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 finalized for distribution on June 1, 2025, and describes features available in PROCESS 4.1, 4.2, 4.3, and 5.0. All features added in earlier releases are available in later releases.

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.

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=c("posemot","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.

The **linsum** option is not available for logistic regression models (i.e., when Y is dichotomous).

Counterfactually-Defined Indirect, Direct, and Total Effects of *X* in Mediation Models with *X* by Mediator Interaction
(added in version 4.2)

With the release of version 4.2, PROCESS can estimate a mediation model that allows interaction between X and mediator(s). When the **xmint** option is toggled on (by adding **xmint=1** to the PROCESS command), PROCESS will generate counterfactually defined natural

indirect, natural direct, and total effects of *X*, as well as the controlled direct effect. This option, which replaces the now discontinued model 74, is available only for simple (one mediator) and parallel multiple mediator models (model 4). For details on the estimation of such a model, the mathematics of counterfactually-defined effects of *X*, as well as several options available for use in tandemwith the **xmint** option, see CCRAM Technical Report 022-04 available through the Resource Hub at the Canadian Centre for Research Analysis and Methods at *haskayne.ucalgary.ca/ccram*.

Selective Exclusion of Cases from the Analysis
(added in version 4.3)

With the release of version 4.3, it is possible to selectively exclude observations in the data from the analysis. This is accomplished by adding the **exclude** option to the PROCESS command, specifying the row number in the data file you would like to exclude following an equal sign. For example, to exclude the observation in the 12th row, add **exclude=12** to the PROCESS command. To exclude more than one observation, list the row numbers of the observations to be excluded. For example, to exclude the observations in rows 12, 14, and 36, use **exclude=12,14,36** in SPSS, **exclude=12 14 36** in SAS, or **exclude=c(12,14,36)** in R.

Regression Diagnostics
(Added in version 5.0)

Regression diagnostics have many uses in regression analysis, from checking for data entry or other forms of clerical errors, to finding cases that are high in influence or that are in some way distorting the analysis, to checking regression assumptions.

Saving Regression Diagnostic Statistics for Examination and Analysis

Most good regression programs can save various regression diagnostic statistics for each case in the analysis. As of the release of this version, so does PROCESS. This is accomplished using **save** option 4, adding **save=4** to the PROCESS command. In the R version of PROCESS, you must also send the output of the save option to an object for storage. For example, in PROCESS for R

```
diagnostics<-PROCESS(data=…,save=4)
```

sends the diagnostic statistics to an object named "diagnostics".

The regression diagnostics PROCESS generates are discussed in Darlington and Hayes (2017) and other good treatments of regression analysis. These include, for each case in the data:

Column name	Diagnostic statistic
pred	the estimate of the outcome (i.e., variable on the left side of the equation)
resid	residual
dresid	deleted residual
stresid	standardized residual,
tresid	t-residual (aka "externally studentized deleted residual").
h	leverage (aka "hat" value)
mahal	Mahalanobis' distance
cook	Cook's distance
dmsres	change in $MS_{residual}$ as a result of the case being included in the analysis.
drsq	change in R^2 as a result of the case being included in the analysis.
dskew	This will contain all 99999 and is a placeholder for a future diagnostic.
dfb_#	dfbetas for the regression constant and each regression coefficient (dfb_0, dfb_1, etc, appearing in order from left to right as the model variables appear in the output from top to bottom, starting with the regression constant).
dfb_ie#	Change in the indirect effect with and without the case in the analysis, calculated as dfb_ie = IE$_{with}$ − IE$_{without}$. There will be as many dfb_ie columns in the diagnostics file as there are indirect effects in the model. This statistic will not be calculated for the total indirect effect.

In models with a mediation component (but no moderation), PROCESS also generates statistics labeled "dfie_#", which is the change in the indirect effect that results when a case is included in the analysis. When a model contains more the one indirect effect, PROCESS will generate as many columns of these dfie statistics for each case as there are indirect effects, with the columns corresponding in order from left to right as the indirect effects appear in the output from top to bottom. In multiple mediator models, there is no dfie calculated or saved for the total indirect effect.

Cases that were excluded from the analysis as a result of missing data will not be included in the diagnostics file. Case numbers in the diagnostics file are numbered in a variable name "casenum" with values that correspond to the row numbers in the original data file being analyzed. If you are not sure which cases PROCESS deleted as a result of missing data, use the **listmiss** option in the PROCESS command. The diagnostics file also contains each case's value(s) for the regressor(s) in the model, making it easier to determine if there is a systematic relationship between any of the diagnostic statistics and the variables in the model.

As a PROCESS command may generate many different regression equations in the output, the **save=4** option may generate more than one file or collection of regression diagnostics. In the SAS version of PROCESS, these resulting files will be named "diagfile#", with "#" sequencing

upward from 1 to *k* where *k* is the number of regression models in the output. The value of "#" tells you from which equation, in the order the equations appear in the output, the diagnostics come from. In the case of a model with no mediation component, only a single equation is generated by PROCESS and so the diagnostic file is called "diagfile" by default. In SAS, you can change the base name of the file using the **diagfile** option. For example, adding **diagfile=diagstat** to the command will name the file(s) "diagstat" (e.g., diagstat1, diagstat2, etc). These files are stored as temporary SAS work files in the current SAS session. To save them for use later, you must save the diagnostic file(s) as SAS data files.

The R version of PROCESS will save the regression diagnostics as a data frame in the named object as a list, unless the PROCESS command generates only one model. For example, if you named the object `diagnostics` and PROCESS generated three regression equations for your model, the diagnostic statistics for the three models will be held in `diagnostics[[1]]`, `diagnostics[[2]]`, and `diagnostics[[3]]`, with the numbers corresponding to the regression equations as they appear in the PROCESS output from top to bottom. If the model requires only one equation, the object will be a simple data frame rather than a list.

The SPSS version generates only one data file in the active SPSS session, with sets of regression diagnostics for each model stacked on top of each other and numbered in the data file in a variable named "equation". Because not all variables in a PROCESS command will be in every equation, when a variable is not in an equation, all the cases values will be set to 999999 in the rows corresponding to that equation. Likewise, not all equations will have the same number of regressors, so the number of dfbeta statistics will vary from equation to equation. Values in dfbeta columns that have no corresponding values in an equation are set to 999999. Note that this data file that PROCESS produces will not be permanently saved until you manually save the data using the graphical user interface or a SAVE command in SPSS syntax. Because the name of variable on the left side of the equation will vary from equation to equation, in the SPSS version of PROCESS, the values of the variable on the right-hand side of the equation will be found in the column labelled "dv". In addition, unlike the SAS and R versions, the SPSS version will not generate regression diagnostic statistics for the total effect model when that model is requested in the output using the **total** option.

If any of the variables used in the PROCESS command are the same as the column names PROCESS tries to use for the diagnostic statistics file (see the table on the prior page), the diagnostics file will not be created. In that case, change the duplicate variable name in the data to avoid the naming conflict and rerun the PROCESS command.

Diagnostics, Assumption Tests, and Casewise Influence

Whereas the **save=4** option saves a set of diagnostic statistics for each case in the analysis to a file, as discussed above, including **diagnose=1** in the PROCESS command generates a section of output for each regression equation containing information useful for testing assumptions and flagging influential cases. Excerpts of an example output are provided below along with an explanation of the information contained in the excerpt.

```
               Min.        Max.
fitted        2.7207      6.4109
residual     -4.8381      3.4266
t-resid      -4.6146      3.2464
```

This section contains the smallest (Min.) and largest (Max.) estimates of the outcome from the model (fitted), residual, and t-residual.

```
Shape of residuals
         Skewness    Kurtosis
Value     -.2708       .5165
se         .0856       .1711
```

This section contains the skew and kurtosis of the residuals along with an estimate of the standard error of skew and kurtosis. A ratio of "Value" relative to its standard error ("se") that exceeds two is diagnostic of a violation of the assumption of normality of the errors in estimation.

```
Bonferroni-corrected p for largest t-residual
     t-resid     p-value     casenum
     -4.6146       .0037    139.0000
```

This section provides a general test of model assumptions. Under the standard assumptions of regression, the t-residuals should follow a $t(df_{residual})$ distribution, each of which has a two-tailed p-value under the null hypothesis that a case's measurement on the outcome variable comes from a normal distribution around the regression line. Because this test is conducted for all cases in the analysis without any a priori expectations as to which cases might be responsible for an assumption violation, a Bonferroni correction to the two-tailed p-value is applied to correct for multiple tests. The output shows the case number in the data file with the smallest Bonferroni-corrected p-value for its t-residual. A p-value less than .05 (or whatever level of significance or alpha-level you desire for the test) leads to a rejection of the null hypothesis that all the regression assumptions are met. Note that as discussed in Darlington and Hayes (2017), this can be quite low in power relative to tests of specific assumptions. A small p-value is diagnostic of an assumption violation of some kind without identifying which assumption, but a large p-value doesn't necessarily mean all assumptions are met. In this example, case 139 is contributing most to the assumption violation. This does not mean this is the only potentially problematic case, however, as output only shows the Bonferroni-corrected p-value for the case's t-residual that is most distant from zero.

```
      Most influential observations
              casenum       dfbeta
constant     142.0000       .0365
negemot       35.0000      -.0054
posemot      139.0000      -.0060
sex          139.0000       .0118
age          139.0000      -.0004
ideology     139.0000       .0084
```

This section of output identifies, for each regression coefficient (as well as the constant), the case that when included in the analysis changes that regression coefficient the most and by how much (i.e., cases with the dfbeta for that regression coefficient most distant from zero). This output shows the regression coefficient for posemot, sex, age, and ideology change the most when case 139 is included. However, the coefficient for negemot changes the most with the inclusion of case 35, and the regression constant changes the most when case 142 is included. Another way of interpreting these dfbeta values is by multiplying them by -1, which then quantifies how much the regression coefficient changes when that case is deleted from the analysis. Cases with especially large dfbeta values (ignoring sign) relative to others can be said to be more or highly influential.

```
Variable tolerance and VIF
              Tol.        VIF
negemot      .8574     1.1664
posemot      .9743     1.0264
sex          .9498     1.0529
age          .9337     1.0710
ideology     .8372     1.1944
```

This section of output provides the tolerance (Tol.) and variance inflation factor (VIF) for each variable in the model. These are both sensitive to the strength of the association between a variable and all of the other variables on the right-hand side of the regression equation. Note that VIF is just the inverse of tolerance (i.e., VIF = 1/Tol.)

```
Breusch-Pagan test of heteroskedasticity
           Chi-sq         df          p
Normal    18.5281     5.0000      .0024
Robust    14.7868     5.0000      .0113
```

This section of output provides the Breusch-Pagan test of heteroskedasticity in two forms. The null hypothesis tested is that the homoskedasticity assumption is met. The row labeled "Normal" is the traditional test that assumes the errors in estimation (manifested as the residuals in the model) are normally distributed. As this test is sensitive to violations of this assumption, the test is the second row labelled "Robust" is more trustworthy when the errors in estimation are not normally distributed. As can be seen, both versions of this test suggest the errors in estimation are heteroskedastic, which is a violation of the assumption of homoskedasticity.

```
Indirect effect(s) of X on Y:
            Effect      BootSE     BootLLCI    BootULCI
TOTAL       -.0029       .0727      -.1449       .1424
resource    -.1175       .0464      -.2122      -.0302
workload     .1146       .0390       .0444       .1973

Cases with greatest influence on indirect effect(s):
            casenum      dfb_ie
resource    37.0000      -.0165
workload    37.0000      -.0131
```

This section of output, found in the summary section toward the bottom of a mediation analysis output, identifies the case(s) that as a result of inclusion in the analysis change the indirect effect(s) the most and by how much. The dfb_ie statistics are calculated as

$$\text{dfb\_ie} = IE_{with} - IE_{without}$$

where IE_{with} is the indirect effect with the case included and $IE_{without}$ is the indirect effect without the case included. Thus, the indirect effect if the case is excluded is

$$IE_{without} = IE_{with} - \text{dfb\_ie}$$

The dfb_ie statistics are thus like dfbetas for the regression coefficients but are for the indirect effects, which are products of regression coefficients. Cases with especially large dfb_ie values (ignoring sign) relative to others can be said to be more or highly influential. In this example, case 37, by its inclusion in the analysis, is changing the indirect effect through both resource and the indirect effect through workload the most. The negative values here mean that the inclusion of case 37 in the analysis moves the indirect effects through resource and workload to the left on the number line. So if case 37 were excluded from the analysis, the indirect effects through resource and workload would be -0.1010 and 0.1276, respectively. In models with more than one mediator, often one case will be more influential on one indirect effect but a different case will be more influential on another indirect effect.

The dfb_ie statistics are not provided for conditional indirect effects in a conditional process analysis.

Sums of Squares, Mean Squares, and Adjusted R^2
(Added in version 5.0)

In a regression analysis, total variation in the outcome variable is broken into regression and residual components. These sources of variation are the total, regression, and residual sums of squares. With the release of version 5.0, these sources of variation will be displayed in the output (under the column SS) along with corresponding degrees of freedom (df) and mean squares (MS) when **ssquares=1** is added to the PROCESS command. An example of the resulting output is below.

```
Model Summary
          R       R-sq    Adj R-sq          F          p       SEest
      .6232      .3883      .3845    102.7169      .0000      1.0673

              SS         df         MS
Regress   585.0188     5.0000   117.0038
Residual  921.5233   809.0000     1.1391
Total    1506.5421   814.0000     1.8508
```

The use of this option also adds adjusted R^2 to the model summary section of output, as above. To avoid the line of output being excessively wide, "df1" and "df2" for the F-ratio ordinarily in

the output when this option is not used is eliminated and can instead by found in the sources of variation table as the regression and residual degrees of freedom.

Note that **ssquares=1** is the default when estimating a model without a mediation or moderation component (i.e., model=0). To eliminate the printing of adjusted R^2 and the sums of squares in this case, add **ssquares=0** to the PROCESS command.

The **ssquares** option is not available in conjunction with errors-in-variables regression.

Crossvalidation Multiple Correlations/"Shrunken *R*"
(Added in version 5.0)

Adding **crossv=1** to the PROCESS command will produce the three estimates of "Shrunken R" discussed in Darlington in Hayes (2017, pp. 181-186). "LvOut1" and "LvOut2" are discussed on page 184 in that order, and "Browne" is discussed on page 185 (see equation 7.1).

```
    Shrunken R estimates
   Browne     LvOut1     LvOut2
    .6177      .6152      .6182
```

Scale-Free and Standardized Measures of Association
(Added in version 5.0)

Until recently, PROCESS had limited features for producing standardized regression weights or measures of association, and then only for mediation models. With the release of version 5.0, various scale-free and standardized measures of association are available as an option for every model that PROCESS can estimate. These are accessed by adding **stand=1** to the PROCESS command. When this is done, each regression equation will include an output such as below. The rows are the variables on the right-hand side of a model equation, and the columns are various measures of scale free, partially, and completely standardized weights for those variables in the model.

```
Scale-free and standardized measures of association
              r          sr          pr      standYX      standY      standX
negemot     .5777       .4585       .5058       .4952       .3240       .6737
posemot     .0430      -.0262      -.0334      -.0265      -.0197      -.0361
ideology   -.4183      -.2219      -.2730      -.2425      -.1604      -.3300
sex        -.0986      -.0036      -.0046      -.0037      -.0074      -.0050
age        -.0971      -.0152      -.0194      -.0157      -.0010      -.0214

           eta-sq    p_eta-sq        f-sq
negemot     .2103       .2558       .3437
posemot     .0007       .0011       .0011
ideology    .0493       .0745       .0805
sex         .0000       .0000       .0000
age         .0002       .0004       .0004
```

The statistics available include the simple or zero-order correlation (r) with the outcome (the variable on the left-hand side of the model equation), the semipartial (sr) and partial

correlation (pr), the completely standardized regression weight (StandYX), and two partially or "semi-" standardized regression weights (StandY and StandX). The two partially standardized weights differ with respect to which variable is standardized. For StandY, only the variable on the left-side of the model is standardized, and for StandX, only the variable on the right-hand side of the equation is standardized (both the left and right-hand variables are standardized for StandYX).

PROCESS also generates three commonly-used quantifications of partial association that are sometimes used as measures of "effect size." These are η^2 (eta-sq) and partial η^2 (p_eta-sq), and Cohen's f^2 (f-sq). Note that η^2 and partial η^2 are just the squares of the semipartial and partial correlations also provided by PROCESS. See Darlington and Hayes (2017) for a discussion of these.

For mediation models, the **stand** option continues to produce complete and partially standardized direct, indirect, and total effects as in prior releases. The completely standardized total, direct, and indirect effects are not provided when *X* is dichotomous, and the partially standardized effects are produced only when *X* is dichotomous or multicategorical.

The **stand** option is not available when the *Y* variable in the model is dichotomous or when used in conjunction with the **robustse** option or in errors-in-variables regression.

Bootstrap Estimate File Column Names
(added in version 5.0)

In earlier releases of PROCESS, save option 1 (**save=1**) produces a data file containing all the bootstrap estimates of every regression coefficient (plus the regression constant). The bootstrap estimates are in the columns of this data set and labeled "col1," "col2," "col3," etc., bootstrap samples are the rows, and a map in the PROCESS output provides a key for knowing which columns corresponds to which regression coefficients in the model equations. See Appendix A of *Introduction to Mediation, Moderation, and Conditional Process Analysis* for a discussion of this **save** option.

In the SAS and R versions of PROCESS version 5.0, the column names in this file now provide the information needed to know which columns contains which bootstrap estimates. The generic "col1," "col2," "col3" column names are no longer used, and the map has been eliminated in the output. In version 5.0, the column names are now in the format "left_right," where "left" is the variable name on the left side of the equation and "right" is the variable name on the right side of the equation.

For example, adding **save=1** to the SAS or R commands on page 93 of *Introduction to Mediation, Moderation, and Conditional Process Analysis* (3rd edition) generates a data file containing 5000 bootstrap estimates of the regression coefficients and constants from the models of `pmi` and `reaction`. The column names in the file are labeled

```
pmi_constant  pmi_cond  reaction_constant  reaction_cond  reaction_pmi
```

Thus, the first and second column contains the bootstrap estimates of the regression constant and regression coefficient for `cond`, respectively, in the model of `pmi`. The third, fourth, and fifth columns contain the regression constant and regression coefficients for `cond` and `pmi`, respectively, in the model of `reaction`. In the R version of PROCESS, the case (i.e., upper or lower) of the column labels will be consistent with the case of the variable names in the data frame being analyzed.

Note that the SPSS version of PROCESS 5.0 still uses the old column naming system and generates the column map in the output, just as in prior releases.

With this new column labeling format, some of the code printed in the 3rd edition of *Introduction to Mediation, Moderation, and Conditional Process Analysis* needs to be modified to work with PROCESS version 5. For the R version, the following changes are needed:

Line 9 of the code starting on page 446:

```
bootind<-boots$negtone_dysfunc*(boots$perform_negtone+boots$perform_int_1*
  modval[i])
```

Line 9 of the code on page 479:

```
bootind<-(boots$justify_frame+boots$justify_int_1*modval[i])*boots$donate_justify
```

Page 614:

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

For the SAS version, the second line of code on page 618 should now be:

```
data boots;set boots;ab=pmi_cond*reaction_pmi;diff=reaction_cond-ab;run;
```

Regression Analysis without Moderation or Mediation
(Added in version 5.0)

PROCESS version 4.1 added some limited features for the estimation of regression models without a moderation or mediation component. The release of version 5.0 both simplifies the command line and greatly expands the ability of PROCESS to estimate ordinary regression models and various extensions. Although formally designated as model 0, all that is needed in the PROCESS command is a single outcome variable after y= and at least one variable after x=, as below

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

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

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

The inclusion of "model=0" in the PROCESS command is optional, as PROCESS will understand what to do when no mediator or moderator is specified in the command line.

When estimating a regression model with no moderation or mediation component, the default setting for the **ssquares** option (described earlier) is 1, meaning that PROCESS will generate a sum of squares table as well as adjusted R^2 for the model. To eliminate this from the output, set the ssquares option to 0 (i.e., **ssquares=0**).

The variables on the right side of the regression equation need not all be entered following **x=**. An alternative option is to include at least one variable in the **x=** list and the remaining regressors as covariates following **cov=** as below.

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

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

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

The model and corresponding output will be identical to when all variables are listed following **x=.** Breaking the variables up in this fashion into **x=** and **cov=** sets can be useful when using the **settest** option discussed below to conduct a test of equality of fit of the models that include and exclude the variables in the **x=** list.

The **mcx** option can be used in model 0, in which case the multicategorical variable should be listed first in the **x=** list, and PROCESS will automatically create category codes as described in Appendix A of *Introduction to Mediation, Moderation, and Conditional Process Analysis*. Any other multicategorical variables on the right side of the equation would have to be already represented in the data with such codes. 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.

Many other features available in all models that PROCESS can estimate are available in model 0, including heteroskedasticity-consistent inference (using the **hc** option), bootstrapping (with the **modelbt** option), as week as cluster-robust standard errors and errors-in-variables regression, discussed later in this document. Some additional features described next are also available in model 0.

All Subsets Regression

The **subsets** option conducts all subsets regression. When this option's toggle is set to 1 (i.e., **subsets=1**), PROCESS generates output containing R^2 and adjusted R^2 for all possible models containing at least one regressor. The output takes the form of a table with the variable names at the top and models occupying the rows, as below. The table entries for each row contain zeros and ones under the variable name. A one in the column designates that the variable in that column is included in the model, and a zero means that variable is excluded. The table rows are sorted in ascending order of the adjusted R^2 for the model.

```
All subsets regression results
    negemot    posemot   ideology       sex        age      R-sq   Adj R-sq
      .0000     1.0000      .0000      .0000      .0000     .0019      .0006
      .0000      .0000      .0000      .0000     1.0000     .0094      .0082
      .0000      .0000      .0000     1.0000      .0000     .0097      .0085
      .0000     1.0000      .0000      .0000     1.0000     .0117      .0092
      .0000     1.0000      .0000     1.0000      .0000     .0123      .0098
      .0000      .0000      .0000     1.0000     1.0000     .0164      .0140
      .0000     1.0000      .0000     1.0000     1.0000     .0193      .0156
      .0000      .0000     1.0000      .0000      .0000     .1750      .1740
      .0000      .0000     1.0000      .0000     1.0000     .1751      .1730
      .0000     1.0000     1.0000      .0000      .0000     .1759      .1739
      .0000     1.0000     1.0000      .0000     1.0000     .1760      .1730
      .0000      .0000     1.0000     1.0000      .0000     .1769      .1748
      .0000      .0000     1.0000     1.0000     1.0000     .1769      .1738
      .0000     1.0000     1.0000     1.0000      .0000     .1781      .1750
      .0000     1.0000     1.0000     1.0000     1.0000     .1781      .1740
     1.0000      .0000      .0000      .0000      .0000     .3338      .3330
     1.0000      .0000      .0000     1.0000      .0000     .3348      .3331
     1.0000     1.0000      .0000      .0000      .0000     .3348      .3331
     1.0000     1.0000      .0000     1.0000      .0000     .3356      .3331
```

```
1.0000      .0000      .0000      .0000     1.0000      .3379      .3363
1.0000      .0000      .0000     1.0000     1.0000      .3384      .3359
1.0000     1.0000      .0000      .0000     1.0000      .3387      .3363
1.0000     1.0000      .0000     1.0000     1.0000      .3391      .3358
1.0000      .0000     1.0000      .0000      .0000      .3873      .3858
1.0000      .0000     1.0000     1.0000      .0000      .3874      .3851
1.0000      .0000     1.0000      .0000     1.0000      .3876      .3853
1.0000      .0000     1.0000     1.0000     1.0000      .3876      .3846
1.0000     1.0000     1.0000      .0000      .0000      .3881      .3858
1.0000     1.0000     1.0000     1.0000      .0000      .3881      .3851
1.0000     1.0000     1.0000      .0000     1.0000      .3883      .3853
1.0000     1.0000     1.0000     1.0000     1.0000      .3883      .3845
```

The number of possible models explodes as the number of regressors increases, and computing time and memory requirements increase accordingly. For this reason, all subsets regression is available only for models that include 15 or fewer regressors. All subsets regression is not available for models that specify moderation or mediation, models with a dichotomous Y, or when used in conjunction with the **cluster** option.

The **subsets** option is available only for model 0.

Dominance Analysis

Dominance analysis is a method for determining the relative importance of regressors in a model. Output from a dominance analysis is requested by specifying **dominate=1** in the PROCESS command line. PROCESS will display a dominance table, as below, discussed in section 8.4 of Darlington and Hayes (2017). The entries in the dominance table are the proportion of the possible subset models in which the variable in the row contributes more to prediction accuracy than the variable in the column. The diagonals of the dominance table are zero, and the cells symmetrically located around the diagonal usually sum to one.

```
Dominance matrix
          negemot  posemot ideology      sex      age
negemot      .000    1.000    1.000    1.000    1.000
posemot      .000     .000     .000     .500     .500
ideology     .000    1.000     .000    1.000    1.000
sex          .000     .500     .000     .000     .500
age          .000     .500     .000     .500     .000
```

Dominance analysis requires a lot of computations that require time and memory. Consequently, dominance analysis is available only for models with 15 or fewer regressors. In addition, dominance analysis is not available for models that specify moderation or mediation, models with a dichotomous *Y*, or when used in conjunction with the **cluster** option.

Spline Regression

PROCESS can conduct spline regression, discussed in section 12.3 of Darlington and Hayes (2017), wherein separate linear models relating one variable to the outcome are estimated between joints defined by user-specified values on the measurement scale. Spline regression is

conducted by using the **spline** option, followed by a list of joint values separated by commas (in SPSS), spaces (in SAS), or using the c() operator (in R). The joint values should be values on the measurement scale of the variable listed first in the **x=** list. For example, the command below

```
process y=govact/x=age negemot posemot ideology sex/spline=30,40,50.
```

```
%process(data=govact,y=govact,x=age negemot posemot ideology sex,spline=30 40 50)
```

```
process(data=glbwarm,y="govact",x=c("age","negemot","negemot","ideology","sex"),
    spline=c(30,40,50))
```

specify splines for the age variable, with the joints defined at ages 30, 40, and 50. Up to 10 joints may be specified when using the spline option. Joint locations must be listed in ascending order of value, with no ties, and all spline segments must contain at least two cases. The variable listed first following **x=** cannot be multicategorical, and so the spline option is incompatible with the **mcx** option. To get a test for the set of variables that define the spline function, use the **settest** option described next. The features of the spline option cannot be accessed through the PROCESS dialog box in SPSS.

The **spline** option can also be used in a mediation analysis without a moderation component (e.g., models 4, 6, 80, 81).

Inference for Sets of Regressors

PROCESS can provide a test that all of the regression coefficients for a subset of the regressors in the model are zero. In a regression model that includes any covariates listed following **cov=** PROCESS automatically provides a test that the partial regression coefficients for all the variables in the **x=** list are equal to zero. This is equivalent to a test of equality of fit of two models, one that includes only the variables in the **cov=** list and a second that includes variables in the **cov=** and the **x=** list. For example, the command

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

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

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

estimates `govact` from `negemot`, `posemot`, `ideology`, `sex`, and `age` while also providing a test of the null hypothesis that the regression coefficients for both negemot and posemot are both equal to zero. The difference in R^2 between the models with and without negemot and posemot is converted to an F-ratio for the test the null hypothesis, as below. PROCESS will also provide the difference in R^2 between the full model and the model excluding posemot and negemot. This will show up in the output next to the F test as "R2-chng."

```
Hypothesis test for variables in X set:
   R2-chng           F         df1         df2          p
     .2114    139.8217      2.0000    809.0000      .0000
```

When only one variable is listed for X and that variable is specified as multicategorical using the **mcx** option, the test is equivalent to a single factor analysis of covariance comparing the group means, adjusting for differences between the groups on all variables following **cov=**. When the variable listed following **y=** is dichotomous, the test printed by PROCESS will be in the form of a likelihood ratio test.

When using the **spline** option, all of the variables that define the spline function for the first variable following **x=** are included in the set, but this test is only conducted when adding **settest=1** to the PROCESS command.

This test for sets of variables is available only for model 0.

Cluster-Robust Standard Errors
(added in version 5.0)

PROCESS cannot do multilevel or "random effects" regression analysis. However, as of the release of PROCESS version 5.0, PROCESS can generate standard errors and conduct inferential tests that account for the within-cluster nonindependence that is characteristic of data with a hierarchical or nested structure. A variable defining the clusters must be specified following **cluster=** and cluster robust standard errors requested by adding **robustse=1** to the PROCESS command. For example, the command

```
process y=vote/x=ideology/m=conflict/model=4/cluster=country/robustse=1.
```

```
%process(data=civic,y=vote,x=ideology,m=conflict,cluster=country,robustse=1)
```

```
process(data=civic,y="vote",x="ideology",m="conflict",cluster="country",robustse=1)
```

identifies `country` as the clustering variable while requesting cluster robust standard errors for inference.

The algorithm for computation of cluster robust standard errors that is implemented in PROCESS is described in Cameron and Miller (2005, pp. 323-325) and mimics the vce(robust) option in Stata and PROC SURVEYREG in SAS. Like Stata, when using the **robustse** option, degrees of freedom for *t*-statistics, residual degrees of freedom (denominator) of *F*-tests, and for the construction of confidence intervals is the number of clusters *g* minus 1 (PROC SURVEYREG uses the number of clusters *g*).

F-tests for the model or subsets of variables in the model are conducted using the cluster-robust covariance matrix of the regression coefficients. However, if the number of clusters is too small relative to the number of variables (the numerator degrees of freedom) used for the test, PROCESS may not be able to conduct the test. In that case, *F*-ratios, degrees of freedom, and p-values for *F*-tests will be listed as "99999" in the output. These should not be interpreted. Consider this a warning that the number of clusters is far too small for reliable inference.

Cluster-robust inference is not available for models that include a dichotomous *Y*. As cluster-robust standard errors also account for heterogeneity of variance in the errors in estimation, the **hc** option cannot be used in conjunction with **robustse**.

Clustered and Stratified Bootstrapping
(added in version 5.0)

By default, PROCESS uses the casewise bootstrap when generating bootstrap estimates and confidence intervals. With the casewise bootstrap, each case in the data has equal probability of being included in a bootstrap sample. With the release of version 5.0, two new bootstrapping options are available. With both options, the user specifies a single clustering/ stratification variable in the data following "**cluster=**" in the PROCESS command that identifies in which cluster or stratum a case resides. A cluster or stratum is operationalized in the data as cases with a common numerical value on the clustering variable. In the rest of this discussion, the term "cluster" is used to refer to both clusters and strata, as the distinction between these often made in the sampling literature is not pertinent to the mechanics of the bootstrapping procedure described below.

With a cluster variable specified, one of two bootstrapping options is implemented depending on the argument following **clusboot=**. Let N be the sample size, k be the number of clusters, and n_j be the number of cases in cluster j. Adding **clusboot=1** to the PROCESS command implements a bootstrapping procedure such that each bootstrap sample will contain cases from all k clusters and with exactly n_j cases from cluster j. Within cluster j, cases in that cluster are randomly sampled with replacement and have the same probability of inclusion in a bootstrap sample as do other cases in cluster j. This procedure ensures that all k clusters are represented in every bootstrap sample, with each bootstrap sample containing exactly n_j cases from cluster j while also ensuring that each bootstrap sample has exactly N cases.

An example of when this bootstrapping procedure can be useful is when an analysis includes groups whose sample sizes were fixed by the researcher in advance. For example, suppose *X* is a multicategorical variable defining three groups and the investigator conducted the study so that 50 cases would be in each group. If 50 cases per group was fixed by design, it would make sense to restrict the bootstrap sampling to ensure that each bootstrap sample also contains 50 cases from each group. Treating the group variable as the cluster variable and using **clusboot** option 1 will guarantee that each bootstrap sample contains 50 cases from group 1, 50 cases from group 2, and 50 cases from group 3.

A second cluster bootstrapping option is available that randomly chooses *k* clusters with replacement and then includes *all* cases in each randomly selected cluster in the bootstrap sample. This option is requested by adding **clusboot=2** to the PROCESS command line. Unlike when using **clusboot** option 1, there is no guarantee that any cases from cluster *j* will appear in a bootstrap sample. Furthermore, a bootstrap sample may contain more or fewer than *N* cases, depending on the size of the clusters that were randomly selected for inclusion.

When using either of these cluster bootstrapping options, the computation of bootstrap confidence intervals (as well as bootstrap standard errors) is conducted in exactly the same manner as when using the casewise bootstrap. For a discussion of the mechanics of bootstrapping and the construction of bootstrap confidence intervals, see chapter 3 of *Introduction to Mediation, Moderation, and Conditional Process Analysis.*

Note that just as is true for the casewise bootstrap, when using the **clusboot** option, the standard errors and confidence intervals for each model in the output are still computed using ordinary OLS regression formulas unless the **robustse** or **hc** options are also used. Bootstrap results (confidence intervals, standard errors, and the mean of the bootstrap estimates) are displayed in the output only in those output columns with "Boot" in the label.

Errors-in-Variables Regression
(Added in version 5.0)

As of PROCESS version 5.0, errors-in-variables regression is available for the estimation of some models PROCESS can estimate. Errors-in-variables regression can be used to reduce or eliminate the bias in the estimation of regression coefficients as well as statistical inference when variables on the right side of a regression equation contain random measurement error. For the formulas used by PROCESS for estimation of errors-in-variables regression coefficients and various standard error options, see Appendix A of Hayes, Allison, and Alexander (in press).

Errors-in-variables regression is estimated by PROCESS whenever the **relx**, **relm**, or **relcov** options are used in the PROCESS command. These options allow the user to enter the assumed or estimated reliability of the variables on the right-hand side of regression equations. For variables in the **x=** list, provide the reliabilities for each variable in the list following option **relx=** and in the same order the variables are listed in the **x=** list. If no reliabilities are provided (that

is, if **relx** is not included in the command), PROCESS assumes the variables in the **x=** list are measured without error (i.e., with reliabilities equal to 1). In SPSS, the reliabilities in the list should be separated by a comma. In SAS, separate the reliabilities with a space. In R, separate the reliabilities with a comma and enclose the entire list in the c() operator. Unknown reliabilities would ordinarily be set to 1 (which in effect treats that variable as it would be treated in ordinary regression analysis assuming no measurement error). The same rules apply for **relm** and **relcov** when entering the reliabilities for the *M* variables in the **m=** list using **relm** and covariates in the **cov=** list using **relcov.** As with **relx**, when the **relm** or **relcov** options are not used, PROCESS assumes these variables contain no random measurement error.

For example, the PROCESS command below

```
process y=withdraw/x=estress/m=affect/cov=ese sex tenure/relx=0.72/relm=0.88
    /relcov=0.94,1,1.
```

```
%process(data=estress,y=withdraw,x=estress,m=affect,cov=ese sex tenure,relx=0.72,
    relm=0.88,relcov=0.94 1 1)
```

```
process(data=estress,y="withdraw",x="estress",m="affect",cov=c("ese","sex",
    "tenure"),relx=0.72,relm=0.88,relcov=c(0.94,1,1))
```

estimates the economic stress mediation analysis described in Chapter 4, section 4.2, of *Introduction to Mediation, Moderation, and Conditional Process Analysis*. The reliability estimates discussed below and in the PROCESS command above are provided in the original *Journal of Organizational Behavior* article. The reliability of the data for economic stress (`estress`), which is *X* in the model, is set to 0.72, and for business related depressed affect (`affect`), the mediator *M*, reliability is set to 0.88. The model includes three covariates. Sex and years in business (`tenure`) are assumed to be measured without any random measurement error and so the reliabilities are set to 1. But the reliability of entrepreneurial self-efficacy (`ese`) is set to its estimated value of 0.94.

The errors-in-variables option can also be useful to ascertain how vulnerable an analysis that assumes perfect reliability is to unaccounted-for measurement error. This can be accomplished by setting the reliabilities to plausible values or values lower than are likely and executing the analysis to see if the results substantively change. If not, then one can conclude that the results that assume perfect reliability are likely invulnerable to unaccounted-for measurement error.

Errors-in-variables regression makes an adjustment to the variances of variables on the right sides of regression equations prior to estimating the regression coefficients and standard errors. This adjustment can produce a variance-covariance matrix of those variables that could not actually exist in nature. This typically occurs when one or more of the assumed reliabilities

entered is small, though it can happen in other circumstances as well. When it does, PROCESS will not estimate the model and an error message is generating stating that one or more of the assumed reliabilities is too small to estimate the model. Note that an impossible data matrix after adjustment can also occur during bootstrapping. When this occurs, the bootstrap sample will be replaced. A warning message at the bottom of the output will indicate how many bootstrap samples were replaced during the bootstrapping procedure. If this number is large, interpret bootstrap results with caution. There are no guidelines or rules of thumb for what counts as "large" in this situation.

The adjustment to the data requires a different approach to estimating standard errors. These approaches, described in Hayes, Allison, and Alexander (in press), are available by including the **eiv** option in the PROCESS command. By default (if no **eiv** option is used or by adding **eiv=3** to the command), PROCESS implements a method that accounts for the sampling variance that results when adjusting for random measurement error and also includes a heteroskedasticity-consistent component based on the HC3 estimator discussed in Long and Ervin (2000). When all the reliabilities are set to 1, the regression coefficients will be the same as those estimated with ordinary least squares, and the standard errors will be equivalent to the heteroskedasticity-consistent HC3 standard error estimator.

Some alternative standard error estimators are also available. By including **eiv=0** in the PROCESS command along with estimated reliabilities, PROCESS uses the method implemented in Stata15 and later releases and discussed in StataCorp (2023). This method includes a heteroskedasticity-consistent component based on the HC0 estimator, also known as the "Huber-White" estimator. A third standard error option implemented in Stata prior to version 15 and discussed in Lockwood and McCaffrey (2020) is available using the **eiv=5** option in the PROCESS command. Like the default approach, this alternative approach adjusts the standard errors for unreliability but does not include a heteroscedasticity-consistent component. When all reliabilities are set to 1, the standard errors produced by this approach will be equivalent to regular OLS standard errors.

Errors-in-variables regression estimation is not available in moderation models (models 1, 2, or 3), in conditional process models that combine moderation and mediation, or in models with a dichotomous *Y*. Furthermore, the following options are not available for use in conjunction with errors-in-variables regression: **crossv**, **diagnose**, **dominate**, **effsize**, **hc**, **modelres**, **robustse**, **spline**, **stand**, **spline**, **ssquares**, **subsets**, and **save** option **4** (saving regression diagnostics).

## Automatic Plot Generation in PROCESS for R
(added in version 5.0)

When estimating a model with a moderation component, the **plot** option generates a table of estimates of the outcome variable (on the left side of an equation) from various combinations of focal predictor and moderator(s). In the R version of PROCESS version 5 or later, the plot option will now also automatically generate a visual depiction of the corresponding model. In

addition, when using the Johnson-Neyman technique (with option **jn=1**) in conjunction with the **plot** option will produce a Johnson-Neyman plot visually depicting the relationship between the moderator and the effect of the focal predictor along with confidence interval endpoints. Note that the production of the Johnson-Neyman plot is sensitive to the probing filter and will only be generated when the *p*-value for the corresponding interaction is below the filter setting (0.10 by default; see the documentation for more information about filter implemented with the **intprobe** option).

For example, executing the command below using the teams data from Chapter 11 of *Introduction to Mediation, Moderation, and Conditional Process Analysis*

```
process(data=teams,x="dysfunc",m="negtone",y="perform",w="negexp",plot=1,jn=1,
    model=14)
```

automatically generates the plots below:
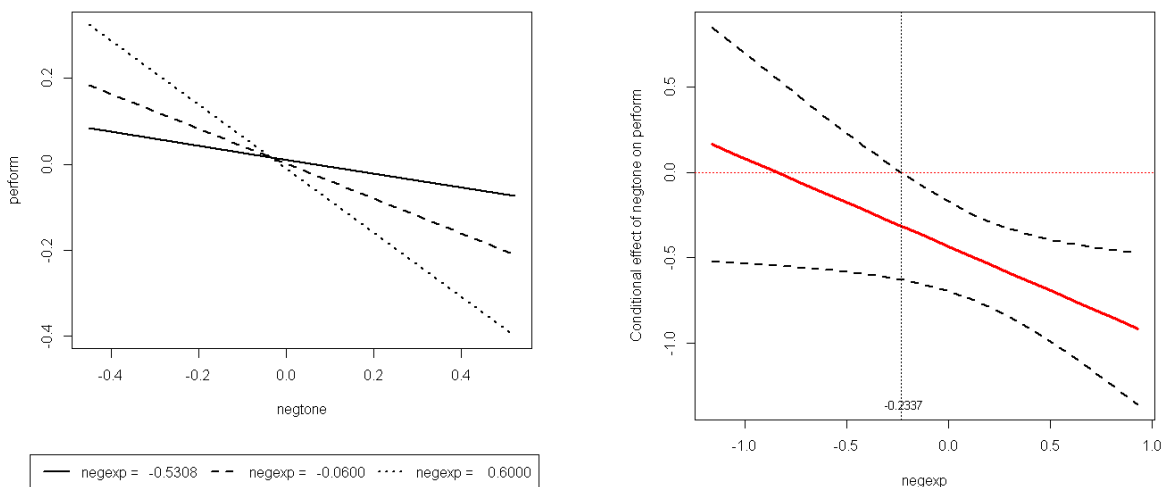
There is no way of modifying the axis labels, scaling of the axes, style or colour of lines, or specifying which variable is placed on the horizontal axis of the plots produced. To customize a plot, paste it into a graphics program and manually modify sections of the plot you wish to modify. For moderation models, PROCESS will always place the focal predictor on the horizontal axis and values of the moderator will determine the lines in the plot, unless the focal predictor is dichotomous or multicategorical, in which case the moderator will be placed on the horizontal axis and groups define the lines.

Note that a visual depiction of the model will only be generated for models or sections of a model with a single moderator. In other words, if more than one variable is specified as moderating a focal predictor's effect in a moderation-only model (i.e., models 2 or 3) or

sections of models (e.g., the effect of *X* on *M* in models 9-13, or the effect of *M* on *Y* in models 16-20), no plot is generated for that effect.

Model Number Defaults

(added in version 5.0)

Prior to the release of version 5.0, a PROCESS command must always contain a model number unless a custom model was being constructed with the use of the **bmatrix** option. With the release of version 5.0, PROCESS will assume model 0, model 1, or model 4 in some circumstances and depending on your PROCESS command, eliminating the need to specify a model number for these models.

If your PROCESS command does not specify a mediator variable *M* or moderator variable *Z* but does include a moderator variable *W*, it will assume you want to estimate a simple moderation model (model 1). Thus, a command such as below will work without a model number:

```
process y=justify/x=frame/w=skeptic.
```

```
%process(data=disaster,y=justify,x=frame,w=skeptic)
```

```
process(data=disaster,y="justify",x="frame",w="skeptic")
```

If your PROCESS command includes no moderator variables (i.e., no *W* or *Z* variable is specified) but at least one mediator variable *M* is specified, PROCESS will assume you want to estimate a simple or parallel multiple mediator model (model 4). If more than one variable is listed as *M*, PROCESS will estimate a parallel multiple mediator model. Thus, a command such as below will work without a model number:

```
process y=reaction/x=cond/m=import pmi.
```

```
%process(data=pmi,y=reaction,x=cond,m=import pmi)
```

```
process(data=pmi,y="reaction",x="negemot",m=c("import","pmi"))
```

If your PROCESS command includes no mediator (*M*) or moderator variables (*W* and *Z*), PROCESS will assume you are estimating a regular OLS or logistic regression model without a

moderation or mediation component (model 0). Thus, the command below works without a model number:

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

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

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

New Bootstrap Performance Information When Saving Output
(added in version 5.0)

As discussed in the documentation, **save** option 2 produces a data file containing the numerical information in the PROCESS output. With the release of version 5, and only when bootstrapping is used to generate any section of the output, the last row of this data file will contain information about the performance of the bootstrapping algorithm. The first column will contain the number of bootstrap samples that had to be replaced during the bootstrapping procedure. The second column contains how many samples were replaced due to a singularity in the bootstrap sample. The last column indicates how many samples were replaced as a result of not being able to apply the errors-in-variables computations on a bootstrap sample.

Output Formatting in SPSS
(added in version 5.0)

By default, the SPSS version of PROCESS produces output in text format. With the release of version 5, a new display option is available. By adding **display=tables** to the PROCESS command in SPSS, certain sections of the output will be in the form of table objects rather than text that can more easily be edited and resized in other documents if desired.

Note that with the release of SPSS v29, IBM changed the default output font for text output such as generated by PROCESS. The new default font will produce sloppy-looking output, with information not properly formatted and spaced. To return the format of the output to pre-v29 form, follow the directions below.

Under the "Edit" menu in SPSS, Choose "Options". The window below will open. Change the font under "Text Output" to "Courier New" and click the "Apply" button and then "OK" at the bottom of the window.
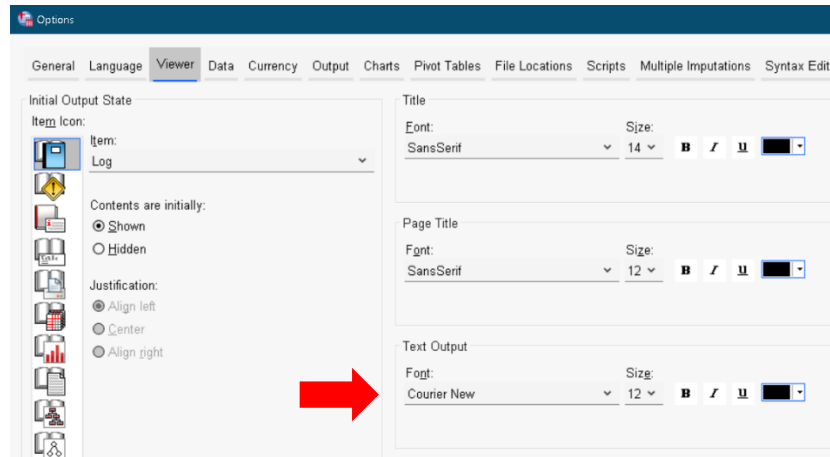
New PROCESS menu is an Extension file
(added in version 5.0)

With the release of version 5.0, SPSS users interested in installing the PROCESS dialog box to set up a model must install a custom dialog extension file (".spe") rather than a custom dialog builder file (".spv"). To do so, select "Extensions"-> "Install Local Extension Bundle.." and choose the .spe file that comes in the PROCESS v5 archive. After doing so, the PROCESS menu can be found under "Analyze"->"Regression". The custom dialog builder file (.spv) has been discontinued as of version 5 and is no longer available.

References

Cameron, A. C., & Miller, D. L. (2015). A practitioner's guide to cluster-robust inference. *Journal of Human Resources, 50*, 317-382.

Hayes, A. F., Allison, P. D., & Alexander, S. M. (in press). Errors-in-variables regression as a viable approach to mediation analysis with random error-tainted measurements: Estimation, effectiveness, and an easy-to-use implementation. *Behavior Research Methods*

Long, J. S., & Ervin, L. H. (2000). Using heteroskedasticity-consistent standard errors in the linear regression model. *American Statistician, 54*, 217-224.

Lockwood, J. R., & McCaffrey, D. F. (2020). Recommendations about estimating errors-in-variables regression in Stata. *The Stata Journal, 20*, 116-130.

StataCorp (2023). *Stata 18 Base Reference Manual*. College Station, TX: Stata Press.