # A tibble: 2 x 3  
## cond mean sd  
## <fct> <dbl> <dbl>  
## 1 -1 3.50 1.80   
## 2 1 5.77 0.865

## Analysis Chunk 4: Study 1 Three-Way Interaction

That concludes all of our preliminary data wrangling/analyses. To test our hypotheses, we ran a 3-way interaction between SDO, ethnic identification, and condition on our formativeness variable. Before running this model, we mean-centered our continuous variables and ensured our dummy-coded condition variable was a factor:

study1f$sdocent <- scale(study1f$sdo, center = TRUE, scale = FALSE)  
study1f$ethcent <- scale(study1f$ethnic, center = TRUE, scale = FALSE)  
study1f$sdocent <- study1f$sdocent[,1]  
study1f$ethcent <- study1f$ethcent[,1]  
study1f$cond2 <- as.factor(study1f$cond)  
  
library(dplyr)  
  
study1f %>%  
 group\_by(cond) %>%  
 summarise(status = mean(status, na.rm = T), form = mean(form, na.rm = T), cat = mean(cat, na.rm = T))

## # A tibble: 2 x 4  
## cond status form cat  
## <fct> <dbl> <dbl> <dbl>  
## 1 -1 3.50 3.72 3.50  
## 2 1 5.77 3.75 3.63

Now we are ready to build the model. In R, after you build a model, you need to run the summary() function on it in order to view the results. You also need to ask for 95% confidence intervals with a separate function.

s1model <- lm(form~sdocent\*ethcent\*cond2, data=study1f)  
  
summary(s1model)

##   
## Call:  
## lm(formula = form ~ sdocent \* ethcent \* cond2, data = study1f)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.9044 -0.3478 0.1996 0.4569 2.2339   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.76489 0.06355 59.241 < 2e-16 \*\*\*  
## sdocent -0.02115 0.05674 -0.373 0.70954   
## ethcent -0.06348 0.03632 -1.748 0.08119 .   
## cond21 -0.05366 0.08811 -0.609 0.54282   
## sdocent:ethcent -0.07986 0.02994 -2.667 0.00793 \*\*   
## sdocent:cond21 0.23785 0.07685 3.095 0.00209 \*\*   
## ethcent:cond21 0.10275 0.05224 1.967 0.04981 \*   
## sdocent:ethcent:cond21 0.10651 0.04200 2.536 0.01156 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9033 on 451 degrees of freedom  
## Multiple R-squared: 0.07447, Adjusted R-squared: 0.0601   
## F-statistic: 5.184 on 7 and 451 DF, p-value: 1.074e-05

confint(s1model) #this gives us the 95% confidence intervals

## 2.5 % 97.5 %  
## (Intercept) 3.639998e+00 3.889789660  
## sdocent -1.326521e-01 0.090358062  
## ethcent -1.348514e-01 0.007898726  
## cond21 -2.268245e-01 0.119499526  
## sdocent:ethcent -1.387086e-01 -0.021012145  
## sdocent:cond21 8.681012e-02 0.388880048  
## ethcent:cond21 8.610435e-05 0.205410417  
## sdocent:ethcent:cond21 2.396489e-02 0.189050083

This gives us the information that we need, but it is still a little difficult to wade through. To make it easier, we can make it all into a little dataframe that has only the information we need for the manuscript (with all the numbers appropriately rounded).

To do this, I’m using the tidy() function from the broom package to pull regression results into a dataframe. We can do the same thing with confidence intervals.

s1modeldf <- tidy(s1model)  
s1cidf <- tidy(confint(s1model))

Then we can bind these two dataframes and change the variable names so they are all one word (and thus easier to work with).

names(s1cidf) <- c("parameter", "lowerci", "upperci")  
s1modeldf <- bind\_cols(s1modeldf, s1cidf) #now I have a dataframe with parameters, estimates, and confidence intercepts  
names(s1modeldf) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci") #and renaming

Finally, we can reduce this to the columns we need for the paper while rounding to the appropriate number of decimals for the paper.

s1modeldf %>%  
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.76 0.000 3.64 3.89  
## 2 sdocent -0.02 0.710 -0.13 0.09  
## 3 ethcent -0.06 0.081 -0.13 0.01  
## 4 cond21 -0.05 0.543 -0.23 0.12  
## 5 sdocent:ethcent -0.08 0.008 -0.14 -0.02  
## 6 sdocent:cond21 0.24 0.002 0.09 0.39  
## 7 ethcent:cond21 0.10 0.050 0.00 0.21  
## 8 sdocent:ethcent:cond21 0.11 0.012 0.02 0.19

Great! Now we need to get the simple slopes. We’ll use the simple\_slopes() function from the reghelper package.

s1simpslope <- simple\_slopes(s1model)  
print(s1simpslope)

## sdocent ethcent cond2 Test Estimate Std. Error t value df  
## 1 -1.165439 -1.702157 sstest -0.2945 0.1539 -1.9127 451  
## 2 0 -1.702157 sstest -0.2286 0.1257 -1.8190 451  
## 3 1.165439 -1.702157 sstest -0.1626 0.2047 -0.7944 451  
## 4 -1.165439 0 sstest -0.3309 0.1231 -2.6882 451  
## 5 0 0 sstest -0.0537 0.0881 -0.6090 451  
## 6 1.165439 0 sstest 0.2235 0.1282 1.7442 451  
## 7 -1.165439 1.702157 sstest -0.3672 0.1908 -1.9251 451  
## 8 0 1.702157 sstest 0.1212 0.1247 0.9721 451  
## 9 1.165439 1.702157 sstest 0.6097 0.1433 4.2552 451  
## 10 -1.165439 sstest -1 0.0296 0.0485 0.6107 451  
## 11 0 sstest -1 -0.0635 0.0363 -1.7478 451  
## 12 1.165439 sstest -1 -0.1565 0.0522 -2.9986 451  
## 13 sstest -1.702157 -1 0.1148 0.0814 1.4100 451  
## 14 sstest 0 -1 -0.0211 0.0567 -0.3727 451  
## 15 sstest 1.702157 -1 -0.1571 0.0708 -2.2199 451  
## 16 -1.165439 sstest 1 0.0082 0.0529 0.1555 451  
## 17 0 sstest 1 0.0393 0.0375 1.0459 451  
## 18 1.165439 sstest 1 0.0703 0.0488 1.4408 451  
## 19 sstest -1.702157 1 0.1713 0.0768 2.2300 451  
## 20 sstest 0 1 0.2167 0.0518 4.1803 451  
## 21 sstest 1.702157 1 0.2621 0.0671 3.9077 451  
## Pr(>|t|) Sig.  
## 1 0.0564141 .  
## 2 0.0695801 .  
## 3 0.4273656   
## 4 0.0074504 \*\*  
## 5 0.5428162   
## 6 0.0818038 .  
## 7 0.0548447 .  
## 8 0.3315169   
## 9 2.542e-05 \*\*\*  
## 10 0.5416871   
## 11 0.0811870 .  
## 12 0.0028619 \*\*  
## 13 0.1592276   
## 14 0.7095404   
## 15 0.0269207 \*  
## 16 0.8765340   
## 17 0.2961637   
## 18 0.1503349   
## 19 0.0262381 \*  
## 20 3.498e-05 \*\*\*  
## 21 0.0001075 \*\*\*

This is great, but this function doesn’t produce the 95% confidence intervals for the coefficients. In order to manually compute the confidence interval, we will use the following formula:  
95% confidence interval = coefficient ± 2(std error) While we’re getting the CIs, we’ll also restructure the results so they are easy to put it into the paper.

First, we will create a reduced dataframe with only the rows (parameters) that we are interested in for the paper. We’ll also change the names of the columns while we are at it so they are all one word (easier to work with)

s1simpsloper <- s1simpslope[c(8, 2, 17, 11, 6, 4, 20, 14), ]   
names(s1simpsloper)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p")

Good, now we create two columns for lower and upper confidence intervals. We’ll also do a few other things to make it easier to copy these values into the paper:  
1. We’ll round the columns in accordance with APA  
2. We’ll make a new parameter column that will make it easy to see which parameter we are looking at  
3. We’ll make select only the columns that we need for the the paper

s1simpsloper %>%   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.12 0.332 -0.13 0.37  
## 2 cond\_at\_low\_eth -0.23 0.070 -0.48 0.02  
## 3 eth\_in\_high\_cond 0.04 0.296 -0.04 0.11  
## 4 eth\_in\_low\_cond -0.06 0.081 -0.14 0.01  
## 5 cond\_at\_high\_sdo 0.22 0.082 -0.03 0.48  
## 6 cond\_at\_low\_sdo -0.33 0.007 -0.58 -0.08  
## 7 sdo\_in\_high\_cond 0.22 0.000 0.11 0.32  
## 8 sdo\_in\_low\_cond -0.02 0.710 -0.13 0.09

And those are our results!

# Study 2 Analyses

Just like Study 1, we start Study 2 by importing the data from a .csv file. This .csv is raw from Qualtrics (with the exception of the suspicion questions and variable labels being removed).

study2 <- read.csv("study 2 school shooting no labels.csv")

Then we filter off the incomplete cases and any case where the participant did not consent to letting us use their data:

#start by filtering out the cases where people didn't consent to letting us use their data  
nrow(study2) # we start with 573 lines of data

## [1] 573

study2 <- study2 %>%  
 filter(data\_yn == 1) #this filters out people who didn't complete the study or didn't let us use data  
nrow(study2) #there are now 501 people in this sample

## [1] 501

## Cleaning Study 2 Data

Now that we have our full dataset ready, we can clean the data and prep it for analyses. Let’s start by creating the condition variable, which again will be contrast coded:

#creating the condition variable; info is stored in both DO-BR-FL\_127 and DO-BR-FL\_12  
study2$cond[study2$DO.BR.FL\_127 == "high Status"] <- 1  
study2$cond[study2$DO.BR.FL\_129 == "high Status"] <- 1  
study2$cond[study2$DO.BR.FL\_127 == "low status"] <- -1  
study2$cond[study2$DO.BR.FL\_129 == "low status"] <- -1  
class(study2$cond)

## [1] "numeric"

#checking the Ns and mean status in each condition  
study2 %>%  
 group\_by(cond) %>%  
 summarise(N = n(), "Mean Status" = mean(status, na.rm = TRUE)) #239 bad cond; 262 good cond; perceptions of status are higher in good condition; (i.e., it's correctly coded)

## # A tibble: 2 x 3  
## cond N `Mean Status`  
## <dbl> <int> <dbl>  
## 1 -1.00 239 2.92  
## 2 1.00 262 4.68

Next we can create the scales and make ethnicity and gender into factors. This is all identical to Study 1. (The one exception is that there were no Native American participants in this sample so that is not a level in the ethnicity factor)

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#SDO  
#reverse code items:  
study2$SDO7\_5 <- 8-study2$SDO7\_5  
study2$SDO7\_6 <- 8-study2$SDO7\_6  
study2$SDO7\_7 <- 8-study2$SDO7\_7  
study2$SDO7\_8 <- 8-study2$SDO7\_8  
study2$SDO7\_13 <- 8-study2$SDO7\_13  
study2$SDO7\_14 <- 8-study2$SDO7\_14  
study2$SDO7\_15 <- 8-study2$SDO7\_15  
study2$SDO7\_16 <- 8-study2$SDO7\_16  
  
#creating the scale  
study2$sdo <- study2 %>%  
 select(starts\_with("SDO7\_")) %>%  
 rowMeans(na.rm = TRUE)   
  
#checking for NAs--there shouldn't really be any bc everyone should have at least some SDO values in this   
anyNA(study2$sdo) #no NAs

## [1] FALSE

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#ETHNIC ID   
#we want to use ethnic\_id2 & ethnic\_id3; those are the only ones in all studies.  
study2$ethnic <- rowMeans(subset(study2, select = c(ethnic\_id2, ethnic\_id3)), na.rm = TRUE)  
summary(study2$ethnic) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 2.50 4.00 3.91 5.50 7.00

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#FORMATIVENESS & CATEGORIZATION  
#hypo 1-4 are categorization; hypo 4-5, 11-12 are formativeness (but not included)  
#compute the scale scale  
study2$form <- rowMeans(subset(study2, select = c(hypo\_4, hypo\_5, hypo\_11, hypo\_12), na.rm = TRUE))  
study2$cat <- rowMeans(subset(study2, select = c(hypo\_1, hypo\_2), na.rm = TRUE))  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#MAKING ETHNIC ID & GENDER FACTORS  
study2$ethnic\_id <- factor(study2$ethnic\_id, labels = c("Black/African American", "Asian/Asian American/Pacific Islander",   
 "White", "Latino/Hispanic American", "Middle Eastern/Arab American",  
 "Biracial/Mixed race (please specify)", "Other (please specify)"))  
  
study2$demo\_gend2 <- factor(study2$demo\_gend, labels = c("male", "female"))

Excellent! The data is ready for analysis.

## Analysis Chunk 5: Study 1 Demographics

nrow(study2) #501 people total in study 2

## [1] 501

#how many white participants?  
study2 %>%   
 filter(ethnic\_id == "White") %>%  
 nrow() #411 White participants

## [1] 411

#selecting white participants who completed teh attention checks; final sample  
study2f <- study2 %>%  
 filter(ethnic\_id == "White" & attn == 5 & attn2 == 3) #making final study 1 dataset  
nrow(study2f) #final sample is 404

## [1] 404

411 - nrow(study2f) #7 people didn't pass attention checks

## [1] 7

#mean age  
  
#checking for outliers  
study2f$demo\_age[study2f$demo\_age > 100 | study2f$demo\_age < 18] #no outliers

## integer(0)

describe(study2f$demo\_age, na.rm = TRUE) #mean age is 35.90; sd = 12.08

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 404 39.56 13.12 37 38.61 13.34 18 79 61 0.6 -0.43  
## se  
## X1 0.65

#gender  
summary(study2f$demo\_gend2) #169 males, 235 females;

## male female   
## 169 235

169/(169+235) #41.8% females

## [1] 0.4183168

## Analysis Chunk 6: Study 2 Scales

#ethnic id correlation  
cor.test(study2f$ethnic\_id2, study2f$ethnic\_id3) #.71 p < .001

##   
## Pearson's product-moment correlation  
##   
## data: study2f$ethnic\_id2 and study2f$ethnic\_id3  
## t = 20.371, df = 402, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.6611105 0.7575906  
## sample estimates:  
## cor   
## 0.7127049

#sdo reliability  
alpha(select(study2f, starts\_with("SDO7\_"))) #sdo reliability is .94

##   
## Reliability analysis   
## Call: alpha(x = select(study2f, starts\_with("SDO7\_")))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.94 0.94 0.95 0.5 16 0.0043 2.7 1.3  
##   
## lower alpha upper 95% confidence boundaries  
## 0.93 0.94 0.95   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## SDO7\_1 0.94 0.94 0.95 0.51 16 0.0045  
## SDO7\_2 0.93 0.94 0.95 0.49 15 0.0048  
## SDO7\_3 0.94 0.94 0.95 0.50 15 0.0047  
## SDO7\_4 0.94 0.94 0.95 0.51 15 0.0045  
## SDO7\_5 0.94 0.94 0.95 0.50 15 0.0046  
## SDO7\_6 0.94 0.94 0.95 0.51 15 0.0045  
## SDO7\_7 0.94 0.94 0.95 0.53 17 0.0041  
## SDO7\_8 0.94 0.94 0.95 0.52 16 0.0043  
## SDO7\_9 0.94 0.94 0.95 0.50 15 0.0046  
## SDO7\_10 0.94 0.94 0.95 0.51 16 0.0044  
## SDO7\_11 0.94 0.94 0.95 0.50 15 0.0047  
## SDO7\_12 0.94 0.94 0.95 0.50 15 0.0046  
## SDO7\_13 0.94 0.94 0.95 0.50 15 0.0046  
## SDO7\_14 0.93 0.94 0.95 0.50 15 0.0047  
## SDO7\_15 0.94 0.94 0.95 0.50 15 0.0047  
## SDO7\_16 0.93 0.94 0.95 0.49 15 0.0048  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## SDO7\_1 404 0.67 0.68 0.65 0.62 2.2 1.6  
## SDO7\_2 404 0.83 0.83 0.83 0.80 2.6 1.7  
## SDO7\_3 404 0.77 0.77 0.76 0.74 2.8 1.8  
## SDO7\_4 404 0.71 0.72 0.70 0.67 2.3 1.6  
## SDO7\_5 404 0.74 0.75 0.73 0.70 2.4 1.5  
## SDO7\_6 404 0.71 0.71 0.69 0.67 2.5 1.7  
## SDO7\_7 404 0.48 0.48 0.43 0.41 2.5 1.8  
## SDO7\_8 404 0.61 0.61 0.57 0.54 3.1 1.9  
## SDO7\_9 404 0.77 0.76 0.75 0.73 2.8 1.8  
## SDO7\_10 404 0.66 0.65 0.62 0.60 3.1 2.0  
## SDO7\_11 404 0.79 0.78 0.77 0.75 2.7 1.8  
## SDO7\_12 404 0.76 0.75 0.73 0.71 3.3 2.0  
## SDO7\_13 404 0.77 0.78 0.77 0.74 2.4 1.5  
## SDO7\_14 404 0.81 0.81 0.81 0.78 2.9 1.7  
## SDO7\_15 404 0.78 0.79 0.78 0.75 2.5 1.6  
## SDO7\_16 404 0.83 0.83 0.83 0.80 2.9 1.8  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## SDO7\_1 0.49 0.21 0.09 0.11 0.06 0.02 0.02 0  
## SDO7\_2 0.40 0.21 0.08 0.15 0.09 0.04 0.03 0  
## SDO7\_3 0.35 0.18 0.12 0.16 0.10 0.06 0.03 0  
## SDO7\_4 0.49 0.19 0.06 0.14 0.07 0.03 0.02 0  
## SDO7\_5 0.40 0.18 0.19 0.14 0.05 0.03 0.01 0  
## SDO7\_6 0.39 0.20 0.15 0.11 0.05 0.06 0.03 0  
## SDO7\_7 0.38 0.22 0.16 0.09 0.05 0.03 0.06 0  
## SDO7\_8 0.29 0.19 0.14 0.16 0.08 0.06 0.08 0  
## SDO7\_9 0.35 0.20 0.15 0.11 0.09 0.06 0.05 0  
## SDO7\_10 0.31 0.20 0.10 0.13 0.11 0.08 0.07 0  
## SDO7\_11 0.34 0.24 0.13 0.10 0.08 0.06 0.05 0  
## SDO7\_12 0.26 0.18 0.10 0.16 0.13 0.09 0.07 0  
## SDO7\_13 0.40 0.21 0.17 0.12 0.04 0.03 0.02 0  
## SDO7\_14 0.29 0.19 0.20 0.15 0.06 0.07 0.04 0  
## SDO7\_15 0.37 0.19 0.18 0.13 0.06 0.04 0.02 0  
## SDO7\_16 0.30 0.19 0.18 0.13 0.06 0.08 0.05 0

#formativeness alpha  
alpha(subset(study2f, select = c(hypo\_4, hypo\_5, hypo\_11,hypo\_12))) #.77 p < .001

##   
## Reliability analysis   
## Call: alpha(x = subset(study2f, select = c(hypo\_4, hypo\_5, hypo\_11,   
## hypo\_12)))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.93 0.93 0.92 0.78 14 0.0053 3.7 1  
##   
## lower alpha upper 95% confidence boundaries  
## 0.92 0.93 0.95   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## hypo\_4 0.93 0.93 0.89 0.81 12.4 0.0065  
## hypo\_5 0.92 0.92 0.89 0.79 11.4 0.0070  
## hypo\_11 0.92 0.92 0.88 0.78 10.8 0.0073  
## hypo\_12 0.90 0.90 0.86 0.75 8.9 0.0087  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## hypo\_4 404 0.89 0.90 0.84 0.81 3.8 1.1  
## hypo\_5 404 0.91 0.91 0.86 0.83 3.7 1.1  
## hypo\_11 404 0.92 0.91 0.87 0.85 3.6 1.2  
## hypo\_12 404 0.94 0.94 0.93 0.89 3.7 1.2  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## hypo\_4 0.06 0.05 0.12 0.67 0.05 0.04 0.01 0  
## hypo\_5 0.06 0.07 0.14 0.59 0.07 0.04 0.01 0  
## hypo\_11 0.08 0.08 0.19 0.53 0.08 0.03 0.01 0  
## hypo\_12 0.08 0.07 0.14 0.61 0.06 0.03 0.01 0

## Analysis Chunk 7: Study 2 Manipulation Check

s2manip <- aov(with(study2f, status ~ cond)) #this seems roughly equivalant to the t-test  
summary(s2manip)

## Df Sum Sq Mean Sq F value Pr(>F)   
## cond 1 305.8 305.8 217.7 <2e-16 \*\*\*  
## Residuals 401 563.1 1.4   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## 1 observation deleted due to missingness

etaSquared(s2manip) #calculating the effect size

## eta.sq eta.sq.part  
## cond 0.3519144 0.3519144

#getting the mean/sd for each condition  
study2f %>%  
 group\_by(cond) %>%  
 summarise(mean = mean(status, na.rm = T), sd = sd(status, na.rm = T))

## # A tibble: 2 x 3  
## cond mean sd  
## <dbl> <dbl> <dbl>  
## 1 -1.00 2.87 1.29  
## 2 1.00 4.61 1.08

## Analysis Chunk 8: Study 2 Three-Way Interaction

First, we mean-center our continuous variables and make sure our dichotomous condition variable is a factor:

study2f$sdocent <- scale(study2f$sdo, center = TRUE, scale = FALSE)  
study2f$ethcent <- scale(study2f$ethnic, center = TRUE, scale = FALSE)  
study2f$sdocent <- study2f$sdocent[,1]  
study2f$ethcent <- study2f$ethcent[,1]  
study2f$cond2 <- as.factor(study2f$cond)  
study2f$condcent <- scale(as.numeric(as.character(study2f$cond)), center= TRUE, scale = FALSE)

Now we can run our 3-way interaction between SDO, ethnic identification, and condition (including the simple slopes).

s2model <- lm(form~sdocent\*ethcent\*cond2, data=study2f)

As before, we’ll reshape these results so can be used more easily (in the context of the paper).

s2modeldf <- tidy(s2model) #pulling out regression results  
s2cidf <- tidy(confint(s2model)) #pulling out CIs  
names(s2cidf) <- c("parameter", "lowerci", "upperci") #renamingh  
  
s2modeldf <- bind\_cols(s2modeldf, s2cidf) #binding regression results and CIs  
names(s2modeldf) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci") #and renaming  
  
s2modeldf %>%  
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>% #rounding  
 select(parameter, B, p, lowerci, upperci) #selecting variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.52 0.000 3.38 3.67  
## 2 sdocent -0.12 0.052 -0.23 0.00  
## 3 ethcent -0.20 0.000 -0.28 -0.11  
## 4 cond21 0.34 0.001 0.14 0.53  
## 5 sdocent:ethcent -0.03 0.319 -0.09 0.03  
## 6 sdocent:cond21 0.18 0.027 0.02 0.34  
## 7 ethcent:cond21 0.25 0.000 0.14 0.37  
## 8 sdocent:ethcent:cond21 0.02 0.564 -0.06 0.10

Now we can do a similar thing with the simple slopes (but we’ll have to manually compute the CIs again).

s2simpslope <- simple\_slopes(s2model) #pulling out the simple slopes to a df  
  
s2simpsloper <- s2simpslope[c(8, 2, 17, 11, 6, 4, 20, 14), ] #pulling out relevant rows  
names(s2simpsloper)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p") #renaming  
  
s2simpsloper %>%   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding according to APA  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",   
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter col  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.78 0.000 0.49 1.06  
## 2 cond\_at\_low\_eth -0.10 0.469 -0.39 0.18  
## 3 eth\_in\_high\_cond 0.06 0.157 -0.02 0.14  
## 4 eth\_in\_low\_cond -0.20 0.000 -0.29 -0.11  
## 5 cond\_at\_high\_sdo 0.57 0.000 0.27 0.86  
## 6 cond\_at\_low\_sdo 0.11 0.454 -0.18 0.39  
## 7 sdo\_in\_high\_cond 0.07 0.242 -0.05 0.18  
## 8 sdo\_in\_low\_cond -0.12 0.052 -0.23 0.00

# Study 3 Analyses

The set-up and analyses of Study 3 was virtually identical to Study 2. We begin by importing the data (again, labels and suspicion question were removed from the csv prior to this import). Then we filter out incomplete cases and people who did not consent to letting us use their data.

study3 <- read.csv(file="study 3 project artist no labels.csv")  
#start by filtering out the cases where people didn't consent to letting us use their data  
nrow(study3) # we start with 591 lines of data

## [1] 591

study3 <- study3 %>%  
 filter(data\_yn == 1) #this filters out people who didn't complete the study or didn't let us use data  
nrow(study3) #there are now 506 people in this sample

## [1] 506

## Cleaning the Data

Similar to prior studies, we next created a contrast-coded condition variable.

study3$cond[study3$DO.BR.FL\_53 == "high status"] <- 1  
study3$cond[study3$DO.BR.FL\_115 == "high status"] <- 1  
study3$cond[study3$DO.BR.FL\_53 == "low Status"] <- -1  
study3$cond[study3$DO.BR.FL\_115 == "low Status"] <- -1  
  
 #changing this to a factor  
summary(study3$cond) #259 bad cond; 247 good cond

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.00000 -1.00000 -1.00000 -0.02372 1.00000 1.00000

study3 %>%  
 group\_by(cond) %>%  
 summarise('n' = n(), 'status' = mean(status, na.rm = TRUE)) #259 bad cond; 247 good cond#perceptions of status are higher in good condition; i.e., it's correctly coded

## # A tibble: 2 x 3  
## cond n status  
## <dbl> <int> <dbl>  
## 1 -1.00 259 2.97  
## 2 1.00 247 4.37

Next, we will create the SDO, ethnic identification, and formativeness and categorization DVs. We’ll also make gender and ethnic identification into factors.

#SDO  
#reverse code items:  
study3$SDO7\_5 <- 8-study3$SDO7\_5  
study3$SDO7\_6 <- 8-study3$SDO7\_6  
study3$SDO7\_7 <- 8-study3$SDO7\_7  
study3$SDO7\_8 <- 8-study3$SDO7\_8  
study3$SDO7\_13 <- 8-study3$SDO7\_13  
study3$SDO7\_14 <- 8-study3$SDO7\_14  
study3$SDO7\_15 <- 8-study3$SDO7\_15  
study3$SDO7\_16 <- 8-study3$SDO7\_16  
  
#creating the scale  
study3$sdo <- study3 %>%  
 select(starts\_with("SDO7\_")) %>%  
 rowMeans(na.rm = TRUE)   
  
#checking for NAs   
anyNA(study3$sdo) #no NAs

## [1] FALSE

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#ETHNIC ID   
#we want to use ethnic\_id2 & ethnic\_id3; those are the only ones in all studies.  
study3$ethnic <- rowMeans(subset(study3, select = c(ethnic\_id2, ethnic\_id3)), na.rm = TRUE)  
summary(study3$ethnic) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.500 4.000 3.954 5.500 7.000

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#FORMATIVENESS & CATEGORIZATION  
#hypo 1-4 are categorization; hypo 4-5, 11-12 are formativeness (but not included)  
#compute the scale scale  
study3$form <- rowMeans(subset(study3, select = c(hypo\_4, hypo\_5, hypo\_11, hypo\_12), na.rm = TRUE))  
study3$cat <- rowMeans(subset(study3, select = c(hypo\_1, hypo\_2), na.rm = TRUE))  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#MAKING ETHNIC ID & GENDER FACTORS  
study3$ethnic\_id <- factor(study3$ethnic\_id, labels = c("Black/African American", "Asian/Asian American/Pacific Islander", "White", "Latino/Hispanic American", "Middle Eastern/Arab American","Native American", "Biracial/Mixed race (please specify)", "Other (please specify)"))  
  
study3$demo\_gend2 <- factor(study3$demo\_gend, labels = c("male", "female"))

Great! With that all done, we are ready to proceed to the Study 3 analyses.

## Analysis Chunk 9: Study 3 Demographics

#total number of people in Study 3   
nrow(study3) #506 people total in study 1

## [1] 506

#how many white participants?  
study3 %>%   
 filter(ethnic\_id == "White") %>%  
 nrow() #379 White participants

## [1] 379

#selecting white participants who completed the attention checks; final sample  
study3f <- study3 %>%  
 filter(ethnic\_id == "White" & attn == 5 & attn2 == 3) #making final study 1 dataset  
nrow(study3f) #final sample is 368

## [1] 368

379 - nrow(study3f) #11 people didn't pass attention checks

## [1] 11

#ages  
#checking for outliers  
study3f$demo\_age[study3f$demo\_age > 100 | study3f$demo\_age < 18] #no outliers

## integer(0)

#looking for mean/standard deviation  
describe(study3f$demo\_age, na.rm = TRUE) #mean age is 38.80; sd = 13.29

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 368 38.8 13.29 35 37.79 13.34 19 73 54 0.61 -0.73  
## se  
## X1 0.69

#gender  
summary(study3f$demo\_gend2) #163 males, 205 females;

## male female   
## 163 205

205/(163+205) #55.7% females

## [1] 0.5570652

## Analysis Chunk 10: Study 3 Scales

#ethnic id correlation  
cor.test(study3f$ethnic\_id2, study3f$ethnic\_id3) #.77 p < .001

##   
## Pearson's product-moment correlation  
##   
## data: study3f$ethnic\_id2 and study3f$ethnic\_id3  
## t = 23.293, df = 366, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.7280506 0.8109320  
## sample estimates:  
## cor   
## 0.7727651

#sdo reliability  
alpha(select(study3f, starts\_with("SDO7\_"))) #sdo reliability is .95

##   
## Reliability analysis   
## Call: alpha(x = select(study3f, starts\_with("SDO7\_")))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.95 0.95 0.96 0.57 21 0.0035 2.6 1.3  
##   
## lower alpha upper 95% confidence boundaries  
## 0.95 0.95 0.96   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## SDO7\_1 0.95 0.95 0.96 0.57 20 0.0036  
## SDO7\_2 0.95 0.95 0.96 0.56 19 0.0038  
## SDO7\_3 0.95 0.95 0.96 0.56 19 0.0038  
## SDO7\_4 0.95 0.95 0.96 0.57 20 0.0037  
## SDO7\_5 0.95 0.95 0.96 0.57 20 0.0036  
## SDO7\_6 0.95 0.95 0.96 0.57 20 0.0037  
## SDO7\_7 0.96 0.96 0.96 0.60 23 0.0033  
## SDO7\_8 0.95 0.96 0.96 0.59 22 0.0034  
## SDO7\_9 0.95 0.95 0.96 0.56 19 0.0039  
## SDO7\_10 0.95 0.95 0.96 0.58 20 0.0036  
## SDO7\_11 0.95 0.95 0.96 0.56 19 0.0038  
## SDO7\_12 0.95 0.95 0.96 0.57 20 0.0037  
## SDO7\_13 0.95 0.95 0.96 0.56 19 0.0038  
## SDO7\_14 0.95 0.95 0.96 0.56 19 0.0038  
## SDO7\_15 0.95 0.95 0.96 0.56 19 0.0038  
## SDO7\_16 0.95 0.95 0.96 0.56 19 0.0039  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## SDO7\_1 368 0.73 0.74 0.72 0.70 2.1 1.5  
## SDO7\_2 368 0.82 0.82 0.81 0.79 2.5 1.7  
## SDO7\_3 368 0.82 0.82 0.81 0.79 2.7 1.8  
## SDO7\_4 368 0.78 0.78 0.78 0.75 2.3 1.7  
## SDO7\_5 368 0.73 0.73 0.71 0.69 2.5 1.7  
## SDO7\_6 368 0.80 0.80 0.79 0.77 2.3 1.5  
## SDO7\_7 368 0.50 0.50 0.44 0.43 2.5 1.8  
## SDO7\_8 368 0.59 0.59 0.55 0.53 2.7 1.8  
## SDO7\_9 368 0.86 0.86 0.86 0.84 2.6 1.8  
## SDO7\_10 368 0.72 0.72 0.69 0.68 2.8 1.8  
## SDO7\_11 368 0.82 0.82 0.81 0.79 2.7 1.8  
## SDO7\_12 368 0.79 0.79 0.78 0.76 3.1 1.9  
## SDO7\_13 368 0.83 0.84 0.83 0.81 2.3 1.5  
## SDO7\_14 368 0.86 0.86 0.85 0.83 2.6 1.7  
## SDO7\_15 368 0.82 0.82 0.81 0.79 2.5 1.6  
## SDO7\_16 368 0.87 0.87 0.87 0.85 2.8 1.8  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## SDO7\_1 0.52 0.19 0.08 0.10 0.07 0.03 0.01 0  
## SDO7\_2 0.43 0.17 0.11 0.14 0.09 0.04 0.03 0  
## SDO7\_3 0.38 0.18 0.07 0.17 0.10 0.07 0.03 0  
## SDO7\_4 0.51 0.15 0.08 0.10 0.09 0.04 0.02 0  
## SDO7\_5 0.41 0.20 0.14 0.13 0.05 0.03 0.04 0  
## SDO7\_6 0.46 0.18 0.14 0.12 0.06 0.02 0.01 0  
## SDO7\_7 0.40 0.20 0.16 0.11 0.04 0.03 0.07 0  
## SDO7\_8 0.36 0.19 0.14 0.16 0.05 0.04 0.05 0  
## SDO7\_9 0.39 0.18 0.13 0.12 0.07 0.05 0.04 0  
## SDO7\_10 0.35 0.20 0.11 0.14 0.09 0.07 0.04 0  
## SDO7\_11 0.38 0.19 0.12 0.12 0.10 0.04 0.05 0  
## SDO7\_12 0.31 0.15 0.12 0.17 0.12 0.07 0.06 0  
## SDO7\_13 0.43 0.23 0.13 0.12 0.05 0.02 0.02 0  
## SDO7\_14 0.36 0.21 0.17 0.11 0.07 0.05 0.03 0  
## SDO7\_15 0.40 0.17 0.17 0.14 0.07 0.04 0.03 0  
## SDO7\_16 0.35 0.18 0.14 0.15 0.07 0.06 0.04 0

#formativeness alpha  
alpha(subset(study3f, select = c(hypo\_4, hypo\_5, hypo\_11,hypo\_12))) #.77 p < .001

##   
## Reliability analysis   
## Call: alpha(x = subset(study3f, select = c(hypo\_4, hypo\_5, hypo\_11,   
## hypo\_12)))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.93 0.93 0.92 0.77 14 0.0059 3.6 1  
##   
## lower alpha upper 95% confidence boundaries  
## 0.92 0.93 0.94   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## hypo\_4 0.92 0.92 0.89 0.80 12.2 0.0069  
## hypo\_5 0.92 0.92 0.89 0.80 11.9 0.0070  
## hypo\_11 0.90 0.90 0.86 0.75 8.9 0.0093  
## hypo\_12 0.90 0.90 0.86 0.74 8.6 0.0094  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## hypo\_4 368 0.88 0.89 0.83 0.80 3.7 1.1  
## hypo\_5 368 0.89 0.89 0.83 0.80 3.6 1.2  
## hypo\_11 368 0.93 0.93 0.91 0.87 3.5 1.2  
## hypo\_12 368 0.94 0.94 0.92 0.88 3.6 1.1  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## hypo\_4 0.05 0.08 0.17 0.60 0.07 0.03 0.01 0  
## hypo\_5 0.06 0.13 0.15 0.52 0.10 0.04 0.01 0  
## hypo\_11 0.08 0.12 0.19 0.51 0.07 0.02 0.01 0  
## hypo\_12 0.06 0.10 0.19 0.53 0.08 0.02 0.01 0

## Analysis Chunk 11: Study 3 Manipulation Check

s3manip <- aov(with(study3f, status ~ cond)) #building the anova model   
summary(s3manip)#extracting results from the model

## Df Sum Sq Mean Sq F value Pr(>F)   
## cond 1 202.1 202.12 124.4 <2e-16 \*\*\*  
## Residuals 366 594.4 1.62   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

etaSquared(s3manip) #calculating the effect size

## eta.sq eta.sq.part  
## cond 0.2537374 0.2537374

#getting the mean/sd for each condition  
study3f %>%  
 group\_by(cond) %>%  
 summarise(mean = mean(status, na.rm = T), sd = sd(status, na.rm = T))

## # A tibble: 2 x 3  
## cond mean sd  
## <dbl> <dbl> <dbl>  
## 1 -1.00 2.84 1.32  
## 2 1.00 4.32 1.23

## Analysis Chunk 12: Study 3 Three-Way Interaction

First, we mean-centered our continuous variables and change our dichotomous condition variable as a factor:

study3f$sdocent <- scale(study3f$sdo, center = TRUE, scale = FALSE)  
study3f$ethcent <- scale(study3f$ethnic, center = TRUE, scale = FALSE)  
study3f$sdocent <- study3f$sdocent[,1]  
study3f$ethcent <- study3f$ethcent[,1]  
study3f$cond2 <- as.factor(study3f$cond)

Now we can run our 3-way interaction between SDO, ethnic identification, and condition (including the simple slopes).

s3model <- lm(form~sdocent\*ethcent\*cond2, data=study3f)

As before, we’ll reshape these results so can be used more easily.

s3modeldf <- tidy(s3model) #pulling out regression results  
s3cidf <- tidy(confint(s3model)) #pulling out CIs  
names(s3cidf) <- c("parameter", "lowerci", "upperci") #renaming  
  
s3modeldf <- bind\_cols(s3modeldf, s3cidf) #binding regression results and CIs  
names(s3modeldf) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci") #and renaming  
  
s3modeldf %>%  
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>% #rounding  
 select(parameter, B, p, lowerci, upperci) #selecting variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.76 0.000 3.61 3.91  
## 2 sdocent -0.03 0.597 -0.15 0.09  
## 3 ethcent 0.01 0.766 -0.07 0.10  
## 4 cond21 -0.37 0.001 -0.59 -0.16  
## 5 sdocent:ethcent 0.04 0.249 -0.02 0.09  
## 6 sdocent:cond21 0.17 0.042 0.01 0.33  
## 7 ethcent:cond21 -0.05 0.420 -0.17 0.07  
## 8 sdocent:ethcent:cond21 -0.04 0.380 -0.12 0.05

Now we’ll look at the simple slopes.

s3simpslope <- simple\_slopes(s3model) #pulling out the simple slopes to a df  
  
s3simpsloper <- s3simpslope[c(8, 2, 17, 11, 6, 4, 20, 14), ] #pulling out relevant rows  
names(s3simpsloper)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p") #renaming  
  
s3simpsloper %>%   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding according to APA  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",   
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter col  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth -0.46 0.003 -0.77 -0.16  
## 2 cond\_at\_low\_eth -0.29 0.065 -0.59 0.02  
## 3 eth\_in\_high\_cond -0.04 0.407 -0.13 0.05  
## 4 eth\_in\_low\_cond 0.01 0.766 -0.07 0.10  
## 5 cond\_at\_high\_sdo -0.15 0.345 -0.46 0.17  
## 6 cond\_at\_low\_sdo -0.60 0.000 -0.90 -0.30  
## 7 sdo\_in\_high\_cond 0.14 0.018 0.02 0.26  
## 8 sdo\_in\_low\_cond -0.03 0.597 -0.15 0.09

# Analyses with Combined Dataset

We were also interested in looking at the effects of all of our studies combined. Given that our research design was extremely similar from study to study and we used all the same measures across studies, we decided to simply merge studies.

## Merging Studies 1-3

Before we merge, we will create a new dataframe for each study that contains only the variables that we plan on using in the final analysis. Study 1 is already only a select subject of variables due to the Study 1a/Study 1b merge, so nothing needs to be selected out of it.

#study 1 variables are already selected; need to rename our Study 1A/1B variable so it  
#doesn't interfere with our new source variable  
study1merge <- study1  
  
#selecting variables for the merge from study 2  
study2merge <- study2 %>%  
 select(V1,demo\_age, demo\_gend, starts\_with("ethnic"), attn, starts\_with("SDO"), status,   
 attn2, starts\_with("hypo"), cond, ethnic, sdo, form, cat, demo\_gend2) #study2 is ready to go  
  
#selecting variables for the merge from study 3  
study3merge <- study3 %>%  
 select(V1,demo\_age, demo\_gend, starts\_with("ethnic"), attn, starts\_with("SDO"), status,   
 attn2, starts\_with("hypo"), cond, ethnic, sdo, form, cat)

We also need to change a couple of the factors to character variables. If you have two factors of different levels, R won’t let you merge them, so we avoid this by making the factors into character variables.

study1merge$cond<- as.character(study1merge$cond)  
study2merge$cond<- as.character(study2merge$cond)  
study3merge$cond<- as.character(study3merge$cond)  
  
study1merge$ethnic\_id <- as.character(study1merge$ethnic\_id)  
study2merge$ethnic\_id <- as.character(study2merge$ethnic\_id)  
study3merge$ethnic\_id <- as.character(study3merge$ethnic\_id)

We will now merge the datasets. The new variable “source” will tell us which study the data came from.

combined <- bind\_rows("study1" = study1merge, "study2" = study2merge, "study3" = study3merge, .id = "source")

## Warning in bind\_rows\_(x, .id): binding character and factor vector,  
## coercing into character vector  
  
## Warning in bind\_rows\_(x, .id): binding character and factor vector,  
## coercing into character vector  
  
## Warning in bind\_rows\_(x, .id): binding character and factor vector,  
## coercing into character vector  
  
## Warning in bind\_rows\_(x, .id): binding character and factor vector,  
## coercing into character vector

nrow(combined) #1628 rows, that's equal to 106 + 113 + 197 +205

## [1] 1628

summary(combined$cond) #checking the conditions --801 bad, 827 good

## Length Class Mode   
## 1628 character character

#study 1: 303 bad, 318 good  
#study 2: 239 bad cond; 262 good cond  
#study 3: 259 bad cond; 247 good cond  
#total should be 801 bad, 827 good; we're good  
  
combined %>%  
 group\_by(cond) %>%  
 summarise('n' = n(), status =mean(status, na.rm = TRUE)) #status is still higher in reputable condition, everything looks reputable

## # A tibble: 2 x 3  
## cond n status  
## <chr> <int> <dbl>  
## 1 -1 801 3.19  
## 2 1 827 5.01

Our merge is now complete! I am going to export this to excel for future reference and double checking.

#exporting this data to Excel.  
write.csv(combined, file = "3 studies combined.csv")

Just a few more things to do before the data is really for analyses. First, I’m going to make ethnic\_id back into a factor:

combined$ethnic\_id <- factor(combined$ethnic\_id, labels = c("Black/African American", "Asian/Asian American/Pacific Islander",   
 "Latino/Hispanic American", "Middle Eastern/Arab American","Native American",  
 "Biracial/Mixed race (please specify)", "Other", "White"))  
  
combined %>%  
 group\_by(ethnic\_id) %>%  
 summarise(n())

## # A tibble: 8 x 2  
## ethnic\_id `n()`  
## <fct> <int>  
## 1 Black/African American 93  
## 2 Asian/Asian American/Pacific Islander 56  
## 3 Latino/Hispanic American 105  
## 4 Middle Eastern/Arab American 86  
## 5 Native American 7  
## 6 Biracial/Mixed race (please specify) 8  
## 7 Other 13  
## 8 White 1260

Next, we’ll rename the combined dataframe as cdf so it’s slightly easier to run analyses on it

cdf <- combined

And finally, before we moving onto to cleaning the data/performing analyses, I like to remove some of the data frames we made in process, just to keep the working environment tidy.

rm(study1merge)  
rm(study2merge)  
rm(study3merge)

## Analysis Chunk 9: Combined Studies Demographics

#total number of people in all studies   
nrow(cdf) #1628 participants total

## [1] 1628

#how many white participants?  
cdf %>%   
 filter(ethnic\_id == "White") %>%  
 nrow() #1260 White participants

## [1] 1260

#selecting white participants who completed the attention checks; final sample  
cdff <- cdf %>%  
 filter(ethnic\_id == "White" & attn == 5 & attn2 == 3) #making final study 1 dataset  
nrow(cdff) #final sample is 1231

## [1] 1231

1260 - nrow(cdff) #29 people didn't pass attention checks

## [1] 29

## Analysis Chunk 14: Three-Way Interaction

First, we mean-center our continuous variables and change our dichotomous condition variable as a factor:

cdff$sdocent <- scale(cdff$sdo, center = TRUE, scale = FALSE)  
cdff$ethcent <- scale(cdff$ethnic, center = TRUE, scale = FALSE)  
cdff$sdocent <- cdff$sdocent[,1]  
cdff$ethcent <- cdff$ethcent[,1]  
cdff$cond2 <- as.factor(cdff$cond)

Now we can run our 3-way interaction between SDO, ethnic identification, and condition (including the simple slopes).

cdfmodel <- lm(form~sdocent\*ethcent\*cond2, data=cdff)  
summary(cdfmodel)

##   
## Call:  
## lm(formula = form ~ sdocent \* ethcent \* cond2, data = cdff)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.2260 -0.4165 0.2001 0.4631 3.6166   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.68084 0.04146 88.790 < 2e-16 \*\*\*  
## sdocent -0.05434 0.03421 -1.588 0.11247   
## ethcent -0.07311 0.02365 -3.092 0.00203 \*\*   
## cond21 -0.00728 0.05763 -0.126 0.89949   
## sdocent:ethcent -0.02076 0.01792 -1.159 0.24688   
## sdocent:cond21 0.20359 0.04713 4.320 1.69e-05 \*\*\*  
## ethcent:cond21 0.09328 0.03342 2.791 0.00533 \*\*   
## sdocent:ethcent:cond21 0.02400 0.02427 0.989 0.32294   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9828 on 1223 degrees of freedom  
## Multiple R-squared: 0.03501, Adjusted R-squared: 0.02948   
## F-statistic: 6.338 on 7 and 1223 DF, p-value: 2.478e-07

confint(cdfmodel) #this gives us the 95% confidence intervals

## 2.5 % 97.5 %  
## (Intercept) 3.59951191 3.76217672  
## sdocent -0.12145062 0.01278019  
## ethcent -0.11950191 -0.02671772  
## cond21 -0.12033522 0.10577586  
## sdocent:ethcent -0.05592727 0.01439918  
## sdocent:cond21 0.11112967 0.29604224  
## ethcent:cond21 0.02771995 0.15884016  
## sdocent:ethcent:cond21 -0.02361738 0.07161646

As before, we’ll reshape these results so can be used more easily.

cdfmodeldf <- tidy(cdfmodel) #pulling out regression results  
cdfcidf <- tidy(confint(cdfmodel)) #pulling out CIs  
names(cdfcidf) <- c("parameter", "lowerci", "upperci") #renaming  
  
cdfmodeldf <- bind\_cols(cdfmodeldf, cdfcidf) #binding regression results and CIs  
names(cdfmodeldf) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci") #and renaming  
  
cdfmodeldf %>%  
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>% #rounding  
 select(parameter, B, p, lowerci, upperci) #selecting variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.68 0.000 3.60 3.76  
## 2 sdocent -0.05 0.112 -0.12 0.01  
## 3 ethcent -0.07 0.002 -0.12 -0.03  
## 4 cond21 -0.01 0.899 -0.12 0.11  
## 5 sdocent:ethcent -0.02 0.247 -0.06 0.01  
## 6 sdocent:cond21 0.20 0.000 0.11 0.30  
## 7 ethcent:cond21 0.09 0.005 0.03 0.16  
## 8 sdocent:ethcent:cond21 0.02 0.323 -0.02 0.07

Now we can do a similar thing with the simple slopes (but we’ll have to manually compute the CIs).

cdfsimpslope <- simple\_slopes(cdfmodel) #pulling out the simple slopes to a df  
  
cdfsimpsloper <- cdfsimpslope[c(8, 2, 17, 11, 6, 4, 20, 14), ] #pulling out relevant rows  
names(cdfsimpsloper)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p") #renaming  
  
cdfsimpsloper %>%   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding according to APA  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",   
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter col  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.15 0.059 -0.01 0.32  
## 2 cond\_at\_low\_eth -0.17 0.039 -0.33 -0.01  
## 3 eth\_in\_high\_cond 0.02 0.393 -0.03 0.07  
## 4 eth\_in\_low\_cond -0.07 0.002 -0.12 -0.03  
## 5 cond\_at\_high\_sdo 0.25 0.003 0.08 0.41  
## 6 cond\_at\_low\_sdo -0.26 0.001 -0.42 -0.10  
## 7 sdo\_in\_high\_cond 0.15 0.000 0.08 0.21  
## 8 sdo\_in\_low\_cond -0.05 0.112 -0.12 0.01

And that’s it for the analyses for the main text!

## Analysis Chunk 15: Descriptive Statistics Table

#STUDY 1  
  
#N per condition  
study1f %>%  
 group\_by(cond2) %>%  
 summarise(n())

## # A tibble: 2 x 2  
## cond2 `n()`  
## <fct> <int>  
## 1 -1 226  
## 2 1 233

#mean and SD for key variables:  
describe(study1f$sdo)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 459 2.55 1.17 2.44 2.48 1.48 1 6.62 5.62 0.48 -0.54  
## se  
## X1 0.05

describe(study1f$ethnic)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 459 3.57 1.7 3.5 3.52 2.22 1 7 6 0.23 -0.91 0.08

describe(study1f$form)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 459 3.74 0.93 4 3.8 0 1 6 5 -0.81 1.5 0.04

describe(study1f$cat)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 459 3.57 1.09 4 3.59 0.74 1 7 6 -0.23 0.73  
## se  
## X1 0.05

library(Hmisc) #I library and remove this package as I need it because otherwise it messes up my "describe()" function

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

##   
## Attaching package: 'Hmisc'

## The following object is masked from 'package:psych':  
##   
## describe

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

study1cortest <-study1f[, c('cond','sdo', 'ethnic', 'form', 'cat')]  
s1cor <- rcorr(as.matrix(study1cortest), type='pearson')  
round(s1cor$r, 2)

## cond sdo ethnic form cat  
## cond 1.00 0.07 0.09 0.01 0.06  
## sdo 0.07 1.00 0.29 0.12 -0.02  
## ethnic 0.09 0.29 1.00 0.01 -0.03  
## form 0.01 0.12 0.01 1.00 0.54  
## cat 0.06 -0.02 -0.03 0.54 1.00

round(s1cor$P, 3)

## cond sdo ethnic form cat  
## cond NA 0.153 0.068 0.789 0.221  
## sdo 0.153 NA 0.000 0.012 0.637  
## ethnic 0.068 0.000 NA 0.837 0.580  
## form 0.789 0.012 0.837 NA 0.000  
## cat 0.221 0.637 0.580 0.000 NA

detach("package:Hmisc", unload=TRUE)   
  
  
#study 2  
#N per condition  
study2f %>%  
 group\_by(cond2) %>%  
 summarise(n())

## # A tibble: 2 x 2  
## cond2 `n()`  
## <fct> <int>  
## 1 -1 186  
## 2 1 218

#mean and SD for key variables:  
describe(study2f$sdo)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 404 2.68 1.27 2.5 2.59 1.58 1 7 6 0.58 -0.16 0.06

describe(study2f$ethnic)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 404 3.65 1.73 3.5 3.59 2.22 1 7 6 0.25 -0.82 0.09

describe(study2f$form)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 404 3.68 1.04 4 3.73 0.37 1 7 6 -0.37 1.73  
## se  
## X1 0.05

describe(study2f$cat)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 404 3.64 1.11 4 3.7 0 1 7 6 -0.47 1.33 0.06

library(Hmisc)

##   
## Attaching package: 'Hmisc'

## The following object is masked from 'package:psych':  
##   
## describe

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

study2cortest <-study2f[, c('cond','sdo', 'ethnic', 'form', 'cat')]  
s2cor <- rcorr(as.matrix(study2cortest), type='pearson')  
round(s2cor$r, 2)

## cond sdo ethnic form cat  
## cond 1.00 0.00 -0.07 0.17 0.13  
## sdo 0.00 1.00 0.20 -0.05 -0.13  
## ethnic -0.07 0.20 1.00 -0.11 -0.20  
## form 0.17 -0.05 -0.11 1.00 0.69  
## cat 0.13 -0.13 -0.20 0.69 1.00

round(s2cor$P, 3)

## cond sdo ethnic form cat  
## cond NA 0.978 0.167 0.000 0.011  
## sdo 0.978 NA 0.000 0.355 0.011  
## ethnic 0.167 0.000 NA 0.032 0.000  
## form 0.000 0.355 0.032 NA 0.000  
## cat 0.011 0.011 0.000 0.000 NA

detach("package:Hmisc", unload=TRUE)   
  
  
##study 3  
#N per condition  
study3f %>%  
 group\_by(cond2) %>%  
 summarise(n())

## # A tibble: 2 x 2  
## cond2 `n()`  
## <fct> <int>  
## 1 -1 189  
## 2 1 179

#mean and SD for key variables:  
describe(study3f$sdo)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 368 2.56 1.32 2.31 2.45 1.58 1 6.75 5.75 0.62 -0.44  
## se  
## X1 0.07

describe(study3f$ethnic)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 368 3.73 1.78 4 3.68 2.22 1 7 6 0.05 -0.95 0.09

describe(study3f$form)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 368 3.58 1.02 4 3.62 0.74 1 7 6 -0.32 1.17  
## se  
## X1 0.05

describe(study3f$cat)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 368 3.56 1.16 4 3.56 0.74 1 7 6 -0.01 0.68  
## se  
## X1 0.06

library(Hmisc)

##   
## Attaching package: 'Hmisc'

## The following object is masked from 'package:psych':  
##   
## describe

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

study3cortest <-study3f[, c('cond','sdo', 'ethnic', 'form', 'cat')]  
s3cor <- rcorr(as.matrix(study3cortest), type='pearson')  
round(s3cor$r, 2)

## cond sdo ethnic form cat  
## cond 1.00 -0.06 -0.04 -0.20 -0.19  
## sdo -0.06 1.00 0.27 0.08 0.02  
## ethnic -0.04 0.27 1.00 0.01 -0.02  
## form -0.20 0.08 0.01 1.00 0.71  
## cat -0.19 0.02 -0.02 0.71 1.00

round(s3cor$P, 3)

## cond sdo ethnic form cat  
## cond NA 0.247 0.411 0.000 0.000  
## sdo 0.247 NA 0.000 0.110 0.661  
## ethnic 0.411 0.000 NA 0.888 0.665  
## form 0.000 0.110 0.888 NA 0.000  
## cat 0.000 0.661 0.665 0.000 NA

detach("package:Hmisc", unload=TRUE)   
  
  
##combined data  
#N per condition  
cdff %>%  
 group\_by(cond2) %>%  
 summarise(n())

## # A tibble: 2 x 2  
## cond2 `n()`  
## <fct> <int>  
## 1 -1 601  
## 2 1 630

#mean and SD for key variables:  
describe(cdff$sdo)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 1231 2.6 1.25 2.44 2.51 1.58 1 7 6 0.57 -0.33  
## se  
## X1 0.04

describe(cdff$ethnic)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 1231 3.65 1.74 3.5 3.59 2.22 1 7 6 0.18 -0.89  
## se  
## X1 0.05

describe(cdff$form)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 1231 3.67 1 4 3.73 0.37 1 7 6 -0.5 1.49 0.03

describe(cdff$cat)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 1231 3.59 1.12 4 3.62 0.74 1 7 6 -0.24 0.9  
## se  
## X1 0.03

library(Hmisc)

##   
## Attaching package: 'Hmisc'

## The following object is masked from 'package:psych':  
##   
## describe

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

cdfcortest <-cdff[, c('cond','sdo', 'ethnic', 'form', 'cat')]  
cdfcor <- rcorr(as.matrix(cdfcortest), type='pearson')  
round(cdfcor$r, 2)

## cond sdo ethnic form cat  
## cond 1.00 0.01 -0.01 0.00 0.00  
## sdo 0.01 1.00 0.25 0.05 -0.04  
## ethnic -0.01 0.25 1.00 -0.03 -0.08  
## form 0.00 0.05 -0.03 1.00 0.65  
## cat 0.00 -0.04 -0.08 0.65 1.00

round(cdfcor$P, 3)

## cond sdo ethnic form cat  
## cond NA 0.824 0.860 0.892 0.900  
## sdo 0.824 NA 0.000 0.084 0.156  
## ethnic 0.860 0.000 NA 0.247 0.005  
## form 0.892 0.084 0.247 NA 0.000  
## cat 0.900 0.156 0.005 0.000 NA

detach("package:Hmisc", unload=TRUE)

## Analysis Chunk 16: Post-test Results (Footnote 8)

#reading in the data  
pretest <- read.csv('pretest from qualtrics 2.csv')  
  
#filtering out incomplete, non-white cases  
pretestr <- pretest %>%  
 filter(ethnic\_id == 3) %>%  
 filter(DO.BR.FL\_57 != "FL\_58|attention/suspicion check") %>%  
 filter(DO.BR.FL\_57 != 'FL\_129|attention/suspicion check') %>%  
 filter(DO.BR.FL\_57 != "attention/suspicion check|FL\_129") %>%  
 filter(DO.BR.FL\_57 != 'attention/suspicion check|FL\_58')  
  
nrow(pretestr) #224 participants

## [1] 224

#reverse code items:  
pretestr$SDO7\_5 <- 8-pretestr$SDO7\_5  
pretestr$SDO7\_6 <- 8-pretestr$SDO7\_6  
pretestr$SDO7\_7 <- 8-pretestr$SDO7\_7  
pretestr$SDO7\_8 <- 8-pretestr$SDO7\_8  
pretestr$SDO7\_13 <- 8-pretestr$SDO7\_13  
pretestr$SDO7\_14 <- 8-pretestr$SDO7\_14  
pretestr$SDO7\_15 <- 8-pretestr$SDO7\_15  
pretestr$SDO7\_16 <- 8-pretestr$SDO7\_16  
  
#creating SDO  
pretestr$sdo <- pretestr %>%  
 select(starts\_with("SDO7\_")) %>%  
 rowMeans(na.rm = TRUE)   
  
#SDO alpha is .94  
psych::alpha(select(pretestr, starts\_with("SDO7\_")))

##   
## Reliability analysis   
## Call: psych::alpha(x = select(pretestr, starts\_with("SDO7\_")))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.94 0.94 0.96 0.51 16 0.0057 2.5 1.2  
##   
## lower alpha upper 95% confidence boundaries  
## 0.93 0.94 0.95   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## SDO7\_1 0.94 0.94 0.95 0.51 16 0.0060  
## SDO7\_2 0.93 0.94 0.95 0.50 15 0.0064  
## SDO7\_3 0.93 0.94 0.95 0.50 15 0.0063  
## SDO7\_4 0.94 0.94 0.95 0.50 15 0.0062  
## SDO7\_5 0.94 0.94 0.95 0.51 16 0.0061  
## SDO7\_6 0.94 0.94 0.95 0.51 16 0.0061  
## SDO7\_7 0.94 0.95 0.96 0.54 17 0.0054  
## SDO7\_8 0.94 0.94 0.96 0.52 16 0.0057  
## SDO7\_9 0.93 0.94 0.95 0.50 15 0.0064  
## SDO7\_10 0.94 0.94 0.95 0.51 16 0.0059  
## SDO7\_11 0.93 0.94 0.95 0.50 15 0.0063  
## SDO7\_12 0.94 0.94 0.95 0.50 15 0.0062  
## SDO7\_13 0.94 0.94 0.95 0.50 15 0.0061  
## SDO7\_14 0.94 0.94 0.95 0.50 15 0.0063  
## SDO7\_15 0.94 0.94 0.95 0.50 15 0.0062  
## SDO7\_16 0.93 0.94 0.95 0.50 15 0.0063  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## SDO7\_1 215 0.69 0.70 0.68 0.65 2.0 1.4  
## SDO7\_2 215 0.83 0.83 0.83 0.80 2.4 1.6  
## SDO7\_3 215 0.80 0.80 0.79 0.77 2.5 1.7  
## SDO7\_4 215 0.75 0.76 0.75 0.71 2.1 1.5  
## SDO7\_5 215 0.71 0.72 0.69 0.67 2.4 1.5  
## SDO7\_6 215 0.70 0.71 0.69 0.66 2.2 1.5  
## SDO7\_7 215 0.45 0.45 0.40 0.37 2.5 1.8  
## SDO7\_8 215 0.59 0.60 0.56 0.53 2.7 1.8  
## SDO7\_9 215 0.83 0.82 0.82 0.80 2.6 1.9  
## SDO7\_10 215 0.69 0.67 0.65 0.63 2.9 2.1  
## SDO7\_11 215 0.81 0.79 0.79 0.77 2.7 1.9  
## SDO7\_12 215 0.77 0.75 0.74 0.72 3.0 2.0  
## SDO7\_13 215 0.74 0.75 0.74 0.71 2.1 1.4  
## SDO7\_14 215 0.80 0.79 0.79 0.76 2.6 1.8  
## SDO7\_15 215 0.77 0.77 0.76 0.73 2.5 1.6  
## SDO7\_16 215 0.80 0.79 0.78 0.76 2.7 1.7  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## SDO7\_1 0.55 0.21 0.08 0.07 0.07 0.02 0.01 0.04  
## SDO7\_2 0.45 0.20 0.08 0.13 0.09 0.03 0.02 0.04  
## SDO7\_3 0.41 0.20 0.09 0.13 0.11 0.03 0.02 0.04  
## SDO7\_4 0.53 0.20 0.08 0.08 0.07 0.03 0.01 0.04  
## SDO7\_5 0.41 0.20 0.18 0.12 0.06 0.01 0.02 0.04  
## SDO7\_6 0.46 0.19 0.15 0.13 0.04 0.01 0.02 0.04  
## SDO7\_7 0.44 0.20 0.13 0.07 0.06 0.06 0.05 0.04  
## SDO7\_8 0.36 0.23 0.09 0.13 0.08 0.05 0.05 0.04  
## SDO7\_9 0.43 0.17 0.08 0.13 0.09 0.04 0.06 0.04  
## SDO7\_10 0.40 0.18 0.08 0.10 0.07 0.07 0.10 0.04  
## SDO7\_11 0.41 0.15 0.12 0.13 0.08 0.05 0.07 0.04  
## SDO7\_12 0.32 0.20 0.10 0.11 0.10 0.10 0.07 0.04  
## SDO7\_13 0.47 0.20 0.14 0.10 0.04 0.02 0.01 0.04  
## SDO7\_14 0.37 0.19 0.18 0.11 0.06 0.05 0.05 0.04  
## SDO7\_15 0.38 0.21 0.16 0.13 0.07 0.02 0.02 0.04  
## SDO7\_16 0.35 0.22 0.14 0.13 0.06 0.06 0.04 0.04

#ALEX (HALF-LATINO, STUDY 1) ANALYSES  
#checking on status:  
describe(pretestr$Q90) #4.92 (1.11)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 216 4.92 1.11 5 4.98 1.48 1 7 6 -0.49 0.55  
## se  
## X1 0.08

#making alex formativeness scale  
pretestr$alex\_form <- pretestr %>%  
 select(Q98:Q104) %>%  
 rowMeans(na.rm = T) #making alex formativeness scale  
psych::alpha(select(pretestr, c(Q98:Q104))) #alpha = .94

##   
## Reliability analysis   
## Call: psych::alpha(x = select(pretestr, c(Q98:Q104)))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.94 0.94 0.92 0.79 15 0.007 3.5 1.1  
##   
## lower alpha upper 95% confidence boundaries  
## 0.92 0.94 0.95   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## Q98 0.91 0.91 0.88 0.78 10.4 0.0104  
## Q100 0.92 0.92 0.89 0.79 11.1 0.0098  
## Q102 0.91 0.91 0.87 0.77 9.8 0.0109  
## Q104 0.93 0.93 0.91 0.82 14.1 0.0077  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## Q98 215 0.93 0.93 0.90 0.87 3.6 1.1  
## Q100 215 0.92 0.92 0.88 0.85 3.5 1.2  
## Q102 215 0.93 0.94 0.92 0.88 3.6 1.1  
## Q104 215 0.89 0.89 0.82 0.80 3.5 1.2  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## Q98 0.04 0.13 0.27 0.38 0.14 0.03 0.00 0.04  
## Q100 0.04 0.15 0.27 0.40 0.09 0.04 0.01 0.04  
## Q102 0.04 0.13 0.25 0.41 0.13 0.04 0.00 0.04  
## Q104 0.06 0.15 0.25 0.38 0.11 0.04 0.01 0.04

describe(pretestr$alex\_form) #mean of 3.54, sd of 1.06

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 215 3.54 1.06 3.75 3.55 1.11 1 7 6 -0.08 0.3  
## se  
## X1 0.07

#ZACK (HALF-ARAB, STUDY 2) ANALYSES  
#checking on status:  
describe(pretestr$Q67) #4.09 (.94)--not bad

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 217 4.09 0.94 4 4.11 0 1 7 6 -0.14 1.31 0.06

#making zack formativeness scale  
pretestr$zack\_form <- pretestr %>%  
 select(Q77:Q83) %>%  
 rowMeans(na.rm = T) #making zack formativeness scale  
psych::alpha(select(pretestr, c(Q77:Q83))) #alpha = .95

##   
## Reliability analysis   
## Call: psych::alpha(x = select(pretestr, c(Q77:Q83)))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.95 0.95 0.94 0.83 19 0.0054 3.8 1.1  
##   
## lower alpha upper 95% confidence boundaries  
## 0.94 0.95 0.96   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## Q77 0.94 0.94 0.91 0.84 15 0.0071  
## Q79 0.94 0.94 0.91 0.83 15 0.0072  
## Q81 0.93 0.93 0.90 0.82 14 0.0077  
## Q83 0.93 0.93 0.90 0.82 14 0.0080  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## Q77 217 0.92 0.93 0.89 0.87 3.9 1.1  
## Q79 217 0.93 0.93 0.89 0.87 3.8 1.2  
## Q81 217 0.94 0.94 0.91 0.89 3.8 1.2  
## Q83 217 0.94 0.94 0.92 0.89 3.9 1.3  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## Q77 0.05 0.06 0.18 0.46 0.21 0.03 0.01 0.03  
## Q79 0.05 0.10 0.19 0.42 0.19 0.04 0.01 0.03  
## Q81 0.05 0.09 0.15 0.46 0.18 0.07 0.00 0.03  
## Q83 0.06 0.06 0.16 0.46 0.18 0.06 0.02 0.03

#mean of formativeness scale  
describe(pretestr$zack\_form) #mean of 3.83, sd of 1.11

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 217 3.83 1.11 4 3.9 0.74 1 6.5 5.5 -0.53 0.49  
## se  
## X1 0.08

#JORDAN (HALF-BLACK, STUDY 3) ANALYSES  
#checking on status:  
describe(pretestr$status) #4.05 (1.15)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 217 4.05 1.15 4 4.09 1.48 1 7 6 -0.31 -0.03  
## se  
## X1 0.08

#making jordon formativeness scale  
pretestr$jordon\_form <- pretestr %>%  
 select(hypo\_3:hypo\_1) %>%  
 rowMeans(na.rm = T) #making jordon formativeness scale  
psych::alpha(select(pretestr, c(hypo\_3:hypo\_1))) #alpha = .95

##   
## Reliability analysis   
## Call: psych::alpha(x = select(pretestr, c(hypo\_3:hypo\_1)))  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd  
## 0.95 0.95 0.93 0.87 20 0.0055 3.5 1.1  
##   
## lower alpha upper 95% confidence boundaries  
## 0.94 0.95 0.96   
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se  
## hypo\_3 0.94 0.94 0.88 0.88 15 0.0086  
## hypo\_2 0.93 0.93 0.86 0.86 13 0.0097  
## hypo\_1 0.93 0.93 0.87 0.87 13 0.0097  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## hypo\_3 217 0.95 0.95 0.91 0.89 3.5 1.1  
## hypo\_2 217 0.96 0.96 0.93 0.90 3.4 1.1  
## hypo\_1 217 0.96 0.96 0.93 0.90 3.4 1.1  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## hypo\_3 0.05 0.16 0.21 0.49 0.06 0.02 0.01 0.03  
## hypo\_2 0.07 0.14 0.18 0.54 0.05 0.01 0.01 0.03  
## hypo\_1 0.06 0.15 0.20 0.52 0.06 0.01 0.01 0.03

describe(pretestr$jordon\_form) #mean of 3.45, sd of 1.06

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 217 3.45 1.06 4 3.5 0.49 1 7 6 -0.32 0.75  
## se  
## X1 0.07

That concludes the analyses for the main text.

### Supplemental Materials Script

The first thing that we need to do is the run the analyses for the pilot and follow-up study from Study 1 separately. We call these Study 1a and Study 1b.

# Analyses for Study 1a

To run the appropriate analyses, we have to reduce the study1a dataframe to include only White participants who passed the attention check.

study1awhite<- study1afull %>%  
 filter(ethnic\_id == 3)  
study1af <- study1awhite %>%  
 filter(attn ==5 & attn2 == 3)

We will also need to create the appropriate scales.

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#SDO  
#reverse code items:  
study1af$SDO7\_5 <- 8-study1af$SDO7\_5  
study1af$SDO7\_6 <- 8-study1af$SDO7\_6  
study1af$SDO7\_7 <- 8-study1af$SDO7\_7  
study1af$SDO7\_8 <- 8-study1af$SDO7\_8  
study1af$SDO7\_13 <- 8-study1af$SDO7\_13  
study1af$SDO7\_14 <- 8-study1af$SDO7\_14  
study1af$SDO7\_15 <- 8-study1af$SDO7\_15  
study1af$SDO7\_16 <- 8-study1af$SDO7\_16  
  
#create the scale  
study1af$sdo <- study1af %>%  
 select(starts\_with("SDO7\_")) %>%  
 rowMeans(na.rm = TRUE)   
  
#checking for NAs (empty cases)--there shouldn't really be any because everyone should have at least some SDO values in this   
anyNA(study1af$sdo) #false means no NAs

## [1] FALSE

#\_\_\_\_\_\_\_\_\_\_\_\_\_#ETHNIC ID   
#we want to use ethnic\_id2 & ethnic\_id3; those are the only ones in all studies.  
study1af$ethnic <- rowMeans(subset(study1af, select = c(ethnic\_id2, ethnic\_id3)), na.rm = TRUE) #making scale  
summary(study1$ethnic) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.500 4.000 3.896 5.500 7.000

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#FORMATIVENESS & CATEGORIZATION  
#hypo 1 and 2 are categorization; hypo 4-5, 11-12 are formativeness (but 11 and 12 are not included in Study 1a so we are not looking at them in this study)  
study1af$form <- rowMeans(subset(study1af, select = c(hypo\_4, hypo\_5), na.rm = TRUE))  
study1af$cat <- rowMeans(subset(study1af, select = c(hypo\_1, hypo\_2), na.rm = TRUE))  
summary(study1af$form) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 3.500 4.000 3.696 4.000 7.000

summary(study1af$cat) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 3.000 4.000 3.473 4.000 7.000

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#MAKING GENDER A FACTOR  
study1af$demo\_gend2 <- factor(study1af$demo\_gend, labels = c("male", "female"))

Next, let’s look at the demographics for this study

#number of participants   
nrow(study1afull) #327 participants

## [1] 327

nrow(study1awhite) #244 white participants

## [1] 244

nrow(study1af) #240 white paritcipants who passed the attention check

## [1] 240

# calculating gender distribution  
study1af %>%  
 group\_by(demo\_gend2) %>%  
 summarise('n' = n(),'%' = n()/nrow(study1af))

## # A tibble: 2 x 3  
## demo\_gend2 n `%`  
## <fct> <int> <dbl>  
## 1 male 82 0.342  
## 2 female 158 0.658

#ages  
#making a factor  
study1af$demo\_age <- as.numeric(as.character(study1af$demo\_age)) #this will throuh a warning because there is one invalid non-numeric response. That is ok because it will just turn it into an NA which is what we want.   
  
#checking for outliers  
study1af$demo\_age[study1af$demo\_age > 100 | study1af$demo\_age < 18]

## [1] 335

#removing outlier  
study1af$demo\_age[study1af$demo\_age > 100 | study1af$demo\_age < 18] <- NA   
  
#calulating mean age  
describe(study1af$demo\_age, na.rm = TRUE) #mean age is 35.90; sd = 12.8

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 239 35.9 12.8 33 34.73 11.86 18 69 51 0.71 -0.37  
## se  
## X1 0.83

Now we can look at the 3-way interaction.  
First, we mean-center our continuous variables. We want the neutral category to be our reference category. In order to do this, we are going to recode the condition variable so that the neutral category has the smallest value. We then make it a factor.

study1af$sdocent <- scale(study1af$sdo, center = TRUE, scale = FALSE)  
study1af$ethcent <- scale(study1af$ethnic, center = TRUE, scale = FALSE)  
study1af$sdocent <- study1af$sdocent[,1]  
study1af$ethcent <- study1af$ethcent[,1]  
#making the 'neutral' category the reference category.  
study1af$cond2 <- ifelse(study1af$condition == 2, 0,  
 ifelse(study1af$condition == 1, 1, 2))  
study1af$cond2 <- as.factor(study1af$cond2)  
levels(study1af$cond2)

## [1] "0" "1" "2"

Now we can run our 3-way interaction between SDO, ethnic identification, and condition (including the simple slopes).

s1amodel <- lm(form~sdocent\*ethcent\*cond2, data=study1af)   
summary(s1amodel)

##   
## Call:  
## lm(formula = form ~ sdocent \* ethcent \* cond2, data = study1af)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.5914 -0.3332 0.1493 0.3835 3.1381   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.65903 0.09273 39.459 <2e-16 \*\*\*  
## sdocent 0.02153 0.08268 0.260 0.7948   
## ethcent -0.03902 0.06163 -0.633 0.5273   
## cond21 0.15535 0.13577 1.144 0.2537   
## cond22 -0.13624 0.13741 -0.992 0.3225   
## sdocent:ethcent 0.05790 0.05475 1.058 0.2914   
## sdocent:cond21 -0.09134 0.12780 -0.715 0.4755   
## sdocent:cond22 0.30886 0.12822 2.409 0.0168 \*   
## ethcent:cond21 -0.02778 0.08770 -0.317 0.7517   
## ethcent:cond22 -0.02385 0.09062 -0.263 0.7926   
## sdocent:ethcent:cond21 -0.14654 0.08456 -1.733 0.0845 .   
## sdocent:ethcent:cond22 0.14160 0.08407 1.684 0.0935 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.8504 on 228 degrees of freedom  
## Multiple R-squared: 0.115, Adjusted R-squared: 0.07225   
## F-statistic: 2.692 on 11 and 228 DF, p-value: 0.002827

confint(s1amodel) #this gives us the 95% confidence intervals

## 2.5 % 97.5 %  
## (Intercept) 3.47631118 3.84175012  
## sdocent -0.14138611 0.18445036  
## ethcent -0.16046109 0.08241838  
## cond21 -0.11218112 0.42288632  
## cond22 -0.40699482 0.13450733  
## sdocent:ethcent -0.04997635 0.16577387  
## sdocent:cond21 -0.34315052 0.16047910  
## sdocent:cond22 0.05620525 0.56151418  
## ethcent:cond21 -0.20058988 0.14502239  
## ethcent:cond22 -0.20240229 0.15470420  
## sdocent:ethcent:cond21 -0.31316820 0.02008337  
## sdocent:ethcent:cond22 -0.02405973 0.30725660

As before, we’ll reshape these results so they can be used more easily.

s1amodeldf <- tidy(s1amodel) #pulling out regression results  
s1amodeldf

## term estimate std.error statistic p.value  
## 1 (Intercept) 3.65903065 0.09273108 39.4585154 7.420276e-104  
## 2 sdocent 0.02153212 0.08268185 0.2604214 7.947738e-01  
## 3 ethcent -0.03902135 0.06163129 -0.6331419 5.272755e-01  
## 4 cond21 0.15535260 0.13577475 1.1441936 2.537432e-01  
## 5 cond22 -0.13624375 0.13740758 -0.9915301 3.224779e-01  
## 6 sdocent:ethcent 0.05789876 0.05474718 1.0575661 2.913725e-01  
## 7 sdocent:cond21 -0.09133571 0.12779732 -0.7146919 4.755306e-01  
## 8 sdocent:cond22 0.30885971 0.12822345 2.4087615 1.680132e-02  
## 9 ethcent:cond21 -0.02778374 0.08770001 -0.3168043 7.516819e-01  
## 10 ethcent:cond22 -0.02384905 0.09061670 -0.2631860 7.926449e-01  
## 11 sdocent:ethcent:cond21 -0.14654242 0.08456345 -1.7329285 8.446049e-02  
## 12 sdocent:ethcent:cond22 0.14159844 0.08407238 1.6842445 9.350285e-02

s1acidf <- tidy(confint(s1amodel)) #pulling out CIs  
names(s1acidf) <- c("parameter", "lowerci", "upperci") #renamingh  
s1amodeldf

## term estimate std.error statistic p.value  
## 1 (Intercept) 3.65903065 0.09273108 39.4585154 7.420276e-104  
## 2 sdocent 0.02153212 0.08268185 0.2604214 7.947738e-01  
## 3 ethcent -0.03902135 0.06163129 -0.6331419 5.272755e-01  
## 4 cond21 0.15535260 0.13577475 1.1441936 2.537432e-01  
## 5 cond22 -0.13624375 0.13740758 -0.9915301 3.224779e-01  
## 6 sdocent:ethcent 0.05789876 0.05474718 1.0575661 2.913725e-01  
## 7 sdocent:cond21 -0.09133571 0.12779732 -0.7146919 4.755306e-01  
## 8 sdocent:cond22 0.30885971 0.12822345 2.4087615 1.680132e-02  
## 9 ethcent:cond21 -0.02778374 0.08770001 -0.3168043 7.516819e-01  
## 10 ethcent:cond22 -0.02384905 0.09061670 -0.2631860 7.926449e-01  
## 11 sdocent:ethcent:cond21 -0.14654242 0.08456345 -1.7329285 8.446049e-02  
## 12 sdocent:ethcent:cond22 0.14159844 0.08407238 1.6842445 9.350285e-02

s1acidf

## parameter lowerci upperci  
## 1 (Intercept) 3.47631118 3.84175012  
## 2 sdocent -0.14138611 0.18445036  
## 3 ethcent -0.16046109 0.08241838  
## 4 cond21 -0.11218112 0.42288632  
## 5 cond22 -0.40699482 0.13450733  
## 6 sdocent:ethcent -0.04997635 0.16577387  
## 7 sdocent:cond21 -0.34315052 0.16047910  
## 8 sdocent:cond22 0.05620525 0.56151418  
## 9 ethcent:cond21 -0.20058988 0.14502239  
## 10 ethcent:cond22 -0.20240229 0.15470420  
## 11 sdocent:ethcent:cond21 -0.31316820 0.02008337  
## 12 sdocent:ethcent:cond22 -0.02405973 0.30725660

s1amodeldf <- bind\_cols(s1amodeldf, s1acidf) #binding regression results and CIs  
names(s1amodeldf) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci") #and renaming  
  
s1amodeldf %>%  
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>% #rounding  
 select(parameter, B, p, lowerci, upperci) #selecting variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.66 0.000 3.48 3.84  
## 2 sdocent 0.02 0.795 -0.14 0.18  
## 3 ethcent -0.04 0.527 -0.16 0.08  
## 4 cond21 0.16 0.254 -0.11 0.42  
## 5 cond22 -0.14 0.322 -0.41 0.13  
## 6 sdocent:ethcent 0.06 0.291 -0.05 0.17  
## 7 sdocent:cond21 -0.09 0.476 -0.34 0.16  
## 8 sdocent:cond22 0.31 0.017 0.06 0.56  
## 9 ethcent:cond21 -0.03 0.752 -0.20 0.15  
## 10 ethcent:cond22 -0.02 0.793 -0.20 0.15  
## 11 sdocent:ethcent:cond21 -0.15 0.084 -0.31 0.02  
## 12 sdocent:ethcent:cond22 0.14 0.094 -0.02 0.31

That’s great. We also need to find out the results that compare the disreputable condition with the reputable condition.

study1af$cond3 <- ifelse(study1af$cond2 == 0, 1, ifelse(study1af$cond2==1, 0, 2))  
study1af$cond3 <- as.factor(study1af$cond3)  
levels(study1af$cond3)

## [1] "0" "1" "2"

#verifying that this worked correctly  
 study1af %>%  
 group\_by(cond3) %>%  
 summarise(n())

## # A tibble: 3 x 2  
## cond3 `n()`  
## <fct> <int>  
## 1 0 76  
## 2 1 86  
## 3 2 78

study1af %>%  
 group\_by(cond2) %>%  
 summarise(n())

## # A tibble: 3 x 2  
## cond2 `n()`  
## <fct> <int>  
## 1 0 86  
## 2 1 76  
## 3 2 78

#yes, run model  
s1amodel2 <- lm(form~sdocent\*ethcent\*cond3, data=study1af)   
summary(s1amodel2)

##   
## Call:  
## lm(formula = form ~ sdocent \* ethcent \* cond3, data = study1af)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.5914 -0.3332 0.1493 0.3835 3.1381   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.814383 0.099175 38.461 < 2e-16 \*\*\*  
## sdocent -0.069804 0.097447 -0.716 0.47452   
## ethcent -0.066805 0.062393 -1.071 0.28543   
## cond31 -0.155353 0.135775 -1.144 0.25374   
## cond32 -0.291596 0.141836 -2.056 0.04093 \*   
## sdocent:ethcent -0.088644 0.064449 -1.375 0.17036   
## sdocent:cond31 0.091336 0.127797 0.715 0.47553   
## sdocent:cond32 0.400195 0.138206 2.896 0.00415 \*\*   
## ethcent:cond31 0.027784 0.087700 0.317 0.75168   
## ethcent:cond32 0.003935 0.091136 0.043 0.96560   
## sdocent:ethcent:cond31 0.146542 0.084563 1.733 0.08446 .   
## sdocent:ethcent:cond32 0.288141 0.090690 3.177 0.00169 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.8504 on 228 degrees of freedom  
## Multiple R-squared: 0.115, Adjusted R-squared: 0.07225   
## F-statistic: 2.692 on 11 and 228 DF, p-value: 0.002827

confint(s1amodel2)

## 2.5 % 97.5 %  
## (Intercept) 3.61896604 4.00980046  
## sdocent -0.26181491 0.12220773  
## ethcent -0.18974554 0.05613534  
## cond31 -0.42288632 0.11218112  
## cond32 -0.57107377 -0.01211892  
## sdocent:ethcent -0.21563623 0.03834892  
## sdocent:cond31 -0.16047910 0.34315052  
## sdocent:cond32 0.12787158 0.67251927  
## ethcent:cond31 -0.14502239 0.20058988  
## ethcent:cond32 -0.17564260 0.18351200  
## sdocent:ethcent:cond31 -0.02008337 0.31316820  
## sdocent:ethcent:cond32 0.10944362 0.46683809

simple\_slopes(s1amodel2)

## sdocent ethcent cond3 Test Estimate Std. Error t value df  
## 1a -1.069293 -1.545169 sstest -0.0538 0.2559 -0.2103 228  
## 1b -1.069293 -1.545169 sstest -0.2495 0.2429 -1.0275 228  
## 2a 0 -1.545169 sstest -0.1983 0.1915 -1.0356 228  
## 2b 0 -1.545169 sstest -0.2977 0.1980 -1.5034 228  
## 3a 1.069293 -1.545169 sstest -0.3427 0.2886 -1.1874 228  
## 3b 1.069293 -1.545169 sstest -0.3458 0.3255 -1.0623 228  
## 4a -1.069293 0 sstest -0.2530 0.1918 -1.3190 228  
## 4b -1.069293 0 sstest -0.7195 0.1990 -3.6160 228  
## 5a 0 0 sstest -0.1554 0.1358 -1.1442 228  
## 5b 0 0 sstest -0.2916 0.1418 -2.0559 228  
## 6a 1.069293 0 sstest -0.0577 0.1934 -0.2982 228  
## 6b 1.069293 0 sstest 0.1363 0.2105 0.6476 228  
## 7a -1.069293 1.545169 sstest -0.4522 0.2895 -1.5618 228  
## 7b -1.069293 1.545169 sstest -1.1895 0.3164 -3.7595 228  
## 8a 0 1.545169 sstest -0.1124 0.1922 -0.5850 228  
## 8b 0 1.545169 sstest -0.2855 0.2017 -1.4154 228  
## 9a 1.069293 1.545169 sstest 0.2274 0.2595 0.8762 228  
## 9b 1.069293 1.545169 sstest 0.6185 0.2681 2.3067 228  
## 10 -1.069293 sstest 0 0.0280 0.0883 0.3168 228  
## 11 0 sstest 0 -0.0668 0.0624 -1.0707 228  
## 12 1.069293 sstest 0 -0.1616 0.0974 -1.6592 228  
## 13 sstest -1.545169 0 0.0672 0.1312 0.5121 228  
## 14 sstest 0 0 -0.0698 0.0974 -0.7163 228  
## 15 sstest 1.545169 0 -0.2068 0.1470 -1.4062 228  
## 16 -1.069293 sstest 1 -0.1009 0.0898 -1.1243 228  
## 17 0 sstest 1 -0.0390 0.0616 -0.6331 228  
## 18 1.069293 sstest 1 0.0229 0.0799 0.2863 228  
## 19 sstest -1.545169 1 -0.0679 0.1257 -0.5404 228  
## 20 sstest 0 1 0.0215 0.0827 0.2604 228  
## 21 sstest 1.545169 1 0.1110 0.1104 1.0056 228  
## 22 -1.069293 sstest 2 -0.2762 0.0945 -2.9221 228  
## 23 0 sstest 2 -0.0629 0.0664 -0.9464 228  
## 24 1.069293 sstest 2 0.1505 0.0959 1.5684 228  
## 25 sstest -1.545169 2 0.0221 0.1437 0.1541 228  
## 26 sstest 0 2 0.3304 0.0980 3.3712 228  
## 27 sstest 1.545169 2 0.6386 0.1342 4.7596 228  
## Pr(>|t|) Sig.  
## 1a 0.8335895   
## 1b 0.3052890   
## 2a 0.3014982   
## 2b 0.1341157   
## 3a 0.2362971   
## 3b 0.2892238   
## 4a 0.1884882   
## 4b 0.0003683 \*\*\*  
## 5a 0.2537432   
## 5b 0.0409344 \*  
## 6a 0.7658128   
## 6b 0.5179133   
## 7a 0.1197226   
## 7b 0.0002164 \*\*\*  
## 8a 0.5591462   
## 8b 0.1583194   
## 9a 0.3818651   
## 9b 0.0219687 \*  
## 10 0.7516560   
## 11 0.2854292   
## 12 0.0984540 .  
## 13 0.6090911   
## 14 0.4745234   
## 15 0.1610315   
## 16 0.2620653   
## 17 0.5272755   
## 18 0.7749065   
## 19 0.5894345   
## 20 0.7947738   
## 21 0.3156922   
## 22 0.0038268 \*\*  
## 23 0.3449399   
## 24 0.1181661   
## 25 0.8777027   
## 26 0.0008790 \*\*\*  
## 27 3.445e-06 \*\*\*

s1amodel2df <- tidy(s1amodel2) #pulling out regression results  
s1aci2df <- tidy(confint(s1amodel2)) #pulling out CIs  
names(s1aci2df) <- c("parameter", "lowerci", "upperci") #renamingh  
  
  
s1amodel2df <- bind\_cols(s1amodel2df, s1aci2df) #binding regression results and CIs  
names(s1amodel2df) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci") #and renaming  
  
s1amodel2df %>%  
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>% #rounding  
 select(parameter, B, p, lowerci, upperci) #selecting variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.81 0.000 3.62 4.01  
## 2 sdocent -0.07 0.475 -0.26 0.12  
## 3 ethcent -0.07 0.285 -0.19 0.06  
## 4 cond31 -0.16 0.254 -0.42 0.11  
## 5 cond32 -0.29 0.041 -0.57 -0.01  
## 6 sdocent:ethcent -0.09 0.170 -0.22 0.04  
## 7 sdocent:cond31 0.09 0.476 -0.16 0.34  
## 8 sdocent:cond32 0.40 0.004 0.13 0.67  
## 9 ethcent:cond31 0.03 0.752 -0.15 0.20  
## 10 ethcent:cond32 0.00 0.966 -0.18 0.18  
## 11 sdocent:ethcent:cond31 0.15 0.084 -0.02 0.31  
## 12 sdocent:ethcent:cond32 0.29 0.002 0.11 0.47

write.csv(study1af, "study\_1a.csv", row.names = F)

Now we can need to get the simple slopes (for both of the models above).

#first set of condition level comparisons  
s1asimpslope <- simple\_slopes(s1amodel) #pulling out the simple slopes to a df  
print(s1asimpslope)

## sdocent ethcent cond2 Test Estimate Std. Error t value df  
## 1a -1.069293 -1.545169 sstest 0.0538 0.2559 0.2103 228  
## 1b -1.069293 -1.545169 sstest -0.1957 0.2583 -0.7576 228  
## 2a 0 -1.545169 sstest 0.1983 0.1915 1.0356 228  
## 2b 0 -1.545169 sstest -0.0994 0.1991 -0.4991 228  
## 3a 1.069293 -1.545169 sstest 0.3427 0.2886 1.1874 228  
## 3b 1.069293 -1.545169 sstest -0.0031 0.3097 -0.0100 228  
## 4a -1.069293 0 sstest 0.2530 0.1918 1.3190 228  
## 4b -1.069293 0 sstest -0.4665 0.1951 -2.3917 228  
## 5a 0 0 sstest 0.1554 0.1358 1.1442 228  
## 5b 0 0 sstest -0.1362 0.1374 -0.9915 228  
## 6a 1.069293 0 sstest 0.0577 0.1934 0.2982 228  
## 6b 1.069293 0 sstest 0.1940 0.1932 1.0044 228  
## 7a -1.069293 1.545169 sstest 0.4522 0.2895 1.5618 228  
## 7b -1.069293 1.545169 sstest -0.7373 0.3008 -2.4508 228  
## 8a 0 1.545169 sstest 0.1124 0.1922 0.5850 228  
## 8b 0 1.545169 sstest -0.1731 0.1932 -0.8960 228  
## 9a 1.069293 1.545169 sstest -0.2274 0.2595 -0.8762 228  
## 9b 1.069293 1.545169 sstest 0.3911 0.2306 1.6964 228  
## 10 -1.069293 sstest 0 -0.1009 0.0898 -1.1243 228  
## 11 0 sstest 0 -0.0390 0.0616 -0.6331 228  
## 12 1.069293 sstest 0 0.0229 0.0799 0.2863 228  
## 13 sstest -1.545169 0 -0.0679 0.1257 -0.5404 228  
## 14 sstest 0 0 0.0215 0.0827 0.2604 228  
## 15 sstest 1.545169 0 0.1110 0.1104 1.0056 228  
## 16 -1.069293 sstest 1 0.0280 0.0883 0.3168 228  
## 17 0 sstest 1 -0.0668 0.0624 -1.0707 228  
## 18 1.069293 sstest 1 -0.1616 0.0974 -1.6592 228  
## 19 sstest -1.545169 1 0.0672 0.1312 0.5121 228  
## 20 sstest 0 1 -0.0698 0.0974 -0.7163 228  
## 21 sstest 1.545169 1 -0.2068 0.1470 -1.4062 228  
## 22 -1.069293 sstest 2 -0.2762 0.0945 -2.9221 228  
## 23 0 sstest 2 -0.0629 0.0664 -0.9464 228  
## 24 1.069293 sstest 2 0.1505 0.0959 1.5684 228  
## 25 sstest -1.545169 2 0.0221 0.1437 0.1541 228  
## 26 sstest 0 2 0.3304 0.0980 3.3712 228  
## 27 sstest 1.545169 2 0.6386 0.1342 4.7596 228  
## Pr(>|t|) Sig.  
## 1a 0.833590   
## 1b 0.449462   
## 2a 0.301498   
## 2b 0.618172   
## 3a 0.236297   
## 3b 0.992059   
## 4a 0.188488   
## 4b 0.017583 \*  
## 5a 0.253743   
## 5b 0.322478   
## 6a 0.765813   
## 6b 0.316249   
## 7a 0.119723   
## 7b 0.015008 \*  
## 8a 0.559146   
## 8b 0.371177   
## 9a 0.381865   
## 9b 0.091179 .  
## 10 0.262065   
## 11 0.527276   
## 12 0.774907   
## 13 0.589435   
## 14 0.794774   
## 15 0.315692   
## 16 0.751656   
## 17 0.285429   
## 18 0.098454 .  
## 19 0.609091   
## 20 0.474523   
## 21 0.161032   
## 22 0.003827 \*\*  
## 23 0.344940   
## 24 0.118166   
## 25 0.877703   
## 26 0.000879 \*\*\*  
## 27 3.445e-06 \*\*\*

s1asimpsloper <- s1asimpslope[c(16, 15, 4, 3, 32, 20, 26, 35, 23, 29, 12, 11, 8, 7), ] #pulling out relevant rows  
names(s1asimpsloper)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p") #renaming  
  
  
s1asimpsloper %>%   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding according to APA  
 mutate(parameter = c("cond\_neut\_v\_high\_at\_high\_eth","cond\_neut\_v\_low\_at\_high\_eth",  
 "cond\_neut\_v\_high\_at\_low\_eth","cond\_neut\_v\_low\_at\_low\_eth",  
 "eth\_in\_high\_cond", "eth\_in\_neut\_cond", "eth\_in\_low\_cond",  
 "sdo\_in\_high\_cond", "sdo\_in\_neut\_cond", "sdo\_in\_low\_cond",   
 "cond\_neut\_v\_high\_at\_high\_sdo","cond\_neut\_v\_low\_at\_high\_sdo",  
 "cond\_neut\_v\_high\_at\_low\_sdo","cond\_neut\_v\_low\_at\_low\_sdo")) %>% #making new parameter col  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_neut\_v\_high\_at\_high\_eth -0.17 0.371 -0.56 0.21  
## 2 cond\_neut\_v\_low\_at\_high\_eth 0.11 0.559 -0.27 0.50  
## 3 cond\_neut\_v\_high\_at\_low\_eth -0.10 0.618 -0.50 0.30  
## 4 cond\_neut\_v\_low\_at\_low\_eth 0.20 0.301 -0.18 0.58  
## 5 eth\_in\_high\_cond -0.06 0.345 -0.20 0.07  
## 6 eth\_in\_neut\_cond -0.04 0.527 -0.16 0.08  
## 7 eth\_in\_low\_cond -0.07 0.285 -0.19 0.06  
## 8 sdo\_in\_high\_cond 0.33 0.001 0.13 0.53  
## 9 sdo\_in\_neut\_cond 0.02 0.795 -0.14 0.19  
## 10 sdo\_in\_low\_cond -0.07 0.475 -0.26 0.13  
## 11 cond\_neut\_v\_high\_at\_high\_sdo 0.19 0.316 -0.19 0.58  
## 12 cond\_neut\_v\_low\_at\_high\_sdo 0.06 0.766 -0.33 0.44  
## 13 cond\_neut\_v\_high\_at\_low\_sdo -0.47 0.018 -0.86 -0.08  
## 14 cond\_neut\_v\_low\_at\_low\_sdo 0.25 0.188 -0.13 0.64

Now we can do the same thing for the second model, which compares the disreputable condition to the reputable condition. It will be abbreviated, though, because we are only looking at four comparisons.

#first set of condition level comparisons  
s1a2simpslope <- simple\_slopes(s1amodel2) #pulling out the simple slopes to a df  
  
  
s1a2simpsloper <- s1a2simpslope[c(16, 4, 12, 8), ] #pulling out relevant rows  
names(s1a2simpsloper)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p") #renaming  
  
  
s1a2simpsloper %>%   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding according to APA  
 mutate(parameter = c("cond\_low\_v\_high\_at\_high\_eth",  
 "cond\_low\_v\_high\_at\_low\_eth",  
 "cond\_low\_v\_high\_at\_high\_sdo",  
 "cond\_low\_v\_high\_at\_low\_sdo")) %>% #making new parameter col  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_low\_v\_high\_at\_high\_eth -0.29 0.158 -0.69 0.12  
## 2 cond\_low\_v\_high\_at\_low\_eth -0.30 0.134 -0.69 0.10  
## 3 cond\_low\_v\_high\_at\_high\_sdo 0.14 0.518 -0.28 0.56  
## 4 cond\_low\_v\_high\_at\_low\_sdo -0.72 0.000 -1.12 -0.32

# Analyses for Study 1b

Now we can do the same thing but for Study 1b.

study1bwhite <- study1b %>%  
 filter(ethnic\_id == 3)  
study1bf <- study1bwhite %>%  
 filter(attn ==5 & attn2 == 3)

We will also need to create the appropriate scales.

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#SDO  
#reverse code items:  
study1bf$SDO7\_5 <- 8-study1bf$SDO7\_5  
study1bf$SDO7\_6 <- 8-study1bf$SDO7\_6  
study1bf$SDO7\_7 <- 8-study1bf$SDO7\_7  
study1bf$SDO7\_8 <- 8-study1bf$SDO7\_8  
study1bf$SDO7\_13 <- 8-study1bf$SDO7\_13  
study1bf$SDO7\_14 <- 8-study1bf$SDO7\_14  
study1bf$SDO7\_15 <- 8-study1bf$SDO7\_15  
study1bf$SDO7\_16 <- 8-study1bf$SDO7\_16  
  
#create the scale  
study1bf$sdo <- study1bf %>%  
 select(starts\_with("SDO7\_")) %>%  
 rowMeans(na.rm = TRUE)   
  
#checking for NAs (empty cases)--there shouldn't really be any because everyone should have at least some SDO values in this   
anyNA(study1bf$sdo) #false means no NAs

## [1] FALSE

#\_\_\_\_\_\_\_\_\_\_\_\_\_#ETHNIC ID   
#we want to use ethnic\_id2 & ethnic\_id3; those are the only ones in all studies.  
study1bf$ethnic <- rowMeans(subset(study1bf, select = c(ethnic\_id2, ethnic\_id3)), na.rm = TRUE) #making scale  
summary(study1$ethnic) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.500 4.000 3.896 5.500 7.000

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#FORMATIVENESS & CATEGORIZATION  
#hypo 1 and 2 are categorization; hypo 4-5, 11-12 are formativeness (but 11 and 12 are not included in Study 1b so we are not looking at them in this study)  
study1bf <- study1bf %>%  
 rename(hypo\_1 = Q78, hypo\_2 = Q80) #renaming these variables  
study1bf$form <- rowMeans(subset(study1bf, select = c(hypo\_4, hypo\_5), na.rm = TRUE))  
study1bf$cat <- rowMeans(subset(study1bf, select = c(hypo\_1, hypo\_2), na.rm = TRUE))  
summary(study1bf$form) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 3.500 4.000 3.748 4.000 6.000

summary(study1bf$cat) #no NAs, great

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 3.000 4.000 3.613 4.000 7.000

#\_\_\_\_\_\_\_\_\_\_\_\_\_  
#MAKING GENDER A FACTOR  
study1bf$demo\_gend2 <- factor(study1bf$demo\_gend, labels = c("male", "female"))

Now for demographics

#number of participants   
nrow(study1b) #402 participants

## [1] 402

nrow(study1bwhite) #313 white participants

## [1] 313

nrow(study1bf) #305 white paritcipants who passed the attention check

## [1] 305

# calculating gender distribution  
study1bf %>%  
 group\_by(demo\_gend2) %>%  
 summarise('n' = n(),'%' = n()/nrow(study1bf))

## # A tibble: 2 x 3  
## demo\_gend2 n `%`  
## <fct> <int> <dbl>  
## 1 male 153 0.502  
## 2 female 152 0.498

#ages  
#making a factor  
study1bf$demo\_age <- as.numeric(as.character(study1bf$demo\_age)) #this will throuh a warning because there is one invalid non-numeric response. That is ok because it will just turn it into an NA which is what we want.   
  
#checking for outliers  
study1bf$demo\_age[study1bf$demo\_age > 100 | study1bf$demo\_age < 18]

## numeric(0)

#there are no outliers  
  
#calulating mean age  
describe(study1bf$demo\_age, na.rm = TRUE) #mean age is 35.90; sd = 12.8

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 305 35.9 11.74 33 34.45 10.38 19 70 51 1.03 0.32  
## se  
## X1 0.67

Now we can look at the 3-way interaction.

First, we mean-centered our continuous variables and change our dichotomous condition variable as a factor:

study1bf$sdocent <- scale(study1bf$sdo, center = TRUE, scale = FALSE)  
study1bf$ethcent <- scale(study1bf$ethnic, center = TRUE, scale = FALSE)  
study1bf$sdocent <- study1bf$sdocent[,1]  
study1bf$ethcent <- study1bf$ethcent[,1]  
study1bf$cond2 <- as.factor(study1bf$cond)

Now we can run our 3-way interaction between SDO, ethnic identification, and condition (including the simple slopes).

s1bmodel <- lm(form~sdocent\*ethcent\*cond2, data=study1bf)

We’ll reshape these results so can be used more easily.

s1bmodeldf <- tidy(s1bmodel) #pulling out regression results  
s1bcidf <- tidy(confint(s1bmodel)) #pulling out CIs  
names(s1bcidf) <- c("parameter", "lowerci", "upperci") #renamingh  
  
s1bmodeldf <- bind\_cols(s1bmodeldf, s1bcidf) #binding regression results and CIs  
names(s1bmodeldf) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci") #and renaming  
  
s1bmodeldf %>%  
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>% #rounding  
 select(parameter, B, p, lowerci, upperci) #selecting variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.73 0.000 3.57 3.90  
## 2 sdocent 0.00 0.966 -0.14 0.14  
## 3 ethcent -0.07 0.128 -0.16 0.02  
## 4 cond21 0.05 0.629 -0.17 0.28  
## 5 sdocent:ethcent -0.08 0.030 -0.15 -0.01  
## 6 sdocent:cond21 0.19 0.050 0.00 0.37  
## 7 ethcent:cond21 0.15 0.019 0.02 0.28  
## 8 sdocent:ethcent:cond21 0.06 0.214 -0.04 0.16

Now we can do a similar thing with the simple slopes (but we’ll have to manually compute the CIs again).

s1bsimpslope <- simple\_slopes(s1bmodel) #pulling out the simple slopes to a df  
  
s1bsimpsloper <- s1bsimpslope[c(8, 2, 17, 11, 6, 4, 20, 14), ] #pulling out relevant rows  
names(s1bsimpsloper)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p") #renaming  
  
s1bsimpsloper %>%   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding according to APA  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",   
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter col  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.32 0.043 0.01 0.64  
## 2 cond\_at\_low\_eth -0.21 0.188 -0.54 0.11  
## 3 eth\_in\_high\_cond 0.08 0.073 -0.01 0.17  
## 4 eth\_in\_low\_cond -0.07 0.128 -0.16 0.02  
## 5 cond\_at\_high\_sdo 0.28 0.090 -0.05 0.61  
## 6 cond\_at\_low\_sdo -0.17 0.270 -0.49 0.14  
## 7 sdo\_in\_high\_cond 0.18 0.003 0.06 0.31  
## 8 sdo\_in\_low\_cond 0.00 0.966 -0.15 0.14

# Study 1 Additional Analyses

First, results for the SDO x condition only (without ethnic identification).

s1model\_supp1 <- lm(form~sdocent\*cond2, data=study1f) #creating model  
s1modeldf\_supp1 <- tidy(s1model\_supp1) #making a df of results  
s1cidf\_supp1 <- tidy(confint(s1model\_supp1)) #making a df of confidence intervals  
names(s1cidf\_supp1) <- c("parameter", "lowerci", "upperci")  
  
s1modeldf\_supp1 <- bind\_cols(s1modeldf\_supp1, s1cidf\_supp1) #binding those dfs  
names(s1modeldf\_supp1) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
  
describe(study1f$form)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 459 3.74 0.93 4 3.8 0 1 6 5 -0.81 1.5 0.04

describe(study2f$form)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 404 3.68 1.04 4 3.73 0.37 1 7 6 -0.37 1.73  
## se  
## X1 0.05

describe(study3f$form)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 368 3.58 1.02 4 3.62 0.74 1 7 6 -0.32 1.17  
## se  
## X1 0.05

describe(cdf$form)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 1627 3.67 1.03 4 3.72 0.37 1 7 6 -0.43 1.42  
## se  
## X1 0.03

summary(s1model\_supp1)

##   
## Call:  
## lm(formula = form ~ sdocent \* cond2, data = study1f)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.8277 -0.3546 0.2121 0.4165 2.1811   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.71786 0.06073 61.216 < 2e-16 \*\*\*  
## sdocent -0.07073 0.05357 -1.320 0.187   
## cond21 0.01071 0.08523 0.126 0.900   
## sdocent:cond21 0.30827 0.07336 4.202 3.18e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9108 on 455 degrees of freedom  
## Multiple R-squared: 0.05066, Adjusted R-squared: 0.0444   
## F-statistic: 8.093 on 3 and 455 DF, p-value: 2.915e-05

describe(study1f$cat)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 459 3.57 1.09 4 3.59 0.74 1 7 6 -0.23 0.73  
## se  
## X1 0.05

describe(study2f$cat)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 404 3.64 1.11 4 3.7 0 1 7 6 -0.47 1.33 0.06

describe(study3f$cat)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 368 3.56 1.16 4 3.56 0.74 1 7 6 -0.01 0.68  
## se  
## X1 0.06

describe(cdf$cat)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 1626 3.59 1.15 4 3.61 0.74 1 7 6 -0.22 0.77  
## se  
## X1 0.03

s1modeldf\_supp1 %>% #select variables, rounding, etc.   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need and reshape

## parameter B p lowerci upperci  
## 1 (Intercept) 3.72 0.000 3.60 3.84  
## 2 sdocent -0.07 0.187 -0.18 0.03  
## 3 cond21 0.01 0.900 -0.16 0.18  
## 4 sdocent:cond21 0.31 0.000 0.16 0.45

#now for simple effects  
s1simpslope\_supp1 <- simple\_slopes(s1model\_supp1)  
  
s1simpsloper\_supp1 <- s1simpslope\_supp1[c(3, 1, 5, 4), ]   
names(s1simpsloper\_supp1)[c(3, 4, 5, 7)] <- c("estimate", "std\_error", "t", "p")  
  
s1simpsloper\_supp1 %>% #adding confidence intervals  
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_sdo 0.37 0.002 0.13 0.61  
## 2 cond\_at\_low\_sdo -0.35 0.004 -0.59 -0.11  
## 3 sdo\_in\_high\_cond 0.24 0.000 0.14 0.34  
## 4 sdo\_in\_low\_cond -0.07 0.187 -0.18 0.04

And now looking at ethnic identification without the SDO control:

s1model\_supp1a <- lm(form~ethcent\*cond2, data=study1f) #creating model  
s1modeldf\_supp1a <- tidy(s1model\_supp1a) #making a df of results  
s1cidf\_supp1a <- tidy(confint(s1model\_supp1a)) #making a df of confidence intervals  
names(s1cidf\_supp1a) <- c("parameter", "lowerci", "upperci")  
  
s1modeldf\_supp1a <- bind\_cols(s1modeldf\_supp1a, s1cidf\_supp1a) #binding those dfs  
names(s1modeldf\_supp1a) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
s1modeldf\_supp1a %>% #select variables, rounding, etc.   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need and reshape

## parameter B p lowerci upperci  
## 1 (Intercept) 3.71 0.000 3.59 3.84  
## 2 ethcent -0.07 0.064 -0.13 0.00  
## 3 cond21 0.02 0.808 -0.15 0.19  
## 4 ethcent:cond21 0.15 0.004 0.05 0.25

#now for simple effects  
s1simpslope\_supp1a <- simple\_slopes(s1model\_supp1a)  
  
s1simpsloper\_supp1a <- s1simpslope\_supp1a[c(3, 1, 5, 4), ]   
names(s1simpsloper\_supp1a)[c(3, 4, 5, 7)] <- c("estimate", "std\_error", "t", "p")  
  
s1simpsloper\_supp1a %>% #adding confidence intervals  
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth","cond\_at\_low\_eth",  
 "eth\_in\_high\_cond","eth\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.27 0.027 0.03 0.52  
## 2 cond\_at\_low\_eth -0.23 0.061 -0.48 0.02  
## 3 eth\_in\_high\_cond 0.08 0.026 0.01 0.16  
## 4 eth\_in\_low\_cond -0.07 0.064 -0.14 0.01

The other additional analysis we’ll do with Study 1 is to look at categorization as a dependent variable.

s1model\_supp2 <- lm(cat~sdocent\*ethcent\*cond2, data=study1f)  
s1modeldf\_supp2 <- tidy(s1model\_supp2) #pull out results  
s1cidf\_supp2 <- tidy(confint(s1model\_supp2)) #pull out CIs  
names(s1cidf\_supp2) <- c("parameter", "lowerci", "upperci")  
summary(s1model\_supp2)

##   
## Call:  
## lm(formula = cat ~ sdocent \* ethcent \* cond2, data = study1f)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.7447 -0.6502 0.3262 0.4616 3.6034   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.53235 0.07638 46.246 <2e-16 \*\*\*  
## sdocent -0.06187 0.06819 -0.907 0.3647   
## ethcent -0.05358 0.04365 -1.227 0.2203   
## cond21 0.09731 0.10590 0.919 0.3587   
## sdocent:ethcent -0.06534 0.03599 -1.815 0.0701 .   
## sdocent:cond21 0.10335 0.09237 1.119 0.2638   
## ethcent:cond21 0.07874 0.06278 1.254 0.2104   
## sdocent:ethcent:cond21 0.04570 0.05048 0.905 0.3657   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.086 on 451 degrees of freedom  
## Multiple R-squared: 0.02117, Adjusted R-squared: 0.005974   
## F-statistic: 1.393 on 7 and 451 DF, p-value: 0.2061

s1modeldf\_supp2 <- bind\_cols(s1modeldf\_supp2, s1cidf\_supp2) #binding the dataframe  
names(s1modeldf\_supp2) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
s1modeldf\_supp2 %>% #rounding   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.53 0.000 3.38 3.68  
## 2 sdocent -0.06 0.365 -0.20 0.07  
## 3 ethcent -0.05 0.220 -0.14 0.03  
## 4 cond21 0.10 0.359 -0.11 0.31  
## 5 sdocent:ethcent -0.07 0.070 -0.14 0.01  
## 6 sdocent:cond21 0.10 0.264 -0.08 0.28  
## 7 ethcent:cond21 0.08 0.210 -0.04 0.20  
## 8 sdocent:ethcent:cond21 0.05 0.366 -0.05 0.14

#adding the simple slopes  
s1simpslope\_supp2 <- simple\_slopes(s1model\_supp2)  
  
s1simpsloper\_supp2 <- s1simpslope\_supp2[c(8, 2, 17, 11, 6, 4, 20, 14), ] #pulling out relevant rows  
names(s1simpsloper\_supp2)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p")  
  
s1simpsloper\_supp2 %>% #restucturing, rounding, etc.   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.23 0.123 -0.07 0.53  
## 2 cond\_at\_low\_eth -0.04 0.808 -0.34 0.27  
## 3 eth\_in\_high\_cond 0.03 0.577 -0.07 0.12  
## 4 eth\_in\_low\_cond -0.05 0.220 -0.14 0.03  
## 5 cond\_at\_high\_sdo 0.22 0.158 -0.09 0.53  
## 6 cond\_at\_low\_sdo -0.02 0.876 -0.32 0.27  
## 7 sdo\_in\_high\_cond 0.04 0.506 -0.08 0.17  
## 8 sdo\_in\_low\_cond -0.06 0.365 -0.20 0.07

# Study 2 Additional Analyses

First, results for the SDO x condition only (without ethnic identification):

s2model\_supp1 <- lm(form~sdocent\*cond2, data=study2f) #creating model  
s2modeldf\_supp1 <- tidy(s2model\_supp1) #making a df of results  
s2cidf\_supp1 <- tidy(confint(s2model\_supp1)) #making a df of confidence intervals  
names(s2cidf\_supp1) <- c("parameter", "lowerci", "upperci")  
  
s2modeldf\_supp1 <- bind\_cols(s2modeldf\_supp1, s2cidf\_supp1) #binding those dfs  
names(s2modeldf\_supp1) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
s2modeldf\_supp1 %>% #select variables, rounding, etc.   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need and reshape

## parameter B p lowerci upperci  
## 1 (Intercept) 3.49 0.000 3.34 3.63  
## 2 sdocent -0.17 0.003 -0.29 -0.06  
## 3 cond21 0.36 0.000 0.16 0.56  
## 4 sdocent:cond21 0.25 0.002 0.09 0.41

#now for simple effects  
s2simpslope\_supp1 <- simple\_slopes(s2model\_supp1)  
  
  
s2simpsloper\_supp1 <- s2simpslope\_supp1[c(3, 1, 5, 4), ]   
names(s2simpsloper\_supp1)[c(3, 4, 5, 7)] <- c("estimate", "std\_error", "t", "p")  
  
s2simpsloper\_supp1 %>% #adding confidence intervals  
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_sdo 0.68 0.000 0.39 0.97  
## 2 cond\_at\_low\_sdo 0.04 0.763 -0.24 0.33  
## 3 sdo\_in\_high\_cond 0.08 0.145 -0.03 0.19  
## 4 sdo\_in\_low\_cond -0.17 0.003 -0.29 -0.05

And now ethnic identification x condition on formativeness:

s2model\_supp1a <- lm(form~ethcent\*cond2, data=study2f) #creating model  
s2modeldf\_supp1a <- tidy(s2model\_supp1a) #making a df of results  
s2cidf\_supp1a <- tidy(confint(s2model\_supp1a)) #making a df of confidence intervals  
names(s2cidf\_supp1a) <- c("parameter", "lowerci", "upperci")  
  
  
s2modeldf\_supp1a <- bind\_cols(s2modeldf\_supp1a, s2cidf\_supp1a) #binding those dfs  
names(s2modeldf\_supp1a) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
s2modeldf\_supp1a %>% #select variables, rounding, etc.   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need and reshape

## parameter B p lowerci upperci  
## 1 (Intercept) 3.51 0.000 3.37 3.66  
## 2 ethcent -0.21 0.000 -0.30 -0.13  
## 3 cond21 0.34 0.001 0.15 0.54  
## 4 ethcent:cond21 0.28 0.000 0.16 0.39

#now for simple effects  
s2simpslope\_supp1a <- simple\_slopes(s2model\_supp1a)  
  
s2simpsloper\_supp1a <- s2simpslope\_supp1a[c(3, 1, 5, 4), ]   
names(s2simpsloper\_supp1a)[c(3, 4, 5, 7)] <- c("estimate", "std\_error", "t", "p")  
  
s2simpsloper\_supp1a %>% #adding confidence intervals  
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth","cond\_at\_low\_eth",  
 "eth\_in\_high\_cond","eth\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.82 0.000 0.54 1.10  
## 2 cond\_at\_low\_eth -0.14 0.333 -0.42 0.15  
## 3 eth\_in\_high\_cond 0.06 0.092 -0.01 0.14  
## 4 eth\_in\_low\_cond -0.21 0.000 -0.30 -0.13

And now we’ll look at categorization as the dependent variable in Study 2.

s2model\_supp2 <- lm(cat~sdocent\*ethcent\*cond2, data=study2f)  
s2modeldf\_supp2 <- tidy(s2model\_supp2) #pull out results  
s2cidf\_supp2 <- tidy(confint(s2model\_supp2)) #pull out CIs  
names(s2cidf\_supp2) <- c("parameter", "lowerci", "upperci")  
  
s2modeldf\_supp2 <- bind\_cols(s2modeldf\_supp2, s2cidf\_supp2) #binding the dataframe  
names(s2modeldf\_supp2) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
s2modeldf\_supp2 %>% #rounding   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.53 0.000 3.38 3.68  
## 2 sdocent -0.18 0.005 -0.30 -0.05  
## 3 ethcent -0.26 0.000 -0.35 -0.17  
## 4 cond21 0.24 0.023 0.03 0.45  
## 5 sdocent:ethcent -0.02 0.565 -0.09 0.05  
## 6 sdocent:cond21 0.18 0.040 0.01 0.35  
## 7 ethcent:cond21 0.27 0.000 0.14 0.39  
## 8 sdocent:ethcent:cond21 0.02 0.694 -0.07 0.10

#adding the simple slopes  
s2simpslope\_supp2 <- simple\_slopes(s2model\_supp2)  
  
s2simpsloper\_supp2 <- s2simpslope\_supp2[c(8, 2, 17, 11, 6, 4, 20, 14), ] #pulling out relevant rows  
names(s2simpsloper\_supp2)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p")  
  
s2simpsloper\_supp2 %>% #restucturing, rounding, etc.   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.70 0.000 0.40 1.01  
## 2 cond\_at\_low\_eth -0.22 0.152 -0.52 0.09  
## 3 eth\_in\_high\_cond 0.01 0.899 -0.08 0.09  
## 4 eth\_in\_low\_cond -0.26 0.000 -0.35 -0.17  
## 5 cond\_at\_high\_sdo 0.47 0.003 0.16 0.78  
## 6 cond\_at\_low\_sdo 0.02 0.901 -0.28 0.32  
## 7 sdo\_in\_high\_cond 0.00 0.979 -0.12 0.12  
## 8 sdo\_in\_low\_cond -0.18 0.005 -0.30 -0.05

# Study 3 Additional Analyses

Like above, we’ll start with the results for the SDO x condition only (without ethnic identification).

s3model\_supp1 <- lm(form~sdocent\*cond2, data=study3f) #creating model  
s3modeldf\_supp1 <- tidy(s3model\_supp1) #making a df of results  
s3cidf\_supp1 <- tidy(confint(s3model\_supp1)) #making a df of confidence intervals  
names(s3cidf\_supp1) <- c("parameter", "lowerci", "upperci")  
  
s3modeldf\_supp1 <- bind\_cols(s3modeldf\_supp1, s3cidf\_supp1) #binding those dfs  
names(s3modeldf\_supp1) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
s3modeldf\_supp1 %>% #select variables, rounding, etc.   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need and reshape

## parameter B p lowerci upperci  
## 1 (Intercept) 3.78 0.000 3.64 3.93  
## 2 sdocent -0.02 0.764 -0.13 0.09  
## 3 cond21 -0.40 0.000 -0.60 -0.19  
## 4 sdocent:cond21 0.14 0.070 -0.01 0.30

#now for simple effects  
s3simpslope\_supp1 <- simple\_slopes(s3model\_supp1)  
  
s3simpsloper\_supp1 <- s3simpslope\_supp1[c(3, 1, 5, 4), ]   
names(s3simpsloper\_supp1)[c(3, 4, 5, 7)] <- c("estimate", "std\_error", "t", "p")  
  
s3simpsloper\_supp1 %>% #adding confidence intervals  
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_sdo -0.21 0.160 -0.50 0.09  
## 2 cond\_at\_low\_sdo -0.59 0.000 -0.88 -0.29  
## 3 sdo\_in\_high\_cond 0.13 0.023 0.02 0.24  
## 4 sdo\_in\_low\_cond -0.02 0.764 -0.13 0.10

Now ethnic identification by condition interaction:

s3model\_supp1a <- lm(form~ethcent\*cond2, data=study3f) #creating model  
s3modeldf\_supp1a <- tidy(s3model\_supp1a) #making a df of results  
s3cidf\_supp1a <- tidy(confint(s3model\_supp1a)) #making a df of confidence intervals  
names(s3cidf\_supp1a) <- c("parameter", "lowerci", "upperci")  
  
s3modeldf\_supp1a <- bind\_cols(s3modeldf\_supp1a, s3cidf\_supp1a) #binding those dfs  
names(s3modeldf\_supp1a) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
s3modeldf\_supp1a %>% #select variables, rounding, etc.   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need and reshape

## parameter B p lowerci upperci  
## 1 (Intercept) 3.78 0.000 3.64 3.93  
## 2 ethcent 0.01 0.825 -0.07 0.09  
## 3 cond21 -0.41 0.000 -0.61 -0.20  
## 4 ethcent:cond21 -0.02 0.728 -0.14 0.10

#now for simple effects  
s3simpslope\_supp1a <- simple\_slopes(s3model\_supp1a)  
  
s3simpsloper\_supp1a <- s3simpslope\_supp1a[c(3, 1, 5, 4), ]   
names(s3simpsloper\_supp1a)[c(3, 4, 5, 7)] <- c("estimate", "std\_error", "t", "p")  
  
s3simpsloper\_supp1a %>% #adding confidence intervals  
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth","cond\_at\_low\_eth",  
 "eth\_in\_high\_cond","eth\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth -0.44 0.003 -0.74 -0.15  
## 2 cond\_at\_low\_eth -0.37 0.013 -0.67 -0.07  
## 3 eth\_in\_high\_cond -0.01 0.787 -0.10 0.07  
## 4 eth\_in\_low\_cond 0.01 0.825 -0.07 0.09

And then Study 3 with categorization as a dependent variable.

s3model\_supp2 <- lm(cat~sdocent\*ethcent\*cond2, data=study3f)  
s3modeldf\_supp2 <- tidy(s3model\_supp2) #pull out results  
s3cidf\_supp2 <- tidy(confint(s3model\_supp2)) #pull out CIs  
names(s3cidf\_supp2) <- c("parameter", "lowerci", "upperci")  
  
s3modeldf\_supp2 <- bind\_cols(s3modeldf\_supp2, s3cidf\_supp2) #binding the dataframe  
names(s3modeldf\_supp2) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
s3modeldf\_supp2 %>% #rounding   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.76 0.000 3.60 3.93  
## 2 sdocent -0.04 0.542 -0.17 0.09  
## 3 ethcent 0.05 0.253 -0.04 0.15  
## 4 cond21 -0.41 0.001 -0.65 -0.17  
## 5 sdocent:ethcent 0.02 0.613 -0.05 0.08  
## 6 sdocent:cond21 0.12 0.199 -0.06 0.31  
## 7 ethcent:cond21 -0.17 0.014 -0.31 -0.03  
## 8 sdocent:ethcent:cond21 -0.06 0.250 -0.15 0.04

#adding the simple slopes  
s3simpslope\_supp2 <- simple\_slopes(s3model\_supp2)  
  
s3simpsloper\_supp2 <- s3simpslope\_supp2[c(8, 2, 17, 11, 6, 4, 20, 14), ] #pulling out relevant rows  
names(s3simpsloper\_supp2)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p")  
  
s3simpsloper\_supp2 %>% #restucturing, rounding, etc.   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth -0.72 0.000 -1.07 -0.37  
## 2 cond\_at\_low\_eth -0.11 0.540 -0.46 0.24  
## 3 eth\_in\_high\_cond -0.12 0.021 -0.22 -0.02  
## 4 eth\_in\_low\_cond 0.05 0.253 -0.04 0.15  
## 5 cond\_at\_high\_sdo -0.25 0.157 -0.61 0.10  
## 6 cond\_at\_low\_sdo -0.57 0.001 -0.92 -0.23  
## 7 sdo\_in\_high\_cond 0.08 0.225 -0.05 0.21  
## 8 sdo\_in\_low\_cond -0.04 0.542 -0.18 0.09

# Combined Data Additional Analyses

First, results for the SDO x condition only (without ethnic identification).

cdmodel\_supp1 <- lm(form~sdocent\*cond2, data=cdff) #creating model  
cdmodeldf\_supp1 <- tidy(cdmodel\_supp1) #making a df of results  
cdcidf\_supp1 <- tidy(confint(cdmodel\_supp1)) #making a df of confidence intervals  
names(cdcidf\_supp1) <- c("parameter", "lowerci", "upperci")  
  
cdmodeldf\_supp1 <- bind\_cols(cdmodeldf\_supp1, cdcidf\_supp1) #binding those dfs  
names(cdmodeldf\_supp1) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
cdmodeldf\_supp1 %>% #select variables, rounding, etc.   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need and reshape

## parameter B p lowerci upperci  
## 1 (Intercept) 3.67 0.000 3.59 3.75  
## 2 sdocent -0.09 0.007 -0.15 -0.02  
## 3 cond21 0.01 0.898 -0.10 0.12  
## 4 sdocent:cond21 0.25 0.000 0.16 0.33

#now for simple effects  
cdsimpslope\_supp1 <- simple\_slopes(cdmodel\_supp1)  
  
cdsimpsloper\_supp1 <- cdsimpslope\_supp1[c(3, 1, 5, 4), ]   
names(cdsimpsloper\_supp1)[c(3, 4, 5, 7)] <- c("estimate", "std\_error", "t", "p")  
  
cdsimpsloper\_supp1 %>% #adding confidence intervals  
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_sdo 0.31 0.000 0.15 0.47  
## 2 cond\_at\_low\_sdo -0.30 0.000 -0.46 -0.14  
## 3 sdo\_in\_high\_cond 0.16 0.000 0.09 0.22  
## 4 sdo\_in\_low\_cond -0.09 0.007 -0.15 -0.02

And now we’ll get the results for ethnic identification interaction without SDO.

cdmodel\_supp1a <- lm(form~ethcent\*cond2, data=cdff) #creating model  
cdmodeldf\_supp1a <- tidy(cdmodel\_supp1a) #making a df of results  
cdcidf\_supp1a <- tidy(confint(cdmodel\_supp1a)) #making a df of confidence intervals  
names(cdcidf\_supp1a) <- c("parameter", "lowerci", "upperci")  
  
cdmodeldf\_supp1a <- bind\_cols(cdmodeldf\_supp1a, cdcidf\_supp1a) #binding those dfs  
names(cdmodeldf\_supp1a) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
cdmodeldf\_supp1a %>% #select variables, rounding, etc.   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need and reshape

## parameter B p lowerci upperci  
## 1 (Intercept) 3.67 0.000 3.59 3.75  
## 2 ethcent -0.08 0.000 -0.13 -0.04  
## 3 cond21 0.01 0.896 -0.10 0.12  
## 4 ethcent:cond21 0.13 0.000 0.07 0.19

#now for simple effecta  
cdsimpslope\_supp1a <- simple\_slopes(cdmodel\_supp1a)  
  
cdsimpsloper\_supp1a <- cdsimpslope\_supp1a[c(3, 1, 5, 4), ]   
names(cdsimpsloper\_supp1a)[c(3, 4, 5, 7)] <- c("estimate", "std\_error", "t", "p")  
  
cdsimpsloper\_supp1a %>% #adding confidence intervals  
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth","cond\_at\_low\_eth",  
 "eth\_in\_high\_cond","eth\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we need

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.23 0.004 0.07 0.39  
## 2 cond\_at\_low\_eth -0.22 0.006 -0.38 -0.06  
## 3 eth\_in\_high\_cond 0.05 0.044 0.00 0.09  
## 4 eth\_in\_low\_cond -0.08 0.000 -0.13 -0.04

The other additional analysis we’ll do with Study 3 is to look at categorization as a dependent variable.

cdmodel\_supp2 <- lm(cat~sdocent\*ethcent\*cond2, data=cdff)  
cdmodeldf\_supp2 <- tidy(cdmodel\_supp2) #pull out results  
cdcidf\_supp2 <- tidy(confint(cdmodel\_supp2)) #pull out CIs  
names(cdcidf\_supp2) <- c("parameter", "lowerci", "upperci")  
  
cdmodeldf\_supp2 <- bind\_cols(cdmodeldf\_supp2, cdcidf\_supp2) #binding the dataframe  
names(cdmodeldf\_supp2) <- c("term", "estimate", "std\_err", "stat", "p", "parameter", "lowerci", "upperci")  
  
cdmodeldf\_supp2 %>% #rounding   
 mutate(B = round(estimate, 2), p = round(p, 3), lowerci = round(lowerci, 2), upperci = round(upperci, 2)) %>%  
 select(parameter, B, p, lowerci, upperci) #select variables we need

## parameter B p lowerci upperci  
## 1 (Intercept) 3.60 0.000 3.51 3.69  
## 2 sdocent -0.08 0.036 -0.16 -0.01  
## 3 ethcent -0.07 0.010 -0.12 -0.02  
## 4 cond21 0.00 0.973 -0.13 0.13  
## 5 sdocent:ethcent -0.02 0.280 -0.06 0.02  
## 6 sdocent:cond21 0.13 0.015 0.02 0.23  
## 7 ethcent:cond21 0.04 0.254 -0.03 0.12  
## 8 sdocent:ethcent:cond21 0.01 0.800 -0.05 0.06

#adding the simple slopes  
cdsimpslope\_supp2 <- simple\_slopes(cdmodel\_supp2)  
  
cdsimpsloper\_supp2 <- cdsimpslope\_supp2[c(8, 2, 17, 11, 6, 4, 20, 14), ] #pulling out relevant rows  
names(cdsimpsloper\_supp2)[c(4, 5, 6, 8)] <- c("estimate", "std\_error", "t", "p")  
  
cdsimpsloper\_supp2 %>% #restucturing, rounding, etc.   
 mutate(lower95ci = round(estimate-(2\*std\_error), 2), upper95ci = round(estimate+(2\*std\_error), 2)) %>% #creating CI values  
 mutate(estimate = round(estimate, 2), p = round(p, 3)) %>% #rounding  
 mutate(parameter = c("cond\_at\_high\_eth", "cond\_at\_low\_eth", "eth\_in\_high\_cond",  
 "eth\_in\_low\_cond", "cond\_at\_high\_sdo","cond\_at\_low\_sdo",  
 "sdo\_in\_high\_cond","sdo\_in\_low\_cond")) %>% #making new parameter column  
 select(parameter, estimate, p, ends\_with("95ci")) #selecting only the variables we ne

## parameter estimate p lower95ci upper95ci  
## 1 cond\_at\_high\_eth 0.08 0.405 -0.11 0.26  
## 2 cond\_at\_low\_eth -0.07 0.433 -0.26 0.11  
## 3 eth\_in\_high\_cond -0.03 0.337 -0.08 0.03  
## 4 eth\_in\_low\_cond -0.07 0.010 -0.12 -0.02  
## 5 cond\_at\_high\_sdo 0.16 0.085 -0.03 0.35  
## 6 cond\_at\_low\_sdo -0.16 0.081 -0.34 0.02  
## 7 sdo\_in\_high\_cond 0.05 0.186 -0.02 0.12  
## 8 sdo\_in\_low\_cond -0.08 0.036 -0.16 0.00

And that’s it!