EMAuxiliary Reference Manual

Version: 0.9.0
Author: Stefan Schneider
Last updated: 2025-11-12

Getting started

Make sure you have completed the following setup steps:

1. **Install Blimp** (a free, standalone program):
   Download from https://www.appliedmissingdata.com/blimp
2. **Install the R interface for Blimp (rblimp)**: run this in your R environment:
   remotes::install_github("blimp-stats/rblimp")
3. **Import the EMAuxiliary** function by sourcing it directly from GitHub. Run this in R:
   source("https://raw.githubusercontent.com/schneids111/EMAuxiliary/main/EMAuxiliary.R")

EMAuxiliary function

```
results <- EMAuxiliary(
  y ~ x + m + x:m + (1 + x | id),        # focal model using lme4-style syntax
  data = dat,                            # dataset
  id = "id",                             # subject ID / clustering variable
  aux = c("aux1", "aux2", "aux3"),       # list of auxiliary variables to include
  center_group = x,                      # optional: within-person centering
  center_grand = m,                      # optional: grand-mean centering
  ordinal = "aux2",                      # optional: ordinal variable
  nominal = "aux3",                      # optional: nominal (unordered categorical) variable
  burn = 10000,                          # optional: number of burn-in iterations
  iter = 10000,                          # optional: number of post-burn-in iterations
)
```

Line-by-Line Walkthrough

Focal Model (y ~ x + m + x:m + (1 + x | id)):
This line specifies your multilevel model using lme4 syntax. Fixed effects and interactions can be written as x1 + x2 + x1:x2 or more compactly as x1 * x2. Random effects follow the usual lme4 format, such as (1 + x | id) for a model with a random intercept and random slope of x.

Dataset (data = dat):
Specifies the dataset.

Clustering Variable (id = "id"):
Specifies the subject identifier for clustering. EMAuxiliary supports two-level models (e.g., observations within persons) with any number of within- or between-person effects and interactions, including cross-level interactions. For three-level models, users must work directly in Blimp.
Auxiliary Variables (aux = c(...)):
Auxiliary variables are specified as additional multivariate dependent variables (DVs) in the multilevel models. EMAuxilairy automatically determines the level of variation for each

auxiliary variable, and regresses them on the corresponding components of the focal model using the following rules:

- **Within-person auxiliaries** (e.g., time of day) are regressed on the within-person components of the outcome and predictor variables.
- **Between-person auxiliaries** (e.g., age) are regressed on the latent between-person components of the outcome and predictor variables.
- **Mixed auxiliaries with variation on both levels** are regressed on both within- and between-person components, including possible cross-level interactions.

This process is handled automatically—no manual preprocessing of auxiliary variables is required.

Centering Predictors (center_group, center_grand):
Users can optionally specify group-mean and grand-mean centering for predictors:

- center_group = x centers x within person.
- center_grand = x centers the person-level mean of x around the grand mean.

Auxiliary variables do *not* need to be centered, as they enter the model as multilevel outcome (not predictor) variables.

Special Feature: Latent Mean Centering

The center_group option in *EMAuxiliary* activates latent within-person centering in Blimp. This feature improves the estimation of within- and between-person effects, especially in the presence of missing data.

When a predictor is listed under center_group, Blimp automatically decomposes it into:

- a latent within-person component, representing deviations from the individual's latent mean, and
- a latent between-person component, representing the individual's latent mean across all occasions.

Both components can then be included as predictors in the multilevel model. The between-person component is referenced as x.mean in the model equation. For example:

- center_group = x

combined with

- y ~ x + x.mean + (1 + x | id)

regresses the outcome y on the latent within-person part of x (x) and the latent between-person part of x (x.mean), estimating a random intercept for y and a random slope for the effect of x on y.

Variable Types (ordinal, nominal):
Non-continuous variables can be declared using the following arguments:

- ordinal = "aux2" — for binary or ordered categorical variables (e.g., gender, education).
- nominal = "aux3" — for unordered categorical variables (e.g., race, region).

Any variable—whether predictor, outcome, or auxiliary—can be specified as ordinal or nominal.
When an ordinal variable is specified, Blimp estimates probit regression models appropriate for binary or ordered outcomes.
When a nominal variable is specified, Blimp estimates multinomial logistic regression models.

Nominal variables are automatically dummy-coded, using the lowest category as the reference group. Note that nominal variables with many categories may substantially increase computational burden.

Bayesian Estimation Settings (burn, iter) and Model Convergence:
Blimp uses Bayesian estimation based on Markov Chain Monte Carlo (MCMC) sampling. Two key parameters control the sampling process:

- burn – Number of initial samples to discard (the *burn-in* period).
- iter – Number of samples to retain after burn-in.

Model convergence is evaluated using the Potential Scale Reduction (PSR) statistic. A PSR value greater than 1.05 suggests that the chains have not mixed well and that additional iterations may be needed. Increasing the burn-in period (e.g., doubling the default value of 10,000) often helps the model reach convergence. Models with larger numbers of auxiliary variables and more iterations will take longer processing times. It is worth the wait!

MCMC algorithms generally sample more efficiently when variables are on comparable scales—typically with variances around 1.0. Large differences in variable scaling can slow mixing and hinder convergence. *EMAuxiliary* automatically standardizes auxiliary variables to stabilize estimation, but large variances in the main model variables (predictors or outcomes) can still cause convergence problems. When detected, *EMAuxiliary* issues a warning and suggests rescaling those variables.

Viewing Output:

```
cat(results$blimp_code)        # show generated BLIMP program code
blimp_print_psr(results)       # PSR section only
blimp_print_focal(results)     #  outcome section for focal analysis model only
rblimp::output(results$fit)    # full output
```

After model estimation, EMAuxiliary provides several options for inspecting results at different levels of detail:

- cat(results$blimp_code)

Displays the Blimp input code that was automatically generated by EMAuxiliary.
Useful for verifying model structure, variable naming, and syntax before re-running the model manually in Blimp if desired.

- blimp_print_psr(results)

Prints the Potential Scale Reduction (PSR) summary section only, allowing quick checks of model convergence.

- blimp_print_focal(results)

Shows the focal model output—that is, the parameter estimates for the primary outcome(s) and predictors, excluding auxiliary-variable regressions.
Helpful when you want to review the main analysis results without the extended multivariate output.

- rblimp::output(results$fit)

Displays the full Blimp output, equivalent to viewing the complete .out file.
This includes PSR statistics, parameter summaries for both focal and auxiliary variables, and full model diagnostics.