# Package: list (via r-universe)

October 26, 2026

**Version** 9.2.6

**Title** Statistical Methods for the Item Count Technique and List Experiment

**Depends** R (>= 3.2.0), utils, sandwich (>= 2.3-3)

**Imports** VGAM (>= 0.9-8), magic (>= 1.5-6), gamlss.dist (>= 4.3-4), MASS (>= 7.3-40), quadprog (>= 1.5-5), corpcor (>= 1.6.7), mvtnorm (>= 1.0-2), coda (>= 0.17-1), stats, arm

**Suggests** testthat (>= 0.9.1), knitr (>= 1.10.5)

**VignetteBuilder** knitr

**Description** Allows researchers to conduct multivariate statistical analyses of survey data with list experiments. This survey methodology is also known as the item count technique or the unmatched count technique and is an alternative to the commonly used randomized response method. The package implements the methods developed by Imai (2011) <doi:10.1198/jasa.2011.ap10415>, Blair and Imai (2012) <doi:10.1093/pan/mpr048>, Blair, Imai, and Lyall (2013) <doi:10.1111/ajps.12086>, Imai, Park, and Greene (2014) <doi:10.1093/pan/mpu017>, Aronow, Coppock, Crawford, and Green (2015) <doi:10.1093/jssam/smu023>, Chou, Imai, and Rosenfeld (2017) <doi:10.1177/0049124117729711>, and Blair, Chou, and Imai (2018) <https://imai.fas.harvard.edu/research/files/listerror.pdf>. This includes a Bayesian MCMC implementation of regression for the standard and multiple sensitive item list experiment designs and a random effects setup, a Bayesian MCMC hierarchical regression model with up to three hierarchical groups, the combined list experiment and endorsement experiment regression model, a joint model of the list experiment that enables the analysis of the list experiment as a predictor in outcome regression models, a method for combining list experiments with direct questions, and methods for diagnosing and adjusting for response error. In addition, the package implements the statistical test that is designed to detect

certain failures of list experiments, and a placebo test for the list experiment using data from direct questions.

**LazyLoad** yes

**LazyData** yes

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**NeedsCompilation** yes

**Author** Graeme Blair [aut, cre], Winston Chou [aut], Kosuke Imai [aut], Bethany Park [ctb], Alexander Coppock [ctb]

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

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

**Date/Publication** 2024-01-11 02:41:43 UTC

**RemoteUrl** https://github.com/cran/list

**RemoteRef** HEAD

**RemoteSha** 2170f2628ef2b549701f62d47c7a9962c842a403

# Contents

affirm . . . 3
combinedListDirect . . . 3
combinedListExps . . . 5
comp.listEndorse . . . 5
ict.hausman.test . . . 6
ict.test . . . 7
ictreg . . . 8
ictreg.joint . . . 16
ictregBayes . . . 19
ictregBayesHier . . . 27
mexico . . . 35
mis . . . 36
multi . . . 37
plot.predict.ictreg . . . 38
predict.ictreg . . . 41
predict.ictreg.joint . . . 45
predict.ictregBayes . . . 49
predict.ictregBayesHier . . . 51
race . . . 53
summary.ictreg . . . 54

**Index** **56**

## affirm *The 1991 National Race and Politics Survey*

### Description

This dataset is a subset of the 1991 National Race and Politics Survey and contains the item count technique or the list experiment. The main question reads as follows: `Now I'm going to read you four things that sometimes make people angry or upset. After I read all (three/four), just tell me HOW MANY of them upset you. (I don't want to know which ones, just how many.) (1) "the federal government increasing the tax on gasoline;" (2) "professional athletes getting million-dollar-plus salaries;" (3) "large corporations polluting the environment;" (4) "black leaders asking the government for affirmative action."`

### Format

A data frame containing the following 6 variables for 1171 observations.

| | | | |
|---|---|---|---|
| y | numeric | the number of items that make respondents angry | 0 - 4 |
| south | numeric | whether or not a respondents live in a southern state | 0 - 1 |
| male | numeric | whether or not a respondent is male | 0 - 1 |
| college | numeric | whether or not a respondent attended some college | 0 - 1 |
| age | numeric | age of a respondent divided by 10 | |
| treat | numeric | treatment status | 0 - 1 |

### Details

where the last item is presented only with the treatment group and the control list only contains the first three items.

### Source

The full data set is available at SDA (Survey Documentation and Analysis; https://sda.berkeley.edu/D3/Natlrace/Doc/nrac.htm)

## combinedListDirect *Combined List Estimator*

### Description

This function implements the combined list estimator described in Aronow, Coppock, Crawford, and Green (2015): Combining List Experiment and Direct Question Estimates of Sensitive Behavior Prevalence

## Usage

```
combinedListDirect(
  formula,
  data = parent.frame(),
  treat = "treat",
  direct = "direct"
)
```

## Arguments

| | |
|---|---|
| formula | an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. Should be of the form Y ~ T + X1 + X2, where Y is the list response, T is the treatment indicator, and X1, X2, etc are pretreatment covariates. It is recommended that T be a numeric variable whose values are 0 for subjects in control and 1 for subjects in treatment. |
| data | an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which combined.list is called. It is good practice to include all variables used in the estimation (list response, treatment indicator, direct response, and optional pre-treatment covariates) in a dataframe rather than calling data from the global environent. |
| treat | a character string giving the name of the treatment variable. Defaults to "treat". |
| direct | a character string giving the name of the direct response variable. Defaults to "direct". The direct response variable itself must only contain the values 0 and 1, where 1 refers to subjects who answered "Yes" to the direct question. |

## Value

a list containing conventional, direct, and combined prevalence estimates with associated standard errors as well as the results of two placebo tests.

## Examples

```
# Load data from Aronow, Coppock, Crawford, and Green (2015)
data("combinedListExps")

# complete case analysis
combinedListExps <- na.omit(combinedListExps)
# Conduct estimation without covariate adjustment
out.1 <- combinedListDirect(list1N ~ list1treat,
                            data = subset(combinedListExps, directsfirst==1),
                            treat = "list1treat", direct = "direct1")
summary(out.1)

# Conduct estimation with covariate adjustment
out.2 <- combinedListDirect(list1N ~ list1treat + gender +
                            ideology + education + race,
                            data = subset(combinedListExps, directsfirst==1),
```

```
                              treat = "list1treat", direct = "direct1")
summary(out.2)
```

---

`combinedListExps` *Five List Experiments with Direct Questions*

---

## Description

A dataset containing the five list experiments in Aronow, Coppock, Crawford, and Green (2015)

A dataset containing the five list experiments in Aronow, Coppock, Crawford, and Green (2015)

## Usage

```
data(combinedListExps)
```

## Format

A data frame with 1023 observations and 23 variables

A data frame with 1023 observations and 23 variables

---

`comp.listEndorse` *Comparing List and Endorsement Experiment Data*

---

## Description

Function to conduct a statistical test with the null hypothesis that there is no difference between the correlation coefficients between list experiment and endorsement experiment data.

## Usage

```
comp.listEndorse(
  y.endorse,
  y.list,
  treat,
  n.draws = 10000,
  alpha = 0.05,
  endorse.mean = FALSE,
  method = "pearson"
)
```

### Arguments

| | |
|---|---|
| y.endorse | A numerical matrix containing the response data for the endorsement experiment. |
| y.list | A numerical vector containing the response data for a list experiment. |
| treat | A numerical vector containing the binary treatment status for the experiments. The treatment assignment must be the same for both experiments to compare across experiments. |
| n.draws | Number of Monte Carlo draws. |
| alpha | Confidence level for the statistical test. |
| endorse.mean | A logical value indicating whether the mean endorsement experiment response is taken across questions. |
| method | The method for calculating the correlation, either Pearson's rho or Kendall's tau. |

### Details

This function allows the user to calculate the correlation between list and endorsement experiment data within the control group and the treatment group, and to conduct a statistical test with the null hypothesis of no difference between the two correlation coefficients.

### Value

comp.listEndorse returns a list with four elements: the correlation statistic (rho or tau) for the treatment group as cor.treat, the correlation statistic for the control group as cor.control, the p.value for the statistical test comparing the two correlation statistics as p.value, and the bootstrapped confidence interval of the difference as ci.

### Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

### References

Blair, Graeme, Jason Lyall and Kosuke Imai. (2014) "Comparing and Combining List and Experiments: Evidence from Afghanistan." American Journal of Political Science. available at http://imai.princeton.edu/research/comp.html

---

## ict.hausman.test *Hausman Specification Test for Two List Experiment Regression Fit Objects*

---

### Description

Hausman Specification Test for Two List Experiment Regression Fit Objects

### Usage

```
ict.hausman.test(ml, nls, abs = FALSE, psd = FALSE)
```

### Arguments

| | |
|---|---|
| ml | Maximum likelihood model fit, for example from ictreg(method = "ml") |
| nls | NLS model fit, for example from ictreg(method = "nls") |
| abs | Flag to override error when Hausman statistic is negative, which may indicate misspecification. Set to FALSE to override. |
| psd | Flag to override error when variance-covariance difference is non-positive semidefinite, suggesting misspecification. Set to TRUE to override. |

### Value

List containing Hausman statistic, p-value, and the degrees of freedom of the test.

---

## ict.test *Item Count Technique*

---

### Description

Function to conduct a statistical test with the null hypothesis that there is no "design effect" in a list experiment, a failure of the experiment.

### Usage

```
ict.test(
  y,
  treat,
  J = NA,
  alpha = 0.05,
  n.draws = 250000,
  gms = TRUE,
  pi.table = TRUE
)
```

### Arguments

| | |
|---|---|
| y | A numerical vector containing the response data for a list experiment. |
| treat | A numerical vector containing the binary treatment status for a list experiment. |
| J | Number of non-sensitive (control) survey items. |
| alpha | Confidence level for the statistical test. |
| n.draws | Number of Monte Carlo draws. |
| gms | A logical value indicating whether the generalized moment selection procedure should be used. |
| pi.table | A logical value indicating whether a table of estimated proportions of respondent types with standard errors is displayed. |

## Details

This function allows the user to perform a statistical test on data from a list experiment or item count technique with the null hypothesis of no design effect. A design effect occurs when an individual's response to the non-sensitive items changes depending upon the respondent's treatment status.

## Value

`ict.test` returns a numerical scalar with the Bonferroni-corrected minimum p-value of the statistical test.

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) "Statistical Analysis of List Experiments." Political Analysis, Vol. 20, No 1 (Winter). available at http://imai.princeton.edu/research/listP.html

## See Also

`ictreg` for list experiment regression based on the assumption of no design effect

## Examples

```
data(affirm)
data(race)

# Conduct test with null hypothesis that there is no design effect
# Replicates results on Blair and Imai (2012) pg. 69

test.value.affirm <- ict.test(affirm$y, affirm$treat, J = 3, gms = TRUE)
print(test.value.affirm)

test.value.race <- ict.test(race$y, race$treat, J = 3, gms = TRUE)
print(test.value.race)
```

---

# ictreg *Item Count Technique*

---

## Description

Function to conduct multivariate regression analyses of survey data with the item count technique, also known as the list experiment and the unmatched count technique.

## Usage

```
ictreg(
  formula,
  data = parent.frame(),
  treat = "treat",
  J,
  method = "ml",
  weights,
  h = NULL,
  group = NULL,
  matrixMethod = "efficient",
  robust = FALSE,
  error = "none",
  overdispersed = FALSE,
  constrained = TRUE,
  floor = FALSE,
  ceiling = FALSE,
  ceiling.fit = "glm",
  floor.fit = "glm",
  ceiling.formula = ~1,
  floor.formula = ~1,
  fit.start = "lm",
  fit.sensitive = "glm",
  fit.nonsensitive = "nls",
  multi.condition = "none",
  maxIter = 5000,
  verbose = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| formula | An object of class "formula": a symbolic description of the model to be fitted. |
| data | A data frame containing the variables in the model |
| treat | Name of treatment indicator as a string. For single sensitive item models, this refers to a binary indicator, and for multiple sensitive item models it refers to a multi-valued variable with zero representing the control condition. This can be an integer (with 0 for the control group) or a factor (with "control" for the control group). |
| J | Number of non-sensitive (control) survey items. |
| method | Method for regression, either `ml` for the Maximum Likelihood (ML) estimation with the Expectation-Maximization algorithm; `lm` for linear model estimation; or `nls` for the Non-linear Least Squares (NLS) estimation with the two-step procedure. |
| weights | Name of the weights variable as a string, if weighted regression is desired. Not implemented for the ceiling/floor models, multiple sensitive item design, or for the modified design. |

| | |
|---|---|
| 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. |
| robust | Robust NLS and ML models that ensure that the estimated proportion of the sensitive trait is close to difference-in-means estimate. |
| error | ML models that model response error processes proposed in Blair, Chou, and Imai (2018). Select either none (standard ML), the default; topcode, which models an error process in which a random subset of respondents chooses the maximal (ceiling) response value, regardless of their truthful response; and uniform, which models an error process in which a subset of respondents chooses their responses at random. |
| overdispersed | Indicator for the presence of overdispersion. If TRUE, the beta-binomial model is used in the EM algorithm, if FALSE the binomial model is used. Not relevant for the NLS or lm methods. |
| constrained | A logical value indicating whether the control group parameters are constrained to be equal. Not relevant for the NLS or lm methods |
| floor | A logical value indicating whether the floor liar model should be used to adjust for the possible presence of respondents dishonestly reporting a negative preference for the sensitive item among those who hold negative views of all the non-sensitive items. |
| ceiling | A logical value indicating whether the ceiling liar model should be used to adjust for the possible presence of respondents dishonestly reporting a negative preference for the sensitive item among those who hold affirmative views of all the non-sensitive items. |
| ceiling.fit | Fit method for the M step in the EM algorithm used to fit the ceiling liar model. glm uses standard logistic regression, while bayesglm uses logistic regression with a weakly informative prior over the parameters. |
| floor.fit | Fit method for the M step in the EM algorithm used to fit the floor liar model. glm uses standard logistic regression, while bayesglm uses logistic regression with a weakly informative prior over the parameters. |
| ceiling.formula | Covariates to include in ceiling liar model. These must be a subset of the covariates used in formula. |
| floor.formula | Covariates to include in floor liar model. These must be a subset of the covariates used in formula. |
| fit.start | Fit method for starting values for standard design ml model. The options are lm, glm, and nls, which use OLS, logistic regression, and non-linear least squares to generate starting values, respectively. The default is nls. |

`fit.sensitive` Fit method for the sensitive item fit for maximum likelihood models. `glm` uses standard logistic regression, while `bayesglm` uses logistic regression with a weakly informative prior over the parameters.

`fit.nonsensitive` Fit method for the non-sensitive item fit for the `nls` method and the starting values for the `ml` method for the `modified` design. Options are `glm` and `nls`, and the default is `nls`.

`multi.condition` For the multiple sensitive item design, covariates representing the estimated count of affirmative responses for each respondent can be included directly as a level variable by choosing `level`, or as indicator variables for each value but one by choosing `indicators`. The default is `none`.

`maxIter` Maximum number of iterations for the Expectation-Maximization algorithm of the ML estimation. The default is 5000.

`verbose` a logical value indicating whether model diagnostics are printed out during fitting.

`...` further arguments to be passed to NLS regression commands.

## Details

This function allows the user to perform regression analysis on data from the item count technique, also known as the list experiment and the unmatched count technique.

Three list experiment designs are accepted by this function: the standard design; the multiple sensitive item standard design; and the modified design proposed by Corstange (2009).

For the standard design, three methods are implemented in this function: the linear model; the Maximum Likelihood (ML) estimation for the Expectation-Maximization (EM) algorithm; the nonlinear least squares (NLS) estimation with the two-step procedure both proposed in Imai (2010); and the Maximum Likelihood (ML) estimator in the presence of two types of dishonest responses, "ceiling" and "floor" liars. The ceiling model, floor model, or both, as described in Blair and Imai (2010) can be activated by using the `ceiling` and `floor` options. The constrained and unconstrained ML models presented in Imai (2010) are available through the `constrained` option, and the user can specify if overdispersion is present in the data for the no liars models using the `overdispersed` option to control whether a beta-binomial or binomial model is used in the EM algorithm to model the item counts.

The modified design and the multiple sensitive item design are automatically detected by the function, and only the binomial model without overdispersion is available.

## Value

`ictreg` returns an object of class "ictreg". The function `summary` is used to obtain a table of the results. The object `ictreg` is a list that contains the following components. Some of these elements are not available depending on which method is used (`lm`, `nls` or `ml`), which design is used (`standard`, `modified`), whether multiple sensitive items are include (`multi`), and whether the constrained model is used (`constrained = TRUE`).

`par.treat` point estimate for effect of covariate on item count fitted on treatment group

| | |
|---|---|
| se.treat | standard error for estimate of effect of covariate on item count fitted on treatment group |
| par.control | point estimate for effect of covariate on item count fitted on control group |
| se.control | standard error for estimate of effect of covariate on item count fitted on control group |
| coef.names | variable names as defined in the data frame |
| design | call indicating whether the standard design as proposed in Imai (2010) or thee modified design as proposed in Corstange (2009) is used |
| method | call of the method used |
| overdispersed | call indicating whether data is overdispersed |
| constrained | call indicating whether the constrained model is used |
| boundary | call indicating whether the floor/ceiling boundary models are used |
| multi | indicator for whether multiple sensitive items were included in the data frame |
| call | the matched call |
| data | the data argument |
| x | the design matrix |
| y | the response vector |
| treat | the vector indicating treatment status |
| J | Number of non-sensitive (control) survey items set by the user or detected. |
| treat.labels | a vector of the names used by the treat vector for the sensitive item or items. This is the names from the treat indicator if it is a factor, or the number of the item if it is numeric. |
| control.label | a vector of the names used by the treat vector for the control items. This is the names from the treat indicator if it is a factor, or the number of the item if it is numeric. |

For the maximum likelihood models, an additional output object is included:

| | |
|---|---|
| pred.post | posterior predicted probability of answering "yes" to the sensitive item. The weights from the E-M algorithm. |

For the floor/ceiling models, several additional output objects are included:

| | |
|---|---|
| ceiling | call indicating whether the assumption of no ceiling liars is relaxed, and ceiling parameters are estimated |
| par.ceiling | point estimate for effect of covariate on whether respondents who answered affirmatively to all non-sensitive items and hold a true affirmative opinion toward the sensitive item lied and reported a negative response to the sensitive item |
| se.ceiling | standard error for estimate for effect of covariate on whether respondents who answered affirmatively to all non-sensitive items and hold a true affirmative opinion toward the sensitive item lied and reported a negative response to the sensitive item |
| floor | call indicating whether the assumption of no floor liars is relaxed, and floor parameters are estimated |

`par.ceiling` point estimate for effect of covariate on whether respondents who answered negatively to all non-sensitive items and hold a true affirmative opinion toward the sensitive item lied and reported a negative response to the sensitive item

`se.ceiling` standard error for estimate for effect of covariate on whether respondents who answered negatively to all non-sensitive items and hold a true affirmative opinion toward the sensitive item lied and reported a negative response to the sensitive item

`coef.names.ceiling` variable names from the ceiling liar model fit, if applicable

`coef.names.floor` variable names from the floor liar model fit, if applicable

For the multiple sensitive item design, the `par.treat` and `se.treat` vectors are returned as lists of vectors, one for each sensitive item.

For the unconstrained model, the `par.control` and `se.control` output is replaced by:

`par.control.phi0` point estimate for effect of covariate on item count fitted on treatment group

`se.control.phi0` standard error for estimate of effect of covariate on item count fitted on treatment group

`par.control.phi1` point estimate for effect of covariate on item count fitted on treatment group

`se.control.phi1` standard error for estimate of effect of covariate on item count fitted on treatment group

Depending upon the estimator requested by the user, model fit statistics are also included:

`llik` the log likelihood of the model, if `ml` is used

`resid.se` the residual standard error, if `nls` or `lm` are used. This will be a scalar if the standard design was used, and a vector if the multiple sensitive item design was used

`resid.df` the residual degrees of freedom, if `nls` or `lm` are used. This will be a scalar if the standard design was used, and a vector if the multiple sensitive item design was used

When using the auxiliary data functionality, the following objects are included:

`aux` logical value indicating whether estimation incorporates auxiliary moments

`nh` integer count of the number of auxiliary moments

`wm` procedure used to estimate the optimal weight matrix

`J.stat` numeric value of the Sargan Hansen overidentifying restriction test statistic

`overid.p` corresponding p-value for the Sargan Hansen test

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) “Statistical Analysis of List Experiments." Political Analysis. Forthcoming. available at http://imai.princeton.edu/research/listP.html

Imai, Kosuke. (2011) “Multivariate Regression Analysis for the Item Count Technique.” Journal of the American Statistical Association, Vol. 106, No. 494 (June), pp. 407-416. available at http://imai.princeton.edu/research/list.html

## See Also

predict.ictreg for fitted values

## Examples

```
data(race)

set.seed(1)

# Calculate list experiment difference in means

diff.in.means.results <- ictreg(y ~ 1, data = race,
               treat = "treat", J=3, method = "lm")

summary(diff.in.means.results)

# Fit linear regression
# Replicates Table 1 Columns 1-2 Imai (2011); note that age is divided by 10

lm.results <- ictreg(y ~ south + age + male + college, data = race,
               treat = "treat", J=3, method = "lm")

summary(lm.results)

# Fit two-step non-linear least squares regression
# Replicates Table 1 Columns 3-4 Imai (2011); note that age is divided by 10

nls.results <- ictreg(y ~ south + age + male + college, data = race,
               treat = "treat", J=3, method = "nls")

summary(nls.results)

## Not run:

# Fit EM algorithm ML model with constraint
# Replicates Table 1 Columns 5-6, Imai (2011); note that age is divided by 10

ml.constrained.results <- ictreg(y ~ south + age + male + college, data = race,
              treat = "treat", J=3, method = "ml",
     overdispersed = FALSE, constrained = TRUE)

summary(ml.constrained.results)
```

```
# Fit EM algorithm ML model with no constraint
# Replicates Table 1 Columns 7-10, Imai (2011); note that age is divided by 10

ml.unconstrained.results <- ictreg(y ~ south + age + male + college, data = race,
          treat = "treat", J=3, method = "ml",
       overdispersed = FALSE, constrained = FALSE)

summary(ml.unconstrained.results)

# Fit EM algorithm ML model for multiple sensitive items
# Replicates Table 3 in Blair and Imai (2010)

multi.results <- ictreg(y ~ male + college + age + south + south:age, treat = "treat",
          J = 3, data = multi, method = "ml",
   multi.condition = "level")

summary(multi.results)

# Fit standard design ML model
# Replicates Table 7 Columns 1-2 in Blair and Imai (2010)

noboundary.results <- ictreg(y ~ age + college + male + south, treat = "treat",
              J = 3, data = affirm, method = "ml",
        overdispersed = FALSE)

summary(noboundary.results)

# Fit standard design ML model with ceiling effects alone
# Replicates Table 7 Columns 3-4 in Blair and Imai (2010)

ceiling.results <- ictreg(y ~ age + college + male + south, treat = "treat",
        J = 3, data = affirm, method = "ml", fit.start = "nls",
     ceiling = TRUE, ceiling.fit = "bayesglm",
     ceiling.formula = ~ age + college + male + south)

summary(ceiling.results)

# Fit standard design ML model with floor effects alone
# Replicates Table 7 Columns 5-6 in Blair and Imai (2010)

floor.results <- ictreg(y ~ age + college + male + south, treat = "treat",
          J = 3, data = affirm, method = "ml", fit.start = "glm",
   floor = TRUE, floor.fit = "bayesglm",
   floor.formula = ~ age + college + male + south)

summary(floor.results)

# Fit standard design ML model with floor and ceiling effects
# Replicates Table 7 Columns 7-8 in Blair and Imai (2010)

both.results <- ictreg(y ~ age + college + male + south, treat = "treat",
              J = 3, data = affirm, method = "ml",
```

```
          floor = TRUE, ceiling = TRUE,
          floor.fit = "bayesglm", ceiling.fit = "bayesglm",
          floor.formula = ~ age + college + male + south,
          ceiling.formula = ~ age + college + male + south)

summary(both.results)

# Response error models (Blair, Imai, and Chou 2018)

top.coded.error <- ictreg(
   y ~ age + college + male + south, treat = "treat",
   J = 3, data = race, method = "ml", error = "topcoded")

uniform.error <- ictreg(
   y ~ age + college + male + south, treat = "treat",
   J = 3, data = race, method = "ml", error = "topcoded")

# Robust models, which constrain sensitive item proportion
#   to difference-in-means estimate

robust.ml <- ictreg(
   y ~ age + college + male + south, treat = "treat",
   J = 3, data = affirm, method = "ml", robust = TRUE)

robust.nls <- ictreg(
   y ~ age + college + male + south, treat = "treat",
   J = 3, data = affirm, method = "nls", robust = TRUE)


## End(Not run)
```

---

`ictreg.joint` *Item Count Technique: Outcome Models*

---

## Description

Function to conduct multivariate regression analyses of survey data with the item count technique, also known as the list experiment, using predicted responses from list experiments as predictors in outcome regression models.

## Usage

```
ictreg.joint(
  formula,
  data = parent.frame(),
  treat = "treat",
  J,
  outcome = "outcome",
```

```
  outcome.reg = "logistic",
  constrained = FALSE,
  maxIter = 1000
)
```

## Arguments

| | |
|---|---|
| formula | An object of class "formula": a symbolic description of the model to be fitted. |
| data | A data frame containing the variables in the model |
| treat | Name of treatment indicator as a string. For single sensitive item models, this refers to a binary indicator, and for multiple sensitive item models it refers to a multi-valued variable with zero representing the control condition. This can be an integer (with 0 for the control group) or a factor (with "control" for the control group). |
| J | Number of non-sensitive (control) survey items. |
| outcome | Name of outcome indicator as a string. |
| outcome.reg | Model for outcome regression. Options are "logistic" or "linear;" default is "logistic". |
| constrained | A logical value indicating whether the control group parameters are constrained to be equal. Default is FALSE. |
| maxIter | Maximum number of iterations for the Expectation-Maximization algorithm of the ML estimation. The default is 1000. |

## Details

This function allows the user to perform regression analysis on survey data with the item count technique, also known as the list experiment, using predicted responses from list experiments as predictors in outcome regression models.

## Value

`ictreg.joint` returns an object of class "ictreg.joint". The function `summary` is used to obtain a table of the results. The object `ictreg.joint` is a list that contains the following components.

| | |
|---|---|
| par.treat | point estimate for effect of covariate on item count fitted on treatment group |
| se.treat | standard error for estimate of effect of covariate on item count fitted on treatment group |
| par.control | point estimate for effect of covariate on item count fitted on control group |
| se.control | standard error for estimate of effect of covariate on item count fitted on control group |
| par.outcome | point estimate for effect of covariate and sensitive item on outcome |
| se.outcome | standard error for estimate of effect of covariate and sensitive item on outcome |
| coef.names | variable names as defined in the data frame |
| constrained | call indicating whether the constrained model is used |
| call | the matched call |

| | |
|---|---|
| data | the data argument |
| outcome.reg | the outcome.reg argument |
| x | the design matrix |
| y | the response vector |
| treat | the vector indicating treatment status |
| J | Number of non-sensitive (control) survey items set by the user or detected. |
| treat.labels | a vector of the names used by the treat vector for the sensitive item or items. This is the names from the treat indicator if it is a factor, or the number of the item if it is numeric. |
| control.label | a vector of the names used by the treat vector for the control items. This is the names from the treat indicator if it is a factor, or the number of the item if it is numeric. |

## References

Imai, Kosuke, Bethany Park, and Kenneth F. Greene. (2014) "Using the Predicted Responses from List Experiments as Explanatory Variables in Regression Models." available at http://imai.princeton.edu/research/files/listExp.pdf

## Examples

```
## Not run:
data(mexico)
loyal <- mexico[mexico$mex.loyal == 1,]
notloyal <- mexico[mexico$mex.loyal == 0,]

## Logistic outcome regression
## (effect of vote-selling on turnout)
## This replicates Table 4 in Imai et al. 2014

loyalreg <- ictreg.joint(formula = mex.y.all ~ mex.male + mex.age + mex.age2 + mex.education +
                         mex.interest + mex.married +
                 mex.wealth + mex.urban + mex.havepropoganda + mex.concurrent, data = loyal,
                     treat = "mex.t", outcome = "mex.votecard", J = 3, constrained = TRUE,
                         outcome.reg = "logistic", maxIter = 1000)
summary(loyalreg)

## Linear outcome regression
## (effect of vote-selling on candidate approval)
## This replicates Table 5 in Imai et al. 2014

approvalreg <- ictreg.joint(formula = mex.y.all ~ mex.male + mex.age + mex.age2 +
                            mex.education +
                            mex.interest + mex.married +
                            mex.urban +
                            mex.cleanelections + mex.cleanelectionsmiss +
                            mex.havepropoganda +
                            mex.wealth + mex.northregion +
                            mex.centralregion + mex.metro + mex.pidpriw2 +
```

```
        mex.pidpanw2 + mex.pidprdw2,
                             data = mexico, treat = "mex.t", outcome = "mex.epnapprove",
                             J = 3, constrained = TRUE,
                             outcome.reg = "linear", maxIter = 1000)


summary(approvalreg)

## End(Not run)
```

---

ictregBayes *Item Count Technique*

---

## Description

Function to conduct multivariate regression analyses of survey data with the item count technique, also known as the list experiment and the unmatched count technique.

## Usage

```
ictregBayes(
  formula,
  data = parent.frame(),
  treat = "treat",
  J,
  constrained.single = "full",
  constrained.multi = TRUE,
  fit.start = "lm",
  n.draws = 10000,
  burnin = 5000,
  thin = 0,
  delta.start,
  psi.start,
  Sigma.start,
  Phi.start,
  delta.mu0,
  psi.mu0,
  delta.A0,
  psi.A0,
  Sigma.df,
  Sigma.scale,
  Phi.df,
  Phi.scale,
  delta.tune,
  psi.tune,
  gamma.tune,
  zeta.tune,
```

```
  formula.mixed,
  group.mixed,
  verbose = TRUE,
  sensitive.model = "logit",
  df = 5,
  endorse.options,
  ...
)
```

## Arguments

| | |
|---|---|
| formula | An object of class "formula": a symbolic description of the model to be fitted. |
| data | A data frame containing the variables in the model |
| treat | Name of treatment indicator as a string. For single sensitive item models, this refers to a binary indicator, and for multiple sensitive item models it refers to a multi-valued variable with zero representing the control condition. This can be an integer (with 0 for the control group) or a factor (with "control" for the control group). |
| J | Number of non-sensitive (control) survey items. This will be set automatically to the maximum value of the outcome variable in the treatment group if no input is sent by the user. |
| constrained.single | A string indicating whether the control group parameters are constrained to be equal in the single sensitive item design, either setting all parameters to be equal (full) or only the intercept (intercept). If neither, set to none. |
| constrained.multi | A logical value indicating whether the non-sensitive item count is included as a predictor in the sensitive item fits for the multiple sensitive item design. |
| fit.start | Fit method for starting values. The options are lm, glm, nls, and ml, which use OLS, logistic regression, non-linear least squares, and maximum likelihood estimation to generate starting values, respectively. The default is lm. |
| n.draws | Number of MCMC iterations after the burnin. |
| burnin | The number of initial MCMC iterations that are discarded. |
| thin | The interval of thinning, in which every other (thin = 1) or more iterations are discarded in the output object |
| delta.start | Optional starting values for the sensitive item fit. This should be a vector with the length of the number of covariates for the single sensitive item design, and either a vector or a list with a vector of starting values for each of the sensitive items. The default runs an ictreg fit with the method set by the fit.start option. |
| psi.start | Optional starting values for the control items fit. This should be a vector of length the number of covariates for the constrained models. The default runs an ictreg fit with the method set by the fit.start option. |
| Sigma.start | Optional starting values for Sigma parameter for mixed effects models for sensitive item. |

| | |
|---|---|
| Phi.start | Optional starting values for the Phi parameter for mixed effects models for control item. |
| delta.mu0 | Optional vector of prior means for the sensitive item fit parameters, a vector of length the number of covariates. |
| psi.mu0 | Optional vector of prior means for the control item fit parameters, a vector of length the number of covariates. |
| delta.A0 | Optional matrix of prior precisions for the sensitive item fit parameters, a matrix of dimension the number of covariates. |
| psi.A0 | Optional matrix of prior precisions for the control items fit parameters, a matrix of dimension the number of covariates. |
| Sigma.df | Optional prior degrees of freedom parameter for mixed effects models for sensitive item. |
| Sigma.scale | Optional prior scale parameter for mixed effects models for sensitive item. |
| Phi.df | Optional prior degress of freedom parameter for mixed effects models for control item. |
| Phi.scale | Optional prior scale parameter for mixed effects models for control item. |
| delta.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.tune | A required vector of tuning parameters for the Metropolis algorithm for the control item fit. This must be set and refined by the user until the acceptance ratios are approximately .4 (reported in the output). |
| gamma.tune | An optional vector of tuning parameters for the Metropolis algorithm for the control item fit for the random effects. This can be set and refined by the user until the acceptance ratios are approximately .4 (reported in the output). |
| zeta.tune | An optional vector of tuning parameters for the Metropolis algorithm for the sensitive item fit for the random effects. This can be set and refined by the user until the acceptance ratios are approximately .4 (reported in the output). |
| 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. |
| group.mixed | A numerical group indicator specifying which group each individual belongs to for a mixed effects model. |
| verbose | A logical value indicating whether model diagnostics are printed out during fitting. |
| sensitive.model | A logical value indicating which model is used for the sensitive item fit, logistic regression (logit, default), robit regression (robit), or probit regression (probit). |
| df | Degrees of freedom for the robit model for the sensitive item fit, only used if robit is set to TRUE. |
| endorse.options | A list of inputs and options for running the combined list experiment and endorsement experiment model. Options documented more fully in endorse package. |
| ... | further arguments to be passed to NLS regression commands. |

## Details

This function allows the user to perform regression analysis on data from the item count technique, also known as the list experiment and the unmatched count technique using a Bayesian MCMC algorithm.

Unlike the maximum likelihood and least squares estimators in the `ictreg` function, the Metropolis algorithm for the Bayesian MCMC estimators in this function must be tuned to work correctly. The `delta.tune` and `psi.tune` are required, and the values, one for each estimated parameter, will need to be manipulated. The output of the `ictregBayes` function, and of the `summary` function run on an `ictregBayes` object display 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.

For the single sensitive item design, the model can constrain all control parameters to be equal (`constrained = "full"`), or just the intercept (`constrained = "intercept"`) or all the control fit parameters can be allowed to vary across the potential sensitive item values (`constrained = "none"`).

For the multiple sensitive item design, the model can include the estimated number of affirmative responses to the control items as a covariate in the sensitive item model fit (`constrained` set to `TRUE`) or exclude it (`FALSE`).

The function also allows the user to perform combined list experiment and endorsement experiment regression. Setting `endorse.options` to a list with the options from the `endorse` package for endorsement experiment regression, the function will return the combined model in which the relationship between covariates and the sensitive item in the list experiment model is set to be identical to the relationship between covariates and support for endorsers in the endorsement experiment model. For more details on endorsement experiment regression, see the help for the `endorse` package.

Convergence is at times difficult to achieve, so we recommend running multiple chains from overdispersed starting values by, for example, running an MLE or linear model using the ictreg() 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

`ictregBayes` returns an object of class "ictregBayes". The function `summary` is used to obtain a table of the results, using the `coda` package. Two attributes are also included, the data ("x"), the call ("call"), which can be extracted using the command, e.g., attr(ictregBayes.object, "x").

| | |
|---|---|
| `mcmc` | an object of class "mcmc" that can be analyzed using the `coda` package. |
| `x` | the design matrix |
| `multi` | a logical value indicating whether the data included multiple sensitive items. |
| `constrained` | a logical or character value indicating whether the control group parameters are constrained to be equal in the single sensitive item design, and whether the non-sensitive item count is included as a predictor in the sensitive item fits for the multiple sensitive item design. |

| | |
|---|---|
| delta.start | Optional starting values for the sensitive item fit. This should be a vector with the length of the number of covariates. The default runs an ictreg fit with the method set by the fit.start option. |
| psi.start | Optional starting values for the control items fit. This should be a vector of length the number of covariates. The default runs an ictreg fit with the method set by the fit.start option. |
| delta.mu0 | Optional vector of prior means for the sensitive item fit parameters, a vector of length the number of covariates. |
| psi.mu0 | Optional vector of prior means for the control item fit parameters, a vector of length the number of covariates. |
| delta.A0 | Optional matrix of prior precisions for the sensitive item fit parameters, a matrix of dimension the number of covariates. |
| psi.A0 | Optional matrix of prior precisions for the control items fit parameters, a matrix of dimension the number of covariates. |
| delta.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.tune | A required vector of tuning parameters for the Metropolis algorithm for the control item fit. This must be set and refined by the user until the acceptance ratios are approximately .4 (reported in the output). |
| J | Number of non-sensitive (control) survey items set by the user or detected. |
| treat.labels | a vector of the names used by the treat vector for the sensitive item or items. This is the names from the treat indicator if it is a factor, or the number of the item if it is numeric. |
| control.label | a vector of the names used by the treat vector for the control items. This is the names from the treat indicator if it is a factor, or the number of the item if it is numeric. |
| call | the matched call |

If the data includes multiple sensitive items, the following object is also included:

| | |
|---|---|
| treat.values | a vector of the values used in the treat vector for the sensitive items, either character or numeric depending on the class of treat. Does not include the value for the control status |

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) "Statistical Analysis of List Experiments." Political Analysis, Vol. 20, No 1 (Winter). available at http://imai.princeton.edu/research/listP.html

Imai, Kosuke. (2011) "Multivariate Regression Analysis for the Item Count Technique." Journal of the American Statistical Association, Vol. 106, No. 494 (June), pp. 407-416. available at http://imai.princeton.edu/research/list.html

Blair, Graeme, Jason Lyall and Kosuke Imai. (2013) "Comparing and Combining List and Experiments: Evidence from Afghanistan." Working paper. available at http://imai.princeton.edu/research/comp.html

## See Also

predict.ictreg for fitted values

## Examples

```
data(race)

## Not run:

## Multiple chain MCMC list experiment regression
## starts with overdispersed MLE starting values

## Standard single sensitive-item design

## Control item parameters fully constrained

mle.estimates <- ictreg(y ~ male + college + age + south, data = race)

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

bayesDraws.1 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[1, 1:5], psi.start = draws[1, 6:10], burnin = 10000,
  n.draws = 100000, delta.tune = diag(.002, 5), psi.tune = diag(.00025, 5),
  constrained.single = "full")

bayesDraws.2 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[2, 1:5], psi.start = draws[2, 6:10], burnin = 10000,
  n.draws = 100000, delta.tune = diag(.002, 5), psi.tune = diag(.00025, 5),
  constrained.single = "full")

bayesDraws.3 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[3, 1:5], psi.start = draws[3, 6:10], burnin = 10000,
  n.draws = 100000, delta.tune = diag(.002, 5), psi.tune = diag(.00025, 5),
  constrained.single = "full")

bayesSingleConstrained <- as.list(bayesDraws.1, bayesDraws.2, bayesDraws.3)

summary(bayesSingleConstrained)

## Control item parameters unconstrained

mle.estimates <- ictreg(y ~ male + college + age + south, data = race,
  constrained = FALSE)

draws <- mvrnorm(n = 3, mu = coef(mle.estimates),
```

```
  Sigma = vcov(mle.estimates) * 9)

bayesDraws.1 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[1, 1:5], psi.start = list(psi0 = draws[1, 6:10],
  psi1 = draws[1, 11:15]), burnin = 10000, n.draws = 100000,
  delta.tune = diag(.002, 5),
  psi.tune = list(psi0 = diag(.0017, 5), psi1 = diag(.00005, 5)),
  constrained.single = "none")

bayesDraws.2 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[2, 1:5], psi.start = list(psi0 = draws[2, 6:10],
  psi1 = draws[2, 11:15]), burnin = 10000, n.draws = 100000,
  delta.tune = diag(.002, 5),
  psi.tune = list(psi0 = diag(.0017, 5), psi1 = diag(.00005, 5)),
  constrained.single = "none")

bayesDraws.3 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[3, 1:5], psi.start = list(psi0 = draws[3, 6:10],
  psi1 = draws[3, 11:15]), burnin = 10000, n.draws = 100000,
  delta.tune = diag(.002, 5),
  psi.tune = list(psi0 = diag(.0017, 5), psi1 = diag(.00005, 5)),
  constrained.single = "none")

bayesSingleUnconstrained <- as.list(bayesDraws.1, bayesDraws.2, bayesDraws.3)

summary(bayesSingleUnconstrained)

## Control item parameters constrained except intercept

mle.estimates <- ictreg(y ~ male + college + age + south, data = race,
  constrained = TRUE)

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

bayesDraws.1 <-  ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[1, 1:5], psi.start = c(draws[1, 6:10],0),
  burnin = 10000, n.draws = 100000, delta.tune = diag(.002, 5),
  psi.tune = diag(.0004, 6), constrained.single = "intercept")

bayesDraws.2 <-  ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[2, 1:5], psi.start = c(draws[2, 6:10],0),
  burnin = 10000, n.draws = 100000, delta.tune = diag(.002, 5),
  psi.tune = diag(.0004, 6), constrained.single = "intercept")

bayesDraws.3 <-  ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[3, 1:5], psi.start = c(draws[3, 6:10],0),
  burnin = 10000, n.draws = 100000, delta.tune = diag(.002, 5),
  psi.tune = diag(.0004, 6), constrained.single = "intercept")

bayesSingleInterceptOnly <- as.list(bayesDraws.1, bayesDraws.2, bayesDraws.3)

summary(bayesSingleInterceptOnly)
```

```
## Multiple sensitive item design

## Constrained (estimated control item count not included in sensitive fit)

mle.estimates.multi <- ictreg(y ~ male + college + age + south, data = multi,
  constrained = TRUE)

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

bayesMultiDraws.1 <- ictregBayes(y ~ male + college + age + south,
  data = multi, delta.start = list(draws[1, 6:10], draws[1, 11:15]),
  psi.start = draws[1, 1:5], burnin = 10000, n.draws = 100000,
  delta.tune = diag(.002, 5), psi.tune = diag(.001, 5),
  constrained.multi = TRUE)

bayesMultiDraws.2 <- ictregBayes(y ~ male + college + age + south,
  data = multi, delta.start = list(draws[2, 6:10], draws[2, 11:15]),
  psi.start = draws[2, 1:5], burnin = 10000, n.draws = 100000,
  delta.tune = diag(.002, 5), psi.tune = diag(.001, 5),
  constrained.multi = TRUE)

bayesMultiDraws.3 <- ictregBayes(y ~ male + college + age + south,
  data = multi, delta.start = list(draws[3, 6:10], draws[3, 11:15]),
  psi.start = draws[3, 1:5], burnin = 10000, n.draws = 100000,
  delta.tune = diag(.002, 5), psi.tune = diag(.001, 5),
  constrained.multi = TRUE)

bayesMultiConstrained <- as.list(bayesMultiDraws.1, bayesMultiDraws.2,
  bayesMultiDraws.3)

summary(bayesMultiConstrained)

## Unconstrained (estimated control item count is included in sensitive fit)

mle.estimates.multi <- ictreg(y ~ male + college + age + south, data = multi,
  constrained = FALSE)

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

bayesMultiDraws.1 <- ictregBayes(y ~ male + college + age + south,
  data = multi, delta.start = list(draws[1, 6:10], draws[1, 11:15]),
  psi.start = draws[1, 1:5], burnin = 50000, n.draws = 300000,
  delta.tune = diag(.0085, 6), psi.tune = diag(.00025, 5),
  constrained.multi = FALSE)

bayesMultiDraws.2 <- ictregBayes(y ~ male + college + age + south,
  data = multi, delta.start = list(draws[2, 6:10], draws[2, 11:15]),
  psi.start = draws[2, 1:5], burnin = 50000, n.draws = 300000,
  delta.tune = diag(.0085, 6), psi.tune = diag(.00025, 5),
  constrained.multi = FALSE)
```

```
bayesMultiDraws.3 <- ictregBayes(y ~ male + college + age + south,
  data = multi, delta.start = list(draws[3, 6:10], draws[3, 11:15]),
  psi.start = draws[3, 1:5], burnin = 50000, n.draws = 300000,
  delta.tune = diag(.0085, 6), psi.tune = diag(.00025, 5),
  constrained.multi = FALSE)

bayesMultiUnconstrained <- as.list(bayesMultiDraws.1, bayesMultiDraws.2,
  bayesMultiDraws.3)

summary(bayesMultiUnconstrained)

## Mixed effects models

## Varying intercepts

mle.estimates <- ictreg(y ~ male + college + age + south, data = race)

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

bayesDraws.1 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[1, 1:5], psi.start = draws[1, 6:10], burnin = 100,
  n.draws = 1000, delta.tune = diag(.002, 5), psi.tune = diag(.00025, 5),
  constrained.single = "full", group.mixed = "state", formula.mixed = ~ 1)

bayesDraws.2 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[2, 1:5], psi.start = draws[2, 6:10], burnin = 10000,
  n.draws = 100000, delta.tune = diag(.002, 5), psi.tune = diag(.00025, 5),
  constrained.single = "full", group.mixed = "state", formula.mixed = ~ 1)

bayesDraws.3 <- ictregBayes(y ~ male + college + age + south, data = race,
  delta.start = draws[3, 1:5], psi.start = draws[3, 6:10], burnin = 10000,
  n.draws = 100000, delta.tune = diag(.002, 5), psi.tune = diag(.00025, 5),
  constrained.single = "full", group.mixed = "state", formula.mixed = ~ 1)

bayesMixed <- as.list(bayesDraws.1, bayesDraws.2, bayesDraws.3)

summary(bayesMixed)


## End(Not run)
```

---

ictregBayesHier *Item Count Technique*

---

## Description

Function to conduct multilevel, multivariate regression analyses of survey data with the item count technique, also known as the list experiment and the unmatched count technique.

## Usage

```
ictregBayesHier(
  formula,
  data = parent.frame(),
  group.level.2,
  group.level.3,
  group.level.4,
  formula.level.2,
  formula.level.3,
  formula.level.4,
  treat = "treat",
  J,
  fit.start = "lm",
  n.draws = 10000,
  burnin = 5000,
  thin = 0,
  delta.start.level.1,
  delta.mu0.level.1,
  delta.A0.level.1,
  delta.start.level.2,
  delta.mu0.level.2,
  delta.A0.level.2,
  delta.start.level.3,
  delta.mu0.level.3,
  delta.A0.level.3,
  delta.start.level.4,
  delta.mu0.level.4,
  delta.A0.level.4,
  sigma.start.level.1,
  sigma.df.level.1,
  sigma.scale.level.1,
  sigma.start.level.2,
  sigma.df.level.2,
  sigma.scale.level.2,
  sigma.start.level.3,
  sigma.df.level.3,
  sigma.scale.level.3,
  sigma.start.level.4,
  sigma.df.level.4,
  sigma.scale.level.4,
  delta.tune,
  alpha.tune,
  verbose = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| formula | An object of class "formula": a symbolic description of the model to be fitted. |
| data | A data frame containing the variables in the model |
| group.level.2 | Name of second level group variable from the data frame indicating which group each individual belongs to as a string |
| group.level.3 | Name of third level group variable from the data frame indicating which group each individual belongs to as a string |
| group.level.4 | Name of fourth level group variable from the data frame indicating which group each individual belongs to as a string |
| formula.level.2 | An object of class "formula" for the second level of the hierarchical model |
| formula.level.3 | An object of class "formula" for the third level of the hierarchical model |
| formula.level.4 | An object of class "formula" for the fourth level of the hierarchical model |
| treat | Name of treatment indicator as a string. For single sensitive item models, this refers to a binary indicator, and for multiple sensitive item models it refers to a multi-valued variable with zero representing the control condition. This can be an integer (with 0 for the control group) or a factor (with "control" for the control group). |
| J | Number of non-sensitive (control) survey items. This will be set automatically to the maximum value of the outcome variable in the treatment group if no input is sent by the user. |
| fit.start | Fit method for starting values. The options are lm, glm, nls, and ml, which use OLS, logistic regression, non-linear least squares, and maximum likelihood estimation to generate starting values, respectively. The default is lm. |
| n.draws | Number of MCMC iterations after the burnin. |
| burnin | The number of initial MCMC iterations that are discarded. |
| thin | The interval of thinning, in which every other (thin = 1) or more iterations are discarded in the output object |
| delta.start.level.1 | Optional starting values for the sensitive item fit. This should be a vector with the length of the number of covariates for the single sensitive item design, and either a vector or a list with a vector of starting values for each of the sensitive items. The default runs an ictreg fit with the method set by the fit.start option. |
| delta.mu0.level.1 | Optional vector of prior means for the sensitive item fit parameters, a vector of length the number of covariates. |
| delta.A0.level.1 | Optional matrix of prior precisions for the sensitive item fit parameters, a matrix of dimension the number of covariates. |

`delta.start.level.2`
: Optional starting values for the sensitive item fit for the second level of the hierarchical model. This should be a vector with the length of the number of covariates for the single sensitive item design, and either a vector or a list with a vector of starting values for each of the sensitive items. The default runs an `ictreg` fit with the method set by the `fit.start` option.

`delta.mu0.level.2`
: Optional vector of prior means for the sensitive item fit parameters for the second level of the hierarchical model, a vector of length the number of covariates.

`delta.A0.level.2`
: Optional matrix of prior precisions for the sensitive item fit parameters for the second level of the hierarchical model, a matrix of dimension the number of covariates.

`delta.start.level.3`
: Optional starting values for the sensitive item fit for the third level of the hierarchical model. This should be a vector with the length of the number of covariates for the single sensitive item design, and either a vector or a list with a vector of starting values for each of the sensitive items. The default runs an `ictreg` fit with the method set by the `fit.start` option.

`delta.mu0.level.3`
: Optional vector of prior means for the sensitive item fit parameters for the third level of the hierarchical model, a vector of length the number of covariates.

`delta.A0.level.3`
: Optional matrix of prior precisions for the sensitive item fit parameters for the third level of the hierarchical model, a matrix of dimension the number of covariates.

`delta.start.level.4`
: Optional starting values for the sensitive item fit for the fourth level of the hierarchical model. This should be a vector with the length of the number of covariates for the single sensitive item design, and either a vector or a list with a vector of starting values for each of the sensitive items. The default runs an `ictreg` fit with the method set by the `fit.start` option.

`delta.mu0.level.4`
: Optional vector of prior means for the sensitive item fit parameters for the fourth level of the hierarchical model, a vector of length the number of covariates.

`delta.A0.level.4`
: Optional matrix of prior precisions for the sensitive item fit parameters for the fourth level of the hierarchical model, a matrix of dimension the number of covariates.

`sigma.start.level.1`
: Optional list of length the number of sensitive items with the starting values for the sigma parameters.

`sigma.df.level.1`
: Optional prior degrees of freedom parameter.

`sigma.scale.level.1`
: Optional prior scale parameter.

`sigma.start.level.2`
: Optional list of length the number of sensitive items with the starting values for the sigma parameters for the second level of the hierarchical model.

`sigma.df.level.2`
: Optional prior degrees of freedom parameter for the second level of the hierarchical model.

`sigma.scale.level.2`
: Optional prior scale parameter for the second level of the hierarchical model.

`sigma.start.level.3`
: Optional list of length the number of sensitive items with the starting values for the sigma parameters for the third level of the hierarchical model.

`sigma.df.level.3`
: Optional prior degrees of freedom parameter for the third level of the hierarchical model.

`sigma.scale.level.3`
: Optional prior scale parameter for the third level of the hierarchical model.

`sigma.start.level.4`
: Optional list of length the number of sensitive items with the starting values for the sigma parameters for the fourth level of the hierarchical model.

`sigma.df.level.4`
: Optional prior degrees of freedom parameter for the fourth level of the hierarchical model.

`sigma.scale.level.4`
: Optional prior scale parameter for the fourth level of the hierarchical model.

`delta.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).

`alpha.tune`
: An optional vector of tuning parameters for the Metropolis algorithm for the random effects.

`verbose`
: A logical value indicating whether model diagnostics are printed out during fitting.

`...`
: further arguments to be passed to NLS regression commands.

## Details

This function allows the user to perform regression analysis on data from the item count technique, also known as the list experiment and the unmatched count technique using a Bayesian MCMC algorithm.

Unlike the maximum likelihood and least squares estimators in the `ictreg` function, the Metropolis algorithm for the Bayesian MCMC estimators in this function must be tuned to work correctly. The `delta.tune` and `psi.tune` are required, and the values, one for each estimated parameter, will need to be manipulated. The output of the `ictregBayes` function, and of the `summary` function run on an `ictregBayes` object display 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.

For the single sensitive item design, the model can constrain all control parameters to be equal (`constrained = "full"`), or just the intercept (`constrained = "intercept"`) or all the control

fit parameters can be allowed to vary across the potential sensitive item values (`constrained = "none"`).

For the multiple sensitive item design, the model can include the estimated number of affirmative responses to the control items as a covariate in the sensitive item model fit (`constrained` set to `TRUE`) or exclude it (`FALSE`).

Convergence is at times difficult to achieve, so we recommend running multiple chains from overdispersed starting values by, for example, running an MLE or linear model using the ictreg() 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

`ictregBayes` returns an object of class "ictregBayes". The function `summary` is used to obtain a table of the results, using the `coda` package. Two attributes are also included, the data ("x"), the call ("call"), which can be extracted using the command, e.g., attr(ictregBayes.object, "x").

| | |
|---|---|
| `mcmc` | an object of class "mcmc" that can be analyzed using the `coda` package. |
| `x` | the design matrix |
| `multi` | a logical value indicating whether the data included multiple sensitive items. |
| `constrained` | a logical or character value indicating whether the control group parameters are constrained to be equal in the single sensitive item design, and whether the non-sensitive item count is included as a predictor in the sensitive item fits for the multiple sensitive item design. |
| `delta.start` | Optional starting values for the sensitive item fit. This should be a vector with the length of the number of covariates. The default runs an `ictreg` fit with the method set by the `fit.start` option. |
| `psi.start` | Optional starting values for the control items fit. This should be a vector of length the number of covariates. The default runs an `ictreg` fit with the method set by the `fit.start` option. |
| `delta.mu0` | Optional vector of prior means for the sensitive item fit parameters, a vector of length the number of covariates. |
| `psi.mu0` | Optional vector of prior means for the control item fit parameters, a vector of length the number of covariates. |
| `delta.A0` | Optional matrix of prior precisions for the sensitive item fit parameters, a matrix of dimension the number of covariates. |
| `psi.A0` | Optional matrix of prior precisions for the control items fit parameters, a matrix of dimension the number of covariates. |
| `delta.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.tune` | A required vector of tuning parameters for the Metropolis algorithm for the control item fit. This must be set and refined by the user until the acceptance ratios are approximately .4 (reported in the output). |

| | |
|---|---|
| J | Number of non-sensitive (control) survey items set by the user or detected. |
| treat.labels | a vector of the names used by the treat vector for the sensitive item or items. This is the names from the treat indicator if it is a factor, or the number of the item if it is numeric. |
| control.label | a vector of the names used by the treat vector for the control items. This is the names from the treat indicator if it is a factor, or the number of the item if it is numeric. |
| call | the matched call |

If the data includes multiple sensitive items, the following object is also included:

| | |
|---|---|
| treat.values | a vector of the values used in the treat vector for the sensitive items, either character or numeric depending on the class of treat. Does not include the value for the control status |

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) "Statistical Analysis of List Experiments." Political Analysis, Vol. 20, No 1 (Winter). available at http://imai.princeton.edu/research/listP.html

Imai, Kosuke. (2011) "Multivariate Regression Analysis for the Item Count Technique." Journal of the American Statistical Association, Vol. 106, No. 494 (June), pp. 407-416. available at http://imai.princeton.edu/research/list.html

## See Also

predict.ictreg for fitted values

## Examples

```
data(race)

## Not run:

## Multiple chain MCMC list experiment regression
## starts with overdispersed MLE starting values

## Multiple item two level hierarchical model - varying intercepts

mle.estimates.multi <- ictreg(y ~ male + college, data = multi,
  constrained = TRUE)

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

bayesDraws.1 <- ictregBayesHier(y ~ male + college,
```

```
                        formula.level.2 = ~ 1,
                  delta.start.level.1 = list(draws[1, 8:9], draws[1, 2:3], draws[1, 5:6]),
                        data = multi, treat = "treat",
                        delta.tune = list(rep(0.005, 2), rep(0.05, 2), rep(0.05, 2)),
                        alpha.tune = rep(0.001, length(unique(multi$state))),
                        J = 3, group.level.2 = "state",
                        n.draws = 100000, burnin = 50000, thin = 100)

bayesDraws.2 <- ictregBayesHier(y ~ male + college,
                        formula.level.2 = ~ 1,
                  delta.start.level.1 = list(draws[2, 8:9], draws[2, 2:3], draws[2, 5:6]),
                        data = multi, treat = "treat",
                        delta.tune = list(rep(0.005, 2), rep(0.05, 2), rep(0.05, 2)),
                        alpha.tune = rep(0.001, length(unique(multi$state))),
                        J = 3, group.level.2 = "state",
                        n.draws = 100000, burnin = 50000, thin = 100)

bayesDraws.3 <- ictregBayesHier(y ~ male + college,
                        formula.level.2 = ~ 1,
                  delta.start.level.1 = list(draws[3, 8:9], draws[3, 2:3], draws[3, 5:6]),
                        data = multi, treat = "treat",
                        delta.tune = list(rep(0.005, 2), rep(0.05, 2), rep(0.05, 2)),
                        alpha.tune = rep(0.001, length(unique(multi$state))),
                        J = 3, group.level.2 = "state",
                        n.draws = 100000, burnin = 50000, thin = 100)

bayesHierTwoLevel <- as.list(bayesDraws.1, bayesDraws.2, bayesDraws.3)

summary(bayesHierTwoLevel)

## Multiple item two level hierarchical model - including covariates

mle.estimates.multi <- ictreg(y ~ male + college, data = multi,
  constrained = TRUE)

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

bayesDraws.1 <- ictregBayesHier(y ~ male + college,
                        formula.level.2 = ~ age,
                  delta.start.level.1 = list(draws[1, 8:9], draws[1, 2:3], draws[1, 5:6]),
                        data = multi, treat = "treat",
                        delta.tune = list(rep(0.005, 2), rep(0.05, 2), rep(0.05, 2)),
                        alpha.tune = rep(0.001, length(unique(multi$state))),
                        J = 3, group.level.2 = "state",
                        n.draws = 100000, burnin = 50000, thin = 100)

bayesDraws.2 <- ictregBayesHier(y ~ male + college,
                        formula.level.2 = ~ age,
                  delta.start.level.1 = list(draws[2, 8:9], draws[2, 2:3], draws[2, 5:6]),
                        data = multi, treat = "treat",
                        delta.tune = list(rep(0.005, 2), rep(0.05, 2), rep(0.05, 2)),
                        alpha.tune = rep(0.001, length(unique(multi$state))),
```

```
                        J = 3, group.level.2 = "state",
                        n.draws = 100000, burnin = 50000, thin = 100)

bayesDraws.3 <- ictregBayesHier(y ~ male + college,
                        formula.level.2 = ~ age,
                  delta.start.level.1 = list(draws[3, 8:9], draws[3, 2:3], draws[3, 5:6]),
                        data = multi, treat = "treat",
                        delta.tune = list(rep(0.005, 2), rep(0.05, 2), rep(0.05, 2)),
                        alpha.tune = rep(0.001, length(unique(multi$state))),
                        J = 3, group.level.2 = "state",
                        n.draws = 100000, burnin = 50000, thin = 100)

bayesHierTwoLevel <- as.list(bayesDraws.1, bayesDraws.2, bayesDraws.3)

summary(bayesHierTwoLevel)


## End(Not run)
```

---

`mexico` *The 2012 Mexico Elections Panel Study*

---

## Description

This dataset is a subset of the 2012 Mexico Elections Panel Study and contains a list experiment question. It reads as follows: `I am going to read you a list of four activities that appear on this card and I want you to tell me how many of these activities you have done in recent weeks. Please don''t tell me which ones, just HOW MANY. The four activities are... (SHOW CARD AND READ) a. See television news that mentions a candidate b. Attend a campaign event c. Exchange your vote for a gift, favor, or access to a service d. Talk about politics with other people`

## Format

A data frame containing the following variables for 1004 observations.

| | | |
|---|---|---|
| y | numeric | the number of items that make respondents angry |
| mex.t | numeric | treatment status |
| mex.male | numeric | whether or not a respondent is male |
| mex.age | numeric | age of a respondent |
| mex.education | numeric | respondent's level of education |
| mex.y.all | numeric | the number of activities that respondent did |
| mex.vote | numeric | respondent's self-reported turnout |
| mex.age2 | numeric | age of a respondent, squared |
| mex.interest | numeric | how interested respondent is in politics |
| mex.married | numeric | indicator for whether respondent is married |

| | | |
|---|---|---|
| mex.pidpanw2 | numeric | indicator for whether respondent identifies with PAN party |
| mex.pidprdw2 | numeric | indicator for whether respondent identifies with PRD party |
| mex.pidpriw2 | numeric | indicator for whether respondent identifies with PRI party |
| mex.votecard | numeric | respondent's enumerator-verified turnout |
| mex.urban | numeric | indicator for whether respondent lives in urban area |
| mex.cleanelections | numeric | indicator for whether respondent thinks elections were clean |
| mex.cleanelectionsmiss | numeric | indicator for whether cleanelections variable was missing |
| mex.metro | numeric | indicator for whether respondent lives in Mexico City metro area |
| mex.centralregion | numeric | indicator for whether respondent lives in Mexico's central region |
| mex.northregion | numeric | indicator for whether respondent lives in Mexico's north region |
| mex.wealth | numeric | scale for respondent's wealth, based on household asset indicators |
| mex.epnapprove | numeric | respondent's approval rating of Enrique Pena-Nieto |
| mex.havepropoganda | numeric | indicator for whether respondent has propaganda outside their home |
| mex.concurrent | numeric | indicator for whether respondent lives in state with concurrent elections |
| mex.loyal | numeric | indicator for whether respondent strongly identifies with the PAN, PRI, or PRD party |
| mex.direct | numeric | indicator for whether respondent directly reports an attempt to buy their vote |

## Details

where item c. is presented only to the treatment group, and the control list only contains the other three items.

## Source

The full data set is available at the Mexico Panel Study website (http://mexicopanelstudy.mit.edu/)

---

`mis` *The 1994 Multi-Investigator Survey*

---

## Description

This dataset is a subset of the 1994 Multi-Investigator Survey and contains the item count technique or the list experiment. The main question reads as follows: `Now I'm going to read you four things that sometimes mak people angry or upset. After I read all (three/four), just tell me HOW MANY of them upset you. (I don't want to know which ones, just how many.) (1) "the federal government increasing the tax on gasoline;" (2) "professional athletes getting million-dollar-plus salaries;" (3) "requiring seatbelts be used when driving;" (4) "large corporations polluting the environment;" (5) "black leaders asking the government for affirmative action."`

## Format

A data frame containing the following 6 variables for 1171 observations.

| | | | |
|---|---|---|---|
| y | numeric | the number of items that make respondents angry | 0 - 4 |
| sensitive | numeric | whether or not the sensitive item (asked directly) makes respondents angry | 0 - 1 |

| | | | |
|---|---|---|---|
| south | numeric | whether or not a respondents live in a southern state | 0 - 1 |
| male | numeric | whether or not a respondent is male | 0 - 1 |
| college | numeric | whether or not a respondent attended some college | 0 - 1 |
| age | numeric | age of a respondent divided by 10 | |
| democrat | numeric | whether not a respondent identifies as a Democrat | 0 - 1 |
| republican | numeric | whether not a respondent identifies as a Republican | 0 - 1 |
| independent | numeric | whether not a respondent identifies as an independent | 0 - 1 |
| treat | numeric | treatment status | 0 - 1 |
| list.data | numeric | indicator for list experiment subset (treatment and control groups) | 0 - 1 |
| sens.data | numeric | indicator for direct sensitive item subset | 0 - 1 |

## Details

where the last item is presented only with the treatment group and the control list only contains the first three items.

The survey also includes a question in which attitudes toward the sensitive item are asked directly. `Now I'm going to ask you about another thing that sometimes makes people angry or upset. Do you get angry or upset when black leaders ask the government for affirmative action?`

## Source

The full data set is available at SDA (Survey Documentation and Analysis; https://sda.berkeley.edu/D3/Multi/Doc/mult.htm)

---

`multi` *The 1991 National Race and Politics Survey*

---

## Description

This dataset is a subset of the 1991 National Race and Politics Survey and contains a list experiment with two sensitive items. The main questions read as follows: `Now I'm going to read you four things that sometimes make people angry or upset. After I read all (three/four), just tell me HOW MANY of them upset you. (I don't want to know which ones, just how many.) (1) "the federal government increasing the tax on gasoline;" (2) "professional athletes getting million-dollar-plus salaries;" (3) "large corporations polluting the environment;" (4) "a black family moving next door to you."`

## Format

A data frame containing the following 6 variables for 1795 observations.

| | | | |
|---|---|---|---|
| y | numeric | the number of items that make respondents angry | 0 - 4 |
| south | numeric | whether or not a respondents live in a southern state | 0 - 1 |
| male | numeric | whether or not a respondent is male | 0 - 1 |

| | | | |
|---|---|---|---|
| college | numeric | whether or not a respondent attended some college | 0 - 1 |
| age | numeric | age of a respondent divided by 10 | |
| treat | numeric | treatment status | 0 - 2 |

## Details

where the last item is presented only with the treatment group and the control list only contains the first three items.

The second sensitive item replaces item (4) with `(4) "black leaders asking the government for affirmative action."`

Treatment status one (treat == 1) is the "black family" item and status two is the "affirmative action" item.

## Source

The full data set is available at SDA (Survey Documentation and Analysis; https://sda.berkeley.edu/D3/Natlrace/Doc/nrac.htm)

---

# plot.predict.ictreg *Plot Method for the Item Count Technique*

---

## Description

Function to plot predictions and confidence intervals of predictions from estimates from multivariate regression analysis of survey data with the item count technique.

## Usage

```
## S3 method for class 'predict.ictreg'
plot(
  x,
  labels = NA,
  axes.ict = TRUE,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = "Estimated Proportion",
  axes = F,
  pch = 19,
  xvec = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| x | object or set of objects of class inheriting from "predict.ictreg". Either a single object from an ictreg() model fit or multiple predict objects combined with the c() function. |
| labels | a vector of labels for each prediction, plotted at the x axis. |
| axes.ict | a switch indicating if custom plot axes are to be used with the user-provided estimate labels. |
| xlim | a title for the y axis. |
| ylim | a title for the y axis. |
| xlab | a title for the x axis. |
| ylab | a title for the y axis. |
| axes | an indicator for whether default plot axes are included. |
| pch | either an integer specifying a symbol or a single character to be used as the default in plotting points. |
| xvec | a vector of x values at which the proportions will be printed. |
| ... | Other graphical parameters to be passed to the plot() command are accepted. |

## Details

plot.predict.ictreg produces plots with estimated population proportions of respondents answering the sensitive item in a list experiment in the affirmative, with confidence intervals.

The function accepts a set of predict.ictreg objects calculated in the following manner:

```
predict(ictreg.object, avg = TRUE, interval = "confidence")
```

For each average prediction, a point estimate and its confidence interval is plotted at equally spaced intervals. The x location of the points can be manipulated with the xvec option.

Either a single predict object can be plotted, or a group of them combined with c(predict.object1, predict.object2). Predict objects with the newdata.diff option, which calculates the mean difference in probability between two datasets, and the direct.glm option, which calculates the mean difference between the mean predicted support for the sensitive item in the list experiment and in a direct survey item, can also be plotted in the same way as other predict objects.

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) "Statistical Analysis of List Experiments." Political Analysis, Vol. 20, No 1 (Winter). available at http://imai.princeton.edu/research/listP.html

Imai, Kosuke. (2011) "Multivariate Regression Analysis for the Item Count Technique." Journal of the American Statistical Association, Vol. 106, No. 494 (June), pp. 407-416. available at http://imai.princeton.edu/research/list.html

## See Also

ictreg for model fitting and predict.ictreg for predictions based on the model fits.

## Examples

```
data(race)
race.south <- race.nonsouth <- race
race.south[, "south"] <- 1
race.nonsouth[, "south"] <- 0

## Not run:

# Fit EM algorithm ML model with constraint
ml.constrained.results <- ictreg(y ~ south + age + male + college,
   data = race, treat = "treat", J=3, method = "ml",
   overdispersed = FALSE, constrained = TRUE)

# Calculate average predictions for respondents in the South
# and the the North of the US for the MLE model, replicating the
# estimates presented in Figure 1, Imai (2011)
avg.pred.south.mle <- predict(ml.constrained.results,
   newdata = race.south, avg = TRUE, interval = "confidence")
avg.pred.nonsouth.mle <- predict(ml.constrained.results,
   newdata = race.nonsouth, avg = TRUE, interval = "confidence")

# A plot of a single estimate and its confidence interval
plot(avg.pred.south.mle, labels = "South")

# A  plot of the two estimates and their confidence intervals
# use c() to combine more than one predict object for plotting
plot(c(avg.pred.south.mle, avg.pred.nonsouth.mle), labels = c("South", "Non-South"))

# The difference option can also be used to simultaneously
# calculate separate estimates of the two sub-groups
# and the estimated difference. This can also be plotted.

avg.pred.diff.mle <- predict(ml.constrained.results,
   newdata = race.south, newdata.diff = race.nonsouth,
   se.fit = TRUE, avg = TRUE, interval="confidence")

plot(avg.pred.diff.mle, labels = c("South", "Non-South", "Difference"))

# Social desirability bias plots

# Estimate logit for direct sensitive question

data(mis)

mis.list <- subset(mis, list.data == 1)

mis.sens <- subset(mis, sens.data == 1)
```

```
# Fit EM algorithm ML model

fit.list <- ictreg(y ~ age + college + male + south,
   J = 4, data = mis.list, method = "ml")

# Fit logistic regression with directly-asked sensitive question

fit.sens <- glm(sensitive ~ age + college + male + south,
   data = mis.sens, family = binomial("logit"))

# Predict difference between response to sensitive item
# under the direct and indirect questions (the list experiment).
# This is an estimate of the revealed social desirability bias
# of respondents. See Blair and Imai (2010).

avg.pred.social.desirability <- predict(fit.list,
   direct.glm = fit.sens, se.fit = TRUE)

plot(avg.pred.social.desirability)


## End(Not run)
```

---

predict.ictreg *Predict Method for Item Count Technique*

---

## Description

Function to calculate predictions and uncertainties of predictions from estimates from multivariate regression analysis of survey data with the item count technique.

## Usage

```
## S3 method for class 'ictreg'
predict(
  object,
  newdata,
  newdata.diff,
  direct.glm,
  newdata.direct,
  se.fit = FALSE,
  interval = c("none", "confidence"),
  level = 0.95,
  avg = FALSE,
  sensitive.item,
  ...
)
```

## Arguments

| | |
|---|---|
| object | Object of class inheriting from "ictreg" |
| newdata | An optional data frame containing data that will be used to make predictions from. If omitted, the data used to fit the regression are used. |
| newdata.diff | An optional data frame used to compare predictions with predictions from the data in the provided newdata data frame. |
| direct.glm | A glm object from a logistic binomial regression predicting responses to a direct survey item regarding the sensitive item. The predictions from the ictreg object are compared to the predictions based on this glm object. |
| newdata.direct | An optional data frame used for predictions from the direct.glm logistic regression fit. |
| se.fit | A switch indicating if standard errors are required. |
| interval | Type of interval calculation. |
| level | Significance level for confidence intervals. |
| avg | A switch indicating if the mean prediction and associated statistics across all obserations in the dataframe will be returned instead of predictions for each observation. |
| sensitive.item | For multiple sensitive item design list experiments, specify which sensitive item fits to use for predictions. Default is the first sensitive item. |
| ... | further arguments to be passed to or from other methods. |

## Details

predict.ictreg produces predicted values, obtained by evaluating the regression function in the frame newdata (which defaults to model.frame(object). If the logical se.fit is TRUE, standard errors of the predictions are calculated. Setting interval specifies computation of confidence intervals at the specified level or no intervals.

If avg is set to TRUE, the mean prediction across all observations in the dataset will be calculated, and if the se.fit option is set to TRUE a standard error for this mean estimate will be provided. The interval option will output confidence intervals instead of only the point estimate if set to TRUE.

Two additional types of mean prediction are also available. The first, if a newdata.diff data frame is provided by the user, calculates the mean predicted values across two datasets, as well as the mean difference in predicted value. Standard errors and confidence intervals can also be added. For difference prediction, avg must be set to TRUE.

The second type of prediction, triggered if a direct.glm object is provided by the user, calculates the mean difference in prediction between predictions based on an ictreg fit and a glm fit from a direct survey item on the sensitive question. This is defined as the revealed social desirability bias in Blair and Imai (2010).

## Value

predict.ictreg produces a vector of predictions or a matrix of predictions and bounds with column names fit, lwr, and upr if interval is set. If se.fit is TRUE, a list with the following components is returned:

| | |
|---|---|
| fit | vector or matrix as above |
| se.fit | standard error of prediction |

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) "Statistical Analysis of List Experiments." Political Analysis, Vol. 20, No 1 (Winter). available at http://imai.princeton.edu/research/listP.html

Imai, Kosuke. (2011) "Multivariate Regression Analysis for the Item Count Technique." Journal of the American Statistical Association, Vol. 106, No. 494 (June), pp. 407-416. available at http://imai.princeton.edu/research/list.html

## See Also

ictreg for model fitting

## Examples

```
data(race)

race.south <- race.nonsouth <- race

race.south[, "south"] <- 1
race.nonsouth[, "south"] <- 0

## Not run:

# Fit EM algorithm ML model with constraint with no covariates

ml.results.south.nocov <- ictreg(y ~ 1,
   data = race[race$south == 1, ], method = "ml", treat = "treat",
   J = 3, overdispersed = FALSE, constrained = TRUE)
ml.results.nonsouth.nocov <- ictreg(y ~ 1,
   data = race[race$south == 0, ], method = "ml", treat = "treat",
   J = 3, overdispersed = FALSE, constrained = TRUE)

# Calculate average predictions for respondents in the South
# and the the North of the US for the MLE no covariates
# model, replicating the estimates presented in Figure 1,
# Imai (2010)

avg.pred.south.nocov <- predict(ml.results.south.nocov,
   newdata = as.data.frame(matrix(1, 1, 1)), se.fit = TRUE,
   avg = TRUE)
avg.pred.nonsouth.nocov <- predict(ml.results.nonsouth.nocov,
   newdata = as.data.frame(matrix(1, 1, 1)), se.fit = TRUE,
   avg = TRUE)
```

```
# Fit linear regression

lm.results <- ictreg(y ~ south + age + male + college,
   data = race, treat = "treat", J=3, method = "lm")

# Calculate average predictions for respondents in the
# South and the the North of the US for the lm model,
# replicating the estimates presented in Figure 1, Imai (2010)

avg.pred.south.lm <- predict(lm.results, newdata = race.south,
   se.fit = TRUE, avg = TRUE)

avg.pred.nonsouth.lm <- predict(lm.results, newdata = race.nonsouth,
   se.fit = TRUE, avg = TRUE)

# Fit two-step non-linear least squares regression

nls.results <- ictreg(y ~ south + age + male + college,
   data = race, treat = "treat", J=3, method = "nls")

# Calculate average predictions for respondents in the South
# and the the North of the US for the NLS model, replicating
# the estimates presented in Figure 1, Imai (2010)

avg.pred.nls <- predict(nls.results, newdata = race.south,
   newdata.diff = race.nonsouth, se.fit = TRUE, avg = TRUE)

# Fit EM algorithm ML model with constraint

ml.constrained.results <- ictreg(y ~ south + age + male + college,
   data = race, treat = "treat", J=3, method = "ml",
   overdispersed = FALSE, constrained = TRUE)

# Calculate average predictions for respondents in the South
# and the the North of the US for the MLE model, replicating the
# estimates presented in Figure 1, Imai (2010)

avg.pred.diff.mle <- predict(ml.constrained.results,
   newdata = race.south, newdata.diff = race.nonsouth,
   se.fit = TRUE, avg = TRUE)

# Calculate average predictions from the item count technique
# regression and from a direct sensitive item modeled with
# a logit.

# Estimate logit for direct sensitive question

data(mis)

mis.list <- subset(mis, list.data == 1)

mis.sens <- subset(mis, sens.data == 1)
```

```
# Fit EM algorithm ML model

fit.list <- ictreg(y ~ age + college + male + south,
   J = 4, data = mis.list, method = "ml")

# Fit logistic regression with directly-asked sensitive question

fit.sens <- glm(sensitive ~ age + college + male + south,
   data = mis.sens, family = binomial("logit"))

# Predict difference between response to sensitive item
# under the direct and indirect questions (the list experiment).
# This is an estimate of the revealed social desirability bias
# of respondents. See Blair and Imai (2010).

avg.pred.social.desirability <- predict(fit.list,
   direct.glm = fit.sens, se.fit = TRUE)


## End(Not run)
```

---

`predict.ictreg.joint` *Predict Method for Item Count Technique, Outcome Regressions*

---

## Description

Function to calculate predictions and uncertainties of predictions from estimates from multivariate regression analysis of survey data with the item count technique, using predicted responses to list experiments as predictors in outcome regressions.

## Usage

```
## S3 method for class 'ictreg.joint'
predict(
  object,
  newdata,
  newdata.diff,
  se.fit = FALSE,
  interval = c("none", "confidence"),
  level = 0.95,
  avg = FALSE,
  sensitive.value = c("0", "1", "both"),
  sensitive.diff = FALSE,
  return.draws = FALSE,
  predict.sensitive = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| object | Object of class inheriting from "ictreg.joint" |
| newdata | An optional data frame containing data that will be used to make predictions from. If omitted, the data used to fit the regression are used. |
| newdata.diff | An optional data frame used to compare predictions with predictions from the data in the provided newdata data frame. |
| se.fit | A switch indicating if standard errors are required. |
| interval | Type of interval calculation. |
| level | Significance level for confidence intervals. |
| avg | A switch indicating if the mean prediction and associated statistics across all obserations in the dataframe will be returned instead of predictions for each observation. |
| sensitive.value | User-specified value for the sensitive item. |
| sensitive.diff | A switch indicating if the difference in predictions when the sensitive item = 1 and when the sensitive item = 0 is calculated. |
| return.draws | A switch indicating if the draws from the simulations used to generate predictions will be returned. |
| predict.sensitive | A switch indicating whether predictions from the sensitive item model are generated. |
| ... | further arguments to be passed to or from other methods. |

## Details

predict.ictreg.joint produces predicted values, obtained by evaluating the regression function in the frame newdata (which defaults to model.frame(object)). By using sensitive.value, users must set the value of z – the latent response to the sensitive item – to be either zero or one, depending on the prediction that the user requires.

Two additional types of mean prediction are also available. The first, if a newdata.diff data frame is provided by the user, calculates the mean predicted values across two datasets, as well as the mean difference in predicted value. Standard errors and confidence intervals are also added. For newdata.diff predictions, sensitive.value must be set to 1 or 0, not "both" (and sensitive.diff must also be set to FALSE). Users may also set the logical sensitive.diff to TRUE and sensitive.value to "both", which will output the mean predicted values across all observations for z = 0 as well as z = 1, in addition to the mean difference in predicted value. Standard errors and confidence intervals are also added. For difference predictions (sensitive.diff and newdata.diff), the option avg must be set to TRUE.

Users can also use the predict.sensitive = TRUE option to generate predictions of responses to the sensitive item, with standard errors and confidence intervals.

NOTE: In order to generate predictions from user-provided data frames (newdata and newdata.diff), users MUST run models using ictreg.joint on data that does not contain any missingness. Further, the data frames provided to predict.ictreg.joint must also not contain any missingness.

## Value

`predict.ictreg.joint` produces a vector of predictions or a matrix of predictions and bounds with column names fit, lwr, and upr if interval is set. If sensitive.value = "both", `predict.ictreg.joint` will produce a list, where the first element corresponds to when the sensitive item = 0 and the second element corresponds to when the sensitive item = 1. If sensitive.diff = TRUE, the third element in the list corresponds to the difference (sensitive = 0 subtracted from sensitive = 1). If se.fit is TRUE, a list with the following components is returned:

| | |
|---|---|
| `fit` | vector or matrix as above. |
| `se.fit` | standard error of prediction(s) |

If return.draws is TRUE, the list includes

| | |
|---|---|
| `draws.predict` | A matrix of draws from a multivariate normal distribution with mean equal to the vector of estimated coefficients from the outcome regression model, and sigma equal to the variance-covariance matrix of the outcome regression model. Rows are observations; colums are 10,000 draws. If sensitive.value = both, will be a list of two elements where each element is a matrix as described; the first matrix will be for when the sensitive item = 0, the second matrix will be for when the sensitive item = 1. If newdata.diff is provided, `draws.predict` will be a list of two elements where each element is a matrix as described; the first matrix will correspond to the newdata data frame; the second matrix will correspond to the newdata.diff data frame. |
| `draws.mean` | The `draws.predict` matrix averaged over all observations; a vector of 10,000 draws. If sensitive.value = both, will be a list of two elements where each element is a vector as described; the first matrix will be for when the sensitive item = 0, the second matrix will be for when the sensitive item = 1. If newdata.diff is provided, `draws.mean` will be a list of two elements where each element is a matrix as described; the first matrix will correspond to the newdata data frame; the second matrix will correspond to the newdata.diff data frame. |
| `sens.diff` | If sensitive.diff = TRUE, a vector of 10,000 draws generated from subtracting the first item in `draws.mean` from the second item. A vector of 10,000 draws. |

If predict.sensitive = TRUE, the list also includes

| | |
|---|---|
| `fitsens` | a vector of predictions and bounds with column names fit, lwr, and upr if interval is set, generated from the sensitive item model. |
| `draws.predict.sens` | A matrix of draws from a multivariate normal distribution with mean equal to the vector of estimated coefficients from the sensitive item model, and sigma equal to the variance-covariance matrix of the sensitive item model. Rows are observations; colums are 10,000 draws (only returned if return.draws is TRUE). If newdata.diff is provided, this will be a list of two matrices as described. The first will correspond to newdata, and the second to newdata.diff. |
| `draws.mean.sens` | The `draws.predict.sens` matrix averaged over all observations; a vector of 10,000 draws (only returned if return.draws is TRUE). If newdata.diff is provided, this will be a list of two matrices as described. The first will correspond to newdata, and the second to newdata.diff. |

## References

Imai, Kosuke, Bethany Park, and Kenneth F. Greene. (2014) "Using the Predicted Responses from List Experiments as Explanatory Variables in Regression Models." available at http://imai.princeton.edu/research/files/listExp.pdf

## Examples

```
data(mexico)
loyal <- mexico[mexico$mex.loyal == 1,]
notloyal <- mexico[mexico$mex.loyal == 0,]

## Not run:

## Logistic outcome regression
## (effect of vote-selling on turnout)
## This replicates Table 4 in Imai et al. 2014

loyalreg <- ictreg.joint(formula = mex.y.all ~ mex.male + mex.age + mex.age2 + mex.education +
                         mex.interest + mex.married +
                mex.wealth + mex.urban + mex.havepropoganda + mex.concurrent, data = loyal,
                    treat = "mex.t", outcome = "mex.votecard", J = 3, constrained = TRUE,
                         outcome.reg = "logistic", maxIter = 1000)


## Linear outcome regression
## (effect of vote-selling on candidate approval)
## This replicates Table 5 in Imai et al. 2014

approvalreg <- ictreg.joint(formula = mex.y.all ~ mex.male + mex.age + mex.age2 +
                            mex.education +
                            mex.interest + mex.married +
                            mex.urban +
                            mex.cleanelections + mex.cleanelectionsmiss +
                            mex.havepropoganda +
                            mex.wealth + mex.northregion +
                            mex.centralregion + mex.metro + mex.pidpriw2 +
       mex.pidpanw2 + mex.pidprdw2,
                            data = mexico, treat = "mex.t", outcome = "mex.epnapprove",
                            J = 3, constrained = TRUE,
                            outcome.reg = "linear", maxIter = 1000)


summary(approvalreg)

## Generate predicted probability of turnout, averaged over the whole sample,
## for vote sellers (z = 1), non-vote sellers (z = 0), and the difference
## between vote sellers and non-vote sellers, in the sample of party supporters.
## This replicates the results in the righthand panel of Figure 2 in Imai et al. 2014

loyalpred <- predict.ictreg.joint(loyalreg, se.fit = TRUE, interval = "confidence",
     level = 0.95, avg = TRUE,
```

```
    sensitive.value = "both",
    sensitive.diff = TRUE, return.draws = TRUE,
        predict.sensitive = TRUE)

loyalpred$fit

## View predicted probability of vote selling, in the sample of party supporters.
## This replicates the results in the lefthand panel of Figure 2 in Imai et al. 2014

loyalpred$fitsens



## End(Not run)
```

---

`predict.ictregBayes` *Predict Method for the Item Count Technique with Bayesian MCMC*

---

## Description

Function to calculate predictions and uncertainties of predictions from estimates from multivariate regression analysis of survey data with the item count technique.

## Usage

```
## S3 method for class 'ictregBayes'
predict(
  object,
  newdata,
  newdata.diff,
  direct.glm,
  se.fit = FALSE,
  interval = c("none", "confidence"),
  level = 0.95,
  sensitive.item,
  ...
)
```

## Arguments

| | |
|---|---|
| `object` | Object of class inheriting from "ictregBayes" or "ictregBayesMulti" |
| `newdata` | An optional data frame containing data that will be used to make predictions from. If omitted, the data used to fit the regression are used. |
| `newdata.diff` | An optional data frame used to compare predictions with predictions from the data in the provided newdata data frame. |

| | |
|---|---|
| direct.glm | A glm object from a logistic binomial regression predicting responses to a direct survey item regarding the sensitive item. The predictions from the ictreg object are compared to the predictions based on this glm object. |
| se.fit | A switch indicating if standard errors are required. |
| interval | Type of interval calculation. |
| level | Significance level for confidence intervals. |
| sensitive.item | For the multiple sensitive item design, the integer indicating which sensitive item coefficients will be used for prediction. |
| ... | further arguments to be passed to or from other methods. |

## Details

predict.ictregBayes produces predicted values, obtained by evaluating the regression function in the frame newdata (which defaults to model.frame(object). If the logical se.fit is TRUE, standard errors of the predictions are calculated. Setting interval specifies computation of confidence intervals at the specified level or no intervals.

The mean prediction across all observations in the dataset is calculated, and if the se.fit option is set to TRUE a standard error for this mean estimate will be provided. The interval option will output confidence intervals instead of only the point estimate if set to TRUE.

Two additional types of mean prediction are also available. The first, if a newdata.diff data frame is provided by the user, calculates the mean predicted values across two datasets, as well as the mean difference in predicted value. Standard errors and confidence intervals can also be added. For difference prediction, avg must be set to TRUE.

The second type of prediction, triggered if a direct.glm object is provided by the user, calculates the mean difference in prediction between predictions based on an ictreg fit and a glm fit from a direct survey item on the sensitive question. This is defined as the revealed social desirability bias in Blair and Imai (2010).

In the multiple sensitive item design, prediction can only be based on the coefficients from one of the sensitive item fits. The sensitive.item option allows you to specify which is used, using integers from 1 to the number of sensitive items.

## Value

predict.ictreg produces a vector of predictions or a matrix of predictions and bounds with column names fit, lwr, and upr if interval is set. If se.fit is TRUE, a list with the following components is returned:

| | |
|---|---|
| fit | vector or matrix as above |
| se.fit | standard error of prediction |

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) "Statistical Analysis of List Experiments." Political Analysis, Vol. 20, No 1 (Winter). available at http://imai.princeton.edu/research/listP.html

Imai, Kosuke. (2011) "Multivariate Regression Analysis for the Item Count Technique." Journal of the American Statistical Association, Vol. 106, No. 494 (June), pp. 407-416. available at http://imai.princeton.edu/research/list.html

## See Also

ictreg for model fitting

## Examples

```
data(race)

## Not run:

bayes.fit <- ictregBayes(y ~ age + college + male + south, data = multi,
  treat = "treat", delta.tune = diag(.002, 5), psi.tune = diag(.00025, 5))

bayes.predict <- predict(bayes.fit, interval = "confidence", se.fit = TRUE)


## End(Not run)
```

---

# predict.ictregBayesHier

*Predict Method for the Item Count Technique with Bayesian Hierarchical Regression*

---

## Description

Function to calculate predictions and uncertainties of predictions from estimates from hierarchical multivariate regression analysis of survey data with the item count technique.

## Usage

```
## S3 method for class 'ictregBayesHier'
predict(
  object,
  newdata,
  se.fit = FALSE,
  interval = c("none", "confidence"),
  level = 0.95,
  sensitive.item,
  ...
)
```

## Arguments

| | |
|---|---|
| object | Object of class inheriting from "ictregBayes" or "ictregBayesMulti" |
| newdata | An optional data frame containing data that will be used to make predictions from. If omitted, the data used to fit the regression are used. |
| se.fit | A switch indicating if standard errors are required. |
| interval | Type of interval calculation. |
| level | Significance level for confidence intervals. |
| sensitive.item | For the multiple sensitive item design, the integer indicating which sensitive item coefficients will be used for prediction. |
| ... | further arguments to be passed to or from other methods. |

## Details

predict.ictregBayesHier produces predicted values, obtained by evaluating the regression function in the frame newdata (which defaults to model.frame(object). If the logical se.fit is TRUE, standard errors of the predictions are calculated. Setting interval specifies computation of confidence intervals at the specified level or no intervals.

The mean prediction across all observations in the dataset is calculated, and if the se.fit option is set to TRUE a standard error for this mean estimate will be provided. The interval option will output confidence intervals instead of only the point estimate if set to TRUE.

In the multiple sensitive item design, prediction can only be based on the coefficients from one of the sensitive item fits. The sensitive.item option allows you to specify which is used, using integers from 1 to the number of sensitive items.

## Value

predict.ictreg produces a vector of predictions or a matrix of predictions and bounds with column names fit, lwr, and upr if interval is set. If se.fit is TRUE, a list with the following components is returned:

| | |
|---|---|
| fit | vector or matrix as above |
| se.fit | standard error of prediction |

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) “Statistical Analysis of List Experiments." Political Analysis, Vol. 20, No 1 (Winter). available at http://imai.princeton.edu/research/listP.html

Imai, Kosuke. (2011) “Multivariate Regression Analysis for the Item Count Technique.” Journal of the American Statistical Association, Vol. 106, No. 494 (June), pp. 407-416. available at http://imai.princeton.edu/research/list.html

## See Also

ictreg for model fitting

## Examples

```
data(race)

## Not run:

mle.estimates.multi <- ictreg(y ~ male + college, data = multi,
  constrained = TRUE)

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

bayes.fit <- ictregBayesHier(y ~ male + college,
                        formula.level.2 = ~ 1,
                  delta.start.level.1 = list(draws[1, 8:9], draws[1, 2:3], draws[1, 5:6]),
                        data = multi, treat = "treat",
                        delta.tune = list(rep(0.005, 2), rep(0.05, 2), rep(0.05, 2)),
                        alpha.tune = rep(0.001, length(unique(multi$state))),
                        J = 3, group.level.2 = "state",
                        n.draws = 100, burnin = 10, thin = 1)

bayes.predict <- predict(bayes.fit, interval = "confidence", se.fit = TRUE)


## End(Not run)
```

---

# race *The 1991 National Race and Politics Survey*

---

## Description

This dataset is a subset of the 1991 National Race and Politics Survey and contains the item count technique or the list experiment. The main question reads as follows: `Now I'm going to read you four things that sometimes make people angry or upset. After I read all (three/four), just tell me HOW MANY of them upset you. (I don't want to know which ones, just how many.) (1) "the federal government increasing the tax on gasoline;" (2) "professional athletes getting million-dollar-plus salaries;" (3) "large corporations polluting the environment;" (4) "a black family moving next door to you."`

## Format

A data frame containing the following 6 variables for 1213 observations.

| | | | |
|---|---|---|---|
| y | numeric | the number of items that make respondents angry | 0 - 4 |
| south | numeric | whether or not a respondents live in a southern state | 0 - 1 |
| male | numeric | whether or not a respondent is male | 0 - 1 |
| college | numeric | whether or not a respondent attended some college | 0 - 1 |
| age | numeric | age of a respondent divided by 10 | |
| treat | numeric | treatment status | 0 - 1 |

## Details

where the last item is presented only with the treatment group and the control list only contains the first three items.

## Source

The full data set is available at SDA (Survey Documentation and Analysis; https://sda.berkeley.edu/D3/Natlrace/Doc/nrac.htm)

---

# summary.ictreg *Summary Method for the Item Count Technique*

---

## Description

Function to summarize results from list experiment regression based on the ictreg() function, and to produce proportions of liars estimates.

## Usage

```
## S3 method for class 'ictreg'
summary(object, boundary.proportions = FALSE, n.draws = 10000, ...)
```

## Arguments

| | |
|---|---|
| object | Object of class inheriting from "ictreg" |
| boundary.proportions | A switch indicating whether, for models with ceiling effects, floor effects, or both (indicated by the floor = TRUE, ceiling = TRUE options in ictreg), the conditional probability of lying and the population proportions of liars are calculated. |
| n.draws | For quasi-Bayesian approximation based predictions, specify the number of Monte Carlo draws. |
| ... | further arguments to be passed to or from other methods. |

## Details

predict.ictreg produces a summary of the results from an ictreg object. It displays the coefficients, standard errors, and fit statistics for any model from ictreg.

predict.ictreg also produces estimates of the conditional probability of lying and of the population proportion of liars for boundary models from ictreg() if ceiling = TRUE or floor = TRUE.

The conditional probability of lying for the ceiling model is the probability that a respondent with true affirmative views of all the sensitive and non-sensitive items lies and responds negatively to the sensitive item. The conditional probability for the floor model is the probability that a respondent lies to conceal her true affirmative views of the sensitive item when she also holds true negative views of all the non-sensitive items. In both cases, the respondent may believe her privacy is not protected, so may conceal her true affirmative views of the sensitive item.

## Author(s)

Graeme Blair, UCLA, <graeme.blair@ucla.edu> and Kosuke Imai, Princeton University, <kimai@princeton.edu>

## References

Blair, Graeme and Kosuke Imai. (2012) "Statistical Analysis of List Experiments." Political Analysis, Vol. 20, No 1 (Winter). available at http://imai.princeton.edu/research/listP.html

Imai, Kosuke. (2011) "Multivariate Regression Analysis for the Item Count Technique." Journal of the American Statistical Association, Vol. 106, No. 494 (June), pp. 407-416. available at http://imai.princeton.edu/research/list.html

## See Also

ictreg for model fitting

## Examples

```
data(race)
## Not run:
# Fit standard design ML model with ceiling effects
# Replicates Table 7 Columns 3-4 in Blair and Imai (2012)

ceiling.results <- ictreg(y ~ age + college + male + south, treat = "treat",
        J = 3, data = affirm, method = "ml", fit.start = "nls",
     ceiling = TRUE, ceiling.fit = "bayesglm",
     ceiling.formula = ~ age + college + male + south)

# Summarize fit object and generate conditional probability
# of ceiling liars the population proportion of ceiling liars,
# both with standard errors.
# Replicates Table 7 Columns 3-4 last row in Blair and Imai (2012)

summary(ceiling.results, boundary.proportions = TRUE)

## End(Not run)
```

# Index


∗ **datasets**
affirm, 3
combinedListExps, 5
mexico, 35
mis, 36
multi, 37
race, 53
∗ **models**
comp.listEndorse, 5
ict.test, 7
ictreg, 8
ictregBayes, 19
ictregBayesHier, 27
plot.predict.ictreg, 38
predict.ictreg, 41
predict.ictregBayes, 49
predict.ictregBayesHier, 51
summary.ictreg, 54
∗ **regression**
comp.listEndorse, 5
ict.test, 7
ictreg, 8
ictregBayes, 19
ictregBayesHier, 27
plot.predict.ictreg, 38
predict.ictreg, 41
predict.ictregBayes, 49
predict.ictregBayesHier, 51
summary.ictreg, 54

affirm, 3

combinedListDirect, 3
combinedListExps, 5
comp.listEndorse, 5

ict *(*ictreg*)*, 8
ict.hausman.test, 6
ict.test, 7
ictreg, 8, *8*, *40*, *43*, *51*, *53*, *55*
ictreg.joint, 16
ictregBayes, 19
ictregBayesHier, 27

list *(*ictreg*)*, 8

mexico, 35
mis, 36
multi, 37

plot.predict.ictreg, 38
predict.ictreg, *14*, *24*, *33*, *40*, 41
predict.ictreg.joint, 45
predict.ictregBayes, 49
predict.ictregBayesHier, 51

race, 53

summary.ictreg, 54