Package 'EMAtools'

August 3, 2017

Type Package

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

Version 0.1.3

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. This is VERY early release software, and more features will be added over time.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports DataCombine,ggplot2,lmerTest, sjstats (>= 0.10.2)

RoxygenNote 6.0.1

NeedsCompilation no

Author Evan Kleiman [aut, cre]

Maintainer Evan Kleiman <ekleiman@fas.harvard.edu>

Repository CRAN

Date/Publication 2017-08-03 21:05:05 UTC

R topics documented:

`ema.powercurve` *Create power curves for EMA data*

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. Defualt 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. Defualt 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. Defualt 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).

Examples

```
## Not run: ema.powercurve(NumbPart=80,days=30,respday=3)
 ## Not run: ema.powercurve(NumbPart=80,NumbResp=200)
  ## Not run: ema.powercurve(NumbPart=80,NumbResp=200,COL.8="orange")
  ## Not run: ema.powercurve(NumbPart=80,NumbResp=200,COL.8="orange",COL.5="#FF5733",COL.3="#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 the event with the most recent momentary datapoint before the event

Usage

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

Arguments

MOMENTARY	a dataframe with momentary (i.e., level-1) data exported from mEMA, should have the following columns (all mEMA default names): KEY, instance_key, subject_id,timestamp
EVENT	a dataframe with event data (i.e., level-2) that should have the following columns (all mEMA default): instance_key subject_id respondent_id timestamp local_date survey_id timezone_offset as well as an "event" column in the last column of the dataframe (can be any name)
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 that contains event data merged into your momentary data. It will have N rows = N rows in the momentary dataset.

Examples

```
## Not run: 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

```
## Not run: data$centeredVAR<-gcenter(data$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 degress of freedom. If you have sufficent level-1 samples, the difference between models will be miniscule.

Examples

```
## Not run: model1<-lmer(DV~IV1+IV2+IV3+(1|subject),data=DATA_1)
 ## Not run: lme.dscore(model1,data=DATA_1,type="lme4")
```

lm_slopes_compare *Compare the slopes of two lme models*

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 comapring slopes.

Examples

```
## Not run: model1<-lmer(DV~IV1+IV2+IV3+(1|subject),data=DATA_1)
## Not run: model2<-lmer(DV~IV1+IV2+IV3+(1|subject),data=DATA_2)
 ## Not run: lm_slopes_compare(model1,model2)
```

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

```
## Not run: data$centeredVAR<-pcenter(data$ID,data$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

```
## Not run: data$centeredVAR<-pmean(data$ID,data$var)
```

Index