PROCESS Documentation Addendum

PROCESS is documented in Appendices A and B of the third edition of *Introduction to Mediation, Moderation, and Conditional Process Analysis*. The addendum to the documentation describes options and features added to PROCESS since the first printing of the book in December 2021. This version of this addendum was produced and uploaded on 14 April 2022.

Missing Data Identification

(Added in version 4.1)

PROCESS uses listwise deletion prior to analysis, meaning that any case in the data file that has missing data on any of the variables in the model will be deleted from the analysis. The resulting sample size *after* listwise deletion is provided at the top of the PROCESS output, and at the bottom PROCESS will provide how many cases with missing data were deleted prior to analysis. However, the user is left in the dark about which cases were deleted from the analysis as a result of missing data. With the release of PROCESS version 4.1, a new option is available that provides information about which cases were deleted. When this option is turned on by adding **listmiss=1** to the PROCESS command, PROCESS will list the case numbers at the bottom of the output, identified by row in the data file, that were deleted from the analysis.

Total Effect of *X* in Mediation Models

(Added in version 4.1)

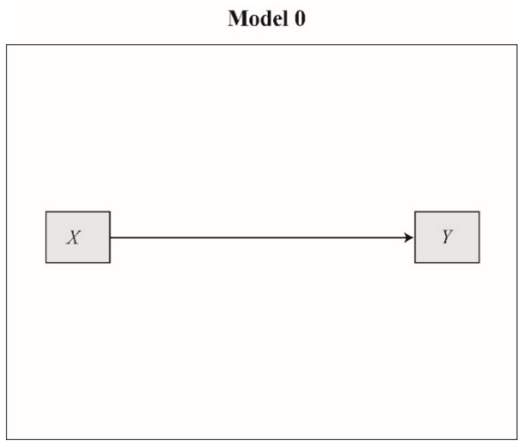
In the typical mediation model, the total effect of *X* on *Y*, which is the sum of the direct and indirect effects of *X*, is estimated in a regression model of *Y* regressed on *X* but not the mediators. But the regression coefficient for *X* in this model will not always be equal to the sum of the direct and indirect effects of *X*, such as when a model includes covariates and the covariates are not included in all equations that are used to estimate the direct and indirect effects of *X*. For this reason, PROCESS will not always produce the total effect of *X* or the model that estimates it when the **total** option is used.

As of the release of version 4.1, for models such as these when the total effect cannot be estimated by regressing *Y* on *X*, PROCESS will now produce the sum of the direct and indirect effects in the output along with a bootstrap confidence interval for inference when the **total** option is used. This option will only generate a point and interval estimate of this sum as well as a bootstrap estimate of the standard error of the sum. Standardized metrics of the sum are not available in this release.

Regression Analysis without Moderation or Mediation

(Added in version 4.1)

Until the release of version 4.1, PROCESS could only estimate models with a mediation or moderation component. With version 4.1, PROCESS can now estimate ordinary regression



models without moderation or mediation of *X*'s effect on *Y*. To estimate such a model, specify **model=0**, provide the consequent (outcome) variable following **y=** and a single antecedent (predictor) variable following **x=**. Other variables, if any, included in the regression model should be listed following **cov=**. Which variable you choose to following **x=** and which as covariate(s) is arbitrary. For example, using the GLBWARM data introduced in Chapter 2, the command below estimates support for government action from negative emotions, positive emotions, ideology, sex, and age.

```
process y=govact/x=negemot/cov=posemot ideology sex age/model=0.
```

```
%process(data=glbwarm,y=govact,x=negemot,cov=posemot ideology sex age,model=0)
```

```
process(data=glbwarm,y="govact",x="negemot",cov=c("ideology","sex","age"),model=0)
```

Heteroscedasticity-consistent inference is available as usual with the **hc** option, as are bootstrap confidence intervals for the regression coefficients using the **modelbt** option. If *X* is a multicategorical, it should be specified as such using the **mcx** option and PROCESS will automatically create category codes as described in Appendix A. Multicategorical variables listed in **cov** must be properly represented with a categorical coding system with the codes (e.g., indicator, Helmert, etc.) generated outside of PROCESS.

Model 0 is not available in the PROCESS graphical user interface for SPSS. In SPSS, it can only be accessed through PROCESS syntax.

Estimation of and Inference for a Weighted Sum of Regression Coefficients

(Added in version 4.1)

Any statistic that can be calculated as a weighted sum of the regression coefficients in a model can be generated using a new **linsum** option in PROCESS available as of version 4.1. This option is available only for models 0, 1, 2, and 3. The weighted sum takes the form

$$\sum_{i=0}^{k} \lambda_i b_i$$

where b_0 is the regression constant, b_1 through b_k are the regression coefficients for the k variables in the model in the order they appear in the regression output for the model of Y, and λ_i are the weights. These weights are listed in sequence 0 to k from left to right following **linsum=** in the PROCESS command and in the same order as the regression weights appear in the PROCESS output from top to bottom.

The estimated value of the consequent variable Y given values on a set of predictor variables in the model is an example of a weighted sum of regression coefficients. For example, using the DISASTER data in Chapter 7 and the PROCESS output in Figure 7.6, the estimated justification for withholding aid for a person in the disaster frame condition (`frame` = 1) with a score of 3 on the skepticism scale (`skeptic` = 3) is

$$\hat{Y} = 1(2.4515) + 1(-0.5625) + 3(0.1051) + 3(0.2012) = 2.8079$$

where the numbers in parentheses are the regression constant and regression coefficients for `frame`, `skeptic`, and the product of `frame` and `skeptic`, and the weights are $\lambda_0 = 1$, $\lambda_1 = 1$, $\lambda_2 = 3$, $\lambda_3 = 3$, for the regression constant and regression coefficients in this same order. In PROCESS, this weighted sum is generated by adding the **linsum** option and the sequence of weights, as in

```
process y=justify/x=frame/w=skeptic/model=1/linsum=1,1,3,3.
```

```
%process(data=disaster,y=justify,x=frame,w=skeptic,model=1,linsum=1 1 3 3)
```

```
process(data=disaster,y="justify",x="frame",w="skeptic",model=1,linsum=c(1,1,3,3))
```

which toward the bottom of the output produces

```
Linear Combination Estimate and Hypothesis Test

Weight vector:
             weight
constant     1.0000
frame        1.0000
skeptic      3.0000
Int_1        3.0000

   Estimate         se          t          p       LLCI       ULCI
     2.8079      .0826    33.9834      .0000     2.6450     2.9708
```

showing the estimate of justification for withholding aid from the model for such a person is 2.8079. The *t* -statistic and *p*-value for the test of the null hypothesis that this weighted sum equals zero, provided in the output, would not be of much interest in this example, but the standard error and confidence interval for the estimate may be. Here, the estimated standard error of the weighted sum is 0.0826 and the 95% confidence interval for the weighted sum is [2.6450, 2.9708].

It is very important that the weights following **linsum=** be in the same order from left to right as the predictors in the model are displayed in PROCESS output from top to bottom, otherwise the weighted sum will not be the sum you wish to construct. In SPSS and R, the weights should be separated by a comma. In SAS, the weights are separated by a space. In R, the comma-delimited sequence of weights should be enclosed in the c() operator.

The **linsum** option can also be used to compare two regression coefficients in a model. For example, in Chapter 2, support for government action is estimated from negative emotions, positive emotions, ideology, sex, and age. The **linsum** option can be utilized to test whether the regression coefficient for negative emotions is equal to the regression coefficient for positive emotions. This comparison would be a weighted sum of regression coefficients of the form

$$(0)b_0 + (1)b_1 + (-1)b_2 + (0)b_3 + (0)b_4 + (0)b_5 = b_1 - b_2$$

where b_0 through b_5 are the regression constant and regression coefficients for negative emotions, positive emotions, ideology, sex, and age, respectively. In terms of the regression coefficients from the model on pages 51-52,

$$0(4.064) + 1(0.441) + (-1)(-0.027) + 0(-0.218) + 0(-0.010) + 0(-0.001) = 0.468$$

In this weighted sum, the weights are 0, 1, -1, 0, 0, and 0 for the regression constant and coefficients for negative emotions, positive emotions, ideology, sex, and age, respectively. In PROCESS, the model and weighted sum is estimated with the command

```
process y=govact/x=negemot/cov=posemot ideology sex age/model=0/
     linsum=0,1,-1,0,0,0.
```

```
%process(data=glbwarm,y=govact,x=negemot,cov=posemot ideology sex age,model=0,
    linsum=0 1 -1 0 0 0)
```

```
process(data=glbwarm,y="govact",x="negemot",cov=ideology sex age,model=0,
    linsum=c(0,1,-1,0,0,0))
```

which generates in the output

```
Linear Combination Estimate and Hypothesis Test

Weight vector:
             weight
constant      .0000
negemot      1.0000
posemot     -1.0000
ideology      .0000
sex           .0000
age           .0000

   Estimate         se          t          p       LLCI       ULCI
      .4676      .0411    11.3856      .0000      .3870      .5482
```

showing that the difference between these regression coefficients is 0.4676 and statistically significant, $t(809) = 11.3856$, $p < .0001$, with a 95% confidence interval of [0.3870, 0.5482]. The degrees of freedom for the t statistic is the residual degrees of freedom for the model, displayed in the PROCESS output in the model summary section under “df2.”

The **linsum** option expects that in a regression model with k predictors (including products created by PROCESS to capture linear moderation), the sequence should contain $k + 1$ weights (the extra weight is for the regression constant). However, when m covariates are listed following **cov=** , weights for all the m covariates can be left out of the sequence if desired. When weights for covariates are not included, PROCESS will automatically set the weights for each covariate to the arithmetic mean of that covariate. When the number of weights in the sequence is neither $k + 1$ nor $k + 1 – m$, a note will be displayed in the output stating that the vector of weights is not correct and no output for the weighted sum will be generated.