Package ‘multibias’

February 23, 2025

Type Package

Title Simultaneous Multi-Bias Adjustment

Version 1.6.3

Maintainer Paul Brendel <pcbrendel@gmail.com>

Description Quantify the causal effect of a binary exposure on a binary outcome with adjustment for multiple biases. The functions can simultaneously adjust for any combination of uncontrolled confounding, exposure/outcome misclassification, and selection bias.
The underlying method generalizes the concept of combining inverse probability of selection weighting with predictive value weighting. Simultaneous multi-bias analysis can be used to enhance the validity and transparency of real-world evidence obtained from observational, longitudinal studies. Based on the work from Paul Brendel, Aracelis Torres, and Onyebuchi Arah (2023) <doi:10.1093/ije/dyad001>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

RoxygenNote 7.3.2

Imports dplyr (>= 1.1.3), lifecycle (>= 1.0.3), magrittr (>= 2.0.3), rlang (>= 1.1.1)

Suggests knitr, rmarkdown, MASS, testthat (>= 3.0.0)

URL https://github.com/pcbrendel/multibias, http://www.paulbrendel.com/multibias/

BugReports https://github.com/pcbrendel/multibias/issues

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Paul Brendel [aut, cre, cph]

Repository CRAN

Date/Publication 2025-02-23 21:50:02 UTC

Contents

`adjust_em` *Adust for exposure misclassification.*

Description

`adjust_em` returns the exposure-outcome odds ratio and confidence interval, adjusted for exposure misclassificaiton.

Usage

```
adjust_em(
  data_observed,
  data_validation = NULL,
  x_model_coefs = NULL,
  level = 0.95
)
```

Arguments

`data_observed` Object of class `data_observed` corresponding to the data to perform bias analysis on.

`data_validation` Object of class `data_validation` corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the true and misclassified exposure corresponding to the observed exposure in `data_observed`.

`x_model_coefs` The regression coefficients corresponding to the model: $logit(P(X = 1)) = \delta_0 + \delta_1 X^* + \delta_2 Y + \delta_{2+j} C_j$, where *X* represents the binary true exposure, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters is therefore 3 + *j*.

`level` Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_em,
  exposure = "Xstar",
  outcome = "Y",
  confounders = "C1"
)

# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_em_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = "C1",
  misclassified_exposure = "Xstar"
)

adjust_em(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using x_model_coefs -------------------------------------------------------
adjust_em(
  data_observed = df_observed,
  x_model_coefs = c(-2.10, 1.62, 0.63, 0.35)
)
```

---

adjust_em_om *Adust for exposure misclassification and outcome misclassification.*

Description

adjust_em_om returns the exposure-outcome odds ratio and confidence interval, adjusted for exposure misclassification and outcome misclassification.

Usage

```
adjust_em_om(
  data_observed,
  data_validation = NULL,
  x_model_coefs = NULL,
  y_model_coefs = NULL,
  x1y0_model_coefs = NULL,
  x0y1_model_coefs = NULL,
  x1y1_model_coefs = NULL,
  level = 0.95
)
```

Arguments

`data_observed` Object of class `data_observed` corresponding to the data to perform bias analysis on.

`data_validation` Object of class `data_validation` corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the true and misclassified exposure and outcome corresponding to the observed exposure and outcome in `data_observed`.

`x_model_coefs` The regression coefficients corresponding to the model: $logit(P(X = 1)) = \delta_0 + \delta_1 X^* + \delta_2 Y^* + \delta 2 + jC_j$, where *X* represents the binary true exposure, *X** is the binary misclassified exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters is therefore 3 + *j*.

`y_model_coefs` The regression coefficients corresponding to the model: $logit(P(Y = 1)) = \beta_0 + \beta_1 X + \beta_2 Y^* + \beta 2 + jC_j$, where *Y* represents the binary true outcome, *X* is the binary exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters is therefore 3 + *j*.

`x1y0_model_coefs` The regression coefficients corresponding to the model: $log(P(X = 1, Y = 0)/P(X = 0, Y = 0)) = \gamma_{1,0} + \gamma_{1,1} X^* + \gamma_{1,2} Y^* + \gamma_{1,2+j} C_j$, where *X* is the binary true exposure, *Y* is the binary true outcome, *X** is the binary misclassified exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

`x0y1_model_coefs` The regression coefficients corresponding to the model: $log(P(X = 0, U = 1)/P(X = 0, U = 0)) = \gamma_{2,0} + \gamma_{2,1} X^* + \gamma_{2,2} Y^* + \gamma_{2,2+j} C_j$, where *X* is the binary true exposure, *Y* is the binary true outcome, *X** is the binary misclassified exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

`x1y1_model_coefs` The regression coefficients corresponding to the model: $log(P(X = 1, Y = 1)/P(X = 0, Y = 0)) = \gamma_{3,0} + \gamma_{3,1} X^* + \gamma_{3,2} Y^* + \gamma_{3,2+j} C_j$, where *X* is the binary true exposure, *Y* is the binary true outcome, *X** is the binary misclassified exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

`level` Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Two different options for the bias parameters are available here: 1) parameters from separate models of X and Y (x_model_coefs and y_model_coefs) or 2) parameters from a joint model of X and Y (x1y0_model_coefs, x0y1_model_coefs, and x1y1_model_coefs).

Values for the regression coefficients can be applied as fixed values or as single draws from a probability distribution (ex: rnorm(1, mean = 2, sd = 1)). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_em_om,
  exposure = "Xstar",
  outcome = "Ystar",
  confounders = "C1"
)

# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_em_om_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = "C1",
  misclassified_exposure = "Xstar",
  misclassified_outcome = "Ystar"
)

adjust_em_om(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using x_model_coefs and y_model_coefs -------------------------------------
adjust_em_om(
  data_observed = df_observed,
  x_model_coefs = c(-2.15, 1.64, 0.35, 0.38),
  y_model_coefs = c(-3.10, 0.63, 1.60, 0.39)
)

# Using x1y0_model_coefs, x0y1_model_coefs, and x1y1_model_coefs ------------
adjust_em_om(
  data_observed = df_observed,
  x1y0_model_coefs = c(-2.18, 1.63, 0.23, 0.36),
  x0y1_model_coefs = c(-3.17, 0.22, 1.60, 0.40),
  x1y1_model_coefs = c(-4.76, 1.82, 1.83, 0.72)
)
```

adjust_em_sel *Adust for exposure misclassification and selection bias.*

Description

adjust_em_sel returns the exposure-outcome odds ratio and confidence interval, adjusted for exposure misclassification and selection bias.

Usage

```
adjust_em_sel(
  data_observed,
  data_validation = NULL,
  x_model_coefs = NULL,
  s_model_coefs = NULL,
  level = 0.95
)
```

Arguments

data_observed
: Object of class data_observed corresponding to the data to perform bias analysis on.

data_validation
: Object of class data_validation corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the true and misclassified exposure, corresponding to the observed exposure in data_observed. There should also be a selection indicator representing whether the observation in data_validation was selected in data_observed.

x_model_coefs
: The regression coefficients corresponding to the model: $logit(P(X = 1)) = \delta_0 + \delta_1 X^* + \delta_2 Y + \delta 2 + jC_j$, where *X* represents the binary true exposure, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters is therefore 3 + *j*.

s_model_coefs
: The regression coefficients corresponding to the model: $logit(P(S = 1)) = \beta_0 + \beta_1 X^* + \beta_2 Y + \beta 2 + jC_j$, where *S* represents binary selection, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters is therefore 3 + *j*.

level
: Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_em_sel,
  exposure = "Xstar",
  outcome = "Y",
  confounders = "C1"
)

# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_em_sel_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = "C1",
  misclassified_exposure = "Xstar",
  selection = "S"
)

adjust_em_sel(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using x_model_coefs and s_model_coefs -------------------------------------
adjust_em_sel(
  data_observed = df_observed,
  x_model_coefs = c(-2.78, 1.62, 0.58, 0.34),
  s_model_coefs = c(0.04, 0.18, 0.92, 0.05)
)
```

---

adjust_om *Adust for outcome misclassification.*

Description

`adjust_om` returns the exposure-outcome odds ratio and confidence interval, adjusted for outcome misclassificaiton.

Usage

```
adjust_om(
  data_observed,
  data_validation = NULL,
  y_model_coefs = NULL,
  level = 0.95
)
```

Arguments

`data_observed` Object of class `data_observed` corresponding to the data to perform bias analysis on.

`data_validation` Object of class `data_validation` corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the true and misclassified outcome corresponding to the observed outcome in `data_observed`.

`y_model_coefs` The regression coefficients corresponding to the model: $logit(P(Y = 1)) = \_delta_0 + \_delta_1 X + \_delta_2 Y^* + \_delta_{2+j} C_j$, where *Y* represents the binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters is therefore 3 + *j*.

`level` Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_om,
  exposure = "X",
  outcome = "Ystar",
  confounders = "C1"
)
# Using validation data -------------------------------------------------------
df_validation <- data_validation(
  data = df_om_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = "C1",
  misclassified_outcome = "Ystar"
)

adjust_om(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using y_model_coefs -------------------------------------------------------
adjust_om(
  data_observed = df_observed,
  y_model_coefs = c(-3.1, 0.6, 1.6, 0.4)
)
```

---

adjust_om_sel *Adust for outcome misclassification and selection bias.*

Description

`adjust_om_sel` returns the exposure-outcome odds ratio and confidence interval, adjusted for outcome misclassification and selection bias.

Usage

```
adjust_om_sel(
  data_observed,
  data_validation = NULL,
  y_model_coefs = NULL,
  s_model_coefs = NULL,
  level = 0.95
)
```

Arguments

- `data_observed` Object of class `data_observed` corresponding to the data to perform bias analysis on.
- `data_validation` Object of class `data_validation` corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the true and misclassified outcome, corresponding to the observed outcome in `data_observed`. There should also be a selection indicator representing whether the observation in `data_validation` was selected in `data_observed`.
- `y_model_coefs` The regression coefficients corresponding to the model: $logit(P(Y = 1)) = \delta_0 + \delta_1 X + \delta_2 Y^* + \delta_{2+j} C_j$, where *Y* represents the binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters is therefore 3 + *j*.
- `s_model_coefs` The regression coefficients corresponding to the model: $logit(P(S = 1)) = \beta_0 + \beta_1 X + \beta_2 Y^* + \beta_{2+j} C_j$, where *S* represents binary selection, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters is therefore 3 + *j*.
- `level` Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_om_sel,
  exposure = "X",
  outcome = "Ystar",
  confounders = "C1"
)

# Using validation data -----------------------------------------------------
```

```
df_validation <- data_validation(
  data = df_om_sel_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = "C1",
  misclassified_outcome = "Ystar",
  selection = "S"
)

adjust_om_sel(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using y_model_coefs and s_model_coefs ------------------------------------
adjust_om_sel(
  data_observed = df_observed,
  y_model_coefs = c(-3.24, 0.58, 1.59, 0.45),
  s_model_coefs = c(0.03, 0.92, 0.12, 0.05)
)
```

`adjust_sel` *Adust for selection bias.*

Description

`adjust_sel` returns the exposure-outcome odds ratio and confidence interval, adjusted for selection bias.

Usage

```
adjust_sel(
  data_observed,
  data_validation = NULL,
  s_model_coefs = NULL,
  level = 0.95
)
```

Arguments

`data_observed` Object of class `data_observed` corresponding to the data to perform bias analysis on.

`data_validation` Object of class `data_validation` corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the selection indicator representing whether the observation was selected in `data_observed`.

`s_model_coefs` The regression coefficients corresponding to the model: $logit(P(S = 1)) = \beta_0 + \beta_1 X + \beta_2 Y$, where *S* represents binary selection, *X* is the exposure, and *Y* is the outcome. The number of parameters is therefore 3.

`level` Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_sel,
  exposure = "X",
  outcome = "Y",
  confounders = "C1"
)

# Using validation data -------------------------------------------------------
df_validation <- data_validation(
  data = df_sel_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = "C1",
  selection = "S"
)

adjust_sel(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using s_model_coefs ---------------------------------------------------------
adjust_sel(
  data_observed = df_observed,
  s_model_coefs = c(0, 0.9, 0.9)
)
```

adjust_uc *Adust for uncontrolled confounding.*

Description

adjust_uc returns the exposure-outcome odds ratio and confidence interval, adjusted for uncontrolled confounding from a binary confounder.

Usage

```
adjust_uc(
  data_observed,
  data_validation = NULL,
  u_model_coefs = NULL,
  level = 0.95
)
```

Arguments

data_observed	Object of class data_observed corresponding to the data to perform bias analysis on.
data_validation	Object of class data_validation corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the confounder missing in data_observed.
u_model_coefs	The regression coefficients corresponding to the model: $logit(P(U = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 Y + \alpha_{2+j} C_j$, where *U* is the binary unmeasured confounder, *X* is the exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.
level	Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: rnorm(1, mean = 2, sd = 1)). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_uc,
  exposure = "X_bi",
  outcome = "Y_bi",
  confounders = c("C1", "C2", "C3")
)

# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_uc_source,
  true_exposure = "X_bi",
  true_outcome = "Y_bi",
  confounders = c("C1", "C2", "C3", "U")
)

adjust_uc(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using u_model_coefs -------------------------------------------------------
adjust_uc(
  data_observed = df_observed,
  u_model_coefs = c(-0.19, 0.61, 0.70, -0.09, 0.10, -0.15)
)
```

`adjust_uc_em` *Adust for uncontrolled confounding and exposure misclassification.*

Description

`adjust_uc_em` returns the exposure-outcome odds ratio and confidence interval, adjusted for uncontrolled confounding and exposure misclassificaiton.

Usage

```
adjust_uc_em(
  data_observed,
  data_validation = NULL,
  u_model_coefs = NULL,
  x_model_coefs = NULL,
  x1u0_model_coefs = NULL,
  x0u1_model_coefs = NULL,
  x1u1_model_coefs = NULL,
  level = 0.95
)
```

Arguments

data_observed
: Object of class data_observed corresponding to the data to perform bias analysis on.

data_validation
: Object of class data_validation corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the true and misclassified exposure corresponding to the observed exposure in data_observed. There should also be data for the confounder missing in data_observed.

u_model_coefs
: The regression coefficients corresponding to the model: $logit(P(U = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 Y$, where *U* is the binary unmeasured confounder, *X* is the binary true exposure, and *Y* is the outcome. The number of parameters therefore equals 3.

x_model_coefs
: The regression coefficients corresponding to the model: $logit(P(X = 1)) = \delta_0 + \delta_1 X^* + \delta_2 Y + \delta_{2+j} C_j$, where *X* represents the binary true exposure, *X** is the binary misclassified exposure, *Y* is the outcome, and *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

x1u0_model_coefs
: The regression coefficients corresponding to the model: $log(P(X = 1, U = 0)/P(X = 0, U = 0)) = \gamma_{1,0} + \gamma_{1,1} X^* + \gamma_{1,2} Y + \gamma_{1,2+j} C_j$, where *X* is the binary true exposure, *U* is the binary unmeasured confounder, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

x0u1_model_coefs
: The regression coefficients corresponding to the model: $log(P(X = 0, U = 1)/P(X = 0, U = 0)) = \gamma_{2,0} + \gamma_{2,1} X^* + \gamma_{2,2} Y + \gamma_{2,2+j} C_j$, where *X* is the binary true exposure, *U* is the binary unmeasured confounder, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

x1u1_model_coefs
: The regression coefficients corresponding to the model: $log(P(X = 1, U = 1)/P(X = 0, U = 0)) = \gamma_{3,0} + \gamma_{3,1} X^* + \gamma_{3,2} Y + \gamma_{3,2+j} C_j$, where *X* is the binary true exposure, *U* is the binary unmeasured confounder, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

level
: Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Two different options for the bias parameters are available here: 1) parameters from separate models of *U* and *X* (u_model_coefs and x_model_coefs) or 2) parameters from a joint model of *U* and *X* (x1u0_model_coefs, x0u1_model_coefs, and x1u1_model_coefs).

Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_uc_em,
  exposure = "Xstar",
  outcome = "Y",
  confounders = "C1"
)

# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_uc_em_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = c("C1", "U"),
  misclassified_exposure = "Xstar",
)

adjust_uc_em(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using u_model_coefs and x_model_coefs -------------------------------------
adjust_uc_em(
  data_observed = df_observed,
  u_model_coefs = c(-0.23, 0.63, 0.66),
  x_model_coefs = c(-2.47, 1.62, 0.73, 0.32)
)

# Using x1u0_model_coefs, x0u1_model_coefs, x1u1_model_coefs ----------------
adjust_uc_em(
  data_observed = df_observed,
  x1u0_model_coefs = c(-2.82, 1.62, 0.68, -0.06),
  x0u1_model_coefs = c(-0.20, 0.00, 0.68, -0.05),
  x1u1_model_coefs = c(-2.36, 1.62, 1.29, 0.27)
)
```

`adjust_uc_em_sel` *Adust for uncontrolled confounding, exposure misclassification, and selection bias.*

Description

`adjust_uc_em_sel` returns the exposure-outcome odds ratio and confidence interval, adjusted for uncontrolled confounding, exposure misclassificaiton, and selection bias.

Usage

```
adjust_uc_em_sel(
  data_observed,
  data_validation = NULL,
  u_model_coefs = NULL,
  x_model_coefs = NULL,
  x1u0_model_coefs = NULL,
  x0u1_model_coefs = NULL,
  x1u1_model_coefs = NULL,
  s_model_coefs = NULL,
  level = 0.95
)
```

Arguments

`data_observed` Object of class `data_observed` corresponding to the data to perform bias analysis on.

`data_validation`
Object of class `data_validation` corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for: 1) the true and misclassified exposure corresponding to the observed exposure in `data_observed`, 2) the confounder missing in `data_observed`, 3) a selection indicator representing whether the observation in `data_validation` was selected in `data_observed`.

`u_model_coefs` The regression coefficients corresponding to the model: $logit(P(U = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 Y$, where *U* is the binary unmeasured confounder, *X* is the binary true exposure, and *Y* is the outcome. The number of parameters therefore equals 3.

`x_model_coefs` The regression coefficients corresponding to the model: $logit(P(X = 1)) = \delta_0 + \delta_1 X^* + \delta_2 Y + \delta_{2+j} C_j$, where *X* represents binary true exposure, X^* is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

`x1u0_model_coefs`
The regression coefficients corresponding to the model: $log(P(X = 1, U = 0)/P(X = 0, U = 0)) = \gamma_{1,0} + \gamma_{1,1} X^* + \gamma_{1,2} Y + \gamma_{1,2+j} C_j$, where *X* is the

binary true exposure, *U* is the binary unmeasured confounder, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

`x0u1_model_coefs`
: The regression coefficients corresponding to the model: $log(P(X = 0, U = 1)/P(X = 0, U = 0)) = \gamma_{2,0} + \gamma_{2,1}X^* + \gamma_{2,2}Y + \gamma_{2,2+j}C_j$, where *X* is the binary true exposure, *U* is the binary unmeasured confounder, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

`x1u1_model_coefs`
: The regression coefficients corresponding to the model: $log(P(X = 1, U = 1)/P(X = 0, U = 0)) = \gamma_{3,0} + \gamma_{3,1}X^* + \gamma_{3,2}Y + \gamma_{3,2+j}C_j$, where *X* is the binary true exposure, *U* is the binary unmeasured confounder, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

`s_model_coefs`
: The regression coefficients corresponding to the model: $logit(P(S = 1)) = \beta_0+\beta_1X^*+\beta_2Y+\beta_{2+j}C_j$, where *S* represents binary selection, *X** is the binary misclassified exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

`level`
: Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Two different options for the bias parameters are availale here: 1) parameters from separate models of *U* and *X* (`u_model_coefs` and `x_model_coefs`) or 2) parameters from a joint model of *U* and *X* (`x1u0_model_coefs`, `x0u1_model_coefs`, and `x1u1_model_coefs`). Both approaches require `s_model_coefs`.

Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_uc_em_sel,
  exposure = "Xstar",
  outcome = "Y",
```

```
  confounders = c("C1", "C2", "C3")
)

# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_uc_em_sel_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = c("C1", "C2", "C3", "U"),
  misclassified_exposure = "Xstar",
  selection = "S"
)

adjust_uc_em_sel(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using u_model_coefs, x_model_coefs, s_model_coefs -------------------------
adjust_uc_em_sel(
  data_observed = df_observed,
  u_model_coefs = c(-0.32, 0.59, 0.69),
  x_model_coefs = c(-2.44, 1.62, 0.72, 0.32, -0.15, 0.85),
  s_model_coefs = c(0.00, 0.26, 0.78, 0.03, -0.02, 0.10)
)

# Using x1u0_model_coefs, x0u1_model_coefs, x1u1_model_coefs, s_model_coefs
adjust_uc_em_sel(
  data_observed = df_observed,
  x1u0_model_coefs = c(-2.78, 1.62, 0.61, 0.36, -0.27, 0.88),
  x0u1_model_coefs = c(-0.17, -0.01, 0.71, -0.08, 0.07, -0.15),
  x1u1_model_coefs = c(-2.36, 1.62, 1.29, 0.25, -0.06, 0.74),
  s_model_coefs = c(0.00, 0.26, 0.78, 0.03, -0.02, 0.10)
)
```

`adjust_uc_om` *Adust for uncontrolled confounding and outcome misclassification.*

Description

`adjust_uc_om` returns the exposure-outcome odds ratio and confidence interval, adjusted for uncontrolled confounding and outcome misclassificaiton.

Usage

```
adjust_uc_om(
  data_observed,
  data_validation = NULL,
  u_model_coefs = NULL,
```

```
  y_model_coefs = NULL,
  u1y0_model_coefs = NULL,
  u0y1_model_coefs = NULL,
  u1y1_model_coefs = NULL,
  level = 0.95
)
```

Arguments

data_observed — Object of class data_observed corresponding to the data to perform bias analysis on.

data_validation — Object of class data_validation corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the true and misclassified outcome corresponding to the observed exposure in data_observed. There should also be data for the confounder missing in data_observed.

u_model_coefs — The regression coefficients corresponding to the model: $logit(P(U = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 Y$, where *U* is the binary unmeasured confounder, *X* is the exposure, *Y* is the binary true outcome. The number of parameters therefore equals 3.

y_model_coefs — The regression coefficients corresponding to the model: $logit(P(Y = 1)) = \delta_0 + \delta_1 X + \delta_2 Y^* + \delta_{2+j} C_j$, where *Y* represents binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

u1y0_model_coefs — The regression coefficients corresponding to the model: $log(P(U = 1, Y = 0)/P(U = 0, Y = 0)) = \gamma_{1,0} + \gamma_{1,1} X + \gamma_{1,2} Y^* + \gamma_{1,2+j} C_j$, where *U* is the binary unmeasured confounder, *Y* is the binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

u0y1_model_coefs — The regression coefficients corresponding to the model: $log(P(U = 0, Y = 1)/P(U = 0, Y = 0)) = \gamma_{2,0} + \gamma_{2,1} X + \gamma_{2,2} Y^* + \gamma_{2,2+j} C_j$, where *U* is the binary unmeasured confounder, *Y* is the binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

u1y1_model_coefs — The regression coefficients corresponding to the model: $log(P(U = 1, Y = 1)/P(U = 0, Y = 0)) = \gamma_{3,0} + \gamma_{3,1} X + \gamma_{3,2} Y^* + \gamma_{3,2+j} C_j$, where *U* is the binary unmeasured confounder, *Y* is the binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders.

level — Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Two different options for the bias parameters are available here: 1) parameters from separate models of U and Y (`u_model_coefs` and `y_model_coefs`) or 2) parameters from a joint model of U and Y (`u1y0_model_coefs`, `u0y1_model_coefs`, and `u1y1_model_coefs`).

Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_uc_om,
  exposure = "X",
  outcome = "Ystar",
  confounders = "C1"
)

# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_uc_om_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = c("C1", "U"),
  misclassified_outcome = "Ystar"
)

adjust_uc_om(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using u_model_coefs and y_model_coefs -------------------------------------
adjust_uc_om(
  data_observed = df_observed,
  u_model_coefs = c(-0.22, 0.61, 0.70),
  y_model_coefs = c(-2.85, 0.73, 1.60, 0.38)
)

# Using u1y0_model_coefs, u0y1_model_coefs, u1y1_model_coefs ----------------
adjust_uc_om(
  data_observed = df_observed,
  u1y0_model_coefs = c(-0.19, 0.61, 0.00, -0.07),
```

```
  u0y1_model_coefs = c(-3.21, 0.60, 1.60, 0.36),
  u1y1_model_coefs = c(-2.72, 1.24, 1.59, 0.34)
)
```

---

adjust_uc_om_sel — *Adust for uncontrolled confounding, outcome misclassification, and selection bias.*

Description

adjust_uc_om_sel returns the exposure-outcome odds ratio and confidence interval, adjusted for uncontrolled confounding, outcome misclassificaiton, and selection bias.

Usage

```
adjust_uc_om_sel(
  data_observed,
  data_validation = NULL,
  u_model_coefs = NULL,
  y_model_coefs = NULL,
  u0y1_model_coefs = NULL,
  u1y0_model_coefs = NULL,
  u1y1_model_coefs = NULL,
  s_model_coefs = NULL,
  level = 0.95
)
```

Arguments

data_observed
: Object of class data_observed corresponding to the data to perform bias analysis on.

data_validation
: Object of class data_validation corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for: 1) the true and misclassified outcome corresponding to the observed outcome in data_observed, 2) the confounder missing in data_observed, 3) a selection indicator representing whether the observation in data_validation was selected in data_observed.

u_model_coefs
: The regression coefficients corresponding to the model: $logit(P(U = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 Y$, where *U* is the binary unmeasured confounder, *X* is the exposure, and *Y* is the binary true outcome. The number of parameters therefore equals 3.

y_model_coefs
: The regression coefficients corresponding to the model: $logit(P(Y = 1)) = \delta_0 + \delta_1 X + \delta_2 Y^* + \delta_{2+j} C_j$, where *Y* represents binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

`u0y1_model_coefs`
: The regression coefficients corresponding to the model: $log(P(U = 0, Y = 1)/P(U = 0, Y = 0)) = \gamma_{2,0} + \gamma_{2,1}X + \gamma_{2,2}Y^* + \gamma_{2,2+j}C_j$, where *U* is the binary unmeasured confounder, *Y* is the binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

`u1y0_model_coefs`
: The regression coefficients corresponding to the model: $log(P(U = 1, Y = 0)/P(U = 0, Y = 0)) = \gamma_{1,0} + \gamma_{1,1}X + \gamma_{1,2}Y^* + \gamma_{1,2+j}C_j$, where *U* is the binary unmeasured confounder, *Y* is the binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

`u1y1_model_coefs`
: The regression coefficients corresponding to the model: $log(P(U = 1, Y = 1)/P(U = 0, Y = 0)) = \gamma_{3,0} + \gamma_{3,1}X + \gamma_{3,2}Y^* + \gamma_{3,2+j}C_j$, where *U* is the binary unmeasured confounder, *Y* is the binary true outcome, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

`s_model_coefs`
: The regression coefficients corresponding to the model: $logit(P(S = 1)) = \beta_0 + \beta_1 X + \beta_2 Y^* + \beta_{2+j}C_j$, where *S* represents binary selection, *X* is the exposure, *Y** is the binary misclassified outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.

`level`
: Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Two different options for the bias parameters are availale here: 1) parameters from separate models of *U* and *Y* (`u_model_coefs` and `y_model_coefs`) or 2) parameters from a joint model of *U* and *Y* (`u1y0_model_coefs`, `u0y1_model_coefs`, and `u1y1_model_coefs`). Both approaches require `s_model_coefs`.

Values for the regression coefficients can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_uc_om_sel,
  exposure = "X",
  outcome = "Ystar",
  confounders = c("C1", "C2", "C3")
)

# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_uc_om_sel_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = c("C1", "C2", "C3", "U"),
  misclassified_outcome = "Ystar",
  selection = "S"
)

adjust_uc_om_sel(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using u_model_coefs, y_model_coefs, s_model_coefs -------------------------
adjust_uc_om_sel(
  data_observed = df_observed,
  u_model_coefs = c(-0.32, 0.59, 0.69),
  y_model_coefs = c(-2.85, 0.71, 1.63, 0.40, -0.85, 0.22),
  s_model_coefs = c(0.00, 0.74, 0.19, 0.02, -0.06, 0.02)
)

# Using u1y0_model_coefs, u0y1_model_coefs, u1y1_model_coefs, s_model_coefs
adjust_uc_om_sel(
  data_observed = df_observed,
  u1y0_model_coefs = c(-0.20, 0.62, 0.01, -0.08, 0.10, -0.15),
  u0y1_model_coefs = c(-3.28, 0.63, 1.65, 0.42, -0.85, 0.26),
  u1y1_model_coefs = c(-2.70, 1.22, 1.64, 0.32, -0.77, 0.09),
  s_model_coefs = c(0.00, 0.74, 0.19, 0.02, -0.06, 0.02)
)
```

`adjust_uc_sel` *Adust for uncontrolled confounding and selection bias.*

Description

`adjust_uc_sel` returns the exposure-outcome odds ratio and confidence interval, adjusted for uncontrolled confounding and exposure misclassificaiton.

Usage

```
adjust_uc_sel(
  data_observed,
  data_validation = NULL,
  u_model_coefs = NULL,
  s_model_coefs = NULL,
  level = 0.95
)
```

Arguments

- `data_observed` Object of class `data_observed` corresponding to the data to perform bias analysis on.
- `data_validation` Object of class `data_validation` corresponding to the validation data used to adjust for bias in the observed data. Here, the validation data should have data for the same variables as in the observed data, plus data for the confounder missing in `data_observed`. There should also be a selection indicator representing whether the observation in `data_validation` was selected in `data_observed`.
- `u_model_coefs` The regression coefficients corresponding to the model: $logit(P(U = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 Y + \alpha_{2+j} C_j$, where *U* is the binary unmeasured confounder, *X* is the exposure, *Y* is the outcome, *C* represents the vector of measured confounders (if any), and *j* corresponds to the number of measured confounders. The number of parameters therefore equals 3 + *j*.
- `s_model_coefs` The regression coefficients corresponding to the model: $logit(P(S = 1)) = \beta_0 + \beta_1 X + \beta_2 Y$, where *S* represents binary selection, *X* is the exposure, and *Y* is the outcome. The number of parameters therefore equals 3.
- `level` Value from 0-1 representing the full range of the confidence interval. Default is 0.95.

Details

Bias adjustment can be performed by inputting either a validation dataset or the necessary bias parameters. Values for the bias parameters can be applied as fixed values or as single draws from a probability distribution (ex: `rnorm(1, mean = 2, sd = 1)`). The latter has the advantage of allowing the researcher to capture the uncertainty in the bias parameter estimates. To incorporate this uncertainty in the estimate and confidence interval, this function should be run in loop across bootstrap samples of the dataframe for analysis. The estimate and confidence interval would then be obtained from the median and quantiles of the distribution of odds ratio estimates.

Value

A list where the first item is the odds ratio estimate of the effect of the exposure on the outcome and the second item is the confidence interval as the vector: (lower bound, upper bound).

Examples

```
df_observed <- data_observed(
  data = df_uc_sel,
  exposure = "X",
  outcome = "Y",
  confounders = c("C1", "C2", "C3")
)
# Using validation data -----------------------------------------------------
df_validation <- data_validation(
  data = df_uc_sel_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = c("C1", "C2", "C3", "U"),
  selection = "S"
)

adjust_uc_sel(
  data_observed = df_observed,
  data_validation = df_validation
)

# Using u_model_coefs and s_model_coefs -------------------------------------
adjust_uc_sel(
  data_observed = df_observed,
  u_model_coefs = c(-0.19, 0.61, 0.72, -0.09, 0.10, -0.15),
  s_model_coefs = c(-0.01, 0.92, 0.94)
)
```

`data_observed` *Represent observed causal data*

Description

`data_observed` combines the observed dataframe with specific identification of the columns corresponding to the exposure, outcome, and confounders. It is an essential input of all `adjust` functions.

Usage

```
data_observed(data, exposure, outcome, confounders = NULL)
```

Arguments

`data`	Dataframe for bias analysis.
`exposure`	String name of the column in `data` corresponding to the exposure variable.
`outcome`	String name of the column in `data` corresponding to the outcome variable.
`confounders`	String name(s) of the column(s) in `data` corresponding to the confounding variable(s).

Examples

```
df <- data_observed(
  data = df_sel,
  exposure = "X",
  outcome = "Y",
  confounders = c("C1", "C2", "C3")
)
```

`data_validation` *Represent validation causal data*

Description

`data_validation` combines the validation dataframe with specific identification of the appropriate columns for bias adjustment, including: true exposure, true outcome, confounders, misclassified exposure, misclassified outcome, and selection. The purpose of validation data is to use an external data source to transport the necessary causal relationships that are missing in the observed data.

Usage

```
data_validation(
  data,
  true_exposure,
  true_outcome,
  confounders = NULL,
  misclassified_exposure = NULL,
  misclassified_outcome = NULL,
  selection = NULL
)
```

Arguments

`data`	Dataframe of validation data
`true_exposure`	String name of the column in `data` corresponding to the true exposure.
`true_outcome`	String name of the column in `data` corresponding to the true outcome.
`confounders`	String name(s) of the column(s) in `data` corresponding to the confounding variable(s).
`misclassified_exposure`	String name of the column in `data` corresponding to the misclassified exposure.
`misclassified_outcome`	String name of the column in `data` corresponding to the misclassified outcome.
`selection`	String name of the column in `data` corresponding to the selection indicator.

Examples

```
df <- data_validation(
  data = df_sel_source,
  true_exposure = "X",
  true_outcome = "Y",
  confounders = c("C1", "C2", "C3"),
  selection = "S"
)
```

`df_em` *Simulated data with exposure misclassification*

Description

Data containing one source of bias, three known confounders, and 100,000 observations. This data is obtained from `df_emc_source` by removing the column *X*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified exposure, *Xstar*, and no data on the true exposure. As seen in `df_emc_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_em
```

Format

A dataframe with 100,000 rows and 5 columns:

Xstar misclassified exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_em_om` *Simulated data with exposure misclassification and outcome misclassification*

Description

Data containing two sources of bias, three known confounders, and 100,000 observations. This data is obtained from `df_emc_omc_source` by removing the columns *X* and *Y*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified exposure, *Xstar*, and a misclassified outcome, *Ystar*. As seen in `df_em_om_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

`df_em_om`

Format

A dataframe with 100,000 rows and 5 columns:

Xstar misclassified exposure, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_em_om_source` *Data source for* `df_em_om`

Description

Data with complete information on the two sources of bias, three known confounders, and 100,000 observations. This data is used to derive `df_em_om` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_em_om`. With this source data, the fitted regression $logit(P(Y=1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

`df_em_om_source`

Format

A dataframe with 100,000 rows and 7 columns:

X true exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

Xstar misclassified exposure, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent df_em_sel *Simulated data with exposure misclassification and selection bias*

Description

Data containing two sources of bias, three known confounders, and 100,000 observations. This data is obtained by sampling with replacement with probability = *S* from `df_em_sel_source` then removing the columns *X* and *S*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified exposure, *Xstar*, and missing data for those not selected into the study (*S*=0). As seen in `df_em_sel_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_em_sel
```

Format

A dataframe with 100,000 rows and 5 columns:

Xstar misclassified exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

df_em_sel_source *Data source for* `df_em_sel`

Description

Data with complete information on the two sources of bias, three known confounders, and 100,000 observations. This data is used to derive `df_em_sel` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_em_sel`. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_em_sel_source
```

Format

A dataframe with 100,000 rows and 7 columns:

X true exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

Xstar misclassified exposure, 1 = present and 0 = absent

S selection, 1 = selected into the study and 0 = not selected into the study

---

`df_em_source` *Data source for* `df_em`

Description

Data with complete information on one sources of bias, three known confounders, and 100,000 observations. This data is used to derive `df_em` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_em`. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_em_source
```

Format

A dataframe with 100,000 rows and 6 columns:

X exposure, 1 = present and 0 = absent

Y true outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

Xstar misclassified exposure, 1 = present and 0 = absent

`df_om` *Simulated data with outcome misclassification*

Description

Data containing one source of bias, three known confounders, and 100,000 observations. This data is obtained from `df_om_source` by removing the column *Y*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified outcome, *Ystar*, and no data on the true outcome. As seen in `df_om_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_om
```

Format

A dataframe with 100,000 rows and 5 columns:

X exposure, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_om_sel` *Simulated data with outcome misclassification and selection bias*

Description

Data containing two sources of bias, a known confounder, and 100,000 observations. This data is obtained by sampling with replacement with probability = *S* from `df_om_sel_source` then removing the columns *Y* and *S*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified outcome, *Ystar*, and missing data for those not selected into the study (*S*=0). As seen in `df_om_sel_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_om_sel
```

Format

A dataframe with 100,000 rows and 5 columns:

X exposure, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_om_sel_source` *Data source for* `df_om_sel`

Description

Data with complete information on the two sources of bias, a known confounder, and 100,000 observations. This data is used to derive `df_om_sel` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_om_sel`. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_om_sel_source
```

Format

A dataframe with 100,000 rows and 7 columns:

X exposure, 1 = present and 0 = absent

Y true outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

S selection, 1 = selected into the study and 0 = not selected into the study

df_om_source — *Data source for* df_om

Description

Data with complete information on one sources of bias, three known confounders, and 100,000 observations. This data is used to derive df_om and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with df_om. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_om_source
```

Format

A dataframe with 100,000 rows and 6 columns:

X exposure, 1 = present and 0 = absent

Y true outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

df_sel — *Simulated data with selection bias*

Description

Data containing one source of bias, three known confounders, and 100,000 observations. This data is obtained by sampling with replacement with probability = *S* from df_sel_source then removing the *S* column. The resulting data corresponds to what a researcher would see in the real-world: missing data for those not selected into the study (*S*=0). As seen in df_sel_source, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_sel
```

Format

A dataframe with 100,000 rows and 5 columns:

X exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_sel_source` *Data source for* `df_sel`

Description

Data with complete information on study selection, three known confounders, and 100,000 observations. This data is used to derive `df_sel` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_sel`. With this source data, the fitted regression $logit(P(Y=1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_sel_source
```

Format

A dataframe with 100,000 rows and 6 columns:

X true exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

S selection, 1 = selected into the study and 0 = not selected into the study

---

`df_uc` *Simulated data with uncontrolled confounding*

Description

Data containing one source of bias, three known confounders, and 100,000 observations. This data is obtained from `df_uc_source` by removing the column *U*. The resulting data corresponds to what a researcher would see in the real-world: information on known confounders (*C1*, *C2*, and *C3*), but not for confounder *U*. As seen in `df_uc_source`, the true, unbiased exposure-outcome effect estimate = 2.

Usage

```
df_uc
```

Format

A dataframe with 100,000 rows and 7 columns:

X_bi binary exposure, 1 = present and 0 = absent

X_cont continuous exposure

Y_bi binary outcome corresponding to exposure *X_bi*, 1 = present and 0 = absent

Y_cont continuous outcome corresponding to exposure *X_cont*

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_uc_em` *Simulated data with uncontrolled confounding and exposure misclassification*

Description

Data containing two sources of bias, three known confounders, and 100,000 observations. This data is obtained from `df_uc_em_source` by removing the columns *X* and *U*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified exposure, *Xstar*, and missing data on a confounder *U*. As seen in `df_uc_em_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_em
```

Format

A dataframe with 100,000 rows and 5 columns:

Xstar misclassified exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_uc_em_sel` *Simulated data with uncontrolled confounding, exposure misclassification, and selection bias*

Description

Data containing three sources of bias, three known confounders, and 100,000 observations. This data is obtained by sampling with replacement with probability = *S* from `df_uc_em_sel_source` then removing the columns *X*, *U*, and *S*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified exposure, *Xstar*; missing data on a confounder *U*; and missing data for those not selected into the study (*S*=0). As seen in `df_uc_em_sel_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_em_sel
```

Format

A dataframe with 100,000 rows and 5 columns:

Xstar misclassified exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_uc_em_sel_source` *Data source for* `df_uc_em_sel`

Description

Data with complete information on the three sources of bias, three known confounders, and 100,000 observations. This data is used to derive `df_uc_em_sel` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_uc_em_sel`. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3 + \alpha_5 U$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_em_sel_source
```

Format

A dataframe with 100,000 rows and 8 columns:

X true exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

U unmeasured confounder, 1 = present and 0 = absent

Xstar misclassified exposure, 1 = present and 0 = absent

S selection, 1 = selected into the study and 0 = not selected into the study

`df_uc_em_source` *Data source for* `df_uc_em`

Description

Data with complete information on the two sources of bias, a known confounder, and 100,000 observations. This data is used to derive `df_uc_em` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_uc_em`. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 U$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_em_source
```

Format

A dataframe with 100,000 rows and 7 columns:

X true exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

U unmeasured confounder, 1 = present and 0 = absent

Xstar misclassified exposure, 1 = present and 0 = absent

`df_uc_om` *Simulated data with uncontrolled confounding and outcome misclassification*

Description

Data containing two sources of bias, three known confounders, and 100,000 observations. This data is obtained from `df_uc_om_source` by removing the columns *Y* and *U*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified outcome, *Ystar*, and missing data on the binary confounder *U*. As seen in `df_uc_omc_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_om
```

Format

A dataframe with 100,000 rows and 5 columns:

X exposure, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent `df_uc_om_sel` *Simulated data with uncontrolled confounding, outcome misclassification, and selection bias*

Description

Data containing three sources of bias, three known confounders, and 100,000 observations. This data is obtained by sampling with replacement with probability = *S* from `df_uc_om_sel_source` then removing the columns *Y*, *U*, and *S*. The resulting data corresponds to what a researcher would see in the real-world: a misclassified outcome, *Ystar*; missing data on a confounder *U*; and missing data for those not selected into the study (*S*=0). As seen in `df_uc_om_sel_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_om_sel
```

Format

A dataframe with 100,000 rows and 5 columns:

X exposure, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_uc_om_sel_source` *Data source for* `df_uc_om_sel`

Description

Data with complete information on the three sources of bias, three known confounders, and 100,000 observations. This data is used to derive `df_uc_om_sel` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_uc_om_sel`. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3 + \alpha_5 U$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_om_sel_source
```

Format

A dataframe with 100,000 rows and 8 columns:

X exposure, 1 = present and 0 = absent

Y true outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

U unmeasured confounder, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

S selection, 1 = selected into the study and 0 = not selected into the study

---

`df_uc_om_source` *Data source for* `df_uc_om`

Description

Data with complete information on the two sources of bias, three known confounders, and 100,000 observations. This data is used to derive `df_uc_om` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_uc_om`. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 U$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_om_source
```

Format

A dataframe with 100,000 rows and 7 columns:

X exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

U unmeasured confounder, 1 = present and 0 = absent

Ystar misclassified outcome, 1 = present and 0 = absent

`df_uc_sel` *Simulated data with uncontrolled confounding and selection bias*

Description

Data containing two sources of bias, three known confounders, and 100,000 observations. This data is obtained by sampling with replacement with probability = S from `df_uc_sel_source` then removing the columns U and S. The resulting data corresponds to what a researcher would see in the real-world: missing data on confounder U; and missing data for those not selected into the study (S=0). As seen in `df_uc_sel_source`, the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_sel
```

Format

A dataframe with 100,000 rows and 5 columns:

X exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

`df_uc_sel_source` *Data source for* `df_uc_sel`

Description

Data with complete information on the two sources of bias, a known confounder, and 100,000 observations. This data is used to derive `df_uc_sel` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_uc_sel`. With this source data, the fitted regression $logit(P(Y = 1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3 + \alpha_5 U$ shows that the true, unbiased exposure-outcome odds ratio = 2.

Usage

```
df_uc_sel_source
```

Format

A dataframe with 100,000 rows and 7 columns:

X true exposure, 1 = present and 0 = absent

Y outcome, 1 = present and 0 = absent

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

U unmeasured confounder, 1 = present and 0 = absent

S selection, 1 = selected into the study and 0 = not selected into the study

---

`df_uc_source` *Data source for* `df_uc`

Description

Data with complete information on one source of bias, three known confounders, and 100,000 observations. This data is used to derive `df_uc` and can be used to obtain bias parameters for purposes of validating the simultaneous multi-bias adjustment method with `df_uc`. With this source data, the fitted regression $logit(P(Y=1)) = \alpha_0 + \alpha_1 X + \alpha_2 C1 + \alpha_3 C2 + \alpha_4 C3 + \alpha_5 U$ shows that the true, unbiased exposure-outcome effect estimate = 2 when:

1. g = logit, Y = *Y_bi*, and X = *X_bi* or
2. g = identity, Y = *Y_cont*, X = *X_cont*.

Usage

```
df_uc_source
```

Format

A dataframe with 100,000 rows and 8 columns:

X_bi binary exposure, 1 = present and 0 = absent

X_cont continuous exposure

Y_bi binary outcome corresponding to exposure *X_bi*, 1 = present and 0 = absent

Y_cont continuous outcome corresponding to exposure *X_cont*

C1 1st confounder, 1 = present and 0 = absent

C2 2nd confounder, 1 = present and 0 = absent

C3 3rd confounder, 1 = present and 0 = absent

U uncontrolled confounder, 1 = present and 0 = absent `evans` *Evans County dataset*

Description

Data from a cohort study in which white males in Evans County were followed for 7 years, with coronary heart disease as the outcome of interest.

Usage

```
evans
```

Format

A dataframe with 609 rows and 9 columns:

ID subject identifiction

CHD outcome variable; 1 = coronary heart disease

AGE age (in years)

CHL cholesterol, mg/dl

SMK 1 = subject has ever smoked

ECG 1 = presence of electrocardiogram abnormality

DBP diastolic blood pressure, mmHg

SBP systolic blood pressure, mmHg

HPT 1 = SBP greater than or equal to 160 or DBP greater than or equal to 95

Source

http://web1.sph.emory.edu/dkleinb/logreg3.htm#data

Index