

PROCESS Documentation Addendum

PROCESS is documented in Appendices A and B of the second edition of *Introduction to Mediation, Moderation, and Conditional Process Analysis*. This addendum to the documentation describes options and features added to PROCESS since the first printing of the book in December 2017. The most recent version of this addendum was produced on December 17, 2020.

Dichotomous Y

(Added in version 3.1)

With the release of version 3.1, PROCESS will accept a dichotomous outcome variable Y . No input is needed from the user to specify that Y is dichotomous; PROCESS figures this out on its own. When PROCESS sees only two values in the variable specified as Y , the estimation of Y is conducted using logistic regression, modeling the probability of the event coded in Y , with the event being modeled represented in the data with the largest numerical code in the variable specified as Y . For example, if Y is coded 0 and 1, the probability of $Y = 1$ is estimated, and all effects should be interpreted accordingly. All regression coefficients for the model of Y are logistic regression coefficients and are on a log-odds metric, including confidence intervals for these coefficients. These can be exponentiated to yield effects on an odds ratio metric.

Inferences for each regression coefficient in the model of Y are based on ratio of the estimate of the coefficient to its standard error. P -values and confidence intervals are based on the assumption of a normally distributed sampling distribution of the regression coefficient. Hence, the hypothesis test that a regression coefficient is equal to zero is equivalent to a Wald test. Inference for the complete model of Y is based a likelihood ratio test with degrees of freedom equal to the number of predictors in the model of Y .

When the model of Y includes a moderation component, the section of output for the model of Y labelled “test for highest-order unconditional interaction(s)” is based on a likelihood ratio test, comparing the fit of the model of Y that includes the interaction compared to a model that excludes it. For single-degree-of-freedom tests of interactions, this test can produce a p -value for the interaction that is different than the p -value produced for the ratio of the regression coefficient to its standard error, as likelihood ratio and Wald tests are equivalent only asymptotically and can produce different results in finite samples.

Although likelihood ratio tests are used for tests of unconditional interactions, tests of conditional interactions as well as tests of conditional effects of X when X is specified as multicategorical are Wald tests.

When the **plot** option is used with a dichotomous Y , estimates of Y for combinations of focal predictor and moderator(s) are estimated log odds of the event for those combinations. PROCESS also produces the estimated probability of the event under the label “prob.”

Indirect effects of X when Y is dichotomous are calculated as always as the product of the effect of X on the mediator M and the effect of mediator M on Y controlling for X . The indirect effect (and direct effect of X) is on a log-odds metric given that the effect of the mediator on Y is a logistic regression coefficient. Because the regression coefficient for X in a model of Y without the mediators included is not equal to the sum of the direct and indirect effects of X , the **total** option is not available with a dichotomous Y . The **effsize** option is also disabled for models with a dichotomous Y .

Logistic regression is computationally more intensive than ordinary least squares regression analysis. When combined with repeated model estimation using bootstrapping for inference about indirect effects, time to produce output can be considerable even in modest sample sizes. Expect PROCESS to take longer to show results than when Y is not dichotomous. A good strategy when estimating a model with indirect effects is to first set the number of bootstrap samples to a smaller number than the default of 5000 (e.g., **boot=1000**), even zero, to make sure the model will estimate and PROCESS will produce output. Once you are satisfied, increase the number of bootstrap samples to generate the final result that you interpret.

In addition, the logistic regression estimation module programmed in PROCESS is more susceptible to matrix inversion errors and also errors produced by “perfect separation” (as it is called in the logistic regression literature). These errors are more likely to occur when X and/or moderators W or Z are multicategorical or when the event is rare or very frequent in the data. If you see INV, EXP, or other errors in the SPSS output or SAS log file, do not interpret the output.

The user has control over the maximum number of iterations before PROCESS provides a solution as well as the convergence criterion. The convergence criterion is met when changes to the logistic regression coefficients result in a change in the likelihood of less than the specified criterion. At that point, the iteration stops and the solution is provided. A failure to converge can sometimes be rectified by increasing the number of iterations or increasing the convergence criterion. However, doing so will increase computational time, perhaps dramatically. This can be done with the **iterate** and **converge** options. The defaults are 100 and .0001, respectively (i.e., **iterate=100** and **converge=.0001**).

Specifying a Moderator as a Covariate

(Added in version 3.2)

On pages 630-632 of *Introduction to Mediation, Moderation, and Conditional Process Analysis*, a trick is described for how to specify a moderator in one equation

as a covariate in another equation. With the release of PROCESS v3.2, this trick is no longer necessary for a dichotomous or continuous moderator. PROCESS now allows a moderator (a variable specified as *W* or *Z*) to be specified as a covariate in the usual way by listing it following **cov=**. By default, such a variable will be added to all model equations except those where it already plays the role of a moderator variable. To include it in only some of the equations, use the **cmatrix** option described in Appendix A.

For example, to estimate the model depicted in Figure B.7 (see page 631), the SPSS PROCESS command is

```
process y=tile/m=wine/x=baby/w=milk/cov=sand tent milk/model=7.
```

In SAS, the equivalent command is

```
%process (data=four,y=tile,m=wine,x=baby,w=milk,cov=sand tent milk,
          model=7);
```

Note that this does not work for a moderator specified as multicategorical using the **mcw** or **mcz** options. To specify a multicategorical moderator in one equation as a covariate in another, the trick on pages 630-632 must still be used after first manually constructing a set of codes (e.g., indicator, Helmert, etc.) representing the multicategorical variable and adding them to the data file.

Standardized Regression Coefficients

(Added in version 3.2)

In mediation-only models (i.e., models with no product terms to capture moderation), the **effsize** option produces the completely and partially standardized direct and indirect effects in the summary section of the PROCESS output (see section 4.3 of *Introduction to Mediation, Moderation, and Conditional Process Analysis*). However, the regression coefficients elsewhere in the output are printed in unstandardized form. The release of version 3.2 adds an option for generating standardized regression coefficients for the regression models of mediators and *Y* provided in the PROCESS output. To generate standardized regression coefficients, add **stand=1** to the PROCESS command. These regression coefficients will be in completely standardized form. But because a completely standardized regression coefficient is not meaningful for a dichotomous or multicategorical variable, if the variable specified as *X* in the PROCESS command is dichotomous or multicategorical, then standardized regression coefficients for *X* (or multicategorical codes representing *X*) will be in partially standardized form. A partially standardized regression coefficient is equal to the unstandardized regression coefficient divided by the standard deviation of the outcome variable (i.e., the variable on the left side of the regression equation). That is, unlike a completely standardized regression coefficient, which removes the metrics of the predictor and the outcome from the scaling of the unstandardized regression coefficient, the partially standardized regression coefficient retains the scaling of *X*. See the discussion about partially and

completely standardized regression coefficients on pages 52-54 of *Introduction to Mediation, Moderation, and Conditional Process Analysis*.

Standardized regression coefficients are not available in models that include any products of variables to represent an interaction. Standardized regression coefficients are also not available when Y is dichotomous. Note that standardized regression coefficients for dichotomous covariates (i.e., variables listed in the **cov** option) will be in completely standardized form. These should not be interpreted or reported.

New Mean Centering Option

(Added in version 3.4)

The **center=1** option available in all earlier releases of PROCESS mean centers continuous and dichotomous variables that are used to construct products of variables when parameterizing an interaction between those variables (codes representing a multicategorical focal predictor or moderator are not centered when the **center** option is used). With the release of version 3.4, a new **center=2** option mean centers only the continuous variables used to form a product. A dichotomous focal predictor or moderator used in the construction of a product is kept in its uncentered form when this option is used. Therefore, if all focal predictors and moderators are dichotomous, the **center=2** option has no effect.

Testing for Interaction Between X and a Mediator

(Added in version 3.4)

An indirect effect of X on Y is a product of at least two effects, such as the effect of X on mediator M and the effect of M on Y when X is held constant. The effect of M is estimated assuming that this effect does not depend on X , the assumption of *no X by M interaction*. PROCESS cannot estimate any model that allows X to moderate any path in a mediation model. However, with the release of version 3.4, PROCESS now provides a test of this no-interaction assumption. By specifying **xmtest=1** in the PROCESS command line, PROCESS will conduct a test of interaction between X and each variable specified as a mediator in models of outcomes causally downstream.

Depending on the complexity of the model, there may be many such tests conducted. For example, in a parallel multiple mediator model with four mediators, there will be four such tests, one for each mediator in the model of Y . In a serial multiple mediator model with two mediators, there will be three tests, one for each mediator in the model of Y , and one for M_1 in the model of M_2 .

This test is conducted by assuming no interaction between X and any mediator in the model specified, and then releasing this constraint for each of the paths leading from a mediator to another variable, while all other effects of the other mediators in the model are assumed to be independent of X . For tests involving a continuous Y , the test takes the form of an F -ratio when comparing the fit of

the constrained and unconstrained model. For tests involving a dichotomous Y , a likelihood ratio test is conducted.

Consider, for example, the output below generated for the parallel multiple model discussed in Chapter 5 by adding `mxtest=1` to the PROCESS command line on page 156:

Test(s) of X by M interaction:

	F	df1	df2	p
M1*X	.0110	1.0000	118.0000	.9166
M2*X	2.2035	1.0000	118.0000	.1404

The first row indicates no statistically significant interaction between experimental condition and topic importance in the model of reaction to the story, $F(1, 118) = 0.011, p = .917$. This test is conducted while fixing the effect of presumed media influence (M_2) to be unmoderated by experimental condition. The second row indicates no interaction between presumed media influence and experimental condition in this same model, $F(1, 118) = 2.204, p = 0.140$, fixing the effect of topic importance (M_1) to be independent of experimental condition. Combined, these tests support the assumption of no interaction between X and either mediator in this parallel multiple mediator model.

Note that this test is not provided for a mediator if the model specifies a three way interaction between that mediator and moderators W and Z (as in, for example, model 18). This test is suppressed in this circumstance because the test of interaction would be conditioned on holding either W or Z (or both) fixed at zero, which may not be meaningful or substantively interpretable.

If evidence of interaction between X and a mediator is found, that interaction can be probed by estimating an equivalent moderation-only model such as model 1 with X specified as the moderator W and M specified as the focal predictor X . The larger the difference in the effect of M across values of X , the less meaningful will be the indirect effect of X . Judgment must be made by the investigator as to whether such interaction jeopardizes the meaningfulness or validity of an inference from a mediation or conditional process analysis. There are methods for mediation analysis that allow for X by M interaction, though these methods are restricted in application to relatively simple models. See section 14.6 for a discussion.

Joint Test of Additive Moderation

(Added in version 3.4)

When an effect in a model is specified as moderated additively by two moderators W and Z (e.g., in model 2, the effect of X on Y ; in model 16, the path from M to Y), PROCESS now provides a joint test of the null hypothesis that neither W nor Z moderates the effect. When the outcome variable (i.e., the variable on the left side of the equation) is continuous, this test takes the form of an F -ratio. When it is dichotomous, a likelihood ratio test is provided.

Consider, for example, the section of output below from model 2, which specifies that the effect of X on Y is moderated additively by both W and Z . The joint test is listed in the row labeled "Both."

Test(s) of highest order unconditional interaction(s):

	R2-chng	F	df1	df2	p
X*W	.0640	14.2636	1.0000	202.0000	.0002
X*Z	.0301	6.7192	1.0000	202.0000	.0102
BOTH	.0697	7.7643	2.0000	202.0000	.0006

In this example, we can reject the null hypothesis that neither W nor Z moderates the effect of X on M , $F(2, 202) = 7.764, p < .001$. Adding the two products to the model allowing the effect of X on M to be linearly moderated by W and Z increases the squared multiple correlation by .070.

Bias-Corrected Bootstrap Confidence Intervals

(Added in version 3.5)

As of version 3.5, bias corrected bootstrap confidence intervals are available as an option for bootstrap inference. By default, bootstrap confidence intervals are generated using the percentile method. To generate bias corrected bootstrap confidence intervals instead, add **bc=1** to the PROCESS command.

Correlations Between Residuals

(Added in version 3.5)

All mediation and conditional process models are constructed from $k + 1$ regression equations, where k is the number of mediators. For some models, the correlations between the residuals of all or some of regression equations are necessarily zero, whereas for others they may be different from zero. The correlation between residuals can sometimes be useful for model evaluation and modification. Adding **modelres=1** to the PROCESS command will generate a matrix of observed correlations between the residuals. This matrix will be printed following the regression equation for the variable specified as Y in the PROCESS command.

Long Variable Names in PROCESS for SPSS

(Added in version 3.5)

Until version 3.5, PROCESS for SPSS has allowed the user to specify variables in the PROCESS command with variables names in the data file containing more than eight characters. A warning against the use of long variable names has always been included in the documentation. This warning emphasizes the potential that PROCESS will confuse variables in the data file that have the same first eight characters in their names, and this can produce incorrect output. However, this warning is sometimes ignored by users who have not read the documentation for PROCESS before using it. For this reason, by default, version 3.5 will no longer allow the inclusion of variables in a PROCESS command that have names longer than eight characters. However, this default can be overridden by adding

longname=1 to the PROCESS command. Users who override this default are taking the risk and accept responsibility that the output generated by PROCESS may be incorrect if the PROCESS command includes variables in the data file that are not unique in the first eight characters of their names.

PROCESS for R

(Available as of version 3.5)

As of version 3.5, PROCESS is available for R in beta version. The R beta version of PROCESS is operated similarly to the SPSS and SAS versions, and most of the instructions described in Appendices A and B of the second edition of *Introduction to Mediation, Moderation, and Conditional Process Analysis* apply to the R version, with only the minor modifications described below. As a beta release, this version is still going through testing, and the user should beware that bugs or other programming imperfections could still exist in the code and produce incorrect results. As with all software, use at your own risk.

Like the SPSS and SAS versions, PROCESS for R is a program file or “script” (process.R) that when executed defines a function called **process** in the existing workspace. Once the PROCESS script is executed (without changing the file whatsoever), then the **process** function is available for use and process.R can be closed. Have patience, as process.R is a large script file and it will take a few minutes for it to execute. When it is complete, a message in the console will appear stating that PROCESS is ready for use. PROCESS was written base R v3.6. PROCESS for R does not rely on and so does not require the installation of any packages prior to use.

Because PROCESS for R is a script that defines a function rather than a package, process.R must be executed each time you begin an R session and intend to use the PROCESS function. To circumvent this, save the workspace after running process.R, either using the existing workspace name or a different one. With the workspace saved, PROCESS will be available for use and will function much like a package whenever using that workspace. PROCESS is not currently available as a package. Beware that a package called “processr” is available on CRAN that attempts to mimic the performance of an older release of PROCESS. It is an unofficial version of PROCESS produced by an R user but without many of the features of the official version described in the second edition of *Introduction to Mediation, Moderation, and Conditional Process Analysis* and this document. It won’t necessarily produce the same information or produce output in the same format.

The syntax structure for PROCESS for R is most similar to the SAS version of PROCESS. The list below enumerates some important distinctions between the SPSS, SAS, and R versions of PROCESS code:

- Like the SPSS versions, PROCESS for R is called with the **process** function. But like the SAS version, all arguments except the call itself (i.e., the word **process**) should be enclosed in parentheses.
- The data file being analyzed must be in the form of an R data frame, and the name of the data frame should be specified in the PROCESS command

as **data=name** where *name* is the name of a data frame containing the data being analyzed.

- PROCESS for R accepts data only in *numeric* format. Thus, for example, if a variable named **sex** were coded M and F in the data, these alphabetic codes must be converted to numeric form (e.g., 0 and 1) prior to their use in a PROCESS command. PROCESS will not accept variables that are designated as factors. The factor designation must be removed from a variable before it can be used in a PROCESS command.
- Missing data must be represented in the data frame as NA.
- Unlike SPSS and SAS, R is a case-sensitive language. The **process** call as well as all options must be in *lower case*. Variable names are also case-sensitive. Variable names used in a PROCESS command should be consistent with the case of the names as stored in the data frame being analyzed.
- Unlike in the SPSS and R versions, PROCESS for R requires variable names be enclosed in double-quotes (e.g., **y="hello"**).
- Like the SAS version, options and specifications must be delimited with a comma (,).
- When more than one variable is listed as a mediator or a covariate, the names of the variables should be separated by a comma and enclosed in the "c()" function, as in **m=c("med1", "med2", "med3")** and **cov=c("cov1", "cov2")**. For instance, the PROCESS for R code for example 4 on page 555 is

```
process (data=chip,y="know",x="educ",m=c("attn","elab"),cov=
c("sex",age"),model=4,contrast=1,normal=1,conf=90,save=1)
```

- When using the **wmodval**, **zmodval**, **xcatcode**, **wcatcode**, **zcatcode**, or **contrast** options, moderator values, contrast weights, and codes defining groups should be separated by a comma and enclosed in the "c()" operator, as in **wmodval=c(2,3,4)** and **contrast=c(1,1,-1,-1)**.
- When specifying a custom model or editing a preprogrammed model using the **bmatrix**, **wmatrix**, **zmatrix**, and **wzmatrix** options, and when assigning covariates to equations using the **cmatrix** option, the zeros and ones in the sequence defining a model matrix or assigning the covariates should be separated by commas and enclosed in the "c()" operator. For example,

```
process (data=study,y="behavior",x="friends",m=c("attract",
"similar"),w="weight",z="bmi",cov=c("tv","cancer"),bmatrix=
c(1,1,1,0,1,1),wmatrix=c(1,1,1,0,0,0),zmatrix=c(1,0,0,0,0,0),
cmatrix=c(1,0,1,1,0,1),mcz=2)
```

- When using the **contrast** option to compare conditional effects of X on Y in models 2 and 3 (see page 342), moderator values should be provided in the order w_1 z_1 w_2 z_2 , separated by commas, and the entire sequence of

values should be enclosed in the "c()" operator. For example, to compare the conditional effect of X on Y when $w = 1$ and $z = 3$ compared to when $w = 2$ and $z = 5$, use **contrast=c(1,3,2,5)**.

- Like in the SAS version, when specifying the number of decimal places in output using the **decimals** option, the "F" that is required in the SPSS version should be left off the *dec* argument. For example, to set 12 characters for numbers with six after the decimal, use **decimals=12.6**.
- The **plot** option in PROCESS for R produces a table of data for visualizing an interaction but, like the SAS version, does not write any R code to produce the plot, nor will PROCESS produce a plot in the R window.
- The default random number generator in R is different than the default random number generator in SPSS and SAS. Thus, bootstrap confidence intervals generated by R will be different even when the same seed is used when estimating the same model using the same data.
- While bootstrapping, PROCESS for R will display a graphical counter indicating progress toward completion of the bootstrap sampling. This counter can be deactivated by adding **progress=0** to the PROCESS command.
- PROCESS for R has three save options, both of which require sending information from PROCESS to an R object using the <- operator when calling PROCESS. Option 1 (i.e., **save=1**) will produce an R data frame containing all the bootstrap estimates of the regression coefficients that define the model. This can be used in the manner described starting on page 573, such as to visualize bootstrap distributions or conduct inferences about quantities that can be expressed in the form of functions of regression coefficients. For example, the R code below conducts the same analysis described on page 574 that constructs a histogram of the bootstrap estimates of the indirect effect and produces a 95% bootstrap confidence interval for the difference between the direct effect and the indirect effect of X . Using this code, the bootstrap estimates are saved in an R data frame named "result."

```
result<-process(data=pmi,y="reaction",x="cond",m="pmi",model=4,
  seed=31216,save=1)
ab<-result$col2*result$col5
hist(ab,breaks=25)
diff<-result$col4-ab
quantile(diff,c(.025,.975))
```

Save option 2 (i.e., **save=2**) will generate a matrix containing the majority of the statistics in the PROCESS output for the model estimated. A version of this matrix will also be printed in the PROCESS output. Entries in this matrix can be given to other R programs for processing of the output generated by PROCESS. For example, the PROCESS for R command

```
result<-process(data=disaster,y="justify",x="skeptical",w="frame",
model=1,plot=1,save=2)
```

from the analysis described in section 8.1 produces a matrix held in “result” that looks like (though to more significant digits of resolution):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.4962	0.2463	0.6609	22.5430	3.0000	207.0000	0.0000
[2,]	2.4515	0.1490	16.4486	0.0000	2.1577	2.7454	NA
[3,]	0.1051	0.0381	2.7559	0.0064	0.0299	0.1803	NA
[4,]	-0.5625	0.2179	-2.5811	0.0105	-0.9921	-0.1328	NA
[5,]	0.2012	0.0553	3.6401	0.0003	0.0922	0.3101	NA
[6,]	0.0482	13.2503	1.0000	207.0000	0.0003	NA	NA
[7,]	0.0000	0.1051	0.0381	2.7559	0.0064	0.0299	0.1803
[8,]	1.0000	0.3063	0.0400	7.6548	0.0000	0.2274	0.3851
[9,]	1.5920	0.0000	2.6188	NA	NA	NA	NA
[10,]	2.8000	0.0000	2.7458	NA	NA	NA	NA
[11,]	5.2000	0.0000	2.9980	NA	NA	NA	NA
[12,]	1.5920	1.0000	2.3766	NA	NA	NA	NA
[13,]	2.8000	1.0000	2.7466	NA	NA	NA	NA
[14,]	5.2000	1.0000	3.4816	NA	NA	NA	NA

Comparing this matrix to the output in Figure 8.2, notice, for example, that R^2 for the model is row 1, column 2 of the matrix, the conditional effect of X when $W = 0$ is in row 7, column 2, and \hat{Y} when $X = 5.2$ and $W = 1$ is in row 14, column 3.

Save option 3 (**save=3**) is a combination of save options 1 and 2. When used in conjunction with bootstrapping (by default for models 4 and higher or any custom model that estimates an indirect effect, unless the **boot** option is set to 0), the resulting object will be a *list*. The first element in this list will be the data frame of bootstrap estimates of regression coefficients, and the second element will be the matrix of statistics shown in the PROCESS output

Note that in models that include interactions, PROCESS output and thus the dimensions of the resulting output matrix ordinarily would be influenced by the p -values for interactions and the **intprobe** setting, which defaults to 0.10. Thus, when using **save** option 2 or 3, the **intprobe** setting will automatically be set to 1 by PROCESS so that the dimensions of the output matrix generated by the **save** option will not depend on the p -values for interactions.

- When PROCESS is used as a computational engine, it may be desirable to suppress the generation of output in the R console when saving output using the **save** option. Output to the console can be suppressed by adding **outscreen=0** to the PROCESS command.

Below are some example commands in PROCESS for R that correspond to equivalent commands for SPSS and SAS found in the second edition of *Introduction to Mediation, Moderation, and Conditional Process Analysis*.

Chapter 3, Page 91:

```
process(data=pmi,y="reaction",x="cond",m="pmi",total=1,normal=1,
        model=4,seed=31216)
```

Chapter 4, page 126:

```
process(data=estress,y="withdraw",x="estress",m="affect",cov=c("ese",
        "sex","tenure"),model=4,seed=100770)
```

Chapter 7, page 238:

```
process(data=disaster,y="justify",x="frame",w="skeptic",model=1,
        jn=1,plot=1)
```

Chapter 8, page 280-281:

```
process(data=glbwarm,y="govact",x="negemot",w="age",cov=c("posemot",
        "ideology","sex"),model=1,jn=1,plot=1,wmodval=c(30,50,70))
```

Chapter 9, page 345:

```
process(data=glbwarm,y="govact",x="negemot",w="sex",z="age",
        cov=c("posemot","ideology"),model=3,zmodval=c(30,50,70),
        contrast=c(1,30,0,50))
```

Chapter 10, page 357:

```
process(data=protest,y="liking",x="protest",w="sexism",mcx=1,model=1,
        plot=1)
```

Appendix B, page 620:

```
process(data=four,y="tile",m=c("wine","tent","sand"),x="baby",
        w="milk",z="hair",bmatrix=c(1,1,0,0,1,1,1,1,0,1),wmatrix=c(1,0,0,0,1,
        0,1,0,0,0),zmatrix=c(1,1,0,0,0,0,0,0,0,0))
```