MEMORE

```
MEMORE Y = depA depB /M = med1A med1B [med2A med2B...]
                                      [/MODEL = {mod}{1**}]
                                      [/MC = {mc}{0**}]
                                      [/BC = {bc}(0**)]
                                      [/NORMAL = {n}(0**)]
                                      [/CONF = {c}{95**}]
                                      [/SAMPLES = {sm}(5000**)]
                                      [/CONTRAST = {cn}(0**)]
                                      [/SAVE = {sv}{0**}]
                                      [/SEED = {sd}{random**}]
                                      [/SERIAL = {s}{0**}]
                                      [/DECIMALS = {dc}{F10.4**}]
                                      [/JN = {jn}{0**}]
                                      [/QUANTILE = {q}{0**}]
                                      [/PLOT = {p}{0**}]
                                      [/CENTER = {ce}{0**}
                                      [/MMODVAL1 = {mm1}]
                                      [/MMODVAL2 = {mm2}]
                                      [/MMODVAL3 = {mm3}].
```

Subcommands in brackets are optional
** Default if subcommand is omitted

Overview

`MEMORE` (pronounced like "memory") is a macro for estimating mediation and moderation models in two-condition repeated measures designs.

Model 1 estimates the total, direct, and indirect effects of *X* on *Y* through one or more mediators *M* in the two-condition or two-occasion within-subjects/repeated measures design. In a path-analytic form using OLS regression as illustrated in Montoya and Hayes (in press), it implements the method described by Judd, Kenny, and McClelland (2001) for single mediators while extending it as described in Montoya and Hayes (in press) to multiple mediators operating in parallel or serial. Along with an estimate of the indirect effect(s), `MEMORE` generates confidence intervals for inference about the indirect effect(s) using bootstrapping, Monte Carlo, or normal theory approaches. `MEMORE` also provides an option that conducts pairwise contrasts between specific indirect effects in models with multiple mediators.

Models 2 and 3 estimate regression coefficients and conditional effects of *X* on *Y* when this relationship is moderated by at least one variable, *M*, in two-condition or two-occasion within-subjects/repeated measures designs. It implements the methods described in Judd, McClelland, and Smith (1996) and Judd, Kenny, and McClelland (2001) for single moderator models. Multiple moderator models generalize these methods to multiple moderator models where the effect of *X* on *Y* is a linear function of each of the moderators $M_1 \ldots M_k$ (Model 2) and a multiplicative function of each of the moderators $M_1 \ldots M_k$ (Model 3). Along with estimates of all regression coefficients, `MEMORE` probes the interaction in both directions (the effect of *X* on *Y* conditional on *M* and the effect of *M* on *Y* conditional on *X*), provides results of the Johnson-Neyman procedure, and provides syntax for plotting conditional effects.

Preparation for Use

The MEMORE.sps file should be opened as a syntax file in SPSS. Once it has been opened, execute the entire file exactly as is. Do not modify the code at all. Once the program is executed, the

MEMORE.sps file window can be closed. Access to the `MEMORE` command is available after activation until quitting SPSS. The MEMORE.sps file must be loaded and reexecuted each time SPSS is opened. To install `MEMORE` permanently in SPSS, install the custom dialog version (see below).

Model Specification

Unlike in between-subjects mediation or moderation analysis, the data file for a within-subjects mediation analysis generally does not contain a column coding the *X* variable. As a result, there is no specification of the *X* variable in the `MEMORE` code. Rather, the *X* variable is represented in the data by two repeated measurements of the mediator(s) and dependent variable in the data file in the case of mediation and repeated measurements of the dependent variable in the case of moderation. It is the repeated measurements that appear in the `MEMORE` code. Moderators for models 2 and 3 are between-subjects variables and should not have repeated-measurements.

Mediation Example: For instance, *X* might be a manipulation of content in a stimulus, with each participant in the study receiving stimulus version A and stimulus version B. Each participant's measurement of the mediator and outcome is collected following exposure to each of the two stimuli. If in the data the mediator measurements are variables `medA` and `medB` following exposure to stimulus A and B, respectively, and the dependent variables measurements are variables `depA` and `depB` following exposure to stimulus A and B, then

```
MEMORE Y=depA depB/M=medA medB/model=1.
```

estimates the direct and total effects of independent variable *X* (the content manipulation) on dependent variable *Y* as well as the indirect effect of *X* on *Y* through mediator *M* and produces a bootstrap confidence interval for the indirect effect based on 5,000 bootstrap samples.

Various options are available in `MEMORE` to control the confidence level, number of samples used for inference, pairwise comparisons between specific indirect effects, and so forth. For example,

```
MEMORE Y=depA depB/M=medA medB/model=1/conf=99/mc=1/samples=10000/save=1.
```

estimates the effects of *X*, produces 99 percent confidence intervals for all model estimates, generates a Monte Carlo confidence interval for the indirect effect based on 10,000 samples, and saves the Monte Carlo estimates to a data file.

`MEMORE` constructs the difference between the two mediator measurements and the difference between the two dependent variable measurements, and these are modeled in accordance with the procedure described in Montoya and Hayes (in press) and Judd et al. (2001). `MEMORE` constructs the difference score as $M_A - M_B$ and $Y_A - Y_B$, where M_A and M_B are the mediator measurements following `M=` and Y_A and Y_B are the dependent variable measurements following `Y=`. The order these are listed in following `M=` and `Y=` matters for the sake of the construction of the difference, and the order must be consistent between the `M=` and `Y=` lists. For instance, if the dependent variable in condition A is listed first following `Y=`, then the mediator in condition A should also be listed first following `M=`. The top of the output will denote how the difference scores were constructed based on the `MEMORE` code submitted. Check this section of the output for consistency with your intentions before interpreting the results.

Moderator Example: *X* might be a manipulation of content in a stimulus, with each participant in the study receiving stimulus version A and stimulus version B. Each participant's measurement of the outcome is collected following exposure to each of the two stimuli. Additionally, a single measurement of the moderator of interest will be collected. If in the data the moderator measurements is the variable `mod` and the dependent variables measurements are variables `depA` and `depB` following exposure to stimulus A and B, then

```
MEMORE Y=depA depB/M=mod/model=2.
```

estimates allow the effect of *X* to vary linearly by *M* and estimate the effect of *X* and *M* on *Y*. MEMORE provides regression coefficient estimates, standard errors, and test statistics. Additionally, MEMORE provides a variety of tables which probe the effect of *X* on *Y* and the effect of *M* on *Y*.

Some of the options which are available in `MEMORE` for mediation are also available in moderation. For example, controlling the confidence level and the number of decimal places printed. Additional, options for moderation allow the user to decide what values will be probed, if the moderator should be centered before analysis, and if syntax for a plot should be printed. For example,

```
MEMORE Y=depA depB/M=mod/conf=99/mmodval1=5/quantiles=1/plot=1/center=1.
```

estimates the effects of *X* and *M* on *Y*, produces 99 percent confidence intervals for all model estimates, conducts all analyses using a mean centered *M*, probes the effect of *X* on *Y* at $M = 5$, probes the effect of *X* on *Y* at the 10th, 25th, 50th, 75th, and 90th quantiles of the observed distribution of *M*, and prints syntax for making plots of conditional effects. .

`MEMORE` constructs the difference between the outcome measurements and models this difference in accordance with the procedure described in Judd et al. (1996, 2001). `MEMORE` constructs the difference score $Y_A – Y_B$, where Y_A and Y_B are the dependent variable measurements following `Y=`. The order these are listed in following `Y=` matters for the sake of the construction of the difference. The top of the output will denote how the difference score was constructed based on the `MEMORE` code submitted. Check this section of the output for consistency with your intentions before interpreting the results.

Multiple Mediators

`MEMORE` can estimate specific indirect effects of *X* on *Y* in models with up to 10 mediators operating in parallel, or two in serial, as well as the total indirect effect of *X* on *Y* aggregated across all mediators. As mediators must come in pairs of measurements, in a model with *k* parallel mediators, there should be 2*k* variables provided in the `M=` list. The pairs should come in sequence of the mediators, with the occasion of measurement within each pair also preserved across pairs in the `M=` list. For instance, suppose three mediators M_1, M_2, and M_3 were each measured following stimulus A and stimulus B. In that case the `MEMORE` command for a parallel multiple mediator model would be

```
MEMORE Y=depA depB/M=med1A med1B med2A med2B med3A med3B.
```

where **`depA`** and **`depB`** are the measurement of the dependent variable following stimulus A and B, `med1A` and `med1B` are the measurements of mediator 1 following stimulus A and B, `med2A` and `med2B` are the measurements of mediator 2 following stimulus A and B, and `med3A` and `med3B` are the measurements of mediator 3 following stimulus A and B. Check the top of the output carefully to make sure `MEMORE` is constructing the difference scores as expected given the order in which the variables in `M=` are listed.

For a discussion of the parallel multiple mediator model, see Preacher and Hayes (2008) or Hayes (2013, Chapter 5).

For the serial mediator model, the order of the two pairs (MEMORE allows only two pairs for a serial model) in the `M=` list dictates the presumed direction of causal flow. The serial mediation model is specified by setting the `s` argument in the `serial` option to 1 (i.e., `serial=1)` in the MEMORE command. Thus, the MEMORE command below estimates a serial multiple mediator model with mediator 1 **(**`med1A` and `med1B`) causally prior to mediator 2 (`med2A` and `med2B`):

```
MEMORE Y=depA depB/M=med1A med1B med2A med2B/serial=1.
```

In the parallel and serial multiple mediator models, all direct and indirect effects are freely estimated. It is not possible to constrain a direct effect to zero using MEMORE.

Multiple Moderators

`MEMORE` can allow the effect of X on Y to depend on up to five moderators. The effect of X on Y can be either a linear function of all the moderator (Model 2) or a multiplicative function of all of the moderators (Model 3). For instance, suppose there were two moderators M_1 and M_2, the `MEMORE` command for a parallel multiple mediator model would be

```
MEMORE Y=depA depB/M=mod1 mod2/model=2.
```

where **`depA`** and **`depB`** are the measurement of the dependent variable following stimulus A and B, `mod1` is the measurement of moderator 1, and `mod2` is the measurement of moderator 2. In this case, the model of the difference of the dependent variables would only include `mod1` and `mod2`. If however, we had specified `model = 3` instead, then the product of `mod1` and `mod2` would also be included in the model of the differences in the outcome variables.

Inference for Indirect Effects

By default, `MEMORE` generates percentile bootstrap confidence interval for inference about the indirect effect based on 5,000 bootstrap samples. Bias corrected bootstrap and Monte Carlo confidence intervals are also available. To generate a Monte Carlo confidence interval instead of a bootstrap confidence interval, use the `MC` option, setting its argument to 1 (i.e., `MC=1`). To generate a bias corrected bootstrap confidence interval, use the BC option, setting its argument to 1 (i.e., `BC=1`). The lower and upper bounds of bootstrap confidence intervals are listed in the output under BootLLCI and BootULCI, respectively, whereas Monte Carlo confidence interval estimates are denoted MCLCCI and MCULCI. In a single run of `MEMORE`, a confidence interval is generated using only one method. The Monte Carlo confidence interval takes precedence when both the bias corrected confidence interval and Monte Carlo method are specified in a `MEMORE` command.

The standard error of the indirect effect is not required for confidence interval construction for the indirect effect when using bootstrapping or Monte Carlo methods. However, `MEMORE` does produce an estimate of the standard error of the indirect effect. This standard error is the standard deviation of the distribution of the bootstrap or Monte Carlo estimates. It appears in the output as BootSE (when using bootstrapping) or MCSE (when using the Monte Carlo method).

The `NORMAL` option generates a test of significance for the indirect effect using the Sobel test (Sobel, 1982). The Sobel test assumes that the sampling distribution of the indirect effect is normal, an assumption which has been shown to be inaccurate. To produce the Sobel test, set the `n` argument in the `NORMAL` option to 1 (i.e., `NORMAL = 1`). By default, `MEMORE` does not produce this test in the output.

Confidence Interval Width

The `c` argument in the `CONF` option specifies the desired confidence for confidence interval-based inference. The default is 95%. Confidence can be specified anywhere between 50 and 99.99% (e.g., `CONF=99` generates 99% confidence intervals). Note that the closer the confidence level requested gets to one, the more bootstrap or Monte Carlo samples are required in order to generate trustworthy confidence intervals for inference about indirect effects. If the number of bootstrap or Monte Carlo samples requested is too small to construct a confidence interval of the desired confidence, the program will not run and an error will appear in the "Analysis Notes and Warnings" section of the output.

Number of Samples for Bootstrap and Monte Carlo Inference

The `SAMPLES` option sets the number of samples used in the generation of bootstrap or Monte Carlo confidence intervals for inference about indirect effects. The `sm` argument defaults to 5000 and can be set to any integer between 1000 and infinity. Any number less than 1000, except zero, is ignored, and the default is implemented. If zero is specified, `MEMORE` generates a Monte Carlo confidence interval for indirect effects based on 5000 samples.

Covariates

There are no options available in `MEMORE` for the inclusion of covariates in the model. When a covariate is fixed across measurements of mediators and outcome (such as gender or some other stable individual difference) and it is assumed that the covariate's effect on each mediator repeatedly measured is the same, as is the covariate's effect on each outcome repeatedly measured, then the effect of the covariate on mediator and covariate differences becomes ignorable and thus the covariate can be excluded from the model.

Pairwise Contrasts Between Specific Indirect Effects

In models with more than one mediator, setting the `cn` argument in the `CONTRAST` option to one (i.e., `CONTRAST=1`) generates pairwise contrasts between all specific indirect effects, including bootstrap or Monte Carlo confidence intervals for inference. When there are only two repeated mediator variables in the model, the contrast between specific indirect effects is listed in the output as `(C1)`. With k repeated mediators in parallel, the $0.5k(k-1)$ possible pairwise contrasts are listed as `(C1)`, `(C2)`, `(C3)`, and so forth, and a key for interpreting which code corresponds to which contrast is provided.

Saving Bootstrap and Monte Carlo estimates

The `SAVE` subcommand generates a data file visible through the SPSS Data Viewer containing regression coefficients produced through bootstrap or Monte Carlo sampling. When bootstrapping, all model regression coefficients are saved. When using the Monte Carlo method, only the model coefficients that define the indirect effect(s) are saved. This file can be used for visualizing sampling distributions or the construction of custom hypothesis tests involving functions of regression coefficients. By default, this file is not created. To activate this option, specify `SAVE=1` in the `MEMORE` command. The file is not permanently saved to a storage device, so this file should be saved for future use if desired. Subsequent runs of `MEMORE` without first saving the file produced by a prior run will erase the old file in favor of the new file.

Seeding the Random Number Generator

Bootstrap and Monte Carlo confidence intervals require random resampling of the data or from theoretical distributions and thus will differ from run to run of `MEMORE` even when the data and model are the same. The `SEED` option can be used to seed the random number generator, thereby allowing for the replication of the output from run to run when analyzing the same data. By default `MEMORE` sets the seed randomly. The `sd` argument in the `SEED` command can be set to any positive integer that is less than or equal to 2,000,000. When this option is used, the random number seed specified is printed in the output for later reference.

Decimal Precision in Output

Output precision, in the form of number of decimal places of resolution, can be set with the `dc` argument in the `DECIMALS` command. The default for `dc` is F10.4, meaning 10 characters and four points to the right of the decimals place. Changing this to, for example, F8.2 will allocate eight characters with two to the right of the decimal point. See the *SPSS Syntax Reference Manual* for additional format options.

Johnson-Neyman Procedure

For moderation models with a continuous moderator, the Johnson-Neyman procedure will find the points along the observed range of the moderator (if they exist) for which the effect of *X* on *Y* is exactly statistically significant, based on a $\alpha =$ 1-`c`/100 level test. By setting the `jn` argument in the `JN` command to 1, the Johnson-Neyman procedure will be implemented and the points of interested printed along with a table of conditional effects which are useful for interpreting the Johnson-Neyman solutions.

Probing Conditional Effects

For moderation models, the default is the probe the effect of *X* on *Y* at three values of the moderator: the mean minus one standard deviation, the mean, and the mean plus one standard deviation. When the `q` argument in the `QUANTILE` command is set to 1, the probed values will instead be the 10^{th}, 25^{th}, 50^{th}, 75^{th}, and 90^{th} quantile of the observed distribution of *M*.
The three MMODVAL commands can be used to specify specific values of the moderators to probe at. The arguments for these commands must be numeric, and if there are multiple moderators, the list should be comma separated. The arguments for the MMODVAL commands should be the same length as the list of M variables. For example if the model has two moderators, then each of the MMODVAL arguments should be two numbers.

```
MEMORE Y=depA depB/M=mod1 mod2/model=2/mmodval1 = 3,1/mmodval2 = 2.3,4.
```

MEMORE will output the conditional effect of *X* on *Y* when `mod1`=3 and `mod2` = 1 as well as the conditional effect of *X* on *Y* when `mod1` = 2.3 and `mod2` = 1. MEMORE will check each of the subsequent `MMODVAL` commands. If there is no point specified for `MMODVAL1` then any information input in `MMODVAL2` will not be used. The first point of interest for probing should be specified as `MMODVAL1` and the second point should be specified as `MMODVAL2` etc.

Plot of Conditional Effects

When the `p` argument of the `PLOT` command is set to 1, `MEMORE` will print a chunk of SPSS syntax which can be copied and pasted into a syntax window. When run this syntax will generate three plots. Each plot has the moderator *M* on the *X* axis, and different outcomes on the *Y* axis. One plot has the difference between the two observations of the outcome variables, and the other two plots have each of the outcome variables separately.

Centering Moderator Variables

When the `ce` argument of the `CENTER` command is set to 1, `MEMORE` will mean center all moderator variables before conducting the regression analyses. Additionally, all conditional effects will be estimated using the new mean centered moderators, and the Johnson-Neyman procedure will be conducted on the mean centered version of the moderators.

MEMORE Custom Dialog

A Custom Dialog version of `MEMORE` can be permanently housed in the SPSS menus. To do so, install the MEMORE.spd (UI Dialog Builder) file by double clicking, right clicking, or opening and install it from within SPSS under the Utilities menu. Administrative access to the computer is required. This should install a new `MEMORE` option under the SPSS "Analyze→Regression" menu. Without administrative access, a local information technology specialist will be required for assistance in setting up administrative access to the computer and installing the dialog file.

Notes

- A case will be deleted from the analysis if missing on any of the variables in the model.
- All variable names must be 8 characters or fewer in length.
- Exactly two variables containing measurements of *Y* must be listed following `Y=`.
- For model 1, mediator measurements must be listed in sets of 2. Listing an odd number of variables in the `M=` list will produce an error.
- Do not use STRING formatted variables in any of your models. Doing so will produce errors. All variables should be in NUMERIC format.

References

Hayes, A. F. (2013). *Introduction to mediation, moderation, and conditional process analysis.* New York: The Guilford Press.

Judd, C. M., Kenny, D. A., & McClelland, G. H. (2001). Estimating and testing mediation and moderation in within-subjects designs. *Psychological Methods, 6*, 115-134.

Judd, C. M., McClelland, G. H., and Smith, E. R. (1996). Testing treatment by covariate interactions when treatment varies within subjects. *Psychological Methods, 1*(4), 366-378.

Montoya, A. K., & Hayes, A. F. (in press). Two condition within-participant statistical mediation analysis: A path-analytic framework. *Psychological Methods.*

Preacher, K. J., & Hayes, A. F. (2008). Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. *Behavior Research Methods, 40*, 879-891.