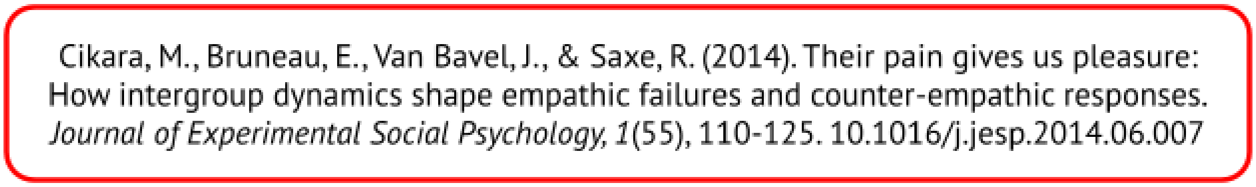
ESTIMATING THE RELIABILITY & ROBUSTNESS OF RESEARCH

# ERROR REVIEW



Cikara, M., Bruneau, E., Van Bavel, J., & Saxe, R. (2014). Their pain gives us pleasure: How intergroup dynamics shape empathic failures and counter-empathic responses. *Journal of Experimental Social Psychology, 1*(55), 110-125. 10.1016/j.jesp.2014.06.007

*reviewed by*

**Øystein Sørensen**, University of Oslo

Jul 2, 2025

*For the sections below, indicate whether you discovered any errors using the dropdown menu. Describe the* ***errors*** *you discovered, the* ***methods*** *that you used to find them, and the* ***amount of time*** *you invested in the search. Refer to specific files to allow verification of your review. For the assessment below, make sure to check if authors have provided* ***supplementary analyses*** *as these may clear concerns arising from the (interpretation of) the primary analyses. If you have written code yourself for the review, please attach it to the report. Please indicate the version of software/packages you used to run the original code and/or your own code.*

## I. METHODS, MEASUREMENT, AND DESIGN

### 1. Design No errors found ▾

Are there errors in the conceptual design of the study? E.g., flawed randomisation technique

*Time spent: 30 minutes*

The study design seems well equipped to test the research hypotheses. In the experiments, participants were randomly assigned to different exposures, and while the particular randomization algorithm was not stated, it is clear from the context that they used a uniform discrete distribution.

One point is unclear to me, but is likely not of relevance to the correctness of the study. In the first paragraph of page 114 they write (my underlinings):

*"On the following page we had the participants indicate the strength of their agreement with a series of five personality items (taken from the NEO PI-R, Costa & McCrae, 1992), ostensibly for the purposes of assigning them to a team. After the participants completed the items, we randomly assigned the participants to a team: the Eagles or the Rattlers."*

Then in the next paragraph they write:

*"Note that assigning the participants to teams based on their personalities and imposing competitive and cooperative functional relations no longer makes ours a minimal groups study."*

The quoted sentence above makes it seem as if they assigned participants based on personalities, but the previous sentences say the opposite. I do still agree that this is not a minimal groups study, but isn't this only because of "*imposing competitive and cooperative functional relations*"?

### 2. Measurement No errors found ▾

Are there any measures, techniques, or devices that were incorrectly applied or inappropriate for the specific task described in the paper?

*Time spent: 10 minutes*

It seems like everything was correctly applied.

### 3. Preregistration Consistency Not applicable ▾

Are there substantial deviations from the preregistration, particularly undisclosed ones?

*Time spent: 0 minutes*

No preregistration reported.

### 4. Sampling No errors found ▾

Is there an error in the sampling strategy? Is the power analysis reproducible? Does the model used for the power analysis match the model in the substantive analyses? Were separate power analyses conducted for all primary analyses?

*Time spent: 20 minutes*

The sampling strategy seems fine. They used Amazon Mechanical Turk to sample participants based in the US between 18 and 60 years of age.

No power analyses were reported. It is unclear why the authors chose exactly the number of participants they did. However, while a power analysis is important for estimating the number of samples required to have a good chance of finding an effect of a hypothesized magnitude, and hence to avoid wasting resources by conducting an overpowered or underpowered study, the lack of a power analysis does not have any direct implications on the correctness of the conclusions reached for a given sample. I hence do not consider this as an error.

It should be noted that the authors did a pilot study to validate that the negative and positive scenarios were properly separated in the absence of teams or competition. However, it is not mentioned anywhere whether the estimates obtained from the pilot were used for estimating power.

### 5. Other Aspects Related to Methods and Measures Not applicable

*Time spent: 0 minutes*

## II. DATA, CODE, AND STATISTICAL ANALYSES

### 1. Code Functionality Errors found

Does the provided code run without the need to make any adjustments and without errors? If not, what steps were needed to get it to run (if it was eventually possible)?

*Time spent: 14 hours*

For this and all remaining points I used SAS OnDemand for Academics which runs SAS version 9.4. It is freely available for academics at https://welcome.oda.sas.com/. All scripts can also be run locally for anyone who has SAS installed. I used R version 4.5.0 on a MacBook Pro (M1) for other calculations.

I received code and data from the authors in directories named study1, study2, study3a, study3b, and study4. This corresponds to Experiments 1, 2, 3a, 3b, and 4 in the paper. Each of these directories contained one .sas file with code and one .csv file with data. The directories also contained some SAS output, but these are not directly relevant for my review, since my job is to run the code to reproduce the results. I had to make many changes to the code to make it run, and the code is attached to this review with the same directory structure. My scripts are named “study1_revised.sas”, “study2_revised.sas”, and so on, and are provided alongside the original scripts and the provided data.

Due to how SAS works, the paths to the data have to be hard-coded, so anyone who wants to rerun the scripts on their own platform needs to change the paths at the top of each script. After making this edit, all my scripts should run without any error.

To answer the first question first: No, the code does not run without making adjustments. To describe the changes I made, I structure this part per experiment, starting with experiment 1.

Experiment 1

The code provided from the authors for experiment 1 can be found in "study1/study1_syntax_effectsize.sas". My revised code can be found in "study1/study1_revised.sas".

The first line of the provided code reads

```
libname study1
'\\andrew\users\users21\mcikara\Desktop\Arbgroups\study1'
```

This does not work in the provided setup, even if I change the path to point to the appropriate folder. The reason is that the libname command requires that a data file with the extension .sas7bdat is available in the provided directory. It should also be noted that SAS does not have a concept of "current working directory", and hence the fact that the path was hard coded was not an error in and off itself.

In order to import the data, I had to change it to the code block shown below. The first part of the path, /home/u64240311/ERROR/, is specific to my setup. Note that I had to manually add the non-default argument guessingrows=MAX to the import step, because without it all string variables become cut after a small number of letters and, e.g., "cooperate" become "coopera" in my dataset and the code for the mixed model would fail.

```
libname study1 "/home/u64240311/ERROR/study1";
proc import
datafile="/home/u64240311/ERROR/study1/ArbitraryGroups_STUDY1_SA
S.csv"
   out=study1.final
   dbms=csv
   replace;
   guessingrows=MAX;
   getnames=yes;
run;
```

Next, I had to edit illegally formatted SAS comments by adding semicolons at the end. In particular I had to change from

```
**THIS IS THE FULL MODEL
```

To

```
**THIS IS THE FULL MODEL;
```

and similarly for the following comments, for which I had to add semicolons at the end, regardless of the asterisks.

```
**THIS WAS HOW WE GENERATED LSMEANS COMPARISONS
***THE REST IS FOR COMPUTING EFFECT SIZE
***COMPUTING F^2 FOR MAIN EFFECT OF COND2*****
***COMPUTING F^2 FOR MAIN EFFECT OF STORY*****
***COMPUTING F^2 FOR MAIN EFFECT OF GROUP*****
***COMPUTING F^2 FOR MAIN EFFECT OF RESPTYPE*****
***COMPUTING F^2 FOR CONDXSTORY*****
***COMPUTING F^2 FOR CONDXGROUP*****
***COMPUTING F^2 FOR CONDXRESPTYPE*****
***COMPUTING F^2 FOR STORYXGROUP*****
***COMPUTING F^2 FOR STORYXRESPTYPE*****
***COMPUTING F^2 FOR GROUPXRESPTYPE*****
***COMPUTING F^2 FOR CONDXSTORYXGROUP*****
***COMPUTING F^2 FOR STORYXGROUPXRESPTYPE*****
***COMPUTING F^2 FOR CONDXSTORYXGROUPXRESPTYPE*****
```

After making these adjustments, fitting of the mixed model for experiment 1 did work.

Generating the least squares means comparisons did however not work. The provided code line was the following:

```
**THIS WAS HOW WE GENERATED LSMEANS COMPARISONS
```

```
lsmeans cond2*story1*group1*resptype / slice =
cond2*story1*resptype diff = all;
```

The lsmeans chunk above is not a SAS command, but instead an argument which has to be provided to proc mixed. To generate the least squares means I hence had to modify the proc mixed call. It originally was this:

```
ods output CovParms = Study1_full;
proc mixed data=study1.final plot(maxpoints=6500)=studentpanel;
class pid cond2 story1 group1 storynum resptype;
model rating = cond2 story1 group1 resptype cond2*story1
cond2*group1 cond2*resptype story1*group1 story1*resptype
group1*resptype cond2*story1*group1 cond2*story1*resptype
cond2*group1*resptype story1*group1*resptype
cond2*story1*group1*resptype / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
run;
quit;
ods output close;
```

I changed it by adding the highlighted line:

```
ods output CovParms = Study1_full;
proc mixed data=study1.final plot(maxpoints=6500)=studentpanel;
class pid cond2 story1 group1 storynum resptype;
model rating = cond2 story1 group1 resptype cond2*story1
cond2*group1 cond2*resptype story1*group1 story1*resptype
group1*resptype cond2*story1*group1 cond2*story1*resptype
cond2*group1*resptype story1*group1*resptype
cond2*story1*group1*resptype / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
lsmeans cond2*story1*group1*resptype / slice =
cond2*story1*resptype diff = all;
run;
quit;
ods output close;
```

Next, in the code used to compute effect sizes, the following block failed:

```
ods output CovParms = Study1_woResptype;
proc mixed data=study1.final plot(maxpoints=6500)=studentpanel;
class pid cond2 story1 group1 storynum ;
model rating = cond2 story1 group1 / solution;
repeated storynum/ subject=pid type=un@cs /*type=un@un*/;
parms /parmsdata=Study1_full hold=2,4;
run;
quit;
ods output close;
```

The error message is given below:

```
ERROR: Two REPEATED effects are required with this covariance
structure.
```

The following fix solves the problem. I did the following:

1. Add resptype to class.
2. Add resptype to repeated.
3. Kept resptype out of model, since we want to estimate a model without resptype.

This tells proc mixed that resptype is part of the repeated measures structure and hence retains correlated residuals for the individuals, but keeps it out of the fixed effects. The corrected code is provided below:

```
ods output CovParms = Study1_woResptype;
proc mixed data=study1.final plot(maxpoints=6500)=studentpanel;
class pid cond2 story1 group1 storynum resptype;
model rating = cond2 story1 group1 / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study1_full hold=2,4;
run;
quit;
ods output close;
```

Further on in the script, in preparation to computing effect sizes, there are DATA statements which remove all rows from the data, and hence make it impossible to actually compute the effect sizes. I'll illustrate the problem for the dataset Study1_full. The same applies to Study1_null and Study1_wo4way.

Consider the following two statements:

```
DATA Study1_full;
set Study1_full;
if CovParm = "resptype UN(1,1)";
run;

DATA Study1_full;
set Study1_full;
if CovParm = "UN(2,2)";
run;
```

The first DATA step filters Study1_full to only contain rows where CovParm equals "resptype UN(1,1)". It hence reduces from 4 to 1 row. Next, the exact same dataset is filtered again, to keep only rows where CovParm equals "UN(2,2)", but these values were removed in the previous step, so we end up with a dataset that contains 0 rows. On top of that, the test CovParm = "UN(2,2)" would return 0 regardless, because SAS puts lots of whitespace in front of the string UN(2,2). I fixed this by replacing the two statements above with the single statement below. Note how I use the strip function to remove any whitespace.

```
DATA Study1_full;
set Study1_full;
if strip(CovParm) = "UN(2,2)" or strip(CovParm) = "resptype
UN(1,1)";
run;
```

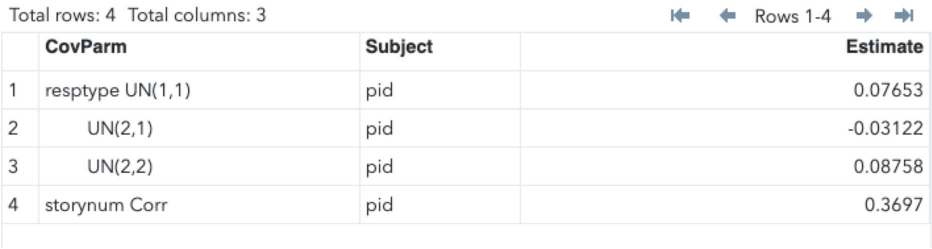
This fix lets us obtain effect size for both the *howgood* and *howbad* condition in a single step.

Next, the script contains lots of code for fitting different models with fixed effects somewhere between the full model and the null model. There are 15 statements (for 15 different models) of the form

```
DATA Study1_woCond;
set Study1_woCond;
```

```
if CovParm = "pid";
run;
```

All of these statements yield 0 rows, because the variable CovParm does not have the value “pid”. The screenshot below shows one such table:



Total rows: 4  Total columns: 3  Rows 1-4

|   | CovParm | Subject | Estimate |
|---|---|---|---:|
| 1 | resptype UN(1,1) | pid | 0.07653 |
| 2 | UN(2,1) | pid | -0.03122 |
| 3 | UN(2,2) | pid | 0.08758 |
| 4 | storynum Corr | pid | 0.3697 |

Presumably, the following would be a correct statement.

```
DATA Study1_woCond;
set Study1_woCond;
if Subject = "pid";
run;
```

However, since Subject = “pid” is true for all rows in all the datasets considered, all these DATA steps should rather be removed, and this is what I ended up doing.

Next, all the DATA steps in which the effect size F^2 is computed give me an error message that the data are not sorted. For example this one:

```
DATA Study1_woCond_op;
merge Study1_full (rename=(Estimate=full)) Study1_Null (rename
=(Estimate=null)) Study1_woCond (rename=(Estimate=woCond)); by
CovParm;
DROP CovParm Subject;
run;
```

```
ERROR: BY variables are not properly sorted on data set
WORK.STUDY1_WOCOND.
```

I had to add proc sort in front for this to work. Again, this had to be added in front of 15 separate data steps, one for each model.

```
proc sort data=Study1_full; by CovParm; run;
proc sort data=Study1_Null; by CovParm; run;
proc sort data=Study1_woCond; by CovParm; run;
DATA Study1_woCond_op;
merge Study1_full (rename=(Estimate=full)) Study1_Null (rename
=(Estimate=null)) Study1_woCond (rename=(Estimate=woCond)); by
CovParm;
DROP CovParm Subject;
run;
```

After this adjustment, computation of effect sizes worked well until the following statement:

```
***COMPUTING F^2 FOR STORYXRESPTYPE;
DATA Study1_woStoryXResptype_op;
merge Study1_full (rename=(Estimate=full)) Study1_Null (rename
=(Estimate=null)) Study1_woStoryXResptype
(rename=(Estimate=woStoryXResptype)); by CovParm;
DROP CovParm Subject;
run;
```

```
ERROR: File WORK.STUDY1_WOSTORYXRESPTYPE.DATA does not exist.
```

Going back up to the fitting of reduced models, it turns out that there was inconsistent naming, since the “wo” part was missing from the model:

The original code is here:

```
ods output CovParms = Study1_StoryXResptype;
proc mixed data=study1.final plot(maxpoints=6500)=studentpanel;
class pid cond2 story1 group1 storynum resptype;
model rating = cond2 story1 group1 resptype cond2*story1
cond2*group1 cond2*resptype story1*group1 group1*resptype /
solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study1_full hold=2,4;
run;
quit;
```

```
ods output close;
```

I had to change it to:

```
ods output CovParms = Study1_woStoryXResptype;
proc mixed data=study1.final plot(maxpoints=6500)=studentpanel;
class pid cond2 story1 group1 storynum resptype;
model rating = cond2 story1 group1 resptype cond2*story1
cond2*group1 cond2*resptype story1*group1 group1*resptype /
solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study1_full hold=2,4;
run;
quit;
ods output close;
```

Next, in the following chunk the variable name for the output dataset is longer than what SAS allows:

```
DATA resultsStudy1_woCondXStoryXResptype; set
Study1_woCondXStoryXResptype_op;
DROP full null woCondXStoryXResptype;
R2full = (null - full)/null;
R2woCondXStoryXResptype = (null - woCondXStoryXResptype)/null;
f2 = (R2full - R2woCondXStoryXResptype)/(1 - R2full);
run;
```

```
ERROR 307-185: The data set name cannot have more than 32 bytes.
```

The changes I made to make this run are shown below:

```
DATA resultsStudy1_woCXSXR; set Study1_woCondXStoryXResptype_op;
DROP full null woCondXStoryXResptype;
R2full = (null - full)/null;
R2woCondXStoryXResptype = (null - woCondXStoryXResptype)/null;
f2 = (R2full - R2woCondXStoryXResptype)/(1 - R2full);
run;
PROC PRINT data=resultsStudy1_woCXSXR;
```

```
run;
```

The same happened here, where I simply show the code before and after.

Before:

```
DATA resultsStudy1_woStoryXGroupXResptype; set
Study1_woStoryXGroupXResptype_op;
DROP full null woStoryXGroupXResptype;
R2full = (null - full)/null;
R2woStoryXGroupXResptype = (null - woStoryXGroupXResptype)/null;
f2 = (R2full - R2woStoryXGroupXResptype)/(1 - R2full);
run;

PROC PRINT data=resultsStudy1_woStoryXGroupXResptype;
run;
```

After:

```
DATA resultsStudy1_woSXGXR; set
Study1_woStoryXGroupXResptype_op;
DROP full null woStoryXGroupXResptype;
R2full = (null - full)/null;
R2woStoryXGroupXResptype = (null - woStoryXGroupXResptype)/null;
f2 = (R2full - R2woStoryXGroupXResptype)/(1 - R2full);
run;

PROC PRINT data=resultsStudy1_woSXGXR;
run;
```

Going back to the DATA statements for computing F^2, since I include both the *howbad* condition and the *howgood* condition, we should not drop CovParm in all the DATA steps similar to the one below, because then we don't know which row corresponds to which condition.

The problematic argument is highlighted below:

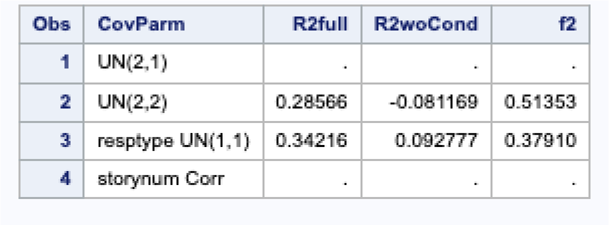
```
DATA Study1_woCond_op;
```

```
merge Study1_full (rename=(Estimate=full)) Study1_Null (rename
=(Estimate=null)) Study1_woCond (rename=(Estimate=woCond)); by
CovParm;
DROP CovParm Subject;
run;
```

I removed the CovParm term in the revised code::

```
DATA Study1_woCond_op;
merge Study1_full (rename=(Estimate=full)) Study1_Null (rename
=(Estimate=null)) Study1_woCond (rename=(Estimate=woCond)); by
CovParm;
DROP Subject;
run;
```

The resulting tables with effect estimates now get the structure shown in the example screenshot below: Since we keep CovParm, we know which effect size corresponds to which condition.



| Obs | CovParm | R2full | R2woCond | f2 |
|---|---|---|---|---|
| 1 | UN(2,1) | . | . | . |
| 2 | UN(2,2) | 0.28566 | -0.081169 | 0.51353 |
| 3 | resptype UN(1,1) | 0.34216 | 0.092777 | 0.37910 |
| 4 | storynum Corr | . | . | . |

After these changes, my full revised script runs in one go and reproduces the results, with the exception of Cohen’s d, as mentioned elsewhere.

In the log, there are no errors or warnings.

Experiment 2

The original code for experiment 2 was provided in “study2/study2_syntax_effectsize.sas” and my revised code is in “study2/study2_revised.sas”. To avoid duplicating too many explanations, I refer to the section on Experiment 1 for details whenever exactly the same issue appears.

First, there were 9 comments in the script not obeying SAS' rules for how to write comments, as semicolons were missing exactly as described for experiment 1. I changed all of these.

Next, the csv data were not imported. The code started with the following line:

```
libname study2
'\\andrew\users\users21\mcikara\Desktop\Arbgroups\study2';
```

I had to change it to the following to be able to load the data:

```
libname study2 "/home/u64240311/ERROR/study2";
proc import
datafile="/home/u64240311/ERROR/study2/ArbitraryGroups_STUDY2_SA
S.csv"
   out=study2.final
   dbms=csv
   replace;
   guessingrows=MAX;
   getnames=yes;
run;
```

Next, in fitting the full model, the model expected two variables named story1 and group1, which were not part of the dataset study2.final.

```
ods output CovParms = Study2_full;
proc mixed data=study2.final plot(maxpoints=6500)=studentpanel;
class pid story1 group1 storynum resptype;
model rating = story1 group1 resptype story1*group1
story1*resptype group1*resptype story1*group1*resptype /
solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
run;
quit;
ods output close;
```

```
ERROR: Variable STORY1 not found.
ERROR: Variable GROUP1 not found.
```

I changed it to story and group – which were part of the data – and it seemed to work:

```
ods output CovParms = Study2_full;
proc mixed data=study2.final plot(maxpoints=6500)=studentpanel;
class pid story group storynum resptype;
model rating = story group resptype story*group story*resptype
group*resptype story*group*resptype / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
run;
quit;
ods output close;
```

After this change, I was able to reproduce all results in Table 2. However, the PROC MIXED statement above does not include a command for computing marginal means, which are reported in the right column on page 116. The command for doing this is not shown anywhere else in the attached code either. I added the marginal means computation with the line highlighted below:

```
ods output CovParms = Study2_full;
proc mixed data=study2.final plot(maxpoints=6500)=studentpanel;
class pid story group storynum resptype;
model rating = story group resptype story*group story*resptype
group*resptype story*group*resptype / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
lsmeans story*group*resptype / slice = story*resptype diff =
all;
run;
quit;
ods output close;
```

After making this fix, I was able to reproduce the results for least squares means reported in the right column on page 116. However, since my encodings of the variables story and group likely had different string values than the original study, the direction of the effect sizes were flipped. This is simply due to SAS handling factor variables in alphabetical order, and does not have any impact for the substantial interpretation of the results.

Moving on, all subsequent smaller models – used to compute effect sizes – also had group1 and story1 in them. I hence did a find-and-replace for the whole script to update this syntax.

Then the following model gave an error:

```
ods output CovParms = Study2_woResptype;
proc mixed data=study2.final plot(maxpoints=6500)=studentpanel;
class pid  story group storynum ;
model rating =  story group / solution;
repeated storynum/ subject=pid type=un@cs /*type=un@un*/;
parms /parmsdata=Study2_full hold=2,4;
run;
quit;
ods output close;
```

```
ERROR: Two REPEATED effects are required with this covariance
structure.
```

The fix has been described previously, and is shown here:

```
ods output CovParms = Study2_woResptype;
proc mixed data=study2.final plot(maxpoints=6500)=studentpanel;
class pid  story group storynum resptype;
model rating =  story group / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study2_full hold=2,4;
run;
quit;
ods output close;
```

Next, I encountered the same problem as in experiment 1, namely that datasets were filtered so they ended up with zero rows. This applied to Study2_full, Study2_Null, and Study2_wo3way. The change for the first of them is shown below.

This is the original code whcih gives 0 rows:

```
DATA Study2_full;
```

```
set Study2_full;
if CovParm = "resptype UN(1,1)";
run;
DATA Study2_full;
set Study2_full;
if CovParm = "UN(2,2)";
run;
```

This is my fix, which gives the rows we need:

```
DATA Study2_full;
set Study2_full;
if strip(CovParm) = "resptype UN(1,1)" or strip(CovParm) =
"UN(2,2)";
run;
```

Next, in 6 DATA steps we had the same problem as previously, that datasets were filtered against a value of CovParm that did not exist, and hence yielded 0 rows. One example is shown below. Here there were also some issues with the “?” symbol, but that could have been easily fixed.

```
DATA Study2_woStory;
set Study2_woStory;
if CovParm = "�pid"�;
run;
```

Again, the solution was to remove these six DATA steps.

Next, in all DATA steps where effect sizes were computed, I got an error because data were not sorted. For the same reason as in Experiment 1, I also removed CovParm from DROP. Below is one example:

```
DATA Study2_woStory_op;
merge Study2_full (rename=(Estimate=full)) Study2_Null (rename
=(Estimate=null)) Study2_woStory (rename=(Estimate=woStory)); by
CovParm;
DROP CovParm Subject;
run;
```

```
ERROR: BY variables are not properly sorted on data set
WORK.STUDY2_FULL.
```

I hence added sort statements in front of each of these statements, to make sure it worked.

```
proc sort data=Study2_full; by CovParm; run;
proc sort data=Study2_null; by CovParm; run;
proc sort data=Study2_woStory; by CovParm; run;
DATA Study2_woStory_op;
merge Study2_full (rename=(Estimate=full)) Study2_Null (rename
=(Estimate=null)) Study2_woStory (rename=(Estimate=woStory)); by
CovParm;
DROP Subject;
run;
```

Next, in this block I get an error message:

```
DATA Study2_woStoryXResptype_op;
merge Study2_full (rename=(Estimate=full)) Study2_Null (rename
=(Estimate=null)) Study2_woStoryXResptype
(rename=(Estimate=woStoryXResptype)); by CovParm;
DROP Subject;
run;
```

```
ERROR: File WORK.STUDY2_WOSTORYXRESPTYPE.DATA does not exist.
```

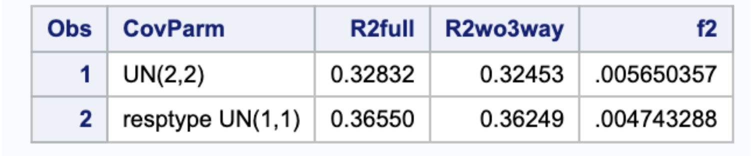
Going back up, this turns out to be because the dataset with model output is improperly named:

```
ods output CovParms = Study2_StoryXResptype;
proc mixed data=study2.final plot(maxpoints=6500)=studentpanel;
class pid  story group storynum resptype;
model rating =  story group resptype story*group group*resptype
/ solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study2_full hold=2,4;
run;
quit;
ods output close;
```

The following change fixed it:

```
ods output CovParms = Study2_woStoryXResptype;
proc mixed data=study2.final plot(maxpoints=6500)=studentpanel;
class pid  story group storynum resptype;
model rating =  story group resptype story*group group*resptype
/ solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study2_full hold=2,4;
run;
quit;
ods output close;
```

After making these changes, I was able to reproduce the effect sizes reported near the top of the right column on page 116. The screenshot below shows this, where UN(1,1) is *howbad* and UN(2,2) is *howgood*.



| Obs | CovParm | R2full | R2wo3way | f2 |
|---|---|---|---|---|
| 1 | UN(2,2) | 0.32832 | 0.32453 | .005650357 |
| 2 | resptype UN(1,1) | 0.36550 | 0.36249 | .004743288 |

Again calculation of Cohen's d reported in the text do not seem to be part of the code.

When running the whole script, the log shows no errors or warnings.

Experiment 3a

The original code can be found in "study3a/study3a_syntax_effectsize.sas" and my revised code in "study3a/study3a_revised.sas". To avoid duplicating too many explanations, I refer to the sections on Experiment 1 and 2 for details whenever exactly the same issue appears.

First, throughout the script the variables have the suffix "4a". I changed this to "3a" to make the variable naming consistent, given that this is experiment 3a.

Next, there were again code comments which were not properly formatted as SAS comments, and I corrected this by adding semicolons.

The csv data were not imported anywhere, so I had to change from

```
libname study3a
'\\andrew\users\users21\mcikara\Desktop\Arbgroups\study3a';
```

to

```
libname study3a "/home/u64240311/ERROR/study3a";
proc import
datafile="/home/u64240311/ERROR/study3a/ArbitraryGroups_STUDY3a_
SAS.csv"
   out=study3a.final
   dbms=csv
   replace;
   guessingrows=MAX;
   getnames=yes;
run;
```

Fitting the full model then did not work, due to wrong variable names:

```
ods output CovParms = Study3a_full;
proc mixed data=study3a.final plot(maxpoints=6500)=studentpanel;
class pid bef_aft story1 group1 storynum resptype;
model rating = bef_aft story1 group1 resptype bef_aft*story1
bef_aft*group1 bef_aft*resptype story1*group1 story1*resptype
group1*resptype bef_aft*story1*group1 bef_aft*story1*resptype
bef_aft*group1*resptype story1*group1*resptype
bef_aft*story1*group1*resptype / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
run;
quit;
ods output close;
```

```
ERROR: Variable STORY1 not found.
ERROR: Variable GROUP1 not found.
```

I changed it to this:

```
ods output CovParms = Study3a_full;
proc mixed data=study3a.final plot(maxpoints=6500)=studentpanel;
class pid bef_aft story group storynum resptype;
model rating = bef_aft story group resptype bef_aft*story
bef_aft*group bef_aft*resptype story*group story*resptype
group*resptype bef_aft*story*group bef_aft*story*resptype
bef_aft*group*resptype story*group*resptype
bef_aft*story*group*resptype / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
run;
quit;
ods output close;
```

I also did a find-and-replace in the whole script, replacing group1 with group and story1 with story, as these variable names were inconsistent throughout

The model did not contain a statement for computing least squares means. Since these effects are reported in Table 3, I added the highlighted line below to achieve this:

```
ods output CovParms = Study3a_full;
proc mixed data=study3a.final plot(maxpoints=6500)=studentpanel;
class pid bef_aft story group storynum resptype;
model rating = bef_aft story group resptype bef_aft*story
bef_aft*group bef_aft*resptype story*group story*resptype
group*resptype bef_aft*story*group bef_aft*story*resptype
bef_aft*group*resptype story*group*resptype
bef_aft*story*group*resptype / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
lsmeans bef_aft*story*group*resptype / slice =
bef_aft*story*resptype diff = all;
run;
quit;
ods output close;
```

After adding this, I was able to reproduce the results reported in the left part of Table 3, except for Cohen's d, which as mentioned previously is never computed in the provided code.

Next, I got an error at this point:

```
ods output CovParms = Study3a_woResptype;
proc mixed data=study3a.final plot(maxpoints=6500)=studentpanel;
class pid bef_aft story group storynum ;
model rating = bef_aft story group / solution;
repeated storynum/ subject=pid type=un@cs /*type=un@un*/;
parms /parmsdata=Study3a_full hold=2,4;
run;
quit;
ods output close;
```

```
ERROR: Two REPEATED effects are required with this covariance
structure.
```

The necessary change is shown below, i.e., resptype need to be added to class and repeated, even though it's not in model.

```
ods output CovParms = Study3a_woResptype;
proc mixed data=study3a.final plot(maxpoints=6500)=studentpanel;
class pid bef_aft story group storynum resptype;
model rating = bef_aft story group / solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study3a_full hold=2,4;
run;
quit;
ods output close;
```

Also as previously, for the subsequent code to run a change of variable names had to be performed, from the original:

```
ods output CovParms = Study3a_StoryXResptype;
proc mixed data=study3a.final plot(maxpoints=6500)=studentpanel;
class pid bef_aft story group storynum resptype;
```

```
model rating = bef_aft story group resptype bef_aft*story
bef_aft*group bef_aft*resptype story*group group*resptype /
solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study3a_full hold=2,4;
run;
quit;
ods output close;
```

To this:

```
ods output CovParms = Study3a_woStoryXResptype;
proc mixed data=study3a.final plot(maxpoints=6500)=studentpanel;
class pid bef_aft story group storynum resptype;
model rating = bef_aft story group resptype bef_aft*story
bef_aft*group bef_aft*resptype story*group group*resptype /
solution;
repeated resptype storynum/ subject=pid type=un@cs
/*type=un@un*/;
parms /parmsdata=Study3a_full hold=2,4;
run;
quit;
ods output close;
```

Next, data statements were only set up to compute the effect sizes for the *howgood* condition:

```
DATA Study3a_full;
set Study3a_full;
if CovParm = "resptype UN(1,1)";
run;
DATA Study3a_Null;
set Study3a_Null;
if CovParm = "resptype UN(1,1)";
run;
```

I made the change described previously:

```
DATA Study3a_full;
```

```
set Study3a_full;
if strip(CovParm) = "resptype UN(1,1)" or strip(CovParm) =
"UN(2,2)";
run;
DATA Study3a_Null;
set Study3a_Null;
if strip(CovParm) = "resptype UN(1,1)" or strip(CovParm) =
"UN(2,2)";
run;
```

Next there were a large number of statements removing all rows in the data, of the type

```
DATA Study3a_woBA;
set Study3a_woBA;
if CovParm = "�pid"�;
run;
```

These were all removed.

I next had to remove CovParm from the DROP argument in all DATA steps like the example below:

```
DATA Study3a_woStory_op;
merge Study3a_full (rename=(Estimate=full)) Study3a_Null (rename
=(Estimate=null)) Study3a_woStory (rename=(Estimate=woStory));
by CovParm;
DROP CovParm Subject;
run;
```

Next, when computing effect sizes I got an error message that by-variables were not properly sorted. Here is one example:

```
DATA Study3a_woBA_op;
merge Study3a_full (rename=(Estimate=full)) Study3a_Null (rename
=(Estimate=null)) Study3a_woBA (rename=(Estimate=woBA)); by
CovParm;
DROP Subject;
Run;
```

```
ERROR: BY variables are not properly sorted on data set
WORK.STUDY3A_FULL.
```

Again, the fix is to add sort statements in front of the merge:

```
proc sort data=Study3a_full; by CovParm; run;
proc sort data=Study3a_null; by CovParm; run;
proc sort data=Study3a_woBA; by CovParm; run;
DATA Study3a_woBA_op;
merge Study3a_full (rename=(Estimate=full)) Study3a_Null (rename
=(Estimate=null)) Study3a_woBA (rename=(Estimate=woBA)); by
CovParm;
DROP Subject;
run;
```

This sort statement was added in front of all DATA steps.

Next, I got an error message about a too long variable name:

```
DATA resultsStudy3a_woBAXStoryXResptype; set
Study3a_woBAXStoryXResptype_op;
DROP full null woBAXStoryXResptype;
R2full = (null - full)/null;
R2woBAXStoryXResptype = (null - woBAXStoryXResptype)/null;
f2 = (R2full - R2woBAXStoryXResptype)/(1 - R2full);
run;
```

```
ERROR 307-185: The data set name cannot have more than 32 bytes.
```

I fixed it with the following change:

```
DATA resultsStudy3a_woBAXSXR; set
Study3a_woBAXStoryXResptype_op;
DROP full null woBAXStoryXResptype;
R2full = (null - full)/null;
R2woBAXStoryXResptype = (null - woBAXStoryXResptype)/null;
f2 = (R2full - R2woBAXStoryXResptype)/(1 - R2full);
run;
```

After making these changes, the whole script ran and the log showed no errors or warnings.

Experiment 3b

The code provided by the authors can be found in “study3b/study3b_syntax_effectsize.sas”. My revised script is in “study3b/study3b_revised.sas”.

Again I had to make major changes for the script to run. The changes were identical to those described for experiment 3a, so I do not repeat them here, and simply state briefly what was done:

- Correct all comments so they become actual comments.
- Change data import statement so that it reads the provided csv file.
- Find-and-replace from “study4b” to “study3b” in all variable names.
- Find-and-replace variable “group1” to “group” and “story1” to “story”.
- Add “lsmeans” option to proc mixed to compute least squares means.
- Change model name Study3b_StoryXResptype to Study3b_woStoryXResptype.
- Add variable “resptype” to “class” and “repeated” argument to model that was missing this.
- Correct data steps which removed all rows from the data.
- Delete data steps which removed all rows from the data.
- Add proc sort in front of data steps for merge to work.
- Not drop CovParm from tables, to be able to distinguish between the *howgood* and *howbad* conditions.

After making these changes, my script ran with no errors or warnings.

Experiment 4

The code provided by the authors can be found in “study4/study4_syntax_effectsize.sas”. My revised script is in “study4/study4_revised.sas”.

Again I had to make major changes for the script to run. Most changes were identical to those described for the experiments above, so I do not repeat them here, and simply state briefly what was done:

- Correct all comments so they become actual comments.

- Change data import statement so that it reads the provided csv file.
- Find-and-replace from “study5” to “study4” in all variable names.
- Find-and-replace variable “group1” to “group” and “story1” to “story”.
- Add “lsmeans” option to proc mixed to compute least squares means.
- Change model name Study4_StoryXResptype to Study4_woStoryXResptype.
- Add variable “resptype” to “class” and “repeated” argument to model that was missing this.
- Correct data steps which removed all rows from the data.
- Delete data steps which removed all rows from the data.
- Add proc sort in front of data steps for merge to work.
- Not drop CovParm from tables, to be able to distinguish between the *howgood* and *howbad* conditions.
- Shorten illegal variable names.

After making these changes, the script ran with no errors or warnings.

## 2. Computational Reproducibility of Reported Statistics Errors found ▾

Is there a clear traceability of reported stats to code? Does the code output match what’s reported in the paper? Are all reported statistics findable within the analysis code?

*Time spent: 2 hours*

There is in principle some traceability of reported stats to code. After making the necessary changes to the code so that it runs – described extensively above – the results mostly show up nice and orderly.

One exception is Cohen’s d, which is reported throughout the paper, but which I cannot find anywhere in the code or the model outputs. That is, I am not able to compute Cohen’s d with the provided code, and the authors do not report how they computed it. As shown in a later section, I was however able to reverse engineer computation of Cohen’s d for mixed model output, and reproduce the values.

There is no information about participants’ sex in the provided data files, and I can hence not reproduce the descriptive statistics for sex distribution. There is also no information about participants’ age in the provided data, and I can hence not reproduce the stated mean value for age. This applies to the sections titled “Participant” on page 113 for experiment 1, on page 115 for experiment 2, on page 118 for experiments 3a and 3b, and on page 119 for experiment 4.

Cronbach's alpha is reported in the section named "Identification questions" on page 114, and in the sections named "Measures" on page 116 for experiment 2, on page 118 for experiments 3a and 3b, and on page 119 for experiment 4. I did not receive the data for reproducing these values.

I cannot reproduce the numbers reported in Appendix A, as I did not receive the data from the pilot study.

Below are some further issues related to each experiment:

Pilot study

The values of Cohen's d reported for the pilot study are computed based on t-tests, although that is not explicitly stated. I am able to reproduce the values d=3.12 and d=9.13 on page 113 using the formula d=(mean - 5.5) / sd, where 5.5 is the midpoint of the scale. I can however not reproduce the value d=13.95 reported for the difference between the positive and negative events. The closest I am able to get is to use the formula $d = t / \sqrt{n}$, which gives the value 13.77. The discrepancy between 13.77 and 13.95 is not due to only rounding, since we have reported t(7)=-38.95, that is, to the nearest two decimals. Hence, both for the lower end 38.945 and the upper end 38.955 we would get 13.77. Of note, I am able to reproduce the value d=17.24 reported in the top of the left column on page 116, using the same formula $d = t / \sqrt{n}$, with t=-59.72 and n=12. Given that the exact same analysis should apply to the values d=13.95 and d=17.24, this suggests that the Cohen's d value reported for the difference on page 113 is an error.

Experiment 1

I did not receive data for reproducing in-group and out-group identification scores, reported on pages 114-115.

The output in Table 1 matches the output from SAS proc mixed except that the denominator degrees of freedom in the first row of Table 1 is stated as 196, whereas the output from SAS is 199, which is also the correct value.

I cannot reproduce the results reported in the last paragraph describing Experiment 1, in the top of the right column on page 115, since I did not have access to the pilot data.

### Experiment 2

I cannot reproduce the results for the pilot study reported near the bottom of page 115 and the first column on page 116 because I did not have access to these data. I can also not reproduce the results related to identification and intergroup differentiation scores, since I did not have these data.

### Experiment 3a

I cannot reproduce the intergroup differentiation scores reported on the top of the left column on page 119, since these data were not provided.

I get different incremental effect sizes f^2 for the four-way interaction reported in the section "Predicting how bad/good the participants felt", in the left column on page 119. The authors reported the effect sizes for this non-significant effect to be 0.0002 for both conditions, whereas I get -0.0002 for both. While negative f^2 sounds counterintuitive, this is possible and not wrong here, since it is defined as f^2 = (R_squared_full_model - R_squared_reduce_model) / (1 - R_squared_full_model).

### Experiment 3b

I get different incremental effect sizes f^2 for the four-way interaction reported in the second paragraph of section "Predicting how bad/good the participants felt" in the left column on page 119. I get -0.0001 for both, and not 0.0001 as the authors report.

In the right part of Table 3 I get standard error 0.02493 for the first row (Empathy for negative events), which when rounded to two decimals becomes 0.02. It is reported to be 0.03. The same happens in the fourth row (Glückschmerz), where I again get 0.02493 which when rounded to two decimals becomes 0.02, but it is reported to be 0.03.

### Experiment 4

In the last sentence of the first paragraph of section "Predicting how bad/good the participants felt" in the left column on page 120, an effect size with value 0.0005 is reported for the *howgood* condition, but I get 0.0006 when running my revised SAS code.

### 3. Data Processing Errors Indeterminable

Are there substantive errors during the preparation or cleaning of data (e.g. duplication of rows during a merge) prior to substantive analyses and hypothesis tests?

*Time spent: 20 minutes*

While I was not able to run the provided code, I did reproduce all results after making the necessary changes. Hence, I have no reason to believe that the authors did errors in this part of the data processing.

However, I only received the finalized datasets for each experiment, after participants who failed the control condition had been removed. I thus do not have a chance to check whether there were any errors in the initial data processing steps.

### 4. Model Misspecification Errors found

Are there any consequential issues with the assumptions or the form of a statistical model (e.g., overfitting, wrong distribution assumption) used to describe data?

*Time spent: 2.5 hours*

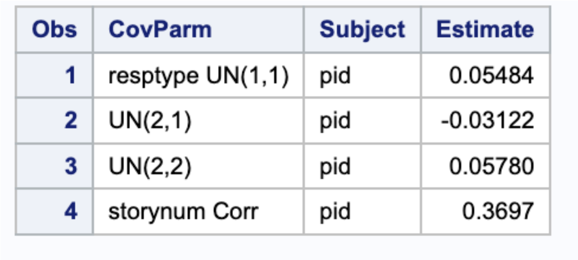
The authors report the value "d" for all the least squares means estimates, and I assume this means Cohen's d. After spending some time searching the literature, I realized that it's not clear how to compute Cohen's d for such marginal estimates from a linear mixed effects model, and the paper nowhere reports how this was done.

**NOTE**: Values of Cohen's d reported for the pilot study, in the bottom paragraph in the right column on page 113 and in the left column on page 116, were not based on mixed models and computed using a different formula.

Based on the reported values, it seems like they used the formula $d = t / \sqrt{df}$, where df is the degrees of freedom and t is the value of the t-statistic. For example, in the left column on page 115 we have df = 199 in all cases, and all the reported values of Cohen's d exactly match this formula. Also, Cohen's d values in the right column on page 116, all Cohen's d values in Table 3 on page 118, and all Cohen's d values on page 120 could be reproduced using this formula.

This approach would be correct for an independent samples t-test with equal sample sizes, where it is normally used. To see why this approach is problematic in the current setting, note that for the "Differences of Least Squares Means" output by SAS, the t-value is computed as $t = \frac{\hat{\beta}}{SE(\hat{\beta})}$, where $\hat{\beta}$ is the predicted value of population mean difference (not directly a parameter in the model) and the denominator is its standard error. Hence Cohen's d with this formula becomes $d = \frac{\hat{\beta}}{\sqrt{df} \times SE(\hat{\beta})}$. That is, the effect size becomes defined as the estimated effect divided by the **uncertainty** of that regression coefficient multiplied by the square root of the number of degrees of freedom. However, the definition of Cohen's d is $\frac{mean\ difference}{standard\ deviation\ of\ the\ outcome}$, and while the numerator $\hat{\beta}$ appropriately defines a mean difference here, $\sqrt{df} \times SE(\hat{\beta})$ does not represent the standard deviation of the outcome. Instead, it is an approximation of the sampling variability of the effect estimate $\hat{\beta}$, i.e., if we sampled datasets repeatedly from the population and computed $\hat{\beta}$, $\sqrt{df} \times SE(\hat{\beta})$ is approximately the standard deviation of the estimates we would get.

An approach that would come closer to Cohen's original definition would be to define the it as $t = \frac{\hat{\beta}}{residual\ standard\ deviation}$. Since $\hat{\beta}$ is in fact a prediction – the expected response value for someone with a certain set of values for the factor variables minus the expected response for someone with a different set of values for the factor variables – the residual standard deviation from the model would represent the actual standard deviation of the outcome, after taking into account the model structure. That is, the residual standard deviation would represent how much the actual value would vary between individuals in the population who have the same values of the factor variables. This is not straightforward here, however, since the models are set up such that the residuals within a given individual are allowed to be correlated. This is given by the argument `type=un@cs` in all the calls to PROC MIXED. This means that we have one residual for each of the two response types, and we have the covariance matrix shown in the screenshot below:



| Obs | CovParm | Subject | Estimate |
|---|---|---|---|
| 1 | resptype UN(1,1) | pid | 0.05484 |
| 2 | UN(2,1) | pid | -0.03122 |
| 3 | UN(2,2) | pid | 0.05780 |
| 4 | storynum Corr | pid | 0.3697 |

UN(1,1) is for the *howbad* condition and UN(2,2) is for the *howgood* condition. Hence, we would have to pool the estimate for UN(1,1) and UN(2,2) to get Cohen's d. I tried this for Experiment 1. The code is provided at the very end of the script "study1/study1_revised.sas".

The table below shows Cohen's d as it is reported in the paper for experiment 1 in the left column on page 115, and Cohen's d using the alternative approach described above. The values acceptably close, and certainly close enough that the substantial interpretation does not differ. Note that both approaches here are approximations, and there is no ground truth to how to compute Cohen's d. I do however still consider this a minor error, since it was not reported how Cohen's d was computed since the approach used is a crude approximation.

| **Reported in paper** | **Alternative calculation** |
|---|---|
| 0.79 | 0.68 |
| 0.70 | 0.62 |
| 0.62 | 0.55 |
| 0.41 | 0.35 |
| 0.37 | 0.38 |
| 0.40 | 0.43 |
| 0.27 | 0.29 |
| 0.06 | 0.11 |
| 0.20 | 0.21 |
| 0.12 | 0.12 |
| 0.11 | 0.12 |
| 0.03 | 0.09 |

An additional point: in the "Stimuli" paragraph at the lower left of page 113, the item-analysis should use a paired t-test, while the remaining t-tests should be for independent samples. It is unclear from the description whether this was done, and I did not receive any code or data to reproduce this. The same applies to Cohen's d for the comparison across positive and negative events (d = 13.95). In contrast to the single sample Cohen's d computed when comparing to the scale's midpoint, this value should be computed using the sample standard deviation of differences.

### 5. Erroneous/Impossible/Inconsistent Statistical Reporting Errors found

Are there inconsistencies between test statistics, degrees of freedom, and p-values? Are there implausible degrees of freedom between compared SEM models? Are there point estimates outside the confidence interval bounds?

*Time spent: 30 minutes*

In Table 1, in the first row for Effect "Functional relations (FR)", the denominator degrees of freedom is given as 196. The correct value here should be 199.

In the right paragraph on page 116, where marginal means are reported, there is some inconsistent reporting. For Schadenfreude out-group>in-group, it is stated "t(94) = 3.06, p<.001". However, the p-value here is 0.0029, which is not less than 0.001. A one-sided p-value would be 0.0029 / 2, which is still not less than 0.001. Hence, correct would be "t(94) = 3.06, p<.01".

Furthermore, there is an instance of inconsistent reporting in the same column on page 116. For empathy for positive events, out-group < unaffiliated, it is stated "t(94) = -3.71, p<.01". While this is mathematically correct, the p-value here is 0.0004, so it should have been reported as p<.001.

On page 119, in the first paragraph in the section titled "Predicting how bad/good the participants felt" it is stated in parentheses "nor were any of the other lower-order interactions including the before-after feedback factor: all ps > .35". While it is correct that none of the interactions were statistically significant, the term "bef_aft * resptype" has a p-value of 0.2609 in the output from PROC MIXED. Hence, the statement "all ps > .35" is not correct.

In the last sentence of the first paragraph of section "Predicting how bad/good the participants felt" in the left column on page 120, a wrong number of degrees of freedom is reported for the F-statistic. It is reported as "F(1,67) = 5.96", but correct would be "F(1,70) = 5.96". In the second paragraph of the same section, a wrong number of degrees of freedom is reported for the difference in marginal means for positive events “t(170) = -7.00”. The correct value should be “t(70) = -7.00”.

### 6. Other Aspects Related to Data or Code Not applicable ▾

*Time spent: 0 minutes*

## III. CLAIMS, PRESENTATION, AND INTERPRETATION

### 1. Interpretation Issues No errors found ▾

Throughout the entire paper, is there an incorrect substantive interpretation of data or statistical tests, causal inference issues, etc.?

*Time spent: 1 hours*

The authors are very careful about their wording, and by and large describe the results of statistical tests perfectly.

Some comments:

On the top of the left column on page 115, they write
*“Note that identification scores did not vary by gender in this or any of the other experiments (all ps > .10), nor did gender moderate the critical interactions. Thus we collapse across gender in all results.”*
Leaving the interpretation of ps>.10 as the absence of an effect aside: this is a randomized controlled trial, and hence by design gender has no impact on *“the critical interactions”*. That is, even if identification scores (or other outcome measures) do vary by gender, the critical interactions will not. So to conclude, the authors are right in *“collapsing across gender in all results”*, but they did not need to do any statistical tests to conclude this.

Since the authors end up doing the correct thing, I don’t label this as an error.

### 2. Overclaiming Generalisability No errors found ▾

Does the paper overclaim the generalisability of the findings with regards to stimuli, situations, populations, etc.? Is there hyping or overselling of the importance or relevance of findings?

*Time spent: 10 minutes*

Everything is fine. The authors are very careful and precise in their interpretations.

### 3. Citation Accuracy No errors found ▾

Are there misrepresentations of substantive claims by cited sources? Inaccurate direct quotes? Incorrectly cited or interpreted estimates? Citations of retracted papers?

*Time spent: 10 minutes*

I did not find any misrepresentations of substantive claims by cited sources.

I asked a large language model to check every single reference in the paper and search the internet to see whether it had been retracted. No retracted papers were found.

### 4. Other Aspects Related to Interpretation No errors found ▾

*Time spent: 0 minutes*