Package ‘BGGM’

February 6, 2020

Type Package
Title Bayesian Gaussian Graphical Models
Version 1.0.0
Date 2020-01-16
Description Fit Bayesian Gaussian graphical models. The methods are separated into two Bayesian approaches for inference: hypothesis testing and estimation. There are extensions for confirmatory hypothesis testing, comparing Gaussian graphical models, and node wise predictability. These methods were recently introduced in the Gaussian graphical model literature, including Williams (2019) <doi:10.31234/osf.io/x8dpr>, Williams and Mulder (2019) <doi:10.31234/osf.io/ypxd8>, Williams, Rast, Pericchi, and Mulder (2019) <doi:10.31234/osf.io/yt386>.

Depends R (>= 3.5.0)
License GPL-2
Imports mvnfast (>= 0.2.5), foreach (>= 1.4.7), doParallel (>= 1.0.15), ggplot2 (>= 3.2.1), mvtnorm (>= 1.0.11), stringr (>= 1.4.0), ggridges (>= 0.5.1), GGally (>= 1.4.0), pracma (>= 2.2.5), network (>= 1.15), bayesplot (>= 1.7.1), sna (>= 2.5), shiny (>= 1.4.0), reshape2 (>= 1.4.3), cowplot (>= 1.0.0), stats, parallel, Matrix, reshape, MASS
Suggests knitr, rmarkdown, dplyr
Encoding UTF-8
LazyData true
RoxygenNote 7.0.2
VignetteBuilder knitr
NeedsCompilation no
Author Donald Williams [aut, cre], Joris Mulder [aut]
Maintainer Donald Williams <drwilliams@ucdavis.edu>
Repository CRAN
Date/Publication 2020-02-06 16:20:22 UTC
R topics documented:

assess_predictability ............................................. 3
bayes_R2 ............................................................... 4
bfi ................................................................. 5
coef.estimate ........................................................... 7
confirm .............................................................. 8
convergence ........................................................... 9
csws ................................................................. 10
diagnostics .............................................................. 12
estimate ................................................................. 13
explore ................................................................. 14
fisher_r2z .............................................................. 16
fisher_z2r .............................................................. 16
fitted.estimate ......................................................... 17
ggm_compare_bf ....................................................... 18
ggm_compare_estimate ................................................ 19
ggm_compare_estimate.default ........................................ 19
ggm_compare_ppc.default ............................................... 21
iri ............................................................... 22
mae ................................................................. 24
map ................................................................. 25
mape ............................................................... 25
mse ................................................................. 26
plot.ggm_compare_ppc .................................................. 27
plot.metric ............................................................ 28
plot.select.estimate ................................................... 29
plot.select.explore .................................................... 31
plot.select.ggm_compare_bf ........................................... 32
plot.summary.estimate .................................................. 34
plot.summary.explore .................................................. 34
plot_adjacency .......................................................... 35
posterior_predict ........................................................ 36
pp_check.estimate ...................................................... 37
predict.estimate ......................................................... 38
print.coef.estimate ...................................................... 39
print.confirm ........................................................... 39
print.estimate .......................................................... 40
print.explore ........................................................... 40
print.fitted.estimate .................................................... 41
print.ggm_compare_bf .................................................... 41
print.ggm_compare_estimate ............................................ 42
print.ggm_compare_ppc .................................................. 42
print.map .............................................................. 43
print.metric ............................................................ 43
print.post.pred ........................................................ 44
print.predict.estimate ................................................... 44
print.select.estimate .................................................... 45
Assess Predictability

**Description**

Compare nodes within networks or between networks. Currently the only option available is Bayesian R2.
bayes_R2

Bayesian Variance Explained (R2)

Description

Compute Bayesian R2. In contrast to the functions `mse`, `mae`, etc., this can be used to compare predictability between nodes within a network or between networks. Also, only posterior predictive R2 is implemented.

Usage

```r
bayes_R2(object, cred = 0.95, iter = 1000, cores = 2, ...)
```
Arguments

- **object**: object of class `estimate`
- **cred**: credible interval width used for selecting the network
- **iter**: iterations used for computing R2
- **cores**: number of cores for parallel computing
- **...**: currently ignored

Value

object of classes `bayes_R2` and `metric`

Examples

```r
# data
Y <- subset(tas, gender == "M")[, -ncol(tas)]

# fit model
fit <- estimate(Y)

# bayes R2
r2 <- bayes_R2(fit, iter = 50)

# print summary
r2

# plot
plot(r2)
```

---

**bfi**

25 Personality items representing 5 factors

**Description**

This data set and the documentation was taken from the `psych` package. Further details can be found in the documentation of the `psych` package.

**Format**

A data frame with 25 variables and 2800 observations (but with missing data)
Details

- A1 Am indifferent to the feelings of others. (q_146)
- A2 Inquire about others’ well-being. (q_1162)
- A3 Know how to comfort others. (q_1206)
- A4 Love children. (q_1364)
- A5 Make people feel at ease. (q_1419)
- C1 Am exacting in my work. (q_124)
- C2 Continue until everything is perfect. (q_530)
- C3 Do things according to a plan. (q_619)
- C4 Do things in a half-way manner. (q_626)
- C5 Waste my time. (q_1949)
- E1 Don’t talk a lot. (q_712)
- E2 Find it difficult to approach others. (q_901)
- E3 Know how to captivate people. (q_1205)
- E4 Make friends easily. (q_1410)
- E5 Take charge. (q_1768)
- N1 Get angry easily. (q_952)
- N2 Get irritated easily. (q_974)
- N3 Have frequent mood swings. (q_1099)
- N4 Often feel blue. (q_1479)
- N5 Panic easily. (q_1505)
- o1 Am full of ideas. (q_128)
- o2 Avoid difficult reading material. (q_316)
- o3 Carry the conversation to a higher level. (q_492)
- o4 Spend time reflecting on things. (q_1738)
- o5 Will not probe deeply into a subject. (q_1964)

- gender Males = 1, Females =2
- education 1 = HS, 2 = finished HS, 3 = some college, 4 = college graduate 5 = graduate degree

References

Description

There is a direct correspondence between the covariance matrix and multiple regression. In the case of GGMs, it is possible to estimate the edge set with multiple regression (i.e., neighborhood selection). In *BGGM*, the precision matrix is first sampled from, and then each draw is converted to the corresponding coefficients and error variances. This results in a posterior distribution. This function can be used to perform Bayesian multiple regression.

Usage

```r
## S3 method for class 'estimate'
coef(object, node = 1, cred = 0.95, iter = 500, ...)
```

Arguments

- `object`: object of class `estimate` (analytic = F)
- `node`: which node to summarize (i.e., the outcome)
- `cred`: credible interval used in the summary output
- `iter`: number of samples used in the conversion.
- `...`: e.g., digits

Value

list of class `coef.estimate`:

- `list inv_2_beta`:
  - `betas` posterior samples for the regression coefficients
  - `sigma` posterior samples for sigma (residual sd)
  - `call match.call()`

- `data frame summary_inv_2_beta`:
  - summary of regression coefficients
  - `call match.call()`

Examples

```r
# p = 10
Y <- BGGM::bfi[,1:10]

# sample posterior
fit <- estimate(Y, iter = 5000)

# precision to regression coefficients(fit, node = 1, cred = 0.95)
```
confirm

Confirmatory Hypothesis Testing

Description

Traditionally, Gaussian graphical models are inherently exploratory. That is, automated model selection is performed. A key aspect of BGGM is the ability to extend inference beyond exploratory and to confirmatory hypothesis testing. This is accomplished by testing equality and/or inequality constraints for sets of edges (partial correlations).

Usage

```r
confirm(Y, hypothesis, prior_sd = 0.25, iter = 25000, cores = 2)
```

Arguments

- `Y`: data matrix (n by p).
- `hypothesis`: hypothesis (or hypotheses) to be tested
- `prior_sd`: hypothesized standard deviation of the prior distribution
- `iter`: posterior and prior samples. 25,000 is the default, as it results in a more stable Bayes factor than using, say, 5,000.
- `cores`: number of cores for parallel computing. The default is 2, but this can be adjusted

Value

list of class confirm:

- `BF_matrix`: matrix of Bayes factors for each hypothesis. Also includes the compliment
- `post_prob`: posterior hypothesis probabilities
- `hypotheses`: hypothesis
- `call`: match.call()
- `p`: number of variables
- `n`: number of observations
- `iter`: number of posterior samples
- `delta`: hyperparameter of matrix-F prior distribution (corresponds to prior_sd)
- `parcors_mat`: partial correlation matrix
- `returned_mats`: contrast matrices

Note

Currently inequality and equality restrictions can be tested. The former is an ordering the respective edge sizes, whereas the latter allows for testing whether certain edges are exactly the same.

see methods(class = "confirm")
Examples

```r
# p = 10
Y <- BGGM::bfi[,1:10]

# hypothesis
hypothesis <- c("1--2 > 1--3 > 1--4 > 1--5")

# test inequality constraint
test_order <- confirm(Y = Y, hypothesis = hypothesis,
                      prior_sd = 0.5, iter = 50000,
                      cores = 2)

# summary
summary(test_order)

# test hypothesized directions

# hypothesis
hypothesis <- c("(1--2, 1--3, 1--4) < 0 < (1--6)")

# test directions
test_directions <- confirm(Y = Y, hypothesis = hypothesis,
                           prior_sd = 0.5, iter = 50000,
                           cores = 2)

# summary
summary(test_directions)
```

---

**convergence**

*MCMC Convergence Plots*

**Description**

Monitor converge of the MCMC sampler

**Usage**

```r
convergence(x, type = "acf", param = "1--2")
```

**Arguments**

- `x`:
  - object of class `estimate` or `explore`
- `type`:
  - `acf` or `trace plot`
- `param`:
  - edge name(s) (e.g., "1--2" or c("1--2", "1--3"))
Value
ggplots

Examples

```r
# plot
Y <- bfi[,1:5]

# fit model
fit <- estimate(Y)

# convergence plots
convergence(fit, type = "trace")
```

---

**csws**  
*Contingencies of Self-Worth Scale (CSWS)*

---

**Description**

A dataset containing items from the Contingencies of Self-Worth Scale (CSWS) scale. There are 35 variables and 680 observations.

**Usage**

data("csws")

**Format**

A data frame with 35 variables and 680 observations (7 point Likert scale)

**Details**

- 1 When I think I look attractive, I feel good about myself
- 2 My self-worth is based on God’s love
- 3 I feel worthwhile when I perform better than others on a task or skill.
- 4 My self-esteem is unrelated to how I feel about the way my body looks.
- 5 Doing something I know is wrong makes me lose my self-respect
- 6 I don’t care if other people have a negative opinion about me.
- 7 Knowing that my family members love me makes me feel good about myself.
- 8 I feel worthwhile when I have God’s love.
- 9 I can’t respect myself if others don’t respect me.
- 10 My self-worth is not influenced by the quality of my relationships with my family members.
- 11 Whenever I follow my moral principles, my sense of self-respect gets a boost.
- 12 Knowing that I am better than others on a task raises my self-esteem.
• 13 My opinion about myself isn’t tied to how well I do in school.
• 14 I couldn’t respect myself if I didn’t live up to a moral code.
• 15 I don’t care what other people think of me.
• 16 When my family members are proud of me, my sense of self-worth increases.
• 17 My self-esteem is influenced by how attractive I think my face or facial features are.
• 18 My self-esteem would suffer if I didn’t have God’s love.
• 19 Doing well in school gives me a sense of self-respect.
• 20 Doing better than others gives me a sense of self-respect.
• 21 My sense of self-worth suffers whenever I think I don’t look good.
• 22 I feel better about myself when I know I’m doing well academically.
• 23 What others think of me has no effect on what I think about myself.
• 24 When I don’t feel loved by my family, my self-esteem goes down.
• 25 My self-worth is affected by how well I do when I am competing with others.
• 26 My self-esteem goes up when I feel that God loves me.
• 27 My self-esteem is influenced by my academic performance.
• 28 My self-esteem would suffer if I did something unethical.
• 29 It is important to my self-respect that I have a family that cares about me.
• 30 My self-esteem does not depend on whether or not I feel attractive.
• 31 When I think that I’m disobeying God, I feel bad about myself.
• 32 My self-worth is influenced by how well I do on competitive tasks.
• 33 I feel bad about myself whenever my academic performance is lacking.
• 34 My self-esteem depends on whether or not I follow my moral/ethical principles.
• 35 My self-esteem depends on the opinions others hold of me.
• gender "M" (male) or "F" (female)

**Note**

There are seven domains
FAMILY SUPPORT: items 7, 10, 16, 24, and 29.
COMPETITION: items 3, 12, 20, 25, and 32.
APPEARANCE: items 1, 4, 17, 21, and 30.
GOD’S LOVE: items 2, 8, 18, 26, and 31.
ACADEMIC COMPETENCE: items 13, 19, 22, 27, and 33.
VIRTUE: items 5, 11, 14, 28, and 34.
APPROVAL FROM OTHERS: items: 6, 9, 15, 23, and 35.

**References**

**Examples**

```r
data("csws")

# labels
csws_lables <- BGGM:::csws_labels
```

---

**diagnostics**

*Regression Diagnostic Plots for estimate Objects*

**Description**

GGMs have a direct correspondence to multiple regression. Hence this function provides diagnostic plots for inspecting the fitted regression models. This allows for visually inspecting assumptions of the model (e.g., normality of the residuals, etc.)

**Usage**

```r
diagnostics(object, iter = 500, ...)
```

**Arguments**

- `object` object of class `estimate`
- `iter` iterations used for computing residuals and fitted values
- `...` currently ignored

**Value**

list of `ggplot` objects (a plot for each node)

**Examples**

```r
# data
Y <- subset(tas, gender == "M")[-ncol(tas)]

# fit model
fit <- estimate(Y)

# diagnostic plot (iter = 10 as an example)
diagnostics(fit, iter = 10)
```
**estimate**

*GGMs with Credible Intervals or the Region of Practical Equivalence*

**Description**
Estimate the conditional (in)dependence structure with credible intervals or the region of practical equivalence. For the former, there is an analytic solution available, whereas for the latter, samples are efficiently drawn from the posterior distribution.

**Usage**

```r
estimate(Y, iter = 5000, analytic = FALSE, ...)
```

**Arguments**

- **Y**
  - data matrix (n by p).
- **iter**
  - number of posterior samples
- **analytic**
  - analytic solution. see notes for further details.
- **...**
  - not used

**Value**

list of class estimate:

- **analytic = TRUE:**
  - fit list of analytic solution estimates
    - `inv_mu` inverse covariance matrix (mean)
    - `inv_var` inverse covariance matrix (variance)
    - `partial` partial correlation matrix
  - `analytic` TRUE
  - `call` match.call()
  - `dat` data matrix
  - `p` number of variables

- **analytic = FALSE:**
  - `parcors_mat` partial correlation matrix
  - `inv_mat` inverse covariance matrix
  - `posterior samples` posterior samples for partial correlations and inverse covariance matrix
  - `p` number of variables
  - `dat` data matrix
  - `iter` number of posterior samples
  - `call` match.call()
  - `analytic` FALSE
Note

The default is to draws samples from the posterior distribution (analytic = FALSE). The samples are required for computing edge differences, Bayesian R2, etc. If the goal is to *only* determined the non-zero effects, this can be accomplished by setting analytic = TRUE. This is accomplished by estimating the posterior mean and variance, from which the credible intervals can computed. Note also sampling is very fast–i.e., less than 1 second with p = 25, n = 2500 and 5,000 samples. There is one function that makes use of the analytic solution. Namely, loocv computes node-wise leave-one-out error (also analytically).

see methods("estimate")

Examples

```r
# p = 20
Y <- BGGM::bfi[, 1:5]

# analytic approach (sample by setting analytic = FALSE)
fit_analytic <- estimate(Y, analytic = TRUE)

# select the graph (edge set E)
E <- select(fit_analytic, ci_width = 0.95)
```

**explore**

*GGMs with Exploratory Bayesian Hypothesis Testing*

Description

Learn the conditional (in)dependence structure with the Bayes factor computed from the matrix-F prior distribution. It is possible to test for only positive or negative edges, as well as two sided hypothesis testing (which is the customary approach). Further there is also an exhaustive option that provides the posterior probability of the null, greater than zero, and less than zero.

Usage

```r
explore(Y, prior_sd = 0.25, iter = 5000, cores = 2, ...)
```

Arguments

- `Y` data matrix (n by p).
- `prior_sd` hypothesized standard deviation of the prior distribution.
- `iter` number of posterior samples.
- `cores` number of cores for parallel computing. The default is 2, but this can be changed.
- `...` currently not used
Value

list of class explore:

- **parcors_mat** partial correlation matrix
- **parcors_sd** partial correlation standard deviations
- **samples** list of prior and posterior samples
  - **fisher_z_post** Fisher z transformed posterior distributions (partial correlations)
  - **pcor_post** partial correlation posterior distributions (not transformed)
  - **inv_cov_post** inverse covariance matrix posterior distribution
  - **pcor_prior** partial correlation prior distribution
  - **fisher_z_prior** Fisher z transformed prior distributions (partial correlations)
- **delta** hyperparameter of matrix-F prior distribution (corresponds to **prior_sd**)
- **iter** number of posterior and prior samples
- **dat** data matrix
- **call** match.call()
- **p** number of variables
- **cores** number of cores
- **edge** number of estimated edges

Note

After sampling from the posterior distribution, use **select** to determine the edge set and **plot** for visualizing the edge set. see methods(class = "explore")

Examples

```r
# p = 10
Y <- BGGM::bfi[,1:10]

# sample posterior
fit <- explore(Y, iter = 500)

# select E
E <- select(fit, BF_cut = 3)

# summarize
summary(E)

# non-zero edges
E$partials_non_zero

# adjacency matrix
E$Adj_10

# null adjacency matrix
E$Adj_01
```
**fisher_r2z**  
*Fisher Z Transformation*

**Description**
Tranform correlations to Fisher's Z

**Usage**
```r
fisher_r2z(r)
```

**Arguments**
- `r` correlation (can be a vector)

**Value**
Fisher Z transformed correlation(s)

**Examples**
```r
fisher_r2z(0.5)
```

---

**fisher_z2r**  
*Fisher Z Back Transformation*

**Description**
Back tranform Fisher's Z to correlations

**Usage**
```r
fisher_z2r(z)
```

**Arguments**
- `z` Fisher Z

**Value**
Correlation (s) (backtransformed)

**Examples**
```r
fisher_z2r(0.5)
```
Description

Fitted Values for estimate Objects

Usage

```r
## S3 method for class 'estimate'
fitted(object, iter = 500, cred = 0.95, summary = TRUE, ...)
```

Arguments

- **object**: object of class `estimate`
- **iter**: iters used to compute the residuals
- **cred**: credible interval used for summarizing
- **summary**: summarize the posterior samples (Default is `TRUE`).
- **...**: currently ignored

Value

3D array of dimensions n (observations), 4 (posterior summary), p (number of nodes)

Examples

```r
# data
Y <- subset(tas, gender == "M")[, -ncol(tas)]

# fit model
fit <- estimate(Y)

# fitted values
fitted(fit, iter = 25)
```
Compare GGMs with Bayesian Hypothesis Testing

Description

Compare GGMs with the Bayes factor. This method allows for assessing (relative) evidence for edge equality or edges differences across any number of groups. Further, confirmatory hypothesis testing can be used to test predictions or expectations regarding difference or similarities in different groups (e.g., male vs. female).

Usage

```r
ggm_compare_bf(..., prior_sd = 0.35, iter = 25000, cores = 2)
```

Arguments

- `...`: data matrices (n by p). Requires at least two.
- `prior_sd`: hypothesized standard deviation for the edges or partial correlations.
- `iter`: number of posterior samples.
- `cores`: number of cores for parallel computing. The default is 2, but this can be adjusted.

Value

- list of class `ggm_compare_bf`
  - `BF_01` Bayes factors in favor of the null hypothesis
  - `p` number of variables
  - `info` list of information about the data matrices
    - `dat` list containing the data matrices
    - `dat_info` sample size for each data matrix
    - `pairwise` matrix of pairwise combinations
  - `iter` number of posterior and prior samples
  - `call` match.call()
  - `delta` hyperparameter of matrix-F prior distribution (corresponds to `prior_sd`)
  - `groups` number of groups
  - `post_samps` matrix of posterior samples
  - `prior_samps` matrix of prior samples

Note

After fitting, use `select` to determine which partial correlations were different or the same (i.e., evidence for the null hypothesis of equality). This assumes `hypothesis = NULL`. If a hypothesis is tested, then use `summary` which provides information including the Bayes factors and posterior probabilities for each hypothesis.
Examples

# assume null is true
Y1 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))
Y2 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))
Y3 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))

bf_ggm <- ggm_compare_bf(Y1, Y2, Y3, prior_sd = .5, iter = 500, cores = 2)

---

ggm_compare_estimate  S3 estimate method

Description

S3 estimate method

Usage

  ggm_compare_estimate(...)

Arguments

  ... currently not used

See Also

  ggm_compare_estimate.default

---

ggm_compare_estimate.default

Compare Edges (Partial Correlations) Between GGMs with the Posterior Distribution

Description

Compare edges (partial correlations) that are estimated from groups to, say, detect differences or equivalences.

Usage

## Default S3 method:

  ggm_compare_estimate(..., iter = 5000)
Arguments

... data matrices. Requires at least two.
iter number of posterior samples

Value

A list of class `ggm_compare_estimate` containing:

- `pcor_diffs` partial correlation differences (posterior distribution)
- `p` number of variable
- `info` list containing information about each group (e.g., sample size, etc.)
- `iter` number of posterior samples
- `call` match.call

Note

The work flow for most functions in `BGGM` is to first fit the model and then select the graph (in this case the differences) with `select`.

See Also

`select.ggm_compare_estimate`

Examples

```r
# data
Y1 <- BGGM::bfi[1:500,1:5]
Y2 <- BGGM::bfi[501:1000, 1:5]

# fit model
fit <- ggm_compare_estimate(Y1, Y2)

# posterior summary of differences
summary(fit)

# select (threshold) with credible intervals
sel <- select(fit)

# summary
summary(sel)

# selected differences
sel$mat_pcor

# adjacency matrix
sel$mat_adj
```
**ggm_compare_ppc.default**

*Compare GGMs with the Posterior Predictive Distribution*

**Description**

Compare GGMs with the posterior predictive distribution. The method assume group equality, and the predictive check allows for testing whether that assumption should be modified—i.e., the GGMs are actually different. The current test statistic available is Kullback-Leibler divergence, which in this case, can be understood as a likelihood ratio for multivariate normal distributions. There are two options: (1) `global` and (2) `nodewise`. The former tests the entire GGM, whereas the latter allows for testing specific nodes (variables) in the model.

**Usage**

```r
ggm_compare_ppc(..., type = "global", iter = 5000, cores = 1)
```

**Arguments**

- `...`: data matrices \((n \times p)\). Requires at least two.
- `type`: `type = "global"` for testing the entire precision matrix. `type = "nodewise"` for testing each node (i.e., variable).
- `iter`: number of replicated data sets. Default is `iter = 1000`.
- `cores`: number of cores for parallel computing. The default is 2, but this can be adjusted.

**Value**

List of class `ggm_compare_ppc`:

- `pvalue`: posterior predictive p-values
- `obs_jsd`: observed symmetric KL divergence (Jensen-Shannon divergence)
- `predictive_risk`: list of predictive distributions
- `info`: list of information about the data matrices
  - `dat`: list containing the data matrices
  - `dat_info`: sample size for each data matrix
  - `pairwise`: matrix of pairwise combinations
- `names`: contrast names (e.g., `Yg1 vs Yg2`)
- `iter`: number of posterior samples
- `type`: "global"
- `call`: `match.call()`
### Note

This method is Bayesian, as it relies on the posterior predictive distribution. That said, there are clear parallels to frequentist testing—e.g., assuming group equality and critical regions. Most importantly, this method CANNOT provide evidence for the null hypothesis. Thus it can only reject the underlying assumption of group equality. For gaining (relative) evidence for the null hypothesis see...

```r
see methods(class = "ggm_compare_ppc")
```

### Examples

```r
# Assume null is true
Y1 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))
Y2 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))
Y3 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))

# global predictive check (iter = 50 for demonstrative purposes)
ggm_ppc <- ggm_compare_ppc(Y1, Y2, Y3,
                           type = "global", iter = 50)
summary(ggm_ppc)
plot(ggm_ppc)

# nodewise

ggm_ppc <- ggm_compare_ppc(Y1, Y2, Y3, type = "nodewise", iter = 50)
plot(ggm_ppc, log = TRUE)
```

---

**iri**  
**Interpersonal Reactivity Index (IRI)**

### Description

A dataset containing items from the Interpersonal Reactivity Index (IRI; an empathy measure). There are 28 variables and 1973 observations

### Usage

```r
data("iri")
```

### Format

A data frame with 28 variables and 1973 observations (5 point Likert scale)
Details

- I daydream and fantasize, with some regularity, about things that might happen to me.
- I often have tender, concerned feelings for people less fortunate than me.
- I sometimes find it difficult to see things from the "other guy's" point of view.
- Sometimes I don’t feel very sorry for other people when they are having problems.
- I really get involved with the feelings of the characters in a novel.
- In emergency situations, I feel apprehensive and ill-at-ease.
- I am usually objective when I watch a movie or play, and I don’t often get completely caught up in it.
- I try to look at everybody’s side of a disagreement before I make a decision.
- When I see someone being taken advantage of, I feel kind of protective towards them.
- I sometimes feel helpless when I am in the middle of a very emotional situation.
- I sometimes try to understand my friends better by imagining how things look from their perspective.
- Becoming extremely involved in a good book or movie is somewhat rare for me.
- When I see someone get hurt, I tend to remain calm.
- Other people’s misfortunes do not usually disturb me a great deal.
- If I’m sure I’m right about something, I don’t waste much time listening to other people’s arguments.
- After seeing a play or movie, I have felt as though I were one of the characters.
- Being in a tense emotional situation scares me.
- When I see someone being treated unfairly, I sometimes don’t feel very much pity for them.
- I am usually pretty effective in dealing with emergencies.
- I imagine how I would feel if the events in the story were happening to me.
- I believe that there are two sides to every question and try to look at them both.
- I would describe myself as a pretty soft-hearted person.
- I can very easily put myself in the place of a leading character.
- I tend to lose control during emergencies.
- When I’m upset at someone, I usually try to "put myself in his shoes" for a while.
- When I am reading an interesting story or novel, I imagine how I would feel if the events in the story were happening to me.
- When I see someone who badly needs help in an emergency, I go to pieces.
- Before criticizing somebody, I try to imagine how I would feel if I were in their place.
- gender "M" (male) or "F" (female)

Note

There are four domains

Fantasy: items 1, 5, 7, 12, 16, 23, 26
Perspective taking: items 3, 8, 11, 15, 21, 25, 28
Empathic concern: items 2, 4, 9, 14, 18, 20, 22
Personal distress: items 6, 10, 13, 17, 19, 24, 27,
References


Examples

data("iri")

# labels
iri_labels <- BGGM:::iri_labels

mae(object, ...)

Arguments

object object of class post.pred or predict.estimate

... currently ignored

Value

object of class metric

Examples

# data
Y <- subset(tas, gender == "M")[-ncol(tas)]

# fit model
fit <- estimate(Y)

# predict (note summary = FALSE)
pred <- predict(fit, iter = 50, summary = FALSE)

mae(pred)

Mean Absolute Error
**map**

*Maximum A Posteriori Precision Matrix*

**Description**
Maximum A Posteriori Precision Matrix

**Usage**

```r
map(Y)
```

**Arguments**

- `Y` data matrix (n by p)

**Value**
object of class `map`

**Examples**

```r
# p = 20
Y <- BGGM::bfi[, 1:5]

# map
map <- map(Y)
map
```

---

**mape**

*Mean Absolute Percentage Error*

**Description**
Mean Absolute Percentage Error

**Usage**

```r
mape(object, ...)
```

**Arguments**

- `object` object of class `post.pred` or `predict.estimate`
- `...` currently ignored

**Value**
object of class `metric`
Examples

# data
Y <- subset(tas, gender == "M")[,-ncol(tas)]

# fit model
fit <- estimate(Y)

# predict (note summary = FALSE)
pred <- predict(fit, iter = 50, summary = FALSE)

mape(pred)

mse

Mean Squared Error

Description

Compute mean squared error for either the observed data or future data. The former is computed by plugging in the observed y (the predicted node). The latter is computed from replicated data sets (posterior predictive y), which results in the posterior predictive mean squared error. Both provide a measure of uncertainty, as the error is computed from the posterior samples. However, the posterior predictive approach fully captures uncertainty.

Usage

mse(object, ...)

Arguments

object object of class post.pred or predict.estimate
...

currently ignored

Value

object of class metric

Examples

# data
Y <- subset(tas, gender == "M")[,-ncol(tas)]

# fit model
fit <- estimate(Y)

# predict (note summary = FALSE)
pred <- predict(fit, iter = 50, summary = FALSE)

mse(pred)
plot.ggm_compare_ppc  Plot ggm_compare_ppc Objects

Description
Plot ggridges for the GGM comparison with posterior predictive KL-divergence. The plots contain the predictive distribution, assuming group equality, as well as the observed KL-divergence. Further, the predictive distributions are conveniently colored to infer whether the null of group equality should be rejected. This is accomplished by having the critical region, corresponding to a desired 'significance' level, shaded in red. Thus, if the observed value is in the red region, this suggests the null hypothesis of group equality should be rejected.

Usage
## S3 method for class 'ggm_compare_ppc'
plot(
  x,
  critical = 0.05,
  col_noncritical = "#84e184A0",
  col_critical = "red",
  point_size = 2,
  log = FALSE,
  ...
)

Arguments
  x object of class ggm_compare_ppc
critical 'significance' level
col_noncritical fill color of the non critical region
col_critical fill color of the critical region (e.g., critical = 0.05)
point_size point size for the observed KL-divergence
log log transformation. useful for small values and skewed predictive distributions
... currently ignored

Value
one object of class ggplot when type = "global". One object for each pairwise contrast when type = "nodewise"

Note
This method is Bayesian, as it relies on the posterior predictive distribution. That said, there are clear parallels to frequentist testing-e.g., assuming group equality and critical regions. Most importantly, this method CANNOT provide evidence for the null hypothesis. Thus it can only reject the underlying assumption of group equality.
References

Williams, D. R., Rast, P., Pericchi, L. R., & Mulder, J. (2019). Comparing Gaussian Graphical Models with the Posterior Predictive Distribution and Bayesian Model Selection. pre print

Examples

# assume group equality
Y1 <- MASS::mvnrnorm(500, rep(0, 16), Sigma = diag(16))
Y2 <- MASS::mvnrnorm(500, rep(0, 16), Sigma = diag(16))
Y3 <- MASS::mvnrnorm(500, rep(0, 16), Sigma = diag(16))

# global
ggm_ppc <- ggm_compare_ppc(Y1, Y2, Y3, type = "global", iter = 50)

# plot
plot(ggm_ppc)

# nodewise
ggm_ppc <- ggm_compare_ppc(Y1, Y2, Y3, type = "nodewise", iter = 50)

plot(ggm_ppc, log = TRUE)
Arguments

- **x**: object of class `metric`
- **type**: "error_bar" or "ridgeline"
- **cred**: credible interval
- **alpha**: transparency of tail region (type = "ridgeline")
- **scale**: overlap of densities (type = "ridgeline")
- **width**: width of error bar end (type = "error_bar")
- **size**: point size (type = "error_bar")
- **color**: point (type = "error_bar") or tail region (type = "ridgeline") color
- **rope**: region of practical equivalence (only for Bayes R2 difference)
- ... currently ignored

Value

ggplot

Examples

```r
# data
Y <- subset(tas, gender == "M")[-ncol(tas)]

# fit model
fit <- estimate(Y)

# posterior predictions
pred <- posterior_predict(fit, iter = 500,
                          summary = FALSE)

# prediction error
error <- mse(pred)

# plot
plot(error)
```

---

**plot.select.estimate**  
*Plot select.estimate Network*

Description

Plot select.estimate Network
Usage

## S3 method for class 'select.estimate'
plot(
  x,
  layout = "circle",
  edge_colors = "classic",
  node_labels = NULL,
  node_labels_color = "black",
  node_groups = NULL,
  node_outer_size = 12,
  node_inner_size = 11,
  alpha = 0.5,
  txt_size = 8,
  edge_multiplier = 1,
  ...
)

Arguments

x  object of class select.estimate
layout  network layout (gplot.layout)
edge_colors  color theme for positive and negative edges
node_labels  node labels
node_labels_color  node labels color
node_groups  node group indicator
node_outer_size  node border size
node_inner_size  node size
alpha  edge transparency
txt_size  node text size
edge_multiplier  constant to change edge width (egde * edge_multiplier)
...  additional arguments (ggnet2)

Value

object of class ggplot

Examples

Y <- BGGM::bfi[, 1:20]

# analytic approach (sample by setting analytic = FALSE)
fit_analytic <- estimate(Y, analytic = TRUE)
# select the graph (edge set E)
E <- select(fit_analytic, ci_width = 0.95)

# plot
plt <- plot(E,
    node_labels = letters[1:20],
    node_labels_color = "white",
    node_groups = rep(c("1", "2", "3", "4"), each = 5),
    edge_colors = "classic",
    alpha = 0.5, palette = "Set2")

plot.select.explore  
Plot select.explore Network

Description
Plot select.explore Network

Usage
## S3 method for class 'select.explore'
plot(
  x,
  layout = "circle",
  edge_colors = "classic",
  node_labels = NULL,
  node_labels_color = "black",
  node_groups = NULL,
  node_outer_size = 12,
  node_inner_size = 11,
  alpha = 0.5,
  txt_size = 8,
  edge_multiplier = 1,
  ...
)

Arguments
x  object of class select.explore
layout  network layout (gplot.layout)
edge_colors  color theme for positive and negative edges
node_labels  node labels
node_labels_color  node labels color
node_groups  node group indicator
node_outer_size
  node border size
node_inner_size
  node size
alpha
  edge transparency
txt_size
  node text size
edge_multiplier
  constant to change edge width (edge * edge_multiplier)
...
  additional arguments (ggnet2)

Value

object of class ggplot

Examples

Y <- BGGM::bfi[1:500, 1:20]

# fit model
fit_explore <- explore(Y)

# select the graph (edge set E)
E <- select(fit_explore)

# plot
plt <- plot(E,
            node_labels = letters[1:20],
            node_labels_color = "white",
            node_groups = rep(c("1", "2", "3", "4"), each = 5),
            edge_colors = "classic", txt_size = 8,
            alpha = 0.5, palette = "Set2")

plot.select.ggm_compare_bf

Plot select.ggm_compare_bf Objects

Description

This function plots the selected graph when comparing GGMs with Bayesian hypothesis testing. There are two heatmap plots in total. The first includes edges for which there was evidence for a difference. The second includes edges for which there is evidence for the null hypothesis of equality. The tiles in the heatmap correspond to the Bayes factor
Usage

```r
## S3 method for class 'select.ggm_compare_bf'
plot(
  x,
  H0_low = "lightblue",
  H0_high = "purple",
  H1_low = "yellow",
  H1_high = "red",
  ...
)
```

Arguments

- `x`: object of class `select.ggm_compare_bf`
- `H0_low`: tile color for the smallest Bayes factors in the null hypothesis heatmap
- `H0_high`: tile color for the largest Bayes factors in the null hypothesis heatmap
- `H1_low`: tile color for the smallest Bayes factors in the alternative hypothesis heatmap
- `H1_high`: tile color for the largest Bayes factors in the alternative hypothesis heatmap
- `...`: currently ignored

Value

list containing two ggplot objects

Examples

```r
# group 1
Y1 <- MASS::mvrnorm(500, mu = rep(0,16),
  Sigma = BGGM::ptsd_cor2,
  empirical = FALSE)

# group 2
Y2 <- MASS::mvrnorm(500, mu = rep(0,16),
  Sigma = BGGM::ptsd_cor2,
  empirical = FALSE)

# fit model
fit <- ggm_compare_bf(Y1, Y2,
  prior_sd = 0.20,
  iter = 50,
  cores = 2)

# select E
E <- select(fit, BF_cut = 3)

# plot E
plot(E)
```
## plot.summary.estimate

### Description

Plot `summary.estimate`

### Usage

```r
## S3 method for class 'summary.estimate'
plot(x, color = "black", width = 0, ...)
```

### Arguments

- `x`: an object of class `summary.estimate`
- `color`: color of error bar
- `width`: width of error bar cap
- `...`: currently ignored

### Value

an object of class `ggplot`

## plot.summary.explore

### Description

Plot `summary.explore`

### Usage

```r
## S3 method for class 'summary.explore'
plot(x, ...)
```

### Arguments

- `x`: an object of class `summary.explore`
- `...`: currently ignored

### Value

an object of class `ggplot`
### Usage

```r
plot_adjacency(
  x, 
  layout = "circle", 
  node_labels = NULL, 
  node_labels_color = "black", 
  node_groups = NULL, 
  node_outer_size = 12, 
  node_inner_size = 11, 
  alpha = 0.5, 
  txt_size = 8, 
  ... 
)
```

### Arguments

- `x` : adjacency matrix
- `layout` : network layout (`gplot.layout`)
- `node_labels` : node labels
- `node_labels_color` : node labels color
- `node_groups` : node group indicator
- `node_outer_size` : node border size
- `node_inner_size` : node size
- `alpha` : edge transparency
- `txt_size` : node text size
- `...` : additional arguments (`ggnet2`)

### Value

Object of class `ggplot`
Examples

```r
# data
Y <- rsa[, -ncol(rsa)]

# fit model
fit_explore <- explore(Y, iter = 1000)

# select the graph (edge set E)
E <- select(fit_explore)

plot_adjacency(E$Adj_10, node_groups = BGGM:::rsa_labels)
```

---

### posterior_predict

**Posterior Predictive Distribution for estimate Objects**

#### Description

Posterior Predictive Distribution for estimate Objects

#### Usage

```r
posterior_predict(
  object, 
  iter = 500, 
  cred = 0.95, 
  newdata = NULL, 
  summary = TRUE, 
  ...
)
```

#### Arguments

- `object`: object of class `estimate`
- `iter`: iters used to compute the residuals
- `cred`: credible interval used for summarizing
- `newdata`: an optional data frame for obtaining predictions (e.g., on test data)
- `summary`: summarize the posterior samples (Default is TRUE). Setting it to FALSE can be used to then compute performance metrics.
- `...`: currently ignored

#### Value

- `summary = TRUE`: 3D array of dimensions n (observations), 4 (posterior summary), p (number of nodes).
- `summary = FALSE`: list containing predictions for each variable
Examples

```r
# data
Y <- subset(tas, gender == "M")[, -ncol(tas)]

# fit model
fit <- estimate(Y)

# predict
posterior_predict(fit, iter = 25)
```

Description

Perform posterior predictive checks with the help of the `bayesplot` package (with code taken from `brms`).

Usage

```r
## S3 method for class 'estimate'
pp_check(object, iter, type = "stat", ...)
```

Arguments

- `object`: object of class `estimate`
- `iter`: number of posterior samples used
- `type`: type of ppc plot. (`type = "xyz"` provides a list of supported types)
- `...`: currently ignored

Details

Further details are provided here `PPC`

Examples

```r
# data
Y <- bfi[, 1:5]

# fit model
fit <- estimate(Y)

pp_check(fit, iter = 50, type = "stat")
```
**predict.estimate**

**Prediction from an estimate Object**

**Description**

Prediction from an estimate Object

**Usage**

```r
## S3 method for class 'estimate'
predict(object, iter = 500, cred = 0.95, newdata = NULL, summary = TRUE, ...)
```

**Arguments**

- `object`: object of class `estimate`
- `iter`: iters used to compute the residuals
- `cred`: credible interval used for summarizing
- `newdata`: an optional data frame for obtaining predictions (e.g., on test data)
- `summary`: summarize the posterior samples (Default is `TRUE`). Setting it to `FALSE` can be used to then compute performance metrics.
- `...`: currently ignored

**Value**

- `summary = TRUE`: 3D array of dimensions n (observations), 4 (posterior summary), p (number of nodes).
- `summary = FALSE`: list containing predictions for each variable

**Examples**

```r
# data
Y <- subset(tas, gender == "M")[, -ncol(tas)]

# fit model
fit <- estimate(Y)

# predict
predict(fit, iter = 25)
```
print.coeff.estimate  

**Print method for coef.estimate objects**

### Description

Print method for coef.estimate objects

### Usage

```r
## S3 method for class 'coef.estimate'
print(x, ...)
```

### Arguments

- `x`  
  An object of class coef.estimate

- `...`  
  currently ignored

### See Also

coef.estimate

---

print.confirm  

**Print Method for confirm Objects**

### Description

Print Method for confirm Objects

### Usage

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

### Arguments

- `x`  
  object of class confirm

- `...`  
  currently ignored
print.estimate

Description

Print method for estimate.default objects

Usage

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

Arguments

x
An object of class estimate

... currently ignored

See Also

select.estimate

Examples

# data
Y <- BGGM::bfi[, 1:5]
# analytic approach (sample by setting analytic = FALSE)
fit <- estimate(Y, analytic = TRUE)
fit

print.explore

Description

Print method for explore objects

Usage

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

Arguments

x
An object of class explore

... currently ignored
print.fitted.estimate  Print Method for fitted.estimate Objects

Description
Print Method for fitted.estimate Objects

Usage
## S3 method for class 'fitted.estimate'
print(x, ...)

Arguments
x  object of class fitted.estimate
... currently ignored

print.ggm_compare_bf  Print method for ggm_compare_bf objects

Description
Print method for ggm_compare_bf objects

Usage
## S3 method for class 'ggm_compare_bf'
print(x, ...)

Arguments
x  An object of class ggm_compare_bf
... currently ignored

See Also
ggm_compare_bf
print.ggm_compare_estimate  

*Print method for ggm_compare_estimate objects*

### Description

Print method for `ggm_compare_estimate` objects

### Usage

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

### Arguments

- **x**: An object of class `ggm_compare_estimate`
- **...**: currently ignored

### See Also

`ggm_compare_estimate.default`

---

print.ggm_compare_ppc  

*Print method for ggm_compare_ppc.default objects*

### Description

Print method for `ggm_compare_ppc.default` objects

### Usage

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

### Arguments

- **x**: An object of class `ggm_compare_ppc`
- **...**: currently ignored
print.map  Print Method for map Objects

Description
Print Method for map Objects

Usage
## S3 method for class 'map'
print(x, ...)

Arguments
x          object of class map
...        currently ignored

print.metric  Print Method for metric Objects

Description
Print Method for metric Objects

Usage
## S3 method for class 'metric'
print(x, ...)

Arguments
x          object of class metric
...        currently ignored
### print.post.pred

#### Print Method for post.pred Objects

**Description**

Print Method for post.pred Objects

**Usage**

```r
## S3 method for class 'post.pred'
print(x, ...)
```

**Arguments**

- `x` object of class post.pred
- `...` currently ignored

---

### print.predict.estimate

#### Print Method for predict.estimate Objects

**Description**

Print Method for predict.estimate Objects

**Usage**

```r
## S3 method for class 'predict.estimate'
print(x, ...)
```

**Arguments**

- `x` object of class predict.estimate
- `...` currently ignored
print.select.estimate  Print method for select.estimate objects

Description
Print method for select.estimate objects

Usage
## S3 method for class 'select.estimate'
print(x, ...)

Arguments
x          An object of class select.estimate
...       currently ignored

See Also
select.estimate

print.select.explore  Print method for select.explore objects

Description
Print method for select.explore objects

Usage
## S3 method for class 'select.explore'
print(x, summarize = FALSE, ...)

Arguments
x          An object of class select.explore
summarize  summarize connectivity
...       currently ignored

See Also
select.explore
print.select.ggm_compare_bf

*Print method for* select.ggm_compare_bf *objects*

**Description**

Print method for select.ggm_compare_bf objects

**Usage**

```r
## S3 method for class 'select.ggm_compare_bf'
print(x, ...)
```

**Arguments**

- `x`: An object of class select.ggm_compare_bf
- `...`: currently ignored

**See Also**

- `select.ggm_compare_bf`

print.select.ggm_compare_estimate

*Print method for* select.ggm_compare_estimate *objects*

**Description**

Print method for select.ggm_compare_estimate objects

**Usage**

```r
## S3 method for class 'select.ggm_compare_estimate'
print(x, ...)
```

**Arguments**

- `x`: An object of class select.ggm_compare_estimate
- `...`: currently ignored

**See Also**

- `select.ggm_compare_estimate`
print.summary.estimate

Summary method for summary.estimate objects

Description
Summary method for summary.estimate objects

Usage

## S3 method for class 'summary.estimate'
print(x, ...)

Arguments

x An object of class summary.estimate
...
currently ignored

See Also

summary.estimate

print.summary.explore  Summary method for summary.explore objects

Description
Summary method for summary.explore objects

Usage

## S3 method for class 'summary.explore'
print(x, ...)

Arguments

x An object of class summary.explore
...
currently ignored

See Also

summary.explore
print.summary.ggm_compare_ppc

Summary method for summary.ggm_compare_ppc objects

Description
Summary method for summary.ggm_compare_ppc objects

Usage
## S3 method for class 'summary.ggm_compare_ppc'
print(x, ...)

Arguments
x               An object of class summary.ggm_compare_ppc
...             currently ignored

See Also
summary.ggm_compare_ppc

print.summary.ggm_estimate_bf

Summary method for summary.ggm_compare_bf objects

Description
Summary method for summary.ggm_compare_bf objects

Usage
## S3 method for class 'summary.ggm_compare_bf'
print(x, ...)

Arguments
x               An object of class summary.ggm_compare_bf
...             currently ignored

See Also
ggm_compare_bf
Summary method for `summary.ggm_compare_estimate` objects

### Description

Summary method for `summary.ggm_compare_estimate` objects

### Usage

```r
## S3 method for class 'summary.ggm_compare_estimate'
print(x, ...)  
```

### Arguments

- **x**: An object of class `summary.ggm_compare_estimate`
- **...**: currently ignored

### See Also

- `ggm_compare_estimate.default`

### Examples

```r
# data
Y1 <- BGGM::bfi[1:500, 1:5]
Y2 <- BGGM::bfi[501:1000, 1:5]

# fit model
fit <- ggm_compare_estimate(Y1, Y2)

# posterior summary of differences
summary(fit)
```

---

Prin Method for `summary.metric` Object

### Description

Print Method for `summary.metric` Object

### Usage

```r
## S3 method for class 'summary.metric'
print(x, digits = 2, ...)  
```

### Examples

```r
# data
Y1 <- BGGM::bfi[1:500, 1:5]
Y2 <- BGGM::bfi[501:1000, 1:5]

# fit model
fit <- ggm_compare_estimate(Y1, Y2)

# posterior summary of differences
summary(fit)
```
Arguments

x: object of class summary.metric

digits: digits used to round the values

... currently ignored

---

**ptsd**  
*post-traumatic stress disorder dataset*

Description

A dataset containing items that measure Post-traumatic stress disorder symptoms. There are 20 variables (p) and 221 observations (n)

Usage

data("ptsd")

Details

- Intrusive Thoughts
- Nightmares
- Flashbacks
- Emotional cue reactivity
- Psychological cue reactivity
- Avoidance of thoughts
- Avoidance of reminders
- Trauma-related amnesia
- Negative beliefs
- Negative trauma-related emotions
- Loss of interest
- Detachment
- Restricted affect
- Irritability/anger
- Self-destructive/reckless behavior
- Hypervigilance
- Exaggerated startle response
- Difficulty concentrating
- Sleep disturbance

References

Format A data frame with 221 rows and 20 variables
Description

A dataset containing items that measure Post-traumatic stress disorder symptoms. There are 16 variables in total. The correlation matrix was estimated from 526 individuals.

Format

A (correlation) matrix with 16 variables

Details

- Intrusive Thoughts
- Nightmares
- Flashbacks
- Physiological/psychological reactivity
- Avoidance of thoughts
- Avoidance of situations
- Amnesia
- Disinterest in activities
- Feeling detached
- Emotional numbing
- Foreshortened future
- Sleep problems
- Irritability
- Concentration problems
- Hypervigilance
- Startle response

References


Examples

data(ptsd_cor1)
Y <- MASS::mvrnorm(n = 965, mu = rep(0, 16),
      Sigma = ptsd_cor1, empirical = TRUE)
Description

A dataset containing items that measure Post-traumatic stress disorder symptoms. There are 16 variables in total. The correlation matrix was estimated from 365 individuals.

Format

A (correlation) matrix with 16 variables

Details

- Intrusive Thoughts
- Nightmares
- Flashbacks
- Physiological/psychological reactivity
- Avoidance of thoughts
- Avoidance of situations
- Amnesia
- Disinterest in activities
- Feeling detached
- Emotional numbing
- Foreshortened future
- Sleep problems
- Irritability
- Concentration problems
- Hypervigilance
- Startle response

References


Examples

data(ptsd_cor2)
Y <- MASS::mvrnorm(n = 365, mu = rep(0, 16),
                  Sigma = ptsd_cor2, empirical = TRUE)
Description

A dataset containing items that measure Post-traumatic stress disorder symptoms. There are 16 variables in total. The correlation matrix was estimated from 926 individuals.

Format

A (correlation) matrix with 16 variables

Details

- Intrusive Thoughts
- Nightmares
- Flashbacks
- Physiological/psychological reactivity
- Avoidance of thoughts
- Avoidance of situations
- Amnesia
- Disinterest in activities
- Feeling detached
- Emotional numbing
- Foreshortened future
- Sleep problems
- Irritability
- Concentration problems
- Hypervigilance
- Startle response

References


Examples

data(ptsd_cor3)
Y <- MASS::mvrnorm(n = 926, mu = rep(0, 16),
   Sigma = ptsd_cor3, empirical = TRUE)
Description
A dataset containing items that measure Post-traumatic stress disorder symptoms. There are 16 variables in total. The correlation matrix was estimated from 965 individuals.

Format
A (correlation) matrix with 16 variables

Details
- Intrusive Thoughts
- Nightmares
- Flashbacks
- Physiological/psychological reactivity
- Avoidance of thoughts
- Avoidance of situations
- Amnesia
- Disinterest in activities
- Feeling detached
- Emotional numbing
- Foreshortened future
- Sleep problems
- Irritability
- Concentration problems
- Hypervigilance
- Startle response

References

Examples
data(ptsd_cor4)
Y <- MASS::mvrnorm(n = 965, mu = rep(0, 16),
                  Sigma = ptsd_cor4, empirical = TRUE)
residuals.estimate

Residuals for estimate Objects

Description

Residuals for estimate Objects

Usage

## S3 method for class 'estimate'
residuals(object, iter = 500, cred = 0.95, summary = TRUE, ...)

Arguments

object object of class estimate
iter iters used to compute the residuals
cred credible interval used for summarizing
summary summarize the posterior samples (Default is TRUE).
... currently ignored

Value

3D array of dimensions n (observations), 4 (posterior summary), p (number of nodes)

Examples

# data
Y <- subset(tas, gender == "M")[-ncol(tas)]

# fit model
fit <- estimate(Y)

# diagnostic plot
residuals(fit, iter = 25)
### rmse

**Root Mean Squared Error**

**Description**

Root Mean Squared Error

**Usage**

\[
\text{rmse}(\text{object}, \ldots)
\]

**Arguments**

- `object`: object of class `post.pred` or `predict.estimate`
- `...`: currently ignored

**Value**

object of class `metric`

**Examples**

```r
# data
Y <- subset(tas, gender == "M")[, -ncol(tas)]

# fit model
fit <- estimate(Y)

# predict (note summary = FALSE)
pred <- predict(fit, iter = 50, summary = FALSE)

rmse(pred)
```

### rsa

**Resilience Scale of Adults (RSA)**

**Description**

A dataset containing items from the Resilience Scale of Adults (RSA). There are 33 items and 675 observations

**Usage**

```r
data("rsa")
```
Format

A data frame with 28 variables and 1973 observations (5 point Likert scale)

Details

- 1 My plans for the future are
- 2 When something unforeseen happens
- 3 My family understanding of what is important in life is
- 4 I feel that my future looks
- 5 My goals
- 6 I can discuss personal issues with
- 7 I feel
- 8 I enjoy being
- 9 Those who are good at encouraging are
- 10 The bonds among my friends
- 11 My personal problems
- 12 When a family member experiences a crisis/emergency
- 13 My family is characterised by
- 14 To be flexible in social settings
- 15 I get support from
- 16 In difficult periods my family
- 17 My judgements and decisions
- 18 New friendships are something
- 19 When needed, I have
- 20 I am at my best when I
- 21 Meeting new people is
- 22 When I am with others
- 23 When I start on new things/projects
- 24 Facing other people, our family acts
- 25 Belief in myself
- 26 For me, thinking of good topics of conversation is
- 27 My close friends/family members
- 28 I am good at
- 29 In my family, we like to
- 30 Rules and regular routines
- 31 In difficult periods I have a tendency to
- 32 My goals for the future are
- 33 Events in my life that I cannot influence
- gender "M" (male) or "F" (female)
Note

There are 6 domains
- Planned future: items 1, 4, 5, 32
- Perception of self: items 2, 11, 17, 25, 31, 33
- Family cohesion: items 3, 7, 13, 16, 24, 29
- Social resources: items 6, 9, 10, 12, 15, 19, 27
- Social Competence: items 8, 14, 18, 21, 22, 26,
- Structured style: items 23, 28, 30

References


Examples

data("rsa")

# labels
rsa_labels <- BGGM:::rsa_labels

---

select | S3 select method

Description

S3 select method

Usage

select(object, ...)

Arguments

object object of class estimate, explore, or ...
... not currently used

Value

select works with the following methods:

- select.estimate
- select.explore
- select.ggm_compare_estimate
Description

This allows for not only estimating the conditional dependence structure, that is non-zero edges, but also the conditional **in**dependence structure (evidence for no relation). For the latter, the region of practical equivalence must be specified.

Usage

```R
## S3 method for class 'estimate'
select(object, cred = 0.95, rope = NULL, prob = 0.95, ...)
```

Arguments

- `object`: object of class `estimate.default`
- `cred`: credible interval width used for the decision rule
- `rope`: region of practical equivalence
- `prob`: posterior probability (see notes)
- `...`: not currently used

Value

An object of class `select.estimate`:

- **analytic = TRUE**:
  - `partials_non_zero`: selected partial correlation matrix
  - `adjacency_non_zero`: adjacency matrix for the selected edges
  - `ci`: credible interval width
  - `analytic`: TRUE
  - `pcors_samples`: posterior samples

- **analytic = FALSE**:
  - `partials_non_zero`: selected partial correlation matrix
  - `adjacency_non_zero`: adjacency matrix for the selected edges
  - `pcor_sd`: posterior standard deviation
  - `ci`: credible interval width
  - `rope`: NULL

- **credible interval**:
  - `partials_non_zero`: selected partial correlation matrix (outside of the rope)
  - `adjacency_non_zero`: adjacency matrix for the selected edges (outside of the rope)
- `partials_zero` partials in the rope
- `adjacency_zero` adjacency in the rope
- `pcor_sd` posterior standard deviation
- `call` match.call()
- `rope` specified rope
- `in_rope` probability in the rope
- `pcors_samples` posterior samples

**Note**

The region of practical equivalence allows for assessing whether an edge is practically zero. In other words, conditional independence ($\rho = 0$). The argument `prob` is then the posterior probability that must be in (practically zero edges) and out (practically zero edges) of the rope.

**Examples**

```r
# Analytic = TRUE
# p = 5
Y <- BGGM::bfi[,1:5]

# analytic solution
fit_analytic <- estimate(Y, analytic = TRUE)

# select E
E <- select(fit_analytic, ci_width = 0.95)

# non-zero partial correlations
E$partials_non_zero

# adjacency matrix
E$adjacency_non_zero
```

---

**select.explore**

*Select Graphical Structure with the Bayes Factor*

**Description**

This allows for not only estimating the conditional dependence structure, that is non-zero edges, but also the conditional independence structure (evidence for no relation).

**Usage**

```r
## S3 method for class 'explore'
select(object, BF_cut = 3, alternative = "two.sided", ...)
```
Arguments

- **object**
  - object of class `explore.default`
- **BF_cut**
  - evidentiary threshold
- **alternative**
  - type of hypothesis (see notes)
- ...
  - currently not used

Value

list of class `select.explore`:

alternative = "two.sided":

- `partials_non_zero` selected partial correlation matrix
- `pcor_mat` partial correlation matrix (non set to zero)
- `pcor_sd` partial correlation standard deviations
- `Adj_10` adjacency matrix for the selected edges (in favor of the alternative)
- `Adj_01` adjacency matrix for the null hypothesis (conditional independence)
- `BF_10` Bayes factors for alternative
- `BF_01` Bayes factors for the null hypothesis
- `BF_cut` evidentiary threshold
- alternative "two.sided"
- code match.call()

alternative = "greater":

- `partials_positive` selected partial correlation matrix
- `pcor_mat` partial correlation matrix (none set to zero)
- `pcor_sd` partial correlation standard deviations
- `Adj_20` adjacency matrix for the selected edges (in favor of the alternative)
- `Adj_01` adjacency matrix for the null hypothesis (conditional independence)
- `BF_20` Bayes factors for alternative
- `BF_01` Bayes factors for the null hypothesis
- `BF_cut` evidentiary threshold
- alternative "greater"
- code match.call()

alternative = "less":

- `partials_negative` selected partial correlation matrix
- `pcor_mat` partial correlation matrix (none set to zero)
- `pcor_sd` partial correlation standard deviations
- `Adj_20` adjacency matrix for the selected edges (in favor of the alternative)
- `Adj_01` adjacency matrix for the null hypothesis (conditional independence)
• BF_20 Bayes factors for alternative
• BF_01 Bayes factors for the null hypothesis
• BF_cut evidentiary threshold
• alternative "less"
• code match.call()

alternative = "exhaustive"
• post_prob data.frame with posterior probabilities for each edge
• neg_mat adjacency matrix for negative edges
• post_mat adjacency matrix for positive edges
• null_mat adjacency matrix for zero (conditional independence)
• "alternative" "exhaustive"
• pcor_mat partial correlation matrix (non set to zero)
• pcor_sd partial correlation standard deviations
• code match.call()
• prob hyp_prob

Note

The alternative can be either greater, less, two.sided, or exhaustive. The argument hyp_prob is used only when alternative = hypothesis. greater and less test directional hypotheses, and thus, the graphical structure will only included edges in that direction (i.e., positive or negative). two.sided is the customary approach, and test for the presence or absence of an edge. exhaustive tests negative vs. positive vs. zero. Here hyp_prob is the posterior probability threshold for the respective hypotheses.

Examples

```r
# p = 10
Y <- BGGM::bfi[,1:10]

# sample posterior
fit <- explore(Y, iter = 5000)

# select E
E <- select(fit, BF_cut = 3)

# summarize
summary(E)

# non-zero edges
E$partials_non_zero

# adjacency matrix
E$Adj_10
```
# null adjacency matrix
E$Adj_01

select.ggm_compare_bf  Select Graphical Structure with the Bayes Factor

Description

Compare GGMs with the Bayes factor. This method allows for assessing (relative) evidence for
degree equality or edges differences across any number of groups. Further, confirmatory hypothesis
testing can be used to test predictions or expectations regarding difference or similarities in different
groups (e.g., male vs. female).

Usage

```r
## S3 method for class 'ggm_compare_bf'
select(object, BF_cut = 3, ...)
```

Arguments

- **object**: object of class `ggm_compare_bf`
- **BF_cut**: evidentiary threshold
- **...**: currently ignored

Value

List of class `select.ggm_compare_bf`

- `BF_10` Bayes factors for the alternative ("not equal")
- `BF_01` Bayes factors for the null hypothesis
- `BF_10_adj` Bayes factor adjacency matrix for the alternative ("not equal")
- `BF_01_adj` Bayes factor adjacency matrix for the null hypothesis
- `adj_10` adjacency matrix for the selected edges (in favor of the "not equal")
- `adj_01` adjacency matrix for the selected edges (in favor of the null hypothesis)
- `pcor_mat_10` partial correlation matrix for the alternative ("not equal")

Note

The test provides relative evidence for whether all groups have the same edge strength (for each
degree in the model). It is possible to test whether many groups are all the same. In this case, if there
is evidence for the alternative (not equal), information is not immediately available for which group
differs from the others. Thus pairwise group contrasts could be used after testing the equality of
more than two groups.
select.ggm_compare_estimate

Edge Differences and (Practical) Equivalence Between GGMs

Description

Edge Differences and (Practical) Equivalence Between GGMs

Usage

```r
## S3 method for class 'ggm_compare_estimate'
select(object, cred = 0.95, rope = NULL, prob = 0.95, ...)
```

Arguments

- `object` object of class `ggm_compare_estimate`
- `cred` credible interval width used for the decision rule
- `rope` region of practical equivalence
- `prob` posterior probability (see notes)
- `...` not currently used

Value

A list of class `select.ggm_compare_estimate`: `rope = NULL`:

- `mat_adj` adjacency matrix (one for each contrast)
- `mat_pcor` selected partial correlations (one for each contrast)
- `call` match.call()
- `object` object of class `ggm_compare_estimate`

Examples

```r
Y1 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))
Y2 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))
Y3 <- MASS::mvrnorm(500, rep(0, 16), Sigma = diag(16))

ggm_bf <- ggm_compare_bf(Y1, Y2, Y3,
                       prior_sd = .20,
                       iter = 500,
                       cores = 2)

# select with BF_cut = 3
ggm_bf_sel <- select(ggm_bf, BF_cut = 3)

# summary
summary(ggm_bf_sel)
```
shiny_bggm

- rope region of practical equivalence
- cred credible interval
- prob posterior probability

Note

prob (posterior probability) is the decision rule for the rope. For example, with rope = 0.1 and prob = 0.95, differences require that 95% of the posterior excludes \( \pm 0.1 \), whereas equivalence requires that 95% of the posterior is within \( \pm 0.1 \).

Examples

```r
# data
Y1 <- BGGM::bfi[1:500,1:5]
Y2 <- BGGM::bfi[501:1000, 1:5]

# fit model
fit <- ggm_compare_estimate(Y1, Y2)

# posterior summary of differences
summary(fit)

# select (threshold) with credible intervals
sel <- select(fit)

# summary
summary(sel)

# selected differences
sel$mat_pcor

# adjacency matrix
sel$mat_adj
```

shiny_bggm shiny_bggm Generic

Description

shiny_bggm Generic

Usage

shiny_bggm(object, ...)

Arguments

- **object**: object of class `explore`
- **...**: currently ignored
**shiny_bggm.explore**  
*Shiny App for explore Objects*

**Description**
Shiny App for explore Objects

**Usage**
```r
## S3 method for class 'explore'
shiny_bggm(object, ...)
```

**Arguments**
- `object` object of class explore
- `...` currently ignored

**Examples**
```
fit <- explore(tas[,1:20],
  prior_sd = 0.20,
  iter = 5000)

sel <- select(fit)

shiny_bggm(fit)
```

---

**summary.confirm**  
*Summary Method for confirm Objects*

**Description**
Summary Method for confirm Objects

**Usage**
```r
## S3 method for class 'confirm'
summary(object, ...)
```

**Arguments**
- `object` object of class confirm
- `...` currently ignored
**summary.estimate**

Summary method for `estimate.default` objects

**Description**

Summary method for `estimate.default` objects

**Usage**

```r
## S3 method for class 'estimate'
summary(object, cred = 0.95, ...)
```

**Arguments**

- `object`: An object of class `estimate`
- `cred`: credible interval width
- `...`: currently ignored

**Value**

A list containing the summarized posterior distributions

```r
# data Y <- BGGM::bfi[, 1:5] # analytic approach (sample by setting analytic = FALSE)
fit <- estimate(Y, analytic = TRUE)
summary(fit)
```

**See Also**

- `select.estimate`

**summary.explore**

Summary method for `explore` objects

**Description**

Summary method for `explore` objects

**Usage**

```r
## S3 method for class 'explore'
summary(object, ...)
```

**Arguments**

- `object`: An object of class `explore`
- `...`: currently ignored

**See Also**

- `explore`
summary.ggm_compare_bf

Summary method for ggm_compare_bf objects

Description
Summary method for ggm_compare_bf objects

Usage
## S3 method for class 'ggm_compare_bf'
summary(object, ...)

Arguments
  object             An object of class ggm_compare_bf
  ...                 currently ignored

Value
A list containing the summarized posterior hypothesis probabilities

See Also
  ggm_compare_bf

summary.ggm_compare_estimate

Summary method for ggm_compare_estimate.default objects

Description
Summary method for ggm_compare_estimate.default objects

Usage
## S3 method for class 'ggm_compare_estimate'
summary(object, cred = 0.95, ...)

Arguments
  object         An object of class ggm_compare_estimate
  cred           credible interval width
  ...            currently ignored
Value

A list containing the summarized posterior distributions

See Also

ggm_compare_estimate.default

Examples

```r
# data
Y1 <- BGGM::bfi[1:500,1:5]
Y2 <- BGGM::bfi[501:1000, 1:5]

# fit model
fit <- ggm_compare_estimate(Y1, Y2)

# posterior summary of differences
summary(fit)
```
summary.metric  Summary Method for metric Objects

Description

Summary Method for metric Objects

Usage

## S3 method for class 'metric'
summary(object, cred = 0.95, ...)

Arguments

object  object of class metric
cred  credible interval
...  currently ignored

summary.select.estimate  Summary method for select.estimate objects

Description

Summary method for select.estimate objects

Usage

## S3 method for class 'select.estimate'
summary(object, summarize = FALSE, ...)

Arguments

object  An object of class select.estimate
summarize  if TRUE partial correlations and credible intervals are provided
...  currently ignored

See Also

select.estimate
**summary.select.explore**

*Summary method for select.explore objects*

**Description**

Summary method for select.explore objects

**Usage**

```r
## S3 method for class 'select.explore'
summary(object, hyp = "H1", log = TRUE, summarize = FALSE, ...)
```

**Arguments**

- `object`: An object of class `select.explore`
- `hyp`: hypothesis to summarize (default is the alternative H1)
- `log`: log scale for the Bayes factors (default is TRUE)
- `summarize`: if TRUE partial correlations and credible intervals are provided
- `...`: currently ignored

**See Also**

- `select.explore`

---

**summary.select.ggm_compare_bf**

*Print method for select.ggm_compare_bf objects*

**Description**

Print method for select.ggm_compare_bf objects

**Usage**

```r
## S3 method for class 'select.ggm_compare_bf'
summary(object, ...)
```

**Arguments**

- `object`: An object of class `select.ggm_compare_bf`
- `...`: currently ignored

**See Also**

- `select.ggm_compare_bf`
Summary method for select.ggm_compare_estimate objects

Description

Summary method for select.ggm_compare_estimate objects

Usage

```r
## S3 method for class 'select.ggm_compare_estimate'
summary(object, ...)
```

Arguments

- `object`: An object of class `select.ggm_compare_estimate`
- `...`: currently ignored

See Also

`select.ggm_compare_estimate`

tas  

Toronto Alexithymia Scale (TAS)

Description

A dataset containing items from the Toronto Alexithymia Scale (TAS). There are 20 variables and 1925 observations

Usage

```r
data("tas")
```

Format

A data frame with 20 variables and 1925 observations (5 point Likert scale)
Details

• 1 I am often confused about what emotion I am feeling
• 2 It is difficult for me to find the right words for my feelings
• 3 I have physical sensations that even doctors don’t understand
• 4 I am able to describe my feelings easily
• 5 I prefer to analyze problems rather than just describe them
• 6 When I am upset, I don’t know if I am sad, frightened, or angry
• 7 I am often puzzled by sensations in my body
• 8 I prefer just to let things happen rather than to understand why they turned out that way
• 9 I have feelings that I can’t quite identify
• 10 Being in touch with emotions is essential
• 11 I find it hard to describe how I feel about people
• 12 People tell me to describe my feelings more
• 13 I don’t know what’s going on inside me
• 14 I often don’t know why I am angry
• 15 I prefer talking to people about their daily activities rather than their feelings
• 16 I prefer to watch “light” entertainment shows rather than psychological dramas
• 17 It is difficult for me to reveal my innermost feelings, even to close friends
• 18 I can feel close to someone, even in moments of silence
• 19 I find examination of my feelings useful in solving personal problems
• 20 Looking for hidden meanings in movies or plays distracts from their enjoyment

gender "M" (male) or "F" (female)

Note

There are three domains
Difficultly identifying feelings: items 1, 3, 6, 7, 9, 13, 14
Difficultly describing feelings: items 2, 4, 11, 12, 17
Externally oriented thinking: items 10, 15, 16, 18, 19

References


Examples

data("tas")

# labels
tas_labels <- BGGM:::tas_labels
Index

*Topic datasets
   bfi, 5
   csws, 10
   iri, 22
   ptsd, 50
   ptsd_cor1, 51
   ptsd_cor2, 52
   ptsd_cor3, 53
   ptsd_cor4, 54
   rsa, 56
   tas, 72

assess_predictability, 3

bayes_R2, 4
bfi, 5

coeff.estimate, 7, 39
confirm, 8
convergence, 9
csws, 10
diagnostics, 12
diagnose, 12

estimate, 13
explore, 14, 41, 67

fisher_r2z, 16
fisher_z2r, 16
fitted.estimate, 17

ggm_compare_bf, 18, 41, 48, 68
ggm_compare_estimate, 19
ggm_compare_estimate.default, 19, 41, 42, 49, 69

ggm_compare_ppc, 69

plot.ggm_compare_ppc, 27
plot.metric, 28
plot.select.estimate, 29
plot.select.explore, 31
plot.select.ggm_compare_bf, 32
plot.summary.estimate, 34
plot.summary.explore, 34
plot_adjacency, 35
posterior_predict, 36

pp_check (pp_check.estimate), 37
PPC, 37
predict.estimate, 38

print.coef.estimate, 39
print.confirm, 39
print.estimate, 40
print.explore, 40
print.fitted.estimate, 41
print.ggm_compare_bf, 41
print.ggm_compare_estimate, 42
print.ggm_compare_ppc, 42
print.map, 43
print.metric, 43
print.predict.estimate, 44
print.predict.estimate, 44
print.select.estimate, 45
print.select.explore, 45
print.select.ggm_compare_bf, 46
print.select.ggm_compare_estimate, 46
print.summary.estimate, 47
print.summary.explore, 47
print.summary.ggm_compare_bf
   (print.summary.ggm_estimate_bf)
INDEX

48
print.summary.ggm_compare_ppc, 48
print.summary.ggm_estimate_bf, 48
print.summary.ggm_estimate_compare, 49
print.summary.metric, 49
ptsd, 50
ptsd_cor1, 51
ptsd_cor2, 52
ptsd_cor3, 53
ptsd_cor4, 54

residuals.estimate, 55
rmse, 56
rsa, 56

select, 20, 58
select.estimate, 40, 45, 58, 59, 67, 70
select.explore, 45, 58, 60, 71
select.ggm_compare_bf, 46, 63, 71
select.ggm_compare_estimate, 20, 46, 58, 64, 72
shiny_bggm, 65
shiny_bggm.explore, 66
summary.confirm, 66
summary.explore, 47, 67
summary.estimate, 47, 67
summary.ggm_compare_bf, 68
summary.ggm_compare_estimate, 68
summary.ggm_compare_ppc, 48, 69
summary.metric, 70
summary.select.estimate, 70
summary.select.explore, 71
summary.select.ggm_compare_bf, 71
summary.select.ggm_compare_estimate, 72

tas, 72