

Package: attrition (via r-universe)

September 21, 2026

Title Addressing Nonignorable Attrition with Double Sampling and Bounds

Version 1.0.0

Description Implements the double-sampling bounds estimator of Coppock, Gerber, Green, and Kern (2017) [doi:10.1017/pan.2016.6](https://doi.org/10.1017/pan.2016.6) for randomized experiments with nonignorable missing outcomes. Provides worst-case (Manski) bounds, double-sampling bounds with analytic variance and Imbens-Manski confidence intervals, Lee (2009) [doi:10.1111/j.1467-937X.2009.00536.x](https://doi.org/10.1111/j.1467-937X.2009.00536.x) trimming bounds with analytic and bootstrap standard errors, covariate adjustment via poststratification, and a sensitivity analysis for violations of the outcome stability assumption.

Depends R (>= 4.1.0)

Imports generics, ggplot2, tibble

Encoding UTF-8

License GPL-3

URL <https://alexandercoppock.com/attrition/>,
<https://github.com/acoppock/attrition>

BugReports <https://github.com/acoppock/attrition/issues>

Suggests dplyr, estimatr, knitr, purrr, rmarkdown, testthat (>= 3.0.0), vavr (>= 1.1.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

LazyData true

Language en-US

Repository <https://acoppock.r-universe.dev>

Date/Publication 2026-09-21 00:32:56 UTC

RemoteUrl <https://github.com/acoppock/attrition>

RemoteRef HEAD
RemoteSha 162f979c44e5d28465d34ffe2d68b413b99ca3d5

Contents

estimator_ds	2
estimator_ds_sens	4
estimator_ev	6
estimator_trim	8
levendusky_replication	11
print.attrition_bounds	13
print.attrition_sensitivity	14
print.attrition_trim	14
sensitivity_ds	15
summary.attrition_bounds	16
summary.attrition_trim	17
tidy.attrition_bounds	18
tidy.attrition_sensitivity	18
tidy.attrition_trim	19

Index 20

estimator_ds	<i>Extreme Value Bounds with Double Sampling</i>
--------------	--

Description

Bounds the average treatment effect when some outcomes are missing and a random sample of the initial nonrespondents was pursued a second time. Because that sample was drawn at random, the outcomes it recovers estimate the mean outcome among all nonrespondents, so only the subjects who refused twice need worst-case treatment and the identification region narrows accordingly. Reports the region with a joint Imbens-Manski confidence interval.

Usage

```
estimator_ds(  
  Y,  
  Z,  
  R1,  
  Attempt,  
  R2,  
  minY,  
  maxY,  
  strata = NULL,  
  alpha = 0.05,  
  data  
)
```

Arguments

Y	The (unquoted) outcome variable, or a formula <code>outcome ~ treatment</code> for use with <code>declare_estimator(.method = estimator_ds)</code> . Must be numeric.
Z	The (unquoted) assignment indicator variable. Must be numeric and take values 0 or 1. Ignored when Y is a formula.
R1	The initial sample response indicator: unquoted column name, or a quoted string column name when using the formula interface. Must be numeric and take values 0 or 1.
Attempt	The follow-up attempt indicator: unquoted column name, or quoted string. Must be numeric and take values 0 or 1.
R2	The follow-up response indicator: unquoted column name, or quoted string. Must be numeric and take values 0 or 1.
minY	The minimum possible value of the outcome (Y) variable.
maxY	The maximum possible value of the outcome (Y) variable.
strata	Stratification variable: unquoted column name or a quoted string column name.
alpha	The desired significance level. 0.05 by default.
data	A dataframe. Must be given by name: data is the last argument, so passing it positionally assigns it to another argument.

Value

A named numeric vector with elements `estimate_lower` and `estimate_upper`, the two ends of the identification region; `std.error_lower` and `std.error_upper`, their standard errors; and `conf.low` and `conf.high`, the joint Imbens-Manski confidence interval. The names are those of the bounds row of `tidy()`, which returns the same quantities as a data frame.

References

- Coppock, Alexander, Alan S. Gerber, Donald P. Green, and Holger L. Kern (2017). Combining Double Sampling and Bounds to Address Nonignorable Missing Outcomes in Randomized Experiments. *Political Analysis* 25(2):188-206. doi:[10.1017/pan.2016.6](https://doi.org/10.1017/pan.2016.6)
- Imbens, Guido W., and Charles F. Manski (2004). Confidence Intervals for Partially Identified Parameters. *Econometrica* 72(6):1845-1857. doi:[10.1111/j.14680262.2004.00555.x](https://doi.org/10.1111/j.14680262.2004.00555.x)
- Manski, Charles F. (1990). Nonparametric Bounds on Treatment Effects. *American Economic Review Papers and Proceedings* 80(2):319-323.
- Neyman, Jerzy (1938). Contribution to the Theory of Sampling Human Populations. *Journal of the American Statistical Association* 33(201):101-116. doi:[10.1080/01621459.1938.10503378](https://doi.org/10.1080/01621459.1938.10503378)
- Hansen, Morris H., and William N. Hurwitz (1946). The Problem of Non-Response in Sample Surveys. *Journal of the American Statistical Association* 41(236):517-529. doi:[10.1080/01621459.1946.10501894](https://doi.org/10.1080/01621459.1946.10501894)
- Miratrix, Luke W., Jasjeet S. Sekhon, and Bin Yu (2013). Adjusting Treatment Effect Estimates by Post-Stratification in Randomized Experiments. *Journal of the Royal Statistical Society, Series B* 75(2):369-396. doi:[10.1111/j.14679868.2012.01048.x](https://doi.org/10.1111/j.14679868.2012.01048.x)

Examples

```

set.seed(343) # For reproducibility
N <- 1000

# Potential Outcomes
Y_0 <- sample(1:5, N, replace=TRUE, prob = c(0.1, 0.3, 0.3, 0.2, 0.1))
Y_1 <- sample(1:5, N, replace=TRUE, prob = c(0.1, 0.1, 0.4, 0.3, 0.1))

R1_0 <- rbinom(N, 1, prob = 0.7)
R1_1 <- rbinom(N, 1, prob = 0.8)

R2_0 <- rbinom(N, 1, prob = 0.9)
R2_1 <- rbinom(N, 1, prob = 0.95)

# Covariate
strata <- as.numeric(Y_0 > 2)

# Random Assignment
Z <- rbinom(N, 1, .5)

# Reveal Initial Sample Outcomes
R1 <- Z*R1_1 + (1-Z)*R1_0 # Initial sample response
Y_star <- Z*Y_1 + (1-Z)*Y_0 # True outcomes
Y <- Y_star
Y[R1==0] <- NA # Mask outcome of non-responders

# Conduct Double Sampling
Attempt <- rep(0, N)
Attempt[is.na(Y)] <- rbinom(sum(is.na(Y)), 1, .5)

R2 <- rep(0, N)
R2[Attempt==1] <- (Z*R2_1 + (1-Z)*R2_0)[Attempt==1]

Y[R2==1 & Attempt==1] <- Y_star[R2==1 & Attempt==1]

df <- data.frame(Y, Z, R1, Attempt, R2, strata)

# Without post-stratification
estimator_ds(Y, Z, R1, Attempt, R2, minY=1, maxY=5, data=df)

# With post-stratification
estimator_ds(Y, Z, R1, Attempt, R2, minY=1, maxY=5, strata=strata, data=df)

```

estimator_ds_sens

Extreme Value Bounds with Double Sampling with Sensitivity

Description

Interpolates between worst-case bounds and ignorability. δ is the fraction of the follow-up nonrespondents whose outcomes are left unmodeled; the remaining $1 - \delta$ are assumed to be

drawn from a distribution with the mean and variance observed among the follow-up respondents. At $\delta = 1$ the estimator reproduces `estimator_ds`, and at $\delta = 0$ it returns a point estimate.

Usage

```
estimator_ds_sens(
  Y,
  Z,
  R1,
  Attempt,
  R2,
  minY,
  maxY,
  delta,
  strata = NULL,
  alpha = 0.05,
  data
)
```

Arguments

Y	The (unquoted) outcome variable, or a formula <code>outcome ~ treatment</code> for use with <code>declare_estimator(.method = estimator_ds_sens)</code> . Must be numeric.
Z	The (unquoted) assignment indicator variable. Must be numeric and take values 0 or 1. Ignored when Y is a formula.
R1	The initial sample response indicator: unquoted column name, or a quoted string column name when using the formula interface. Must be numeric and take values 0 or 1.
Attempt	The follow-up attempt indicator: unquoted column name, or quoted string. Must be numeric and take values 0 or 1.
R2	The follow-up response indicator: unquoted column name, or quoted string. Must be numeric and take values 0 or 1.
minY	The minimum possible value of the outcome (Y) variable.
maxY	The maximum possible value of the outcome (Y) variable.
delta	Sensitivity parameter in $[0, 1]$. At $\delta = 1$ worst-case bounds apply; at $\delta = 0$ ignorability holds for all follow-up non-responders.
strata	Stratification variable: unquoted column name or a quoted string column name.
alpha	The desired significance level. 0.05 by default.
data	A dataframe. Must be given by name: data is the last argument, so passing it positionally assigns it to another argument.

Value

A named numeric vector with elements `estimate_lower` and `estimate_upper`, the two ends of the identification region; `std.error_lower` and `std.error_upper`, their standard errors; and `conf.low` and `conf.high`, the joint Imbens-Manski confidence interval. The names are those of the bounds row of `tidy()`, which returns the same quantities as a data frame.

References

- Coppock, Alexander, Alan S. Gerber, Donald P. Green, and Holger L. Kern (2017). Combining Double Sampling and Bounds to Address Nonignorable Missing Outcomes in Randomized Experiments. *Political Analysis* 25(2):188-206. doi:[10.1017/pan.2016.6](https://doi.org/10.1017/pan.2016.6)
- Imbens, Guido W., and Charles F. Manski (2004). Confidence Intervals for Partially Identified Parameters. *Econometrica* 72(6):1845-1857. doi:[10.1111/j.14680262.2004.00555.x](https://doi.org/10.1111/j.14680262.2004.00555.x)

Examples

```
set.seed(343)
N <- 1000
Y_0 <- sample(1:5, N, replace = TRUE, prob = c(0.1, 0.3, 0.3, 0.2, 0.1))
Y_1 <- sample(1:5, N, replace = TRUE, prob = c(0.1, 0.1, 0.4, 0.3, 0.1))
Z <- rbinom(N, 1, 0.5)
Y_star <- Z * Y_1 + (1 - Z) * Y_0
R1 <- rbinom(N, 1, prob = 0.7 + 0.1 * Z)
Y <- Y_star
Y[R1 == 0] <- NA

# Follow up intensively with a random half of the initial non-responders
Attempt <- rep(0, N)
Attempt[R1 == 0] <- rbinom(sum(R1 == 0), 1, 0.5)
R2 <- rep(0, N)
R2[Attempt == 1] <- rbinom(sum(Attempt == 1), 1, 0.9)
Y[Attempt == 1 & R2 == 1] <- Y_star[Attempt == 1 & R2 == 1]
df <- data.frame(Y, Z, R1, Attempt, R2)

# delta = 1 reproduces the worst-case double-sampling bounds
estimator_ds_sens(Y, Z, R1, Attempt, R2, minY = 1, maxY = 5, delta = 1, data = df)

# delta = 0 assumes ignorability among follow-up non-responders
estimator_ds_sens(Y, Z, R1, Attempt, R2, minY = 1, maxY = 5, delta = 0, data = df)
```

estimator_ev

Extreme Value (Manski) Bounds

Description

Bounds the average treatment effect when some outcomes are missing and nothing is assumed about why. Filling every missing outcome in the treatment group with the lowest value the outcome can take and every missing outcome in the control group with the highest gives the smallest average effect the data can support; reversing the fills gives the largest. Reports the resulting identification region with a joint Imbens-Manski confidence interval.

Usage

```
estimator_ev(Y, Z, R, minY, maxY, strata = NULL, alpha = 0.05, data)
```

Arguments

Y	The (unquoted) outcome variable, or a formula <code>outcome ~ treatment</code> for use with <code>declare_estimator(.method = estimator_ev)</code> . Must be numeric.
Z	The (unquoted) assignment indicator variable. Must be numeric and take values 0 or 1. Ignored when Y is a formula.
R	The response indicator variable: unquoted column name, or a quoted string column name when using the formula interface. Must be numeric and take values 0 or 1.
minY	The minimum possible value of the outcome (Y) variable.
maxY	The maximum possible value of the outcome (Y) variable.
strata	Stratification variable: unquoted column name or a quoted string column name.
alpha	The desired significance level. 0.05 by default.
data	A dataframe. Must be given by name: data is the last argument, so passing it positionally assigns it to another argument.

Value

A named numeric vector with elements `estimate_lower` and `estimate_upper`, the two ends of the identification region; `std.error_lower` and `std.error_upper`, their standard errors; and `conf.low` and `conf.high`, the joint Imbens-Manski confidence interval. The names are those of the bounds row of `tidy()`, which returns the same quantities as a data frame.

References

- Manski, Charles F. (1990). Nonparametric Bounds on Treatment Effects. *American Economic Review Papers and Proceedings* 80(2):319-323.
- Imbens, Guido W., and Charles F. Manski (2004). Confidence Intervals for Partially Identified Parameters. *Econometrica* 72(6):1845-1857. doi:[10.1111/j.14680262.2004.00555.x](https://doi.org/10.1111/j.14680262.2004.00555.x)
- Miratrix, Luke W., Jasjeet S. Sekhon, and Bin Yu (2013). Adjusting Treatment Effect Estimates by Post-Stratification in Randomized Experiments. *Journal of the Royal Statistical Society, Series B* 75(2):369-396. doi:[10.1111/j.14679868.2012.01048.x](https://doi.org/10.1111/j.14679868.2012.01048.x)
- Coppock, Alexander, Alan S. Gerber, Donald P. Green, and Holger L. Kern (2017). Combining Double Sampling and Bounds to Address Nonignorable Missing Outcomes in Randomized Experiments. *Political Analysis* 25(2):188-206. doi:[10.1017/pan.2016.6](https://doi.org/10.1017/pan.2016.6)

Examples

```
set.seed(343)
N <- 1000
Y_0 <- sample(1:5, N, replace = TRUE, prob = c(0.1, 0.3, 0.3, 0.2, 0.1))
Y_1 <- sample(1:5, N, replace = TRUE, prob = c(0.1, 0.1, 0.4, 0.3, 0.1))
Z <- rbinom(N, 1, 0.5)
Y_star <- Z * Y_1 + (1 - Z) * Y_0

# Treated units respond at a higher rate, so the missingness is nonignorable
R <- rbinom(N, 1, prob = 0.7 + 0.1 * Z)
```

```

Y <- Y_star
Y[R == 0] <- NA
df <- data.frame(Y, Z, R)

estimator_ev(Y, Z, R, minY = 1, maxY = 5, data = df)

# Equivalently, via the formula interface
estimator_ev(Y ~ Z, R = "R", minY = 1, maxY = 5, data = df)

```

estimator_trim

Trimming Bounds

Description

Bounds the average treatment effect among the subjects who would report an outcome under either assignment, the always-reporters, by trimming a tail of the arm with respondents to spare. The outcome need not be bounded, which is what distinguishes this from [estimator_ev](#). Two separate choices shape the estimate: which response arguments are supplied picks the design, and monotonicity picks the selection assumption, which decides which arm is trimmed and by how much.

Usage

```

estimator_trim(
  Y,
  Z,
  R = NULL,
  R1 = NULL,
  Attempt = NULL,
  R2 = NULL,
  monotonicity = c("treatment_increases_response", "treatment_decreases_response",
    "none"),
  alpha = 0.05,
  se = c("analytic", "bootstrap", "none"),
  sims = 1000,
  data
)

```

Arguments

- | | |
|---|--|
| Y | The (unquoted) outcome variable, or a formula <code>outcome ~ treatment</code> for use with <code>declare_estimator(.method = estimator_trim)</code> . Must be numeric. |
| Z | The (unquoted) assignment indicator variable. Must be numeric and take values 0 or 1. Ignored when Y is a formula. |
| R | The single-stage response indicator: unquoted column name, or a quoted string column name when using the formula interface. Must be numeric and take values 0 or 1. Supply either R (single-stage) or R1/Attempt/R2 (double-sampling). |

R1	The initial sample response indicator. Unquoted or quoted string column name. Must be numeric and take values 0 or 1.
Attempt	The follow-up attempt indicator. Unquoted or quoted string column name. Must be numeric and take values 0 or 1.
R2	The follow-up response indicator. Unquoted or quoted string column name. Must be numeric and take values 0 or 1.
monotonicity	The selection assumption, which is separate from the choice of design and is available on both paths. "treatment_increases_response" assumes $R_i(1) \geq R_i(0)$, which makes the control respondents the always-reporters and trims the treatment group; "treatment_decreases_response" assumes the reverse and trims the control group; "none" assumes neither and trims both groups, by the largest share of each that could fail to be always-reporters. The default is "treatment_increases_response" on the single-stage R path, which is Lee (2009), and "none" on the double-sampling path, which is what the follow-up makes affordable; either default gives way to an explicit value. The assumption is the researcher's to make rather than the data's to choose, so nothing here picks a direction from the observed response rates.
alpha	The desired significance level. 0.05 by default.
se	How to obtain standard errors. "analytic" (the default) uses the closed-form asymptotic variance of Lee (2009), Proposition 3, which covers the single-stage, unweighted, one-group-trimmed case and so is available only for the single-stage path under an assumed direction. "bootstrap" resamples units within treatment arm and works everywhere. "none" returns bounds alone. Asking for analytic standard errors where they do not apply is an error rather than a silent substitution.
sims	Number of bootstrap replicates when se = "bootstrap". 1000 by default.
data	A dataframe. Must be given by name: data is the last argument, so passing it positionally assigns it to another argument.

Details

The analytic variance has four contributions: the variance of the retained (trimmed) outcomes, the variance from estimating the trimming threshold, the variance from estimating the trimming proportion, and the variance of the control-group respondent mean. The third of these is often the largest, so treating the trimming proportion as known would understate the uncertainty substantially.

Lee's derivation assumes the bounds are interior points, which fails when the two response rates are equal: the trimming proportion is then zero, the bounds collapse to a point, and the standard errors are not trustworthy. This case warns.

The design and the assumption are separate choices, and `estimator_trim` takes them separately. Which response arguments are supplied picks the design: R is the single-stage estimator, R1, Attempt and R2 the double-sampling one, which recovers outcomes from a random sample of the nonrespondents and so has many fewer subjects left to trim for. `monotonicity` picks the assumption, in either design.

Monotonicity has a direction, and the two directions are not two ways of writing the same assumption: each names a different group as the always-reporters and trims the other, so they give different bounds on the same data, and only one of them is consistent with any given pair of response rates.

The estimator under the reverse direction is the forward estimator run with the arms relabelled, so the bounds it returns are negated and swapped back onto the original contrast.

monotonicity = "none" assumes only random assignment. The share of always-reporters is then bounded below by the Frechet-Hoeffding bound $1 - f_0 - f_1$, where f_z is the missingness rate in arm z , and each arm is trimmed by the largest share of its respondents that could fail to be always-reporters: $f_0/(1 - f_1)$ of the treatment group and $f_1/(1 - f_0)$ of the control group. These are the sharp bounds of Imai (2008), Proposition 1, building on Zhang and Rubin (2003) and Horowitz and Manski (1995). They exist only while $f_0 + f_1 < 1$; beyond that nothing keeps the always-reporter share away from zero, and the estimator says so rather than returning a number. Because both arms are trimmed by the same rule, this case has no direction to set and is invariant to which arm is called treatment.

Value

A named numeric vector leading with the same six elements as the bounding estimators, in the same order whichever path was taken: estimate_lower and estimate_upper, the two trimming bounds; std.error_lower and std.error_upper, their standard errors; and conf.low and conf.high, the joint Imbens-Manski confidence interval. The intermediate quantities used to build them follow, and differ between the two paths. All six lead elements are NA, with a warning naming the other direction, when monotonicity is violated in the direction assumed. Pass to `tidy()` for a data frame.

References

- Horowitz, Joel L., and Charles F. Manski (1995). Identification and Robustness with Contaminated and Corrupted Data. *Econometrica* 63(2):281-302.
- Imai, Kosuke (2008). Sharp Bounds on the Causal Effects in Randomized Experiments with "Truncation-by-Death". *Statistics & Probability Letters* 78(2):144-149.
- Lee, David S. (2009). Training, Wages, and Sample Selection: Estimating Sharp Bounds on Treatment Effects. *Review of Economic Studies* 76(3):1071-1102.
- Tauchmann, Harald (2014). Lee (2009) Treatment-Effect Bounds for Nonrandom Sample Selection. *Stata Journal* 14(4):884-894. doi:10.1177/1536867X1401400411
- Zhang, Junni L., and Donald B. Rubin (2003). Estimation of Causal Effects via Principal Stratification When Some Outcomes are Truncated by "Death". *Journal of Educational and Behavioral Statistics* 28(4):353-368.

Examples

```
set.seed(343)
N <- 1000
Y_0 <- sample(1:5, N, replace = TRUE, prob = c(0.1, 0.3, 0.3, 0.2, 0.1))
Y_1 <- sample(1:5, N, replace = TRUE, prob = c(0.1, 0.1, 0.4, 0.3, 0.1))
Z <- rbinom(N, 1, 0.5)
Y_star <- Z * Y_1 + (1 - Z) * Y_0

# Treated units respond at a higher rate, so the missingness is nonignorable
R <- rbinom(N, 1, prob = 0.7 + 0.1 * Z)
Y <- Y_star
Y[R == 0] <- NA
df <- data.frame(Y, Z, R)
```

```

# Single-stage: trimming bounds under monotonicity, with Lee (2009) standard errors
estimator_trim(Y = Y, Z = Z, R = R, data = df)

# Bootstrap standard errors instead
estimator_trim(Y = Y, Z = Z, R = R, se = "bootstrap", sims = 200, data = df)

# The other direction, on data built the other way round: here treatment
# lowers response, so the control group holds the extra respondents and is
# the group trimmed
R_rev <- rbinom(N, 1, prob = 0.8 - 0.1 * Z)
Y_rev <- Y_star
Y_rev[R_rev == 0] <- NA
df_rev <- data.frame(Y = Y_rev, Z = Z, R = R_rev)

estimator_trim(Y = Y, Z = Z, R = R,
               monotonicity = "treatment_decreases_response", data = df_rev)

# No direction at all: both groups trimmed, randomization the only assumption.
# Wider, and available in either design.
estimator_trim(Y = Y, Z = Z, R = R, monotonicity = "none",
               se = "bootstrap", sims = 200, data = df)

```

levendusky_replication

Perceived polarization under double sampling

Description

A two-wave survey experiment on Amazon Mechanical Turk, replicating Levendusky and Malhotra (2016) and reported as the application in Coppock, Gerber, Green, and Kern (2017). Subjects read a news article describing the electorate either as sharply divided (the polarized condition) or as focused on common ground (the moderate condition). Outcomes were measured immediately in Wave 1 and again in a Wave 2 survey ten days later.

Usage

```
levendusky_replication
```

Format

A tibble with 1,980 rows and 10 columns:

X_party_id Party identification: Dem, Ind, or Rep. The poststratification variable used in Table 3.

Z_condition The condition as assigned: Moderate or Polarized.

Z Polarized (1) versus moderate (0). The contrast analyzed throughout.

R1 Responded in the Wave 2 initial sample.

Attempt Selected for the follow-up sample and offered the higher incentive.

R2 Responded to the follow-up attempt.

Y_polarization_w1 Perceived polarization at Wave 1, from 0 to 6.

Y_polarization_w2 Perceived polarization at Wave 2, from 0 to 6. The dependent variable throughout, and the one with missing values.

Y_extremity_w1 Perceived extremity at Wave 1, from 0 to 3.

Y_extremity_w2 Perceived extremity at Wave 2, from 0 to 3.

Perceived polarization is built from a battery of policy questions. Subjects gave their own view and then guessed how a typical Democratic voter and a typical Republican voter would answer. The outcome is the average absolute difference between the two guesses.

The response indicators rather than the missing values define who responded. Sixteen subjects have a Wave 2 outcome recorded despite `R1 == 0` and `Attempt == 0`, which is why 448 outcomes are NA where 536 subjects did not respond. They are partial completions: they started the Wave 2 survey and abandoned it, and the pre-analysis plan averages the policy items a subject did answer, so answering even one of the four yields a value for the outcome. Every subject who completed the survey answered all four; ten of these sixteen answered fewer, and their missingness across the rest of the Wave 2 questionnaire follows the order the questions were asked. R1 marks completing the survey, and every estimator here keys on R1, Attempt, and R2, as the paper does, so those sixteen outcomes go unused.

They are shipped rather than dropped, because the paper counts them. Each enters every bound as part of its arm's size and its nonrespondent count, and none enters through its outcome, so the naive difference in means among respondents is identical whether they are kept or dropped while the bounds are not: dropping them moves the first column of Table 3 from (-1.54, 1.71) to (-1.50, 1.68) and narrows every interval in the table. Narrowing an identification region by deleting subjects whose outcomes are unknown narrows it without learning anything, and breaking off partway through a survey is itself a post-treatment behavior, split seven and nine across the two arms here. A researcher who wants to depart from the paper's handling can drop them with `subset(levendusky_replication, !(R1 == 0 & Attempt == 0 & !is.na(Y_polarization_w2)))`, or count them as respondents, which uses the answers they did give and narrows the first column further, to (-1.49, 1.66).

Details

Wave 2 is where the attrition happens, and it is what makes the data useful here. Of the 1,980 subjects, 1,444 responded in Wave 2. Exactly 50 nonrespondents were then drawn at random from each condition and offered \$4.00 rather than the original \$1.00 to participate. Of those 100 subjects, 72 completed the survey. The follow-up sample is small, and that is the point: because it is a random sample of the nonrespondents, the outcomes it recovers stand in for the outcomes of every nonrespondent, and the worst-case bounds narrow sharply.

The experiment ran a third condition, a placebo group that read nothing on the topic. It is not analyzed in the paper and is not shipped here, so the data are the polarized-versus-moderate contrast the paper reports and nothing else. `data-raw/levendusky_replication.R` builds them from the archive.

Source

Coppock, Alexander, Alan S. Gerber, Donald P. Green, and Holger L. Kern (2016). Replication Data for: Combining double sampling and bounds to address non-ignorable missing out-

comes in randomized experiments. Harvard Dataverse. doi:10.7910/DVN/AQB4MP, file 2887314 (levendusky_mturk_clean.csv).

References

Coppock, Alexander, Alan S. Gerber, Donald P. Green, and Holger L. Kern (2017). Combining Double Sampling and Bounds to Address Nonignorable Missing Outcomes in Randomized Experiments. *Political Analysis* 25(2):188-206. doi:10.1017/pan.2016.6

Levendusky, Matthew, and Neil Malhotra (2016). Does Media Coverage of Partisan Polarization Affect Political Attitudes? *Political Communication* 33(2):283-301.

Examples

```
# Table 1: attrition by condition
with(levendusky_replication, table(Z_condition, R1))

# Table 3, column 2
estimator_ds(Y_polarization_w2, Z, R1, Attempt, R2,
             minY = 0, maxY = 6, data = levendusky_replication)
```

```
print.attrition_bounds
```

Print bounds

Description

Prints the named vector of results on its own. ‘print.default()’ would otherwise append the class vector to every result, which is noise.

Usage

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

Arguments

x	An object of class “attrition_bounds”, produced by [estimator_ev()], [estimator_ds()], or [estimator_ds_sens()].
...	Passed to ‘print.default()’.

Value

‘x’, invisibly.

```
print.attrition_sensitivity
```

Print a sensitivity analysis

Description

Reports δ^* , the smallest value of the sensitivity parameter at which the confidence interval reaches zero, and names the components of the object. Printing the list itself would draw the plot, which is not what a glance at the result should do.

Usage

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

Arguments

x	An object of class "attrition_sensitivity", produced by [sensitivity_ds()].
...	Unused; included for S3 compatibility.

Value

‘x’, invisibly.

```
print.attrition_trim
```

Print trimming bounds

Description

Print trimming bounds

Usage

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

Arguments

x	An object of class "attrition_trim", produced by [estimator_trim()].
...	Passed to ‘print.default()’.

Value

‘x’, invisibly.

sensitivity_ds

*Sensitivity Analysis***Description**

Searches over delta, the sensitivity parameter of [estimator_ds_sens](#), for delta*: the smallest value at which the confidence interval starts to include zero. A delta* near zero means the finding rests on assuming away nearly all of the missingness; a delta* near one means it survives almost any amount. Returns the search, a plot of it, and delta* itself, which is NA when the interval already includes zero under ignorability.

Usage

```
sensitivity_ds(
  Y,
  Z,
  R1,
  Attempt,
  R2,
  minY,
  maxY,
  sims = 100,
  strata = NULL,
  alpha = 0.05,
  data
)
```

Arguments

Y	The (unquoted) outcome variable, or a formula outcome ~ treatment. Must be numeric.
Z	The (unquoted) assignment indicator variable. Must be numeric and take values 0 or 1. Ignored when Y is a formula.
R1	The initial sample response indicator: unquoted column name, or a quoted string column name when using the formula interface. Must be numeric and take values 0 or 1.
Attempt	The follow-up attempt indicator: unquoted column name, or quoted string. Must be numeric and take values 0 or 1.
R2	The follow-up response indicator: unquoted column name, or quoted string. Must be numeric and take values 0 or 1.
minY	The minimum possible value of the outcome (Y) variable.
maxY	The maximum possible value of the outcome (Y) variable.
sims	Number of values of delta at which to evaluate the bounds. Defaults to 100.
strata	Stratification variable: unquoted column name or a quoted string column name.

alpha	The desired significance level. 0.05 by default.
data	A dataframe. Must be given by name: data is the last argument, so passing it positionally assigns it to another argument.

Value

An object of class "attrition_sensitivity": a list with three elements, `sensitivity_plot`, a ggplot object; `sims_df`, a data frame of bounds and confidence intervals at each value of delta; and `delta_star`, a single number giving delta*, or NA when no delta* exists, which happens when the confidence interval already contains zero at delta = 0. Printing reports delta*; `tidy()` returns `sims_df`.

References

Coppock, Alexander, Alan S. Gerber, Donald P. Green, and Holger L. Kern (2017). Combining Double Sampling and Bounds to Address Nonignorable Missing Outcomes in Randomized Experiments. *Political Analysis* 25(2):188-206. doi:[10.1017/pan.2016.6](https://doi.org/10.1017/pan.2016.6)

Examples

```
set.seed(343)
N <- 1000
Y_0 <- sample(1:5, N, replace = TRUE, prob = c(0.1, 0.3, 0.3, 0.2, 0.1))
Y_1 <- sample(1:5, N, replace = TRUE, prob = c(0.1, 0.1, 0.4, 0.3, 0.1))
Z <- rbinom(N, 1, 0.5)
Y_star <- Z * Y_1 + (1 - Z) * Y_0
R1 <- rbinom(N, 1, prob = 0.7 + 0.1 * Z)
Y <- Y_star
Y[R1 == 0] <- NA

# Follow up intensively with a random half of the initial non-responders
Attempt <- rep(0, N)
Attempt[R1 == 0] <- rbinom(sum(R1 == 0), 1, 0.5)
R2 <- rep(0, N)
R2[Attempt == 1] <- rbinom(sum(Attempt == 1), 1, 0.9)
Y[Attempt == 1 & R2 == 1] <- Y_star[Attempt == 1 & R2 == 1]
df <- data.frame(Y, Z, R1, Attempt, R2)

sens <- sensitivity_ds(Y, Z, R1, Attempt, R2, minY = 1, maxY = 5,
                      sims = 20, data = df)

sens
sens$sensitivity_plot
sens$delta_star
```

summary.attrition_bounds

Summarize bounds

Description

Reports the identification region, its standard errors, and the joint Imbens-Manski interval with a line naming the estimand and the assumptions that produced them. Where `[print()]` shows the returned vector, `'summary()'` says what the numbers are of.

Usage

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

Arguments

<code>object</code>	An object of class <code>"attrition_bounds"</code> , produced by <code>[estimator_ev()]</code> , <code>[estimator_ds()]</code> , or <code>[estimator_ds_sens()]</code> .
<code>...</code>	Unused; included for S3 compatibility.

Value

The `[tidy()]` data frame, invisibly. The report is printed.

Examples

```
summary(estimator_ev(Y = Y_polarization_w2, Z = Z, R = R1,
                     minY = 0, maxY = 6, data = lewendusky_replication))
```

```
summary.attrition_trim
```

Summarize trimming bounds

Description

Reports the identification region, its standard errors, and the joint Imbens-Manski interval, together with the design and the selection assumption that produced them. The two are separate choices in `[estimator_trim()]` and both change what the numbers mean, so both are named here.

Usage

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

Arguments

<code>object</code>	An object of class <code>"attrition_trim"</code> , produced by <code>[estimator_trim()]</code> .
<code>...</code>	Unused; included for S3 compatibility.

Value

The `[tidy()]` data frame, invisibly. The report is printed.

Examples

```
summary(estimator_trim(Y = Y_polarization_w2, Z = Z, R = R1,
                      monotonicity = "treatment_decreases_response",
                      data = levendusky_replication))
```

tidy.attrition_bounds *Tidy an attrition bounds object*

Description

Returns a three-row tibble. The ‘bounds’ row is the estimator’s own named vector transcribed, so ‘estimate_lower’, ‘estimate_upper’, ‘std.error_lower’, ‘std.error_upper’, ‘conf.low’ and ‘conf.high’ are the same names in both places. ‘estimate’ and ‘std.error’ are ‘NA’ on that row, because bounds do not yield a single point estimate; the ‘lower_bound’ and ‘upper_bound’ rows carry one endpoint each in broom’s long form, which is what DeclareDesign selects on with ‘term’.

Usage

```
## S3 method for class 'attrition_bounds'
tidy(x, ...)
```

Arguments

x	An object of class “attrition_bounds” (produced by [estimator_ev()] or [estimator_ds()]).
...	Unused; included for S3 compatibility.

Value

A [tibble::tibble()] with columns ‘term’, ‘estimate’, ‘std.error’, ‘conf.low’, ‘conf.high’, ‘estimate_lower’, ‘estimate_upper’, ‘std.error_lower’, ‘std.error_upper’, ‘outcome’.

tidy.attrition_sensitivity
Tidy a sensitivity analysis

Description

Returns the bounds and joint Imbens-Manski interval at every value of the sensitivity parameter, one row per ‘delta’, under the same names the estimators return.

Usage

```
## S3 method for class 'attrition_sensitivity'
tidy(x, ...)
```

Arguments

`x` An object of class `"attrition_sensitivity"` (produced by `[sensitivity_ds()]`).
`...` Unused; included for S3 compatibility.

Value

A `[tibble::tibble()]` with columns `'delta'`, `'estimate_lower'`, `'estimate_upper'`, `'std.error_lower'`, `'std.error_upper'`, `'conf.low'`, `'conf.high'`, `'outcome'`.

<code>tidy.attrition_trim</code>	<i>Tidy a trimming bounds object</i>
----------------------------------	--------------------------------------

Description

Returns a three-row tibble with the same columns as `[tidy.attrition_bounds()]`. Standard errors and the joint Imbens-Manski confidence interval are filled in when `[estimator_trim()]` was called with `'se = "analytic"'` or `'se = "bootstrap"'`, and are `'NA'` when it was called with `'se = "none"'` or when monotonicity failed.

Usage

```
## S3 method for class 'attrition_trim'
tidy(x, ...)
```

Arguments

`x` An object of class `"attrition_trim"` (produced by `[estimator_trim()]`).
`...` Unused; included for S3 compatibility.

Value

A `[tibble::tibble()]` with columns `'term'`, `'estimate'`, `'std.error'`, `'conf.low'`, `'conf.high'`, `'estimate_lower'`, `'estimate_upper'`, `'std.error_lower'`, `'std.error_upper'`, `'outcome'`.

Index

* datasets

levendusky_replication, 11

estimator_ds, 2, 5

estimator_ds_sens, 4, 15

estimator_ev, 6, 8

estimator_trim, 8

levendusky_replication, 11

print.attrition_bounds, 13

print.attrition_sensitivity, 14

print.attrition_trim, 14

sensitivity_ds, 15

summary.attrition_bounds, 16

summary.attrition_trim, 17

tidy(), 3, 5, 7, 10, 16

tidy.attrition_bounds, 18

tidy.attrition_sensitivity, 18

tidy.attrition_trim, 19