

# Package ‘EMAtools’

June 22, 2026

**Type** Package

**Title** Data Management Tools for Real-Time Monitoring/Ecological  
Momentary Assessment Data

**Version** 0.1.6

**Description** Do data management functions common in real-time monitoring (also called: ecological momentary assessment, experience sampling, micro-longitudinal) data, including creating power curves for multilevel data, centering on participant means and merging event-level data into momentary data sets where you need the events to correspond to the nearest data point in the momentary data. For background on this data type see Shiffman, Stone and Hufford (2008) <[doi:10.1146/annurev.clinpsy.3.022806.091415](https://doi.org/10.1146/annurev.clinpsy.3.022806.091415)>, and on the centering methods see Enders and Tofghi (2007) <[doi:10.1037/1082-989X.12.2.121](https://doi.org/10.1037/1082-989X.12.2.121)>. This is VERY early release software, and more features will be added over time.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** ggplot2, lmerTest, sjstats (>= 0.10.2), anytime, plyr

**Suggests** lme4

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Evan Kleiman [aut, cre]

**Maintainer** Evan Kleiman <[evan.kleiman@rutgers.edu](mailto:evan.kleiman@rutgers.edu)>

**Repository** CRAN

**Date/Publication** 2026-06-22 16:00:20 UTC

## Contents

ema.powercurve . . . . .	2
eventmerge . . . . .	3
gcenter . . . . .	4
lme.dscore . . . . .	4
lm_slopes_compare . . . . .	5
ObsNumbs . . . . .	6
pcenter . . . . .	7
pmean . . . . .	7

---

ema.powercurve	<i>Create power curves for EMA data</i>
----------------	---

---

### Description

This allows you to estimate power to detect an effect at three standard effect sizes ( $d = 0.2, 0.5,$  and  $0.8$ ). It uses the `smpsize_lmm` function from `sjstats` to generate data for the curves and `ggplot2` to plot them.

### Usage

```
ema.powercurve(
  NumbPart,
  NumbResp,
  days,
  respday,
  Est_ICC = 0.05,
  COL.8 = "red",
  COL.5 = "blue",
  COL.2 = "green"
)
```

### Arguments

NumbPart	Total number of participants (i.e., level-2 unit)
NumbResp	Total max number of responses per participant (e.g., number of days * number of responses per day). You can either enter this OR enter number of days and number of responses per day manually. If all are entered, it will default to NumbResp.
days	Maximum number of days in study.
respday	Maximum number of responses per day.
Est_ICC	Estimated model ICC. Defaults to .05, but you should use a priori information from prior studies.
COL.8	Color of line for large ( $d=.8$ ) effect size. Default is red, but you can specify colors by name or by hex code (make sure to put colors in quotation marks).
COL.5	Color of line for medium ( $d=.5$ ) effect size. Default is blue, but you can specify colors by name or by hex code (make sure to put colors in quotation marks).
COL.2	Color of line for small ( $d=.2$ ) effect size. Default is green, but you can specify colors by name or by hex code (make sure to put colors in quotation marks).

### Value

A `ggplot` object that displays power curves at three effect sizes ( $d=.2,.5,.8$ ). You can use this like any other `ggplot` object (e.g., by adding other `ggplot` objects to it)

## Examples

```
ema.powercurve(NumbPart = 80, days = 30, respday = 3)
ema.powercurve(NumbPart = 80, NumbResp = 200)
ema.powercurve(NumbPart = 80, NumbResp = 200, COL.8 = "orange")
ema.powercurve(NumbPart = 80, NumbResp = 200, COL.8 = "orange",
               COL.5 = "#FF5733", COL.2 = "#8E44AD")
```

---

 eventmerge

---

*Merge Mobile EMA (mEMA) event-level data into momentary data*


---

## Description

This allows you to merge event-level data (e.g., yes/no to an event) into momentary data, placing each event on the most recent momentary datapoint at or before the event (for the same subject). The match is based on subject and timestamp, so it does not depend on the row order of either dataset or on how the mEMA KEY is constructed.

## Usage

```
eventmerge(MOMENTARY, EVENT, eventNAME = "eventYN")
```

## Arguments

MOMENTARY	a dataframe with momentary (i.e., level-1) data exported from mEMA. It must contain a "subject_id" column and a numeric "timestamp" column (other mEMA columns such as KEY and instance_key may be present and are passed through unchanged).
EVENT	a dataframe with event data (i.e., level-2). It must contain a "subject_id" column, a numeric "timestamp" column, and the event indicator in the LAST column (which can have any name). If a "local_date" column is present it is carried through as "date_event". Any other columns (e.g., respondent_id, survey_id, timezone_offset) are ignored.
eventNAME	variable name for your event in the final merged dataset (does not have to match last column in EVENT dataset, but can). Defaults to "eventYN".

## Value

A dataframe equal to MOMENTARY (same rows, same order) with three columns added: the event indicator (named by eventNAME, 0 where no event maps to that momentary point), "timestamp\_event" (the timestamp of the matched event, NA otherwise) and "date\_event" (the event's local\_date, NA otherwise). It has N rows = N rows in the momentary dataset. If more than one event maps to the same momentary datapoint, the most recent event is kept.

**Examples**

```
MOMENTARYdata <- data.frame(
  subject_id = c(1, 1, 1),
  timestamp = c(100, 200, 300))
EVENTdata <- data.frame(
  subject_id = c(1, 1),
  timestamp = c(150, 250),
  eventYN = c(1, 1))
newDATA <- eventmerge(MOMENTARYdata, EVENTdata, eventNAME = "eventYN")
```

---

gcenter

*Centering on grand-means*


---

**Description**

This function allows you to center on grand-means.

**Usage**

```
gcenter(var)
```

**Arguments**

var                    name of variable to be centered

**Value**

A column in your dataframe (with grand-mean centered data)

**Examples**

```
df <- data.frame(var = c(1, 3, 5, 9))
df$centeredVAR <- gcenter(df$var)
```

---

lme.dscore

*Calculate d scores from an lme4 or nlme object*


---

**Description**

This will calculate Cohen's D for each effect in an lme4 object.

**Usage**

```
lme.dscore(mod, data, type)
```

**Arguments**

mod	An lme4 or nlme object
data	The dataset the lme4 or nlme object was drawn from
type	Either "lme4" or "nlme"

**Value**

A table of d-scores.

**Note**

lme4 and nlme models will produce slightly different estimates. This is because when using type="lme4", the numerator DF will be calculated using the Satterthwaite approximations to degrees of freedom (via the lmerTest package), whereas nlme includes Kenward-Roger numerator degrees of freedom. If you have sufficient level-1 samples, the difference between models will be miniscule.

**Examples**

```
data(sleepstudy, package = "lme4")
model1 <- Reaction ~ Days + (1 | Subject)
lme.dscore(model1, data = sleepstudy, type = "lme4")
```

---

lm_slopes_compare	<i>Compare the slopes of two lme models</i>
-------------------	---

---

**Description**

This allows you to compare two lme4 models that have the same fixed predictors but differ in other ways (e.g., from different datasets, different random effects). It will produce a Z score a p-value for each effect.

**Usage**

```
lm_slopes_compare(VAR1, VAR2)
```

**Arguments**

VAR1	An lme4 object
VAR2	An lme4 object that has the same variables, in the same order as VAR1.

**Value**

Z-tests comparing slopes.

**Examples**

```

data(sleepstudy, package = "lme4")
model1 <- lmerTest::lmer(Reaction ~ Days + (1 | Subject), data = sleepstudy)
model2 <- lmerTest::lmer(Reaction ~ Days + (1 | Subject),
                        data = sleepstudy[sleepstudy$Days < 8, ])
lm_slopes_compare(model1, model2)

```

---

ObsNumbs

*Create observation numbers in your data*


---

**Description**

This function allows you number your observations by participant (P), by day (D) or by participant and day (PD) using a unix (numeric) timestamp

**Usage**

```
ObsNumbs(ID, TS, BY = c("P", "D", "PD"))
```

**Arguments**

ID	name of ID variable
TS	name unix (numeric) timestamp (if you have a date-time object, try converting it with <code>as.numeric(as.POSIXct())</code> )
BY	participant (P), by day (D), or by participant and day (PD). P will create a column of just 1 through n responses for each participant. D will create a day-level sequential value. PD will create an observation within each day. This might be most useful by first using the <code>BY=D</code> option, which will give you day number to use in tandem with observation #.

**Value**

A column in your dataframe (with person-centered data)

**Examples**

```

df <- data.frame(
  ID = c(1, 1, 1, 2, 2),
  TS = as.numeric(as.POSIXct(
    c("2020-01-01 09:00", "2020-01-01 17:00", "2020-01-02 09:00",
      "2020-01-01 10:00", "2020-01-02 10:00"), tz = "UTC")))
df$ObsNumb <- ObsNumbs(df$ID, df$TS, BY = "P")
df$DayNumb <- ObsNumbs(df$ID, df$TS, BY = "D")
df$ObsNumb_D <- ObsNumbs(df$ID, df$TS, BY = "PD")

```

---

pcenter                      *Centering on person-means*

---

**Description**

This function allows you to center on person-means (also called "centering within clusters")

**Usage**

```
pcenter(ID, var)
```

**Arguments**

ID	name of ID variable
var	name of variable to be centered

**Value**

A column in your dataframe (with person-centered data)

**Examples**

```
df <- data.frame(ID = c(1, 1, 2, 2), var = c(1, 3, 5, 9))  
df$centeredVAR <- pcenter(df$ID, df$var)
```

---

pmean                      *Centering on person-means*

---

**Description**

This function allows you calculate person-level means. This will create a level-2 variable that can be used in tandem with person-centered means. This is useful if you are interested in both the within-person and between-person effects.

**Usage**

```
pmean(ID, var)
```

**Arguments**

ID	name of ID variable
var	name of variable to be centered

**Value**

A column in your dataframe (with person-level means)

**Examples**

```
df <- data.frame(ID = c(1, 1, 2, 2), var = c(1, 3, 5, 9))  
df$personMEAN <- pmean(df$ID, df$var)
```

# Index

- \* **Cohen's**
    - lme.dscore, 4
  - \* **Compare**
    - lm\_slopes\_compare, 5
  - \* **D**
    - lme.dscore, 4
  - \* **Numbers**
    - ObsNumbs, 6
  - \* **Observation**
    - ObsNumbs, 6
  - \* **analysis**
    - ema.powercurve, 2
  - \* **centering**
    - gcenter, 4
    - pcenter, 7
    - pmean, 7
  - \* **merging**
    - eventmerge, 3
  - \* **power**
    - ema.powercurve, 2
  - \* **slopes**
    - lm\_slopes\_compare, 5
- ema.powercurve, 2
- eventmerge, 3
- gcenter, 4
- lm\_slopes\_compare, 5
- lme.dscore, 4
- ObsNumbs, 6
- pcenter, 7
- pmean, 7