Package: rr (via r-universe)

November 2, 2026

Version 1.4.2

Title Statistical Methods for the Randomized Response Technique

Depends R (>= 3.0.0), utils

Imports MASS, arm, coda, magic

Description Enables researchers to conduct multivariate statistical analyses of survey data with randomized response technique items from several designs, including mirrored question, forced question, and unrelated question. This includes regression with the randomized response as the outcome and logistic regression with the randomized response item as a predictor. In addition, tools for conducting power analysis for designing randomized response items are included. The package implements methods described in Blair, Imai, and Zhou (2015) "Design and Analysis of the Randomized Response Technique," Journal of the American Statistical Association <https://graemeblair.com/papers/randresp.pdf>.

LazyLoad yes

LazyData yes

License GPL (>= 3)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation yes

Author Graeme Blair [aut, cre], Yang-Yang Zhou [aut], Kosuke Imai [aut], Winston Chou [ctb]

Maintainer Graeme Blair <graeme.blair@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2024-01-17 02:41:43 UTC

RemoteUrl https://github.com/cran/rr

RemoteRef HEAD

RemoteSha 5a82266c32877445c9c4e4c1f9dec00149a5f79b

Contents

rr-package . . . 2
nigeria . . . 3
nigeria-data . . . 4
power.rr.plot . . . 5
power.rr.test . . . 7
predict.rrreg . . . 9
predict.rrreg.predictor . . . 11
rrreg . . . 14
rrreg.bayes . . . 16
rrreg.predictor . . . 20

Index **24**

`rr-package` *R Package for the Randomized Response Technique*

Description

`rr` implements methods developed by Blair, Imai, and Zhou (2015) such as multivariate regression and power analysis for the randomized response technique. Randomized response is a survey technique that introduces random noise to reduce potential bias from non-response and social desirability when asking questions about sensitive behaviors and beliefs. The current version of this package conducts multivariate regression analyses for the sensitive item under four standard randomized response designs: mirrored question, forced response, disguised response, and unrelated question. Second, it generates predicted probabilities of answering affirmatively to the sensitive item for each respondent. Third, it also allows users to use the sensitive item as a predictor in an outcome regression under the forced response design. Additionally, it implements power analyses to help improve research design. In future versions, this package will extend to new modified designs that are based on less stringent assumptions than those of the standard designs, specifically to allow for non-compliance and unknown distribution to the unrelated question under the unrelated question design.

Author(s)

Graeme Blair, Experiments in Governance and Politics, Columbia University <graeme.blair@gmail.com>, https://graemeblair.com

Kosuke Imai, Departments of Government and Statistics, Harvard University <kimai@harvard.edu>, https://imai.fas.harvard.edu

Yang-Yang Zhou, Department of Political Science, University of British Columbia <yangyang.zhou@ubc.ca>, https://www.yangyangzhou.com

Maintainer: Graeme Blair <graeme.blair@gmail.com>

References

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2015) "Design and Analysis of the Randomized Response Technique." *Journal of the American Statistical Association.* Available at https://graemeblair.com/papers/randresp.pdf.

`nigeria` *Nigeria Randomized Response Survey Experiment on Social Connections to Armed Groups*

Description

This data set is a subset of the data from the randomized response technique survey experiment conducted in Nigeria to study civilian contact with armed groups. The survey was implemented by Blair (2014).

Usage

```
data(nigeria)
```

Format

A data frame containing 2457 observations. The variables are:

- `Quesid`: Survey ID of civilian respondent.
- `rr.q1`: Randomized response survey item using the Forced Response Design asking the respondent whether they hold direct social connections with members of armed groups. 0 if no connection; 1 if connection.
- `cov.age`: Age of the respondent.
- `cov.asset.index`: The number of assets owned by the respondent from an index of nine assets including radio, T.V., motorbike, car, mobile phone, refrigerator, goat, chicken, and cow.
- `cov.married`: Marital status. 0 if single; 1 if married.
- `cov.education`: Education level of the respondent. 1 if no school; 2 if started primary school; 3 if finished primary school; 4 if started secondary school; 5 if finished secondary school; 6 if started polytechnic or college; 7 if finished polytechnic or college; 8 if started university; 9 if finished university; 10 if received graduate (masters or Ph.D) education.
- `cov.female`: Gender. 0 if male; 1 if female.
- `civic`: Whether or not the respondent is a member of a civic group in their communities, such as youth groups , women's groups, or community development committees.

Source

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) Replication data for: Design and Analysis of the Randomized Response Technique.

References

Blair, G. (2014). "Why do civilians hold bargaining power in state revenue conflicts? Evidence from Nigeria."

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) "Design and Analysis of the Randomized Response Technique." *Working Paper.* Available at http://imai.princeton.edu/research/randresp.html.

`nigeria-data` *Nigeria Randomized Response Survey Experiment on Social Connections to Armed Groups*

Description

This data set is a subset of the data from the randomized response technique survey experiment conducted in Nigeria to study civilian contact with armed groups. The survey was implemented by Blair (2014).

Usage

```
data(nigeria)
```

Format

A data frame containing 2457 observations. The variables are:

- `Quesid`: Survey ID of civilian respondent.
- `rr.q1`: Randomized response survey item using the Forced Response Design asking the respondent whether they hold direct social connections with members of armed groups. 0 if no connection; 1 if connection.
- `cov.age`: Age of the respondent.
- `cov.asset.index`: The number of assets owned by the respondent from an index of nine assets including radio, T.V., motorbike, car, mobile phone, refrigerator, goat, chicken, and cow.
- `cov.married`: Marital status. 0 if single; 1 if married.
- `cov.education`: Education level of the respondent. 1 if no school; 2 if started primary school; 3 if finished primary school; 4 if started secondary school; 5 if finished secondary school; 6 if started polytechnic or college; 7 if finished polytechnic or college; 8 if started university; 9 if finished university; 10 if received graduate (masters or Ph.D) education.
- `cov.female`: Gender. 0 if male; 1 if female.
- `civic`: Whether or not the respondent is a member of a civic group in their communities, such as youth groups , women's groups, or community development committees.

Source

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) Replication data for: Design and Analysis of the Randomized Response Technique.

References

Blair, G. (2014). "Why do civilians hold bargaining power in state revenue conflicts? Evidence from Nigeria."

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) "Design and Analysis of the Randomized Response Technique." *Working Paper.* Available at http://imai.princeton.edu/research/randresp.html.

`power.rr.plot` *Power Analysis Plot for Randomized Response*

Description

`power.rr.plot` generates a power analysis plot for randomized response survey designs.

Usage

```
power.rr.plot(p, p0, p1, q, design, n.seq, r, presp.seq, presp.null =
NULL, sig.level, prespT.seq, prespC.seq, prespT.null = NULL, prespC.null,
type = c("one.sample", "two.sample"), alternative = c("one.sided",
"two.sided"), solve.tolerance = .Machine$double.eps, legend = TRUE, legend.x
= "bottomright", legend.y, par = TRUE, ...)
```

Arguments

`p`	The probability of receiving the sensitive question (Mirrored Question Design, Unrelated Question Design); the probability of answering truthfully (Forced Response Design); the probability of selecting a red card from the 'yes' stack (Disguised Response Design).
`p0`	The probability of forced 'no' (Forced Response Design).
`p1`	The probability of forced 'yes' (Forced Response Design).
`q`	The probability of answering 'yes' to the unrelated question, which is assumed to be independent of covariates (Unrelated Question Design).
`design`	Call of design (including modified designs) used: "forced-known", "mirrored", "disguised", "unrelated-known", "forced-unknown", and "unrelated-unknown".
`n.seq`	A sequence of number of observations or sample sizes.
`r`	For the modified designs only (i.e. "forced-unknown" for Forced Response with Unknown Probability and "unrelated-unknown" for Unrelated Question with Unknown Probability), `r` is the proportion of respondents allocated to the first group, which is the group that is directed to answer the sensitive question truthfully with probability `p` as opposed to the second group which is directed to answer the sensitive question truthfully with probability `1-p`.
`presp.seq`	For a one sample test, a sequence of probabilities of possessing the sensitive trait under the alternative hypothesis.

presp.null	For a one sample test, the probability of possessing the sensitive trait under the null hypothesis. The default is NULL meaning zero probability of possessing the sensitive trait.
sig.level	Significance level (Type I error probability).
prespT.seq	For a two sample test, a sequence of probabilities of the treated group possessing the sensitive trait under the alternative hypothesis.
prespC.seq	For a two sample test, a sequence of probabitilies of the control group possessing the sensitive trait under the alternative hypothesis.
prespT.null	For a two sample test, the probability of the treated group possessing the sensitive trait under the null hypothesis. The default is NULL meaning there is no difference between the treated and control groups, specifically that prespT.null is the same as prespC.null, the probability of the control group possessing the sensitive trait under the null hypothesis.
prespC.null	For a two sample test, the probability of the control group possessing the sensitive trait under the null hypothesis.
type	One or two sample test. For a two sample test, the alternative and null hypotheses refer to the difference between the two samples of the probabilities of possessing the sensitive trait.
alternative	One or two sided test.
solve.tolerance	When standard errors are calculated, this option specifies the tolerance of the matrix inversion operation solve.
legend	Indicator of whether to include a legend of sample sizes. The default is TRUE.
legend.x	Placement on the x-axis of the legend. The default is "bottomright".
legend.y	Placement on the y-axis of the legend.
par	Option to set or query graphical parameters within the function. The default is TRUE.
...	Additional arguments to be passed to par()

Details

This function generates a power analysis plot for randomized response survey designs, both for the standard designs ("forced-known", "mirrored", "disguised", "unrelated-known") and modified designs ("forced-unknown", and "unrelated -unknown"). The x-axis shows the population proportions with the sensitive trait; the y-axis shows the statistical power; and different sample sizes are shown as different lines in grayscale.

Value

Power curve plot

References

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) "Design and Analysis of the Randomized Response Technique." *Working Paper.* Available at http://imai.princeton.edu/research/randresp.html.

Examples

```
## Generate a power plot for the forced design with known
## probabilities of 2/3 in truth-telling group, 1/6 forced to say "yes"
## and 1/6 forced to say "no", varying the number of respondents from
## 250 to 2500 and the population proportion of respondents
## possessing the sensitive trait from 0 to .15.

presp.seq <- seq(from = 0, to = .15, by = .0025)
n.seq <- c(250, 500, 1000, 2000, 2500)
power.rr.plot(p = 2/3, p1 = 1/6, p0 = 1/6, n.seq = n.seq,
              presp.seq = presp.seq, presp.null = 0,
              design = "forced-known", sig.level = .01,
              type = "one.sample",
              alternative = "one.sided", legend = TRUE)


## Replicates the results for Figure 2 in Blair, Imai, and Zhou (2014)
```

`power.rr.test` *Power Analysis for Randomized Response*

Description

`power.rr.test` is used to conduct power analysis for randomized response survey designs.

Usage

```
power.rr.test(p, p0, p1, q, design, n = NULL, r, presp, presp.null =
NULL, sig.level, prespT, prespC, prespT.null = NULL, prespC.null, power =
NULL, type = c("one.sample", "two.sample"), alternative = c("one.sided",
"two.sided"), solve.tolerance = .Machine$double.eps)
```

Arguments

`p`	The probability of receiving the sensitive question (Mirrored Question Design, Unrelated Question Design); the probability of answering truthfully (Forced Response Design); the probability of selecting a red card from the 'yes' stack (Disguised Response Design).
`p0`	The probability of forced 'no' (Forced Response Design).
`p1`	The probability of forced 'yes' (Forced Response Design).
`q`	The probability of answering 'yes' to the unrelated question, which is assumed to be independent of covariates (Unrelated Question Design).
`design`	Call of design (including modified designs) used: "forced-known", "mirrored", "disguised", "unrelated-known", "forced-unknown", and "unrelated-unknown".
`n`	Number of observations. Exactly one of 'n' or 'power' must be NULL.

r	For the modified designs only (i.e. "forced-unknown" for Forced Response with Unknown Probability and "unrelated-unknown" for Unrelated Question with Unknown Probability), r is the proportion of respondents allocated to the first group, which is the group that is directed to answer the sensitive question truthfully with probability p as opposed to the second group which is directed to answer the sensitive question truthfully with probability 1-p.
presp	For a one sample test, the probability of possessing the sensitive trait under the alternative hypothesis.
presp.null	For a one sample test, the probability of possessing the sensitive trait under the null hypothesis. The default is NULL meaning zero probability of possessing the sensitive trait.
sig.level	Significance level (Type I error probability).
prespT	For a two sample test, the probability of the treated group possessing the sensitive trait under the alternative hypothesis.
prespC	For a two sample test, the probability of the control group possessing the sensitive trait under the alternative hypothesis.
prespT.null	For a two sample test, the probability of the treated group possessing the sensitive trait under the null hypothesis. The default is NULL meaning there is no difference between the treated and control groups, specifically that prespT.null is the same as prespC.null, the probability of the control group possessing the sensitive trait under the null hypothesis.
prespC.null	For a two sample test, the probability of the control group possessing the sensitive trait under the null hypothesis.
power	Power of test (Type II error probability). Exactly one of 'n' or 'power' must be NULL.
type	One or two sample test. For a two sample test, the alternative and null hypotheses refer to the difference between the two samples of the probabilities of possessing the sensitive trait.
alternative	One or two sided test.
solve.tolerance	When standard errors are calculated, this option specifies the tolerance of the matrix inversion operation solve.

Details

This function allows users to conduct power analysis for randomized response survey designs, both for the standard designs ("forced-known", "mirrored", "disguised", "unrelated-known") and modified designs ("forced-unknown", and "unrelated -unknown").

Value

power.rr.test contains the following components (the inclusion of some components such as the design parameters are dependent upon the design used):

n	Point estimates for the effects of covariates on the randomized response item.

r	Standard errors for estimates of the effects of covariates on the randomized response item.
presp	For a one sample test, the probability of possessing the sensitive trait under the alternative hypothesis. For a two sample test, the difference between the probabilities of possessing the sensitive trait for the treated and control groups under the alternative hypothesis.
presp.null	For a one sample test, the probability of possessing the sensitive trait under the null hypothesis. For a two sample test, the difference between the probabilities of possessing the sensitive trait for the treated and control groups under the null hypothesis.
sig.level	Significance level (Type I error probability).
power	Power of test (Type II error probability).
type	One or two sample test.
alternative	One or two sided test.

References

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2015) "Design and Analysis of the Randomized Response Technique." *Journal of the American Statistical Association.* Available at https://graemeblair.com/papers/randresp.pdf.

Examples

```
## Calculate the power to detect a sensitive item proportion of .2
## with the forced design with known probabilities of 2/3 in truth-telling group,
## 1/6 forced to say "yes" and 1/6 forced to say "no" and sample size of 200.

power.rr.test(p = 2/3, p1 = 1/6, p0 = 1/6, n = 200,
              presp = .2, presp.null = 0,
              design = "forced-known", sig.level = .01,
              type = "one.sample", alternative = "one.sided")
```

---

predict.rrreg *Predicted Probabilities for Randomized Response Regression*

Description

predict.rrreg is used to generate predicted probabilities from a multivariate regression object of survey data using randomized response methods.

Usage

```
## S3 method for class 'rrreg'
predict(object, given.y = FALSE, alpha = .05, n.sims =
1000, avg = FALSE, newdata = NULL, quasi.bayes = FALSE, keep.draws = FALSE,
...)
```

Arguments

object	An object of class "rrreg" generated by the rrreg() function.
given.y	Indicator of whether to use "y" the response vector to calculate the posterior prediction of latent responses. Default is FALSE, which simply generates fitted values using the logistic regression.
alpha	Confidence level for the hypothesis test to generate upper and lower confidence intervals. Default is .05.
n.sims	Number of sampled draws for quasi-bayesian predicted probability estimation. Default is 1000.
avg	Whether to output the mean of the predicted probabilities and uncertainty estimates. Default is FALSE.
newdata	Optional new data frame of covariates provided by the user. Otherwise, the original data frame from the "rreg" object is used.
quasi.bayes	Option to use Monte Carlo simulations to generate uncertainty estimates for predicted probabilities. Default is FALSE.
keep.draws	Option to return the Monte Carlos draws of the quantity of interest, for use in calculating differences for example.
...	Further arguments to be passed to predict.rrreg() command.

Details

This function allows users to generate predicted probabilities for the randomized response item given an object of class "rrreg" from the rrreg() function. Four standard designs are accepted by this function: mirrored question, forced response, disguised response, and unrelated question. The design, already specified in the "rrreg" object, is then directly inputted into this function.

Value

predict.rrreg returns predicted probabilities either for each observation in the data frame or the average over all observations. The output is a list that contains the following components:

est	Predicted probabilities for the randomized response item generated either using fitted values, posterior predictions, or quasi-Bayesian simulations. If avg is set to TRUE, the output will only include the mean estimate.
se	Standard errors for the predicted probabilities of the randomized response item generated using Monte Carlo simulations. If quasi.bayes is set to FALSE, no standard errors will be outputted.
ci.lower	Estimates for the lower confidence interval. If quasi.bayes is set to FALSE, no confidence interval estimate will be outputted.
ci.upper	Estimates for the upper confidence interval. If quasi.bayes is set to FALSE, no confidence interval estimate will be outputted.
qoi.draws	Monte Carlos draws of the quantity of interest, returned only if keep.draws is set to TRUE.

References

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) "Design and Analysis of the Randomized Response Technique." *Working Paper.* Available at http://imai.princeton.edu/research/randresp.html.

See Also

rrreg to conduct multivariate regression analyses in order to generate predicted probabilities for the randomized response item.

Examples

```
data(nigeria)

set.seed(1)

## Define design parameters
p <- 2/3  # probability of answering honestly in Forced Response Design
p1 <- 1/6 # probability of forced 'yes'
p0 <- 1/6 # probability of forced 'no'

## Fit linear regression on the randomized response item of
## whether citizen respondents had direct social contacts to armed groups

rr.q1.reg.obj <- rrreg(rr.q1 ~ cov.asset.index + cov.married + I(cov.age/10) +
                       I((cov.age/10)^2) + cov.education + cov.female,
                       data = nigeria, p = p, p1 = p1, p0 = p0,
                       design = "forced-known")

## Generate the mean predicted probability of having social contacts to
## armed groups across respondents using quasi-Bayesian simulations.

rr.q1.reg.pred <- predict(rr.q1.reg.obj, given.y = FALSE,
                                avg = TRUE, quasi.bayes = TRUE,
                                n.sims = 10000)

## Replicates Table 3 in Blair, Imai, and Zhou (2014)
```

---

predict.rrreg.predictor

Predicted Probabilities for Randomized Response as a Regression Predictor

---

Description

predict.rrreg.predictor is used to generate predicted probabilities from a multivariate regression object of survey data using the randomized response item as a predictor for an additional outcome.

Usage

```
## S3 method for class 'rrreg.predictor'
predict(object, fix.z = NULL, alpha = .05,
n.sims = 1000, avg = FALSE, newdata = NULL, quasi.bayes = FALSE, keep.draws
= FALSE, ...)
```

Arguments

object	An object of class "rrreg.predictor" generated by the rrreg.predictor() function.
fix.z	An optional value or vector of values between 0 and 1 that the user inputs as the proportion of respondents with the sensitive trait or probability that each respondent has the sensitive trait, respectively. If the user inputs a vector of values, the vector must be the length of the data from the "rrreg.predictor" object. Default is NULL in which case predicted probabilities are generated for the randomized response item.
alpha	Confidence level for the hypothesis test to generate upper and lower confidence intervals. Default is .05.
n.sims	Number of sampled draws for quasi-bayesian predicted probability estimation. Default is 1000.
avg	Whether to output the mean of the predicted probabilities and uncertainty estimates. Default is FALSE.
newdata	Optional new data frame of covariates provided by the user. Otherwise, the original data frame from the "rreg" object is used.
quasi.bayes	Option to use Monte Carlo simulations to generate uncertainty estimates for predicted probabilities. Default is FALSE meaning no uncertainty estimates are outputted.
keep.draws	Option to return the Monte Carlos draws of the quantity of interest, for use in calculating differences for example.
...	Further arguments to be passed to predict.rrreg.predictor() command.

Details

This function allows users to generate predicted probabilities for the additional outcome variables with the randomized response item as a covariate given an object of class "rrreg.predictor" from the rrreg.predictor() function. Four standard designs are accepted by this function: mirrored question, forced response, disguised response, and unrelated question. The design, already specified in the "rrreg.predictor" object, is then directly inputted into this function.

Value

predict.rrreg.predictor returns predicted probabilities either for each observation in the data frame or the average over all observations. The output is a list that contains the following components:

est	Predicted probabilities of the additional outcome variable given the randomized response item as a predictor generated either using fitted values or quasi-Bayesian simulations. If avg is set to TRUE, the output will only include the mean estimate.
se	Standard errors for the predicted probabilities of the additional outcome variable given the randomized response item as a predictor generated using Monte Carlo simulations. If quasi.bayes is set to FALSE, no standard errors will be outputted.
ci.lower	Estimates for the lower confidence interval. If quasi.bayes is set to FALSE, no confidence interval estimate will be outputted.
ci.upper	Estimates for the upper confidence interval. If quasi.bayes is set to FALSE, no confidence interval estimate will be outputted.
qoi.draws	Monte Carlos draws of the quantity of interest, returned only if keep.draws is set to TRUE.

References

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) "Design and Analysis of the Randomized Response Technique." *Working Paper.* Available at http://imai.princeton.edu/research/randresp.html.

See Also

rrreg.predictor to conduct multivariate regression analyses with the randomized response as predictor in order to generate predicted probabilities.

Examples

```
data(nigeria)

## Define design parameters


set.seed(44)

p <- 2/3  # probability of answering honestly in Forced Response Design
p1 <- 1/6 # probability of forced 'yes'
p0 <- 1/6 # probability of forced 'no'

## Fit joint model of responses to an outcome regression of joining a civic
## group and the randomized response item of having a militant social connection

rr.q1.pred.obj <-
    rrreg.predictor(civic ~ cov.asset.index + cov.married + I(cov.age/10) +
              I((cov.age/10)^2) + cov.education + cov.female
              + rr.q1, rr.item = "rr.q1", parstart = FALSE, estconv = TRUE,
              data = nigeria, verbose = FALSE, optim = TRUE,
              p = p, p1 = p1, p0 = p0, design = "forced-known")

## Generate predicted probabilities for the likelihood of joining
```

```
## a civic group across respondents using quasi-Bayesian simulations.

rr.q1.rrreg.predictor.pred <- predict(rr.q1.pred.obj,
                                  avg = TRUE, quasi.bayes = TRUE,
                                  n.sims = 1000)
```

---

rrreg *Randomized Response Regression*

Description

rrreg is used to conduct multivariate regression analyses of survey data using randomized response methods.

Usage

```
rrreg(formula, p, p0, p1, q, design, data, start = NULL,
h = NULL, group = NULL, matrixMethod = "efficient",
maxIter = 10000, verbose = FALSE, optim = FALSE, em.converge = 10^(-8),
glmMaxIter = 10000, solve.tolerance = .Machine$double.eps)
```

Arguments

formula	An object of class "formula": a symbolic description of the model to be fitted.
p	The probability of receiving the sensitive question (Mirrored Question Design, Unrelated Question Design); the probability of answering truthfully (Forced Response Design); the probability of selecting a red card from the 'yes' stack (Disguised Response Design). For "mirrored" and "disguised" designs, p cannot equal .5.
p0	The probability of forced 'no' (Forced Response Design).
p1	The probability of forced 'yes' (Forced Response Design).
q	The probability of answering 'yes' to the unrelated question, which is assumed to be independent of covariates (Unrelated Question Design).
design	One of the four standard designs: "forced-known", "mirrored", "disguised", or "unrelated-known".
data	A data frame containing the variables in the model.
start	Optional starting values of coefficient estimates for the Expectation-Maximization (EM) algorithm.
h	Auxiliary data functionality. Optional named numeric vector with length equal to number of groups. Names correspond to group labels and values correspond to auxiliary moments.
group	Auxiliary data functionality. Optional character vector of group labels with length equal to number of observations.

`matrixMethod` Auxiliary data functionality. Procedure for estimating optimal weighting matrix for generalized method of moments. One of "efficient" for two-step feasible and "cue" for continuously updating. Default is "efficient". Only relevant if `h` and `group` are specified.

`maxIter` Maximum number of iterations for the Expectation-Maximization algorithm. The default is `10000`.

`verbose` A logical value indicating whether model diagnostics counting the number of EM iterations are printed out. The default is `FALSE`.

`optim` A logical value indicating whether to use the quasi-Newton "BFGS" method to calculate the variance-covariance matrix and standard errors. The default is `FALSE`.

`em.converge` A value specifying the satisfactory degree of convergence under the EM algorithm. The default is `10^(-8)`.

`glmMaxIter` A value specifying the maximum number of iterations to run the EM algorithm. The default is `10000`.

`solve.tolerance`
When standard errors are calculated, this option specifies the tolerance of the matrix inversion operation solve.

Details

This function allows users to perform multivariate regression analysis on data from the randomized response technique. Four standard designs are accepted by this function: mirrored question, forced response, disguised response, and unrelated question. The method implemented by this function is the Maximum Likelihood (ML) estimation for the Expectation-Maximization (EM) algorithm.

Value

`rrreg` returns an object of class "rrreg". The function `summary` is used to obtain a table of the results. The object `rrreg` is a list that contains the following components (the inclusion of some components such as the design parameters are dependent upon the design used):

`est` Point estimates for the effects of covariates on the randomized response item.

`vcov` Variance-covariance matrix for the effects of covariates on the randomized response item.

`se` Standard errors for estimates of the effects of covariates on the randomized response item.

`data` The `data` argument.

`coef.names` Variable names as defined in the data frame.

`x` The model matrix of covariates.

`y` The randomized response vector.

`design` Call of standard design used: "forced-known", "mirrored", "disguised", or "unrelated-known".

`p` The `p` argument.

`p0` The `p0` argument.

p1	The p1 argument.
q	The q argument.
call	The matched call.

References

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) "Design and Analysis of the Randomized Response Technique." Working Paper. Available at http://imai.princeton.edu/research/randresp.html.

See Also

predict.rrreg for predicted probabilities.

Examples

```
data(nigeria)

set.seed(1)

## Define design parameters
p <- 2/3  # probability of answering honestly in Forced Response Design
p1 <- 1/6 # probability of forced 'yes'
p0 <- 1/6 # probability of forced 'no'

## Fit linear regression on the randomized response item of whether
## citizen respondents had direct social contacts to armed groups

rr.q1.reg.obj <- rrreg(rr.q1 ~ cov.asset.index + cov.married +
                    I(cov.age/10) + I((cov.age/10)^2) + cov.education + cov.female,
                    data = nigeria, p = p, p1 = p1, p0 = p0,
                    design = "forced-known")

summary(rr.q1.reg.obj)

## Replicates Table 3 in Blair, Imai, and Zhou (2014)
```

---

rrreg.bayes *Bayesian Randomized Response Regression*

Description

Function to conduct multivariate regression analyses of survey data with the randomized response technique using Bayesian MCMC.

Usage

```
rrreg.bayes(formula, p, p0, p1, design, data, group.mixed,
formula.mixed = ~1, verbose = FALSE, n.draws = 10000, burnin = 5000, thin =
1, beta.start, beta.mu0, beta.A0, beta.tune, Psi.start, Psi.df, Psi.scale,
Psi.tune)
```

Arguments

formula	An object of class "formula": a symbolic description of the model to be fitted.
p	The probability of receiving the sensitive question (Mirrored Question Design, Unrelated Question Design); the probability of answering truthfully (Forced Response Design); the probability of selecting a red card from the 'yes' stack (Disguised Response Design).
p0	The probability of forced 'no' (Forced Response Design).
p1	The probability of forced 'yes' (Forced Response Design).
design	Character indicating the design. Currently only "forced-known" is supported.
data	A data frame containing the variables in the model.
group.mixed	A string indicating the variable name of a numerical group indicator specifying which group each individual belongs to for a mixed effects model.
formula.mixed	To specify a mixed effects model, include this formula object for the group-level fit. ~1 allows intercepts to vary, and including covariates in the formula allows the slopes to vary also.
verbose	A logical value indicating whether model diagnostics are printed out during fitting.
n.draws	Number of MCMC iterations.
burnin	The number of initial MCMC iterations that are discarded.
thin	The interval of thinning between consecutive retained iterations (1 for no thinning).
beta.start	Optional starting values for the sensitive item fit. This should be a vector of length the number of covariates.
beta.mu0	Optional vector of prior means for the sensitive item fit parameters, a vector of length the number of covariates.
beta.A0	Optional matrix of prior precisions for the sensitive item fit parameters, a matrix of dimension the number of covariates.
beta.tune	A required vector of tuning parameters for the Metropolis algorithm for the sensitive item fit. This must be set and refined by the user until the acceptance ratios are approximately .4 (reported in the output).
Psi.start	Optional starting values for the variance of the random effects in the mixed effects models. This should be a scalar.
Psi.df	Optional prior degrees of freedom parameter for the variance of the random effects in the mixed effects models.
Psi.scale	Optional prior scale parameter for the variance of the random effects in the mixed effects models.

`Psi.tune` A required vector of tuning parameters for the Metropolis algorithm for variance of the random effects in the mixed effects models. This must be set and refined by the user until the acceptance ratios are approximately .4 (reported in the output).

Details

This function allows the user to perform regression analysis on data from the randomized response technique using a Bayesian MCMC algorithm.

The Metropolis algorithm for the Bayesian MCMC estimators in this function must be tuned to work correctly. The `beta.tune` and, for the mixed effects model `Psi.tune`, are required, and the values, one for each estimated parameter, will need to be manipulated. The output of the `rrreg.bayes` function displays the acceptance ratios from the Metropolis algorithm. If these values are far from 0.4, the tuning parameters should be changed until the ratios approach 0.4.

Convergence is at times difficult to achieve, so we recommend running multiple chains from overdispersed starting values by, for example, running an MLE using the rrreg() function, and then generating a set of overdispersed starting values using those estimates and their estimated variance-covariance matrix. An example is provided below for each of the possible designs. Running `summary()` after such a procedure will output the Gelman-Rubin convergence statistics in addition to the estimates. If the G-R statistics are all below 1.1, the model is said to have converged.

Value

`rrreg.bayes` returns an object of class "rrreg.bayes". The function `summary` is used to obtain a table of the results.

`beta`	The coefficients for the sensitive item fit. An object of class "mcmc" that can be analyzed using the `coda` package.
`data`	The `data` argument.
`coef.names`	Variable names as defined in the data frame.
`x`	The model matrix of covariates.
`y`	The randomized response vector.
`design`	Call of standard design used: "forced-known", "mirrored", "disguised", or "unrelated-known".
`p`	The `p` argument.
`p0`	The `p0` argument.
`p1`	The `p1` argument.
`beta.tune`	The `beta.tune` argument.
`mixed`	Indicator for whether a mixed effects model was run.
`call`	the matched call.

If a mixed-effects model is used, then several additional objects are included:

`Psi`	The coefficients for the group-level fit. An object of class "mcmc" that can be analyzed using the `coda` package.

gamma	The random effects estimates. An object of class "mcmc" that can be analyzed using the coda package.
coef.names.mixed	Variable names for the predictors for the second-level model
z	The predictors for the second-level model.
groups	A vector of group indicators.
Psi.tune	The Psi.tune argument.

References

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) "Design and Analysis of the Randomized Response Technique." *Working Paper.* Available at http://imai.princeton.edu/research/randresp.html.

Examples

```
data(nigeria)

## Define design parameters
p <- 2/3  # probability of answering honestly in Forced Response Design
p1 <- 1/6 # probability of forced 'yes'
p0 <- 1/6 # probability of forced 'no'

## run three chains with overdispersed starting values

set.seed(1)

## starting values constructed from MLE model
mle.estimates <- rrreg(rr.q1 ~ cov.asset.index + cov.married +
                         I(cov.age/10) + I((cov.age/10)^2) + cov.education + cov.female,
                         data = nigeria,
                       p = p, p1 = p1, p0 = p0,
                       design = "forced-known")



library(MASS)
draws <- mvrnorm(n = 3, mu = coef(mle.estimates),
  Sigma = vcov(mle.estimates) * 9)

## run three chains
bayes.1 <- rrreg.bayes(rr.q1 ~ cov.asset.index + cov.married +
                         I(cov.age/10) + I((cov.age/10)^2) + cov.education + cov.female,
                       data = nigeria, p = p, p1 = p1, p0 = p0,
                       beta.tune = .0001, beta.start = draws[1,],
                       design = "forced-known")
bayes.2 <- rrreg.bayes(rr.q1 ~ cov.asset.index + cov.married +
                         I(cov.age/10) + I((cov.age/10)^2) + cov.education + cov.female,
                       data = nigeria, p = p, p1 = p1, p0 = p0,
                       beta.tune = .0001, beta.start = draws[2,],
                       design = "forced-known")
```

```
bayes.3 <- rrreg.bayes(rr.q1 ~ cov.asset.index + cov.married +
                         I(cov.age/10) + I((cov.age/10)^2) + cov.education + cov.female,
                       data = nigeria, p = p, p1 = p1, p0 = p0,
                       beta.tune = .0001, beta.start = draws[3,],
                       design = "forced-known")

bayes <- as.list(bayes.1, bayes.2, bayes.3)

summary(bayes)
```

`rrreg.predictor` *Randomized Response as a Regression Predictor*

Description

`rrreg.predictor` is used to jointly model the randomized response item as both outcome and predictor for an additional outcome given a set of covariates.

Usage

```
rrreg.predictor(formula, p, p0, p1, q, design, data, rr.item,
model.outcome = "logistic", fit.sens = "bayesglm", fit.outcome = "bayesglm",
bstart = NULL, tstart = NULL, parstart = TRUE, maxIter = 10000, verbose =
FALSE, optim = FALSE, em.converge = 10^(-4), glmMaxIter = 20000, estconv =
TRUE, solve.tolerance = .Machine$double.eps)
```

Arguments

`formula`	An object of class "formula": a symbolic description of the model to be fitted with the randomized response item as one of the covariates.
`p`	The probability of receiving the sensitive question (Mirrored Question Design, Unrelated Question Design); the probability of answering truthfully (Forced Response Design); the probability of selecting a red card from the 'yes' stack (Disguised Response Design).
`p0`	The probability of forced 'no' (Forced Response Design).
`p1`	The probability of forced 'yes' (Forced Response Design).
`q`	The probability of answering 'yes' to the unrelated question, which is assumed to be independent of covariates (Unrelated Question Design).
`design`	One of the four standard designs: "forced-known", "mirrored", "disguised", or "unrelated-known".
`data`	A data frame containing the variables in the model. Observations with missingness are list-wise deleted.

`rr.item` A string containing the name of the randomized response item variable in the data frame.

`model.outcome` Currently the function only allows for logistic regression, meaning the outcome variable must be binary.

`fit.sens` Indicator for whether to use Bayesian generalized linear modeling (bayesglm) in the Maximization step for the Expectation-Maximization (EM) algorithm to generate coefficients for the randomized response item as the outcome. Default is `"bayesglm"`; otherwise input `"glm"`.

`fit.outcome` Indicator for whether to use Bayesian generalized linear modeling (bayesglm) in the Maximization step for the EM algorithm to generate coefficients for the outcome variable given in the formula with the randomized response item as a covariate. Default is `"bayesglm"`; otherwise input `"glm"`.

`bstart` Optional starting values of coefficient estimates for the randomized response item as outcome for the EM algorithm.

`tstart` Optional starting values of coefficient estimates for the outcome variable given in the formula for the EM algorithm.

`parstart` Option to use the function `rrreg` to generate starting values of coefficient estimates for the randomized response item as outcome for the EM algorithm. The default is `TRUE`, but if starting estimates are inputted by the user in `bstart`, this option is overidden.

`maxIter` Maximum number of iterations for the Expectation-Maximization algorithm. The default is `10000`.

`verbose` A logical value indicating whether model diagnostics counting the number of EM iterations are printed out. The default is `FALSE`.

`optim` A logical value indicating whether to use the quasi-Newton "BFGS" method to calculate the variance-covariance matrix and standard errors. The default is `FALSE`.

`em.converge` A value specifying the satisfactory degree of convergence under the EM algorithm. The default is `10^(-4)`.

`glmMaxIter` A value specifying the maximum number of iterations to run the EM algorithm. The default is `20000` .

`estconv` Option to base convergence on the absolute value of the difference between subsequent coefficients generated through the EM algorithm rather than the subsequent log-likelihoods. The default is `TRUE`.

`solve.tolerance`
When standard errors are calculated, this option specifies the tolerance of the matrix inversion operation solve.

Details

This function allows users to perform multivariate regression analysis with the randomized response item as a predictor for a separate outcome of interest. It does so by jointly modeling the randomized response item as both outcome and predictor for an additional outcome given the same set of covariates. Four standard designs are accepted by this function: mirrored question, forced response, disguised response, and unrelated question.

Value

`rrreg.predictor` returns an object of class "rrpredreg" associated with the randomized response item as predictor. The object `rrpredreg` is a list that contains the following components (the inclusion of some components such as the design parameters are dependent upon the design used):

`est.t`	Point estimates for the effects of the randomized response item as predictor and other covariates on the separate outcome variable specified in the formula.
`se.t`	Standard errors for estimates of the effects of the randomized response item as predictor and other covariates on the separate outcome variable specified in formula.
`est.b`	Point estimates for the effects of covariates on the randomized response item.
`vcov`	Variance-covariance matrix for estimates of the effects of the randomized response item as predictor and other covariates on the separate outcome variable specified in formula as well as for estimates of the effects of covariates on the randomized response item.
`se.b`	Standard errors for estimates of the effects of covariates on the randomized response item.
`data`	The `data` argument.
`coef.names`	Variable names as defined in the data frame.
`x`	The model matrix of covariates.
`y`	The randomized response vector.
`o`	The separate outcome of interest vector.
`design`	Call of standard design used: "forced-known", "mirrored", "disguised", or "unrelated-known".
`p`	The `p` argument.
`p0`	The `p0` argument.
`p1`	The `p1` argument.
`q`	The `q` argument.
`call`	The matched call.

References

Blair, Graeme, Kosuke Imai and Yang-Yang Zhou. (2014) "Design and Analysis of the Randomized Response Technique." *Working Paper.* Available at http://imai.princeton.edu/research/randresp.html.

See Also

`rrreg` for multivariate regression.

Examples

```
data(nigeria)

## Define design parameters

set.seed(44)

p <- 2/3  # probability of answering honestly in Forced Response Design
p1 <- 1/6 # probability of forced 'yes'
p0 <- 1/6 # probability of forced 'no'

## Fit joint model of responses to an outcome regression of joining a civic
## group and the randomized response item of having a militant social connection
rr.q1.pred.obj <-
    rrreg.predictor(civic ~ cov.asset.index + cov.married + I(cov.age/10) +
              I((cov.age/10)^2) + cov.education + cov.female
              + rr.q1, rr.item = "rr.q1", parstart = FALSE, estconv = TRUE,
              data = nigeria, verbose = FALSE, optim = TRUE,
              p = p, p1 = p1, p0 = p0, design = "forced-known")

summary(rr.q1.pred.obj)

## Replicates Table 4 in Blair, Imai, and Zhou (2014)
```

Index

∗ **analysis**
power.rr.plot, 5
power.rr.test, 7
∗ **dataset**
nigeria, 3
nigeria-data, 4
∗ **fitted**
predict.rrreg, 9
predict.rrreg.predictor, 11
∗ **joint**
rrreg.predictor, 20
∗ **model**
rrreg.predictor, 20
∗ **package**
rr-package, 2
∗ **power**
power.rr.plot, 5
power.rr.test, 7
∗ **predicted**
predict.rrreg, 9
predict.rrreg.predictor, 11
∗ **predictor**
rrreg.predictor, 20
∗ **probabilities**
predict.rrreg, 9
predict.rrreg.predictor, 11
∗ **regression**
rrreg, 14
rrreg.predictor, 20
∗ **values**
predict.rrreg, 9
predict.rrreg.predictor, 11

nigeria, 3
nigeria-data, 4

power.rr.plot, 5
power.rr.test, 7
predict.rrreg, 9, *16*
predict.rrreg.predictor, 11
rr *(*rr-package*)*, 2
rr-package, 2
rrreg, *11*, 14, *22*
rrreg.bayes, 16
rrreg.predictor, *13*, 20