Package 'flps'

November 2, 2023

Type Package

Title Fully-Latent Principal Stratification

Version 1.0.0

Date 2023-10-14

Description
Simulation and analysis of Fully-Latent Principal Stratification (FLPS) with measurement models. Sales & Pane (2019). <doi:10.1214/18-AOAS1196>. This package is supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305D210036.

License GPL-3

Imports rstan, Rcpp (>= 1.0.8.3), methods, mirt, MASS, utils, stats, mvtnorm, ggplot2, glue

Suggests testthat (>= 3.0.0), knitr, rmarkdown, lavaan, data.table

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

VignetteBuilder knitr

Collate 'flps-package.r' 'load_rstan.r' 'make_FLPSdata.r' 'run_FLPS.r' 'priors.r' 'sim_gen_measurement.r' 'sim_gen_structure.r' 'sim_gen_standata.r' 'sim_make_data.r' 'z_utils.r' 'z_validate.r' 'plot_function.r' 'summary_function.r' 'data_desc.r' 'zzz.r'

URL https://sooyongl.github.io/flps/

NeedsCompilation no

Author Sooyong Lee [aut, cre], Adam Sales [aut], Hyeon-Ah Kang [aut], Tiffany Whittaker [aut]

Maintainer Sooyong Lee <sooyongl09@utexas.edu>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2023-11-02 22:30:02 UTC

R topics documented:

`flps-package` *Fully latent principal stratification*

Description

The FLPS package conducts Bayesian analysis for fully latent principal stratification via rstan.

Details

The 'flps' package.

Author(s)

Sooyong Lee <sooyongl09@gmail.com>

References

Sales, A. C., & Pane, J. F. (2019). The role of mastery learning in an intelligent tutoring system: Principal stratification on a latent variable. The Annals of Applied Statistics, 13(1), 420-443. Lee, S., Adam, S., Kang, H.-A., & Whittaker, T. A. (2022). Fully latent principal stratification: Combining ps with model-based measurement models. In The annual meeting of the psychometric society (pp. 287–298).

`binary` *binary.rda*

Description

A data set containing binary items information. Example data regenerated from CTA1

Usage

```
binary
```

Format

A data frame with variables:

schid School ID

id Student ID

sex 0 = boys; 1 = girls

race 0 = White; 1 = Others

pretest Pre test scores

stdscore Standardized scores

cm_sex Cluster-mean of sex

cm_race Cluster-mean of race

cm_pretest Cluster-mean of Pre test scores

cm_stdscore Cluster-mean of of Standardized scores

trt Treatment assignment; 0 = control, 1 = treatment

Y Outcome

q1 Binary item

q2 Binary item

q3 Binary item

q4 Binary item

q5 Binary item

q6 Binary item

q7 Binary item

q8 Binary item

q9 Binary item

q10 Binary item

q11 Binary item

q12 Binary item

q13 Binary item

q14 Binary item

q15 Binary item

q16 Binary item

q17 Binary item

q18 Binary item

q19 Binary item

q20 Binary item

Source

CTA1

Examples

```
data(binary)
summary(binary)
```

`continuous` *continuous.rda*

Description

A data set containing continuous items information. Example data regenerated from CTA1

Usage

```
continuous
```

Format

A data frame with variables:

schid School ID

id Student ID

sex 0 = boys; 1 = girls

race 0 = White; 1 = Others

pretest Pre test scores

stdscore Standardized scores

cm_sex Cluster-mean of sex

cm_race Cluster-mean of race

cm_pretest Cluster-mean of Pre test scores

cm_stdscore Cluster-mean of of Standardized scores

trt Treatment assignment; 0 = control, 1 = treatment

Y Outcome

q1 Continuous item

q2 Continuous item

q3 Continuous item

q4 Continuous item

q5 Continuous item

q6 Continuous item

q7 Continuous item

q8 Continuous item

q9 Continuous item

q10 Continuous item

q11 Continuous item

q12 Continuous item

q13 Continuous item

q14 Continuous item

q15 Continuous item

q16 Continuous item

q17 Continuous item

q18 Continuous item

q19 Continuous item

q20 Continuous item

Source

CTA1

Examples

```
data(continuous)
summary(continuous)
```

`flps_plot` *Make plots related to FLPS models*

Description

Make plots related to FLPS models

Usage

```
flps_plot(object, type = "latent")
```

Arguments

object	a flps object
type	a character indicating the type of plots

Value

A ggplot object that can be further customized using the **ggplot2** package.

---

graded *graded.rda*

Description

A data set containing graded response items information. Example data regenerated from CTA1

Usage

```
graded
```

Format

A data frame with variables:

schid School ID

id Student ID

sex 0 = boys; 1 = girls

race 0 = White; 1 = Others

pretest Pre test scores

stdscore Standardized scores

cm_sex Cluster-mean of sex

cm_race Cluster-mean of race

cm_pretest Cluster-mean of Pre test scores

cm_stdscore Cluster-mean of of Standardized scores

trt Treatment assignment; 0 = control, 1 = treatment

Y Outcome

q1 Graded response item

q2 Graded response item

q3 Graded response item

q4 Graded response item

q5 Graded response item

q6 Graded response item

q7 Graded response item
q8 Graded response item
q9 Graded response item
q10 Graded response item
q11 Graded response item
q12 Graded response item
q13 Graded response item
q14 Graded response item
q15 Graded response item
q16 Graded response item
q17 Graded response item
q18 Graded response item
q19 Graded response item
q20 Graded response item

Source

CTA1

Examples

```
data(graded)
summary(graded)
```

`importModel` *Import compiled Stan object*

Description

Import compiled Stan object

Usage

```
importModel(lv_type, multilevel = FALSE, lv_randomeffect = FALSE)
```

Arguments

`lv_type`	a character indicating the type of FLPS model.
`multilevel`	a logical indicating multilevel Stan model.
`lv_randomeffect`	A logical indicating whether to estimate random effects for latent variables.

Value

a Stan compiled stanmodel object generated by `modelBuilder`

makeInpData *Generate a matrix style data for simulation*

Description

makeInpData is a function for generating a data based on the given information.

Usage

```
makeInpData(
  N,
  R2Y,
  R2eta,
  omega,
  tau0,
  tau1,
  betaL,
  betaY,
  linear = TRUE,
  ydist = "n",
  lambda,
  nitem,
  nfac = 1,
  lvmodel,
  fcovmat,
  item.missing = TRUE,
  misspec = FALSE,
  cov.res = 0,
  relsize = 0.6
)
```

Arguments

N	a numeric indicating sample size.
R2Y	a numeric indicating predictive power of covariates.
R2eta	a numeric indicating Predictive power of latent variable
omega	a numeric indicating the size of effect of latent factor on the outcome.
tau0	a numeric indicating the size of difference in the outcome between the treatment and the control.
tau1	a numeric indicating the principal effect
betaL	a numeric vector indicating the effects of covariates on the latent factor
betaY	a numeric vector indicating the effects of covariates on the outcome
linear	a logical whether the relationship between the outcome and covariates is linear (default is TRUE).

ydist	a character indicating the outcome distribution (default is n).
lambda	a numeric indicating the mean of Worked problems/person. (extent to which covariates predict eta).
nitem	a numeric indicating the number of maximum measurement items given to students.
nfac	a numeric indicating the number of latent factors
lvmodel	a character specifying a type of latent variable model.
fcovmat	a matrix indicating the variance-covariance matrix of latent factors when nfac > 1
item.missing	a logical to make the measurement item data missing for the control group (default is TRUE).
misspec	a logical to allow cross-loadings across latent factors when nfac > 1 (default is FALSE).
cov.res	a logical to allow for residual correlations (only for CFA model) (default is 0).
relsize	a numeric indicating the degree to which the latent factor explain the variances of continuous items (only for CFA model) (default is 0.6).

Value

a list containing all the data related to population values and running FLPS.

Examples

```
sdat <- makeInpData(
N       = 200,  # sample size
R2Y     = 0.2,  # r^2 of outcome
R2eta   = 0.5,  # r^2 of eta by one covariates
omega   = 0.2,  # the effect of eta
tau0    = 0.13, # direct effect
tau1    = -0.06,# interaction effect between Z and eta
betaL   = 0.2,
betaY   = 0.4,
lambda  = 0.8,  # the proportion of administered items
nitem    = 10,   # the total number of items
nfac    = 1,    # the number of latent factors
lvmodel = '2pl' )
```

---

makeSimData *Generate Fully Latent Principal Stratification data for simulation*

Description

makeInpData is a function for generating a data based on the given information.

Usage

```
makeSimData(
  N,
  R2Y,
  R2eta,
  omega,
  tau0,
  tau1,
  betaL,
  betaY,
  linear = TRUE,
  ydist = "n",
  lambda,
  nitem,
  nfac,
  lvmodel,
  fcovmat,
  item.missing = TRUE,
  misspec = FALSE,
  cov.res = 0,
  relsize = 0.6
)
```

Arguments

N	a numeric indicating sample size.
R2Y	a numeric indicating predictive power of covariates.
R2eta	a numeric indicating Predictive power of latent variable
omega	a numeric indicating the size of effect of latent factor on the outcome.
tau0	a numeric indicating the size of difference in the outcome between the treatment and the control.
tau1	a numeric indicating the principal effect
betaL	a numeric vector indicating the effects of covariates on the latent factor
betaY	a numeric vector indicating the effects of covariates on the outcome
linear	a logical whether the relationship between the outcome and covariates is linear (default is TRUE).
ydist	a character indicating the outcome distribution (default is n).
lambda	a numeric indicating the mean of Worked problems/person. (extent to which covariates predict eta).
nitem	a numeric indicating the number of maximum measurement items given to students.
nfac	a numeric indicating the number of latent factors
lvmodel	a character specifying a type of latent variable model.

fcovmat	a matrix indicating the variance-covariance matrix of latent factors when nfac > 1
item.missing	a logical to make the measurement item data missing for the control group (default is TRUE).
misspec	a logical to allow cross-loadings across latent factors when nfac > 1 (default is FALSE).
cov.res	a logical to allow for residual correlations (only for CFA model) (default is 0).
relsize	a numeric indicating the degree to which the latent factor explain the variances of continuous items (only for CFA model) (default is 0.6).

Value

a list containing all the data related to population values and running FLPS.

Examples

```
sdat <- makeSimData(
N       = 200,  # sample size
R2Y     = 0.2,  # r^2 of outcome
R2eta   = 0.5,  # r^2 of eta by one covariates
omega   = 0.2,  # the effect of eta
tau0    = 0.13, # direct effect
tau1    = -0.06,# interaction effect between Z and eta
betaL   = 0.2,
betaY   = 0.4,
lambda  = 0.8,  # the proportion of administered items
nitem    = 10,   # the total number of items
nfac    = 1,    # the number of latent factors
lvmodel = '2pl' )
```

---

modelBuilder *Generate compiled Stan object to facilitate the analysis*

Description

Generate compiled Stan object to facilitate the analysis

Usage

```
modelBuilder(lv_type, multilevel = FALSE, lv_randomeffect = FALSE)
```

Arguments

lv_type	A character string specifying the type of FLPS model
multilevel	a logical indicating multilevel Stan model.
lv_randomeffect	A logical indicating whether to estimate random effects for latent variables.

Value

There's no return, but the compiled objects are saved in the package root directory.

`plot.flps` *Plot*

Description

Plot

Usage

```
## S3 method for class 'flps'
plot(x, type = NULL, pars = c("tau0", "tau1"), ...)
```

Arguments

`x`	an object of class `flps`
`type`	a string for the type of plot
`pars`	a character vector indicating the target parameters
`...`	additional options for future development

Value

A `ggplot` object that can be further customized using the **ggplot2** package.

`print.flps` *Print results*

Description

Print results

Usage

```
## S3 method for class 'flps'
print(x, ...)
```

Arguments

`x`	an object of class `flps`
`...`	additional options for future development

Value

Results of FLPS model are printed via the **rstan** package.

runFLPS *Conduct fully latent principal stratification*

Description

Conduct fully latent principal stratification

Usage

```
runFLPS(
  inp_data = NULL,
  compiled_stan = NULL,
  outcome = NULL,
  trt = NULL,
  covariate = NULL,
  lv_model = NULL,
  lv_type = NULL,
  multilevel = FALSE,
  lv_randomeffect = FALSE,
  priors_input = NULL,
  stan_options = list(),
  ...
)
```

Arguments

inp_data	A matrix or data frame containing the input data.
compiled_stan	An object of S4 class stanmodel produced by the modelBuilder function.
outcome	A character string specifying the outcome variable's name.
trt	A character string specifying the treatment or control group variable's name.
covariate	A character string specifying the covariate variable names.
lv_model	A description of the latent variable model using syntax akin to the **lavaan** package. Key operators include: • =~ : Denotes associations between factors and indicators (e.g., F1 =~ v1 + v2 + v3). All indicators associated with the corresponding factor should be written in the same line with +. • + : Specifies a series of indicators.
lv_type	A character string indicating the type of latent variable models.
multilevel	A logical indicating if a multilevel structure is present.
lv_randomeffect	A logical indicating whether to estimate random effects for latent variables.
priors_input	A list specifying the priors or defaults to N(0, 5) if not provided. Relevant parameters: tau0 (group difference), tau1 (principal effects), and omega (effect of latent factors on outcome). Ensure that the lengths of tau1 and omega match the number of factors. Examples:

- `list(tau0 = c(0, 1), tau1 = c(0.5, 1))` : Mean and variance for normal priors.
- `list(tau1 = list(c(0.5, 1), c(-0.4, 1)))` : For two factors.

`stan_options`	A list of options for [rstan::stan()], specified as 'name = value'.
`...`	Additional parameters for the latent variable models, such as `nclass = 2`.

Value

An object of class `flps` encompassing a `stanfit` object. Components include:

`call`	Function call with arguments.
`inp_data`	The input data frame provided.
`flps_model`	The Stan syntax used in [rstan::stan()].
`flps_data`	Data list used for [rstan::stan()].
`flps_fit`	Resulting `stanfit` object.
`time`	A numeric; Time taken for computation

See Also

[rstan::stan()]

Examples

```
inp_data <- flps::makeInpData(
  N       = 200,
  R2Y     = 0.2,
  R2eta   = 0.5,
  omega   = 0.2,
  tau0    = 0.23,
  tau1    = -0.16,
  betaL   = 0.1,
  betaY   = 0.2,
  lambda  = 0.8,
  nitem    = 10,
  nfac    = 1,
  lvmodel = 'rasch' )

res <- runFLPS(
   inp_data = inp_data,
   outcome = "Y",
   trt = "Z",
   covariate = c("X1"),
   lv_type = "rasch",
   lv_model = "F =~ v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8 + v9 + v10",
   stan_options = list(iter = 1000, warmup = 500, cores = 1, chains = 2)
   )
```

`summary.flps` *Summarize the results*

Description

Summarize the results

Usage

```
## S3 method for class 'flps'
summary(object, type = "all", ...)
```

Arguments

`object`	an object of class `flps`
`type`	a string for the part of FLPS model
`...`	additional options for future development

Value

Summary of FLPS model are printed via the **rstan** package.

Index