

Package ‘confoundr’

October 12, 2022

Type Package

Title Diagnostics for Confounding of Time-Varying and Other Joint Exposures

Version 1.2

Date 2019-08-23

Maintainer John W. Jackson <john.jackson@jhu.edu>

Description Implements three covariate-balance diagnostics for time-varying confounding and selection-bias in complex longitudinal data, as described in Jackson (2016) <doi:10.1097/EDE.0000000000000547> and Jackson (2019) <doi:10.1093/aje/kwz136>. Diagnostic 1 assesses measured confounding/selection-bias, diagnostic 2 assesses exposure-covariate feedback, and diagnostic 3 assesses residual confounding/selection-bias after inverse probability weighting or propensity score stratification. All diagnostics appropriately account for exposure history, can be adapted to assess a particular depth of covariate history, and can be implemented in right-censored data. Balance assessments can be obtained for all times, selected-times, or averaged across person-time. The balance measures are reported as tables or plots. These diagnostics can be applied to the study of multivariate exposures including time-varying exposures, direct effects, interaction, and censoring.

Depends R (>= 3.1.0),

License GPL-3

Imports grid,gridExtra (>= 2.3),scales (>= 0.5.0),Rmpfr (>= 0.7-0),dplyr (>= 0.7.5),tidyr (>= 0.8.1),ggplot2 (>= 2.2.1),magrittr (>= 1.5),stringr (>= 1.3.1),purrr (>= 0.2.5),rlang (>= 0.2.1),stats

LazyData TRUE

VignetteBuilder knitr

Suggests covr, knitr, rmarkdown, testthat, broom (>= 0.4.4)

BugReports <https://github.com/jwjackson/confoundr/issues>

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Author John W. Jackson [aut, cre],
Erin Schnellinger [ctb],
Linda Valeri [ctb]

Repository CRAN

Date/Publication 2019-09-20 04:40:02 UTC

R topics documented:

apply.scope	2
balance	4
catie_sim	6
diagnose	8
example_sml	11
lengthen	12
makehistory.one	14
makehistory.two	15
makeplot	17
omit.history	20
toy_long	21
toy_long_dropoutN	22
toy_long_dropoutY	23
toy_wide_censN	24
toy_wide_censY	26
toy_wide_dropoutN	28
toy_wide_dropoutY	30
widen	32
Index	34

apply.scope	<i>Function to subset the output table from balance() or diagnose() to covariate balance metrics at a certain distance (e.g. a certain recency) or produce estimates that average over person-time.</i>
-------------	---

Description

Function to subset the output table from balance() or diagnose() to covariate balance metrics at a certain distance (e.g. a certain recency) or produce estimates that average over person-time.

Usage

```
apply.scope(input, diagnostic, approach, scope = "all",
  average.over = NULL, periods = NULL, list.distance = NULL,
  recency = NULL, sort.order = "alphabetical",
  ignore.missing.metric = "no", metric = "SMD")
```

Arguments

input	dataframe output by diagnose() or balance() function
diagnostic	diagnostic of interest e.g. 1, 2, or 3
approach	adjustment method e.g. "none" or "weight" or "stratify"
scope	report the entire trellis e.g. "all", the diagonal e.g. "recent", or a summary e.g. "average"
average.over	summary level for average metrics e.g. standardize over "values" or "history" or "time" or "distance"
periods	a list of contiguous segments of relative distance to pool over e.g. list(0,1:4,5:10) would yield summaries over three segments
list.distance	a vector of distances to retain after averaging over time e.g. c(0,2)
recency	an integer for the relative distance between exposures and covariate measurements to focus on (e.g. 0 would represent the same timing). The default is 0 for Diagnostics 1 and 3, and 1 for Diagnostic 2
sort.order	vector of root names for all covariates listed in the order in which they should appear in the table (and also plot) e.g. c("n","m","o","l","p"). To display covariates in alphabetical order (the default), leave blank or type "alphabetical"
ignore.missing.metric	"yes" or "no" depending on whether the user wishes to estimate averages over person-time when there are missing values of the mean difference or standardized mean difference. Missing values for the standardized mean difference can occur when, for example, there is no covariate variation within levels of exposure-history and measurement times. If this argument is set to "no" and there are missing values, the average will also be missing. If set to "yes" an average will be produced that ignores missing values.
metric	the metric for which the user wishes to ignore missing values as specified in the 'ignore.missing.metric' argument.

Details

In most cases this helper function will not be needed by the user, unless omit.history() is called after diagnose() with scope="all" and one desires to subsequently average metrics over time or distance. When using the balance(), diagnose(), or apply.scope() functions, specifying average.over="average" and average.over="time" will return balance metrics for each "distance" value. The output can be subset to specific distances of interest e.g. k=0 and k=2 by supplying a vector to list.distance e.g. c(0,2) but this is optional. Specifying average.over="distance", you can opt to average within segments of distance using the periods argument (leaving this blank will average over all distance values). The periods argument requires a list of contiguous numeric vectors e.g. list(0,1:4,5:10). For Diagnostic 3 this would report metrics at time t, averages over times t-1 to t-4, and averages over times t-5 to t-10. For Diagnostics 1 and 3 the entire range should lie between 0 and t. For Diagnostic 2 the entire range should lie between 1 and t.

Value

A covariate balance table. See the balance() function for details.

balance	<i>Function to create a balance table for a specified diagnostic. Takes input from <code>lengthen()</code> or <code>omit.history()</code>.</i>
---------	--

Description

Function to create a balance table for a specified diagnostic. Takes input from `lengthen()` or `omit.history()`.

Usage

```
balance(input, diagnostic, approach = "none", censoring, scope,
  times.exposure, times.covariate, exposure, history = NULL,
  weight.exposure = NULL, weight.censor = NULL, strata = NULL,
  recency = NULL, average.over = NULL, periods = NULL,
  list.distance = NULL, sort.order = "alphabetical", loop = "no",
  ignore.missing.metric = "no", metric = "SMD", sd.ref = "no")
```

Arguments

<code>input</code>	a restructured tidy dataframe output from <code>lengthen()</code> or <code>omit.history()</code>
<code>diagnostic</code>	diagnostic of interest e.g. 1, 2, or 3
<code>approach</code>	adjustment method e.g. "none" or "weight" or "stratify"
<code>censoring</code>	use censoring indicators/weights e.g. "yes" or "no"
<code>scope</code>	report the entire trellis e.g. "all", the diagonal e.g. "recent", or a summary e.g. "average"
<code>times.exposure</code>	vector of exposure measurement times e.g. <code>c(0,1,2)</code>
<code>times.covariate</code>	vector of covariate measurement times e.g. <code>c(0,1,2)</code>
<code>exposure</code>	root name of exposure e.g. "a"
<code>history</code>	root name of exposure history e.g. "h"
<code>weight.exposure</code>	root name of IP exposure weights e.g. "wa"
<code>weight.censor</code>	root name of IP censoring weights e.g. "ws"
<code>strata</code>	root name of propensity-score strata e.g. "e"
<code>recency</code>	an integer for the relative distance between exposures and covariate measurements to focus on (e.g. 0 would represent the same timing). The default is 0 for Diagnostics 1 and 3, and 1 for Diagnostic 2
<code>average.over</code>	summary level for average metrics e.g. standardize over "values" or "history" or "time" or "distance"
<code>periods</code>	a list of contiguous segments of relative distance to pool over e.g. <code>list(0,1:4,5:10)</code> would yield summaries over three segments
<code>list.distance</code>	a vector of distances to retain after averaging over time e.g. <code>c(0,2)</code>

<code>sort.order</code>	vector of root names for all covariates listed in the order in which they should appear in the table (and also plot) e.g. <code>c("n","m","o","l","p")</code> . To display covariates in alphabetical order (the default), leave blank or type "alphabetical"
<code>loop</code>	a housekeeping argument the user can ignore. It is automatically set when the <code>balance</code> function is called by the <code>diagnose()</code> function described later. The default is set to "no".
<code>ignore.missing.metric</code>	"yes" or "no" depending on whether the user wishes to estimate averages over person-time when there are missing values of the mean difference or standardized mean difference. Missing values for the standardized mean difference can occur when, for example, there is no covariate variation within levels of exposure-history and measurement times. If this argument is set to "no" and there are missing values, the average will also be missing. If set to "yes" an average will be produced that ignores missing values.
<code>metric</code>	the metric for which the user wishes to ignore missing values as specified in the 'ignore.missing.metric' argument.
<code>sd.ref</code>	"yes" or "no" depending on whether the user wishes to use the standard deviation of the reference group when calculating the SMD.

Details

When using the `balance()`, `diagnose()`, or `apply.scope()` functions, specifying `average.over="average"` and `average.over="time"` will return balance metrics for each "distance" value. The output can be subset to specific distances of interest e.g. `k=0` and `k=2` by supplying a vector to `list.distance` e.g. `c(0,2)` but this is optional. Specifying `average.over="distance"`, you can opt to average within segments of distance using the `periods` argument (leaving this blank will average over all distance values). The `periods` argument requires a list of contiguous numeric vectors e.g. `list(0,1:4,5:10)`. For Diagnostic 3 this would report metrics at time `t`, averages over times `t-1` to `t-4`, and averages over times `t-5` to `t-10`. For Diagnostics 1 and 3 the entire range should lie between 0 and `t`. For Diagnostic 2 the entire range should lie between 1 and `t`.

Value

A dataframe depicting a covariate balance table. If the argument `scope` does not equal "average" the returned table reports the mean difference `D` as well as the standardized mean difference `SMD` across levels of exposure, for each comparison of a non-referent value of exposure `E` vs. the referent value (the lowest value by default) at each pairing of exposure measurement times `time.covariate` and covariate measurement times `time.covariate` within levels of exposure history `H` (and/or strata `S`). The sample size of the non-referent group `Nexp` and the sample size summed across the non-referent and referent groups `N` used in the computation of `D` or `SMD` are also provided within levels of `H` and/or `S`. If the argument `scope` equals "average" and the argument `average.over` equals either "values" or "history" or "strata" the format is the same with the averaged over column removed. If the argument `scope` equals "average" and the argument `average.over` equals "time" then a column `distance` indicating the time between exposure and covariate measurements will be included. If the argument for `scope` equals "average" and the argument for `average.over` equals "distance" then the columns `period.start` and `period.end` indicating the beginning and end of person-time segments will appear.

Examples

```
# Simulate the output of lengthen() or omit.history()
id <- as.numeric(rep(c(1,1,1,2,2,2), 70))
time.exposure <- as.numeric(rep(c(0,1,2), 140))
a <- as.character(rep(c(0,1,1,1,0,0), 70))
h <- as.character(rep(c("H","H0","H01","H","H0","H01"), 70))
name.cov <- as.character(c(rep("n",60), rep("l",180), rep("m",180)))
time.covariate <- as.numeric(rep(c(rep(0,7), rep(1,7), rep(2,7)), 60))
value.cov <- as.numeric(rnorm(420, 2, 3))

mydata.long.omit <- data.frame(id, time.exposure, a, h,
                              name.cov, time.covariate, value.cov)

# Run the balance() function
mytable <- balance(input=mydata.long.omit,
                  diagnostic=1,
                  approach="none",
                  censoring="no",
                  scope="all",
                  times.exposure=c(0,1,2),
                  times.covariate=c(0,1),
                  sort.order=c("l", "m", "n"),
                  exposure="a",
                  history="h"
                  )
```

catie_sim

Simulated data, loosely based on the Clinical Antipsychotic Trial of Intervention Effectiveness (CATIE) study.

Description

1430 completely hypothetical persons with schizophrenia randomized to one of five antipsychotics and followed for up to 18 months. Note that the simulation did not build in sequential randomization, as was done in the trial.

Usage

```
data(catie_sim)
```

Format

A dataframe with 8,632 rows and 62 variables:

CATIEID person id

time month of study visit (0, 1, 3, 6, 9, 12, 15, 18)

td tardive dyskinesia stratum

zprcort ziprasidone cohort stratum
race 1:white, 2:black, 3:other
age.grp 1:18-24, 2:25-34, 3:45-44, 4:45-54, 5:55-67
educ.bin high school graduate
site.ro study site, research only
site.sh study site, state mental health
site.uc study site, university center
site.va study site, veterans affairs
treat.grp antipsychotic 1:ola, 2:que, 3:ris, 4:per, 5:zip
cs14 drug use scale
cs16 clinical global impressions (CGI) severity scale
calg1 calgary depression scale
weight in lbs
epsmean Simpson-Agnes extrapyramidal symptoms
qoltot quality of life total score
pansstotal positive and negative syndrome scale (PANSS)
phase.change.vis switch to new antipsychotic
white race dummy variable, white
black race dummy variable, black
other race dummy variable, other
age.grp.1824 age group dummy variable, 18-24 years
age.grp.2534 age group dummy variable, 25-34 years
age.grp.3544 age group dummy variable, 35-44 years
age.grp.4554 age group dummy variable, 45-54 years
age.grp.5567 age group dummy variable, 55-67 years
Bpansstotal pansstotal at time 0
Bcs14 cs14 at time 0
Bcs16 cs16 at time 0
Bcalg1 calg1 at time 0
Bqoltot qoltot at time 0
Chg.pansstotal change in pansstotal
pct.gain percent weight gain
phase.change.cum number of switches to antipsychotic
phase.change.cum.rec time-varying version of ever switch to antipsychotic
lead.pansstotal pansstotal at next visit
treat.grp.ola olanzapine arm dummy
treat.grp.que quetiapine arm dummy

treat.grp.ris risperidone arm dummy
treat.grp.per perphenazine arm dummy
treat.grp.zip ziprasidone arm dummy
studydisc last visit (1=yes, 0 otherwise)
num.x probability of treatment arm
den.x probability of treatment arm given baseline covariates
wx.b stabilized iptw for treatment arm
num.po probability studydisc=1, given treat.grp & baseline covariates, common model
den.po probability studydisc=1, given treat.grp & baseline and time-varying covariates, common model
num.tr probability studydisc=1, given treat.grp & baseline covariates, treat.grp specific model
den.tr probability studydisc=1, given treat.grp & baseline and time-varying covariates, treat.grp specific model
wpo stabilized ipcw, from common model, not truncated
wtr stabilized ipcw, from treat.grp specific model, not truncated
wpo stabilized ipcw, truncated 99th tile
wtr stabilized ipcw, from treat.grp specific model, truncated 99th tile
wpo stabilized ipcw, from common model, truncated 95th tile
wtr stabilized ipcw, from treat.grp specific model, truncated 95th tile
wpo stabilized ipcw, from common model, truncated 90th tile
wtr stabilized ipcw, from treat.grp specific model, truncated 90th tile

References

Lieberman JA, Stroup TS, McEvoy JP, Swartz MS, Rosenheck RA, Perkins DO, Keefe RS, Davis SM, Davis CE, Lebowitz BD, Severe J, Hsiao JK; Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE) Investigators. Effectiveness of antipsychotic drugs in patients with chronic schizophrenia. *N Engl J Med.* 2005 Sep 22;353(12):1209-23. Epub 2005 Sep 19. Erratum in: *N Engl J Med.* 2010 Sep 9;363(11):1092-3. PubMed PMID: 16172203.

diagnose

Function to loop over the lengthen() and balance() functions.

Description

Function to loop over the lengthen() and balance() functions.

Usage

```
diagnose(input, diagnostic, approach = "none", scope, censoring, id,
  times.exposure, times.covariate, exposure, temporal.covariate,
  static.covariate = NULL, history = NULL, weight.exposure = NULL,
  censor = NULL, weight.censor = NULL, strata = NULL,
  recency = NULL, average.over = NULL, periods = NULL,
  list.distance = NULL, sort.order = "alphabetical", loop = "no",
  ignore.missing.metric = "no", metric = "SMD", sd.ref = "no")
```

Arguments

input	dataframe in wide format (e.g., indexed by person)
diagnostic	diagnostic of interest e.g. 1, 2, or 3
approach	adjustment method e.g. "none" or "weight" or "stratify"
scope	report the entire trellis e.g. "all", the diagonal e.g. "recent", or a summary e.g. "average"
censoring	use censoring indicators/weights e.g. "yes" or "no"
id	unique observation identifier e.g. "id"
times.exposure	vector of exposure measurement times e.g. c(0,1,2)
times.covariate	vector of covariate measurement times e.g. c(0,1,2)
exposure	root name of exposure e.g. "a"
temporal.covariate	a vector of root names for covariates whose values change over time e.g. c("l","m","n","o","p")
static.covariate	a vector of root names for covariates whose values do not change (covariates listed here should not appear in the temporal.covariate argument)
history	the root name for history measurements e.g. "h"
weight.exposure	the root name for exposure weights e.g. "wa"
censor	the root name for censoring indicators e.g. "s"
weight.censor	the root name for censoring weights e.g. "ws"
strata	the root name for propensity-score strata e.g. "e"
recency	an integer for the relative distance between exposures and covariate measurements to focus on (e.g. 0 would represent the same timing). The default is 0 for Diagnostics 1 and 3, and 1 for Diagnostic 2
average.over	summary level for average metrics e.g. standardize over "values" or "history" or "time" or "distance"
periods	a list of contiguous segments of relative distance to pool over e.g. list(0,1:4,5:10) would yield summaries over three segments
list.distance	a vector of distances to retain after averaging over time e.g. c(0,2)

<code>sort.order</code>	vector of root names for all covariates listed in the order in which they should appear in the table (and also plot) e.g. <code>c("n","m","o","l","p")</code> . To display covariates in alphabetical order (the default), leave blank or type "alphabetical"
<code>loop</code>	"yes" to iteratively apply <code>balance()</code> and <code>lengthen()</code> or "no" to process all covariates and measurement times at once.
<code>ignore.missing.metric</code>	"yes" or "no" depending on whether the user wishes to estimate averages over person-time when there are missing values of the mean difference or standardized mean difference. Missing values for the standardized mean difference can occur when, for example, there is no covariate variation within levels of exposure-history and measurement times. If this argument is set to "no" and there are missing values, the average will also be missing. If set to "yes" an average will be produced that ignores missing values.
<code>metric</code>	the metric for which the user wishes to ignore missing values as specified in the <code>'ignore.missing.metric'</code> argument.
<code>sd.ref</code>	"yes" or "no" depending on whether the user wishes to use the standard deviation of the reference group when calculating the SMD.

Details

When using the `balance()`, `diagnose()`, or `apply.scope()` functions, specifying `average.over="average"` and `average.over="time"` will return balance metrics for each "distance" value. The output can be subset to specific distances of interest e.g. `k=0` and `k=2` by supplying a vector to `list.distance` e.g. `c(0,2)` but this is optional. Specifying `average.over="distance"`, you can opt to average within segments of distance using the `periods` argument (leaving this blank will average over all distance values). The `periods` argument requires a list of contiguous numeric vectors e.g. `list(0,1:4,5:10)`. For Diagnostic 3 this would report metrics at time `t`, averages over times `t-1` to `t-4`, and averages over times `t-5` to `t-10`. For Diagnostics 1 and 3 the entire range should lie between 0 and `t`. For Diagnostic 2 the entire range should lie between 1 and `t`.

Value

A covariate balance table. See the `balance()` function for details.

Examples

```
# This example uses the included "example_sml.rda" data set

diagnose(input=example_sml,
  diagnostic=1,
  censoring="no",
  approach="none",
  scope="all",
  id="id",
  times.exposure=c(0,1,2),
  times.covariate=c(0,1,2),
  exposure="a",
  temporal.covariate=c("l","m","n"),
  static.covariate=c("o","p"),
```

```

sort.order="alphabetical",
history="h",
ignore.missing.metric="no",
loop="yes",
sd.ref="no")

```

example_sml

Artificial data set used to illustrate the functionality of confoundr.

Description

The example_sml data set contains 10,000 records and 38 variables. These variables include time-varying exposures, outcomes, and covariates, along with strata and censoring indicators. Time-varying inverse-probability-of-exposure weights and censoring weights are available as well.

Usage

```
data(example_sml)
```

Format

A data frame with 10,000 rows and 38 variables:

X1 row label, can be ignored

id subject ID

a_0 exposure measurement at time 0

a_1 exposure measurement at time 1

a_2 exposure measurement at time 2

l_0 covariate measurement at time 0

l_1 covariate measurement at time 1

l_2 covariate measurement at time 2

m_0 covariate measurement at time 0

m_1 covariate measurement at time 1

m_2 covariate measurement at time 2

n_0 covariate measurement at time 0

n_1 covariate measurement at time 1

n_2 covariate measurement at time 2

o_0 covariate measurement at time 0

o_1 covariate measurement at time 1

o_2 covariate measurement at time 2

p_0 covariate measurement at time 0

p_1 covariate measurement at time 1

p_2 covariate measurement at time 2
wa_0 inverse probability of exposure weight at time 0
wa_1 inverse probability of exposure weight at time 1
wa_2 inverse probability of exposure weight at time 2
wax_0 cumulative inverse probability weight of exposure at time 0
wax_1 cumulative inverse probability weight of exposure at time 1
wax_2 cumulative inverse probability weight of exposure at time 2
wsx_0 cumulative inverse probability of censoring weight at time 0
wsx_1 cumulative inverse probability of censoring weight at time 1
wsx_2 cumulative inverse probability of censoring weight at time 2
e5_0 propensity score strata at time 0
e5_1 propensity score strata at time 1
e5_2 propensity score strata at time 2
h_0 exposure history at time 0
h_1 exposure history at time 1
h_2 exposure history at time 2
s_0 censoring indicator at time 0
s_1 censoring indicator at time 1
s_2 censoring indicator at time 2

lengthen

Function to create a "tidy" dataframe where the key observation is the pairing of exposure and covariate measurement times

Description

Function to create a "tidy" dataframe where the key observation is the pairing of exposure and covariate measurement times

Usage

```
lengthen(input, diagnostic, censoring, id, times.exposure, times.covariate,
         exposure, temporal.covariate, static.covariate = NULL,
         history = NULL, weight.exposure = NULL, censor = NULL,
         weight.censor = NULL, strata = NULL)
```

Arguments

<code>input</code>	dataframe in wide format (e.g., indexed by person)
<code>diagnostic</code>	diagnostic of interest e.g. 1, 2, or 3
<code>censoring</code>	use censoring indicators/weights e.g. "yes" or "no"
<code>id</code>	unique observation identifier e.g. "id"
<code>times.exposure</code>	a vector of exposure measurement times e.g. c(0,1,2)
<code>times.covariate</code>	a vector of covariate measurement times e.g. c(0,1,2)
<code>exposure</code>	the root name for exposure measurements e.g. "a"
<code>temporal.covariate</code>	a vector of root names for covariates whose values change over time e.g. c("l","m","n","o","p")
<code>static.covariate</code>	a vector of root names for covariates whose values do not change (covariates listed here should not appear in the <code>temporal.covariate</code> argument)
<code>history</code>	the root name for history measurements e.g. "h"
<code>weight.exposure</code>	the root name for exposure weights e.g. "wa"
<code>censor</code>	the root name for censoring indicators e.g. "s"
<code>weight.censor</code>	the root name for censoring weights e.g. "ws"
<code>strata</code>	the root name for propensity-score strata e.g. "e"

Details

The input dataset should have one record per observation (wide format) with the timing of variables indexed by an underscore followed by the time index (underscores should NOT appear anywhere else in the variable name). Any indexing scheme can be used (e.g. "var_1", "var_4", "var_9"), but it may be easiest to assign zero as the baseline index and increase it by one the unit for each subsequent measurement (e.g. "var_0", "var_1", "var_2"). You can use `widen()` to transform a person-time dataset into this format. The common referent value—to which all other exposure levels are compared—should be coded as the lowest value. Data with artificial censoring rules should contain a vector of time-indexed censoring indicators (1=censored, 0 otherwise).

Value

A "tidy" dataframe where each record is indexed by the observation identifier, exposure measurement time, exposure value, covariate name, covariate measurement time and possibly exposure history and/or propensity score strata. Weights for exposure and/or censoring will appear as additional columns. The dataframe will be restricted to the uncensored if censoring rules were applied.

Examples

```
# Simulate wide data set with history
id <- as.numeric(c(1, 2))
a_0 <- as.numeric(c(0, 1))
a_1 <- as.numeric(c(1, 1))
```

```

a_2 <- as.numeric(c(1, 0))
l_0 <- as.numeric(rbinom(2, 1, 0.5))
l_1 <- as.numeric(rbinom(2, 1, 0.5))
l_2 <- as.numeric(rbinom(2, 1, 0.5))
m_0 <- as.numeric(rbinom(2, 1, 0.5))
m_1 <- as.numeric(rbinom(2, 1, 0.5))
m_2 <- as.numeric(rbinom(2, 1, 0.5))
n_0 <- as.numeric(rbinom(2, 1, 0.5))
n_1 <- as.numeric(rbinom(2, 1, 0.5))
n_2 <- as.numeric(rbinom(2, 1, 0.5))
h_0 <- as.character(c("H", "H"))
h_1 <- as.character(c("H0", "H1"))
h_2 <- as.character(c("H01", "H11"))

mydata.history <- data.frame(id, a_0, a_1, a_2,
                             l_0, l_1, l_2,
                             m_0, m_1, m_2,
                             n_0, n_1, n_2,
                             h_0, h_1, h_2,
                             stringsAsFactors=FALSE)

# Run the lengthen() function
mydata.long <- lengthen(input=mydata.history,
                        diagnostic=1,
                        censoring="no",
                        id="id",
                        times.exposure=c(0,1,2),
                        times.covariate=c(0,1,2),
                        exposure="a",
                        temporal.covariate=c("l", "m"),
                        static.covariate=c("n"),
                        history="h"
                        )

```

makehistory.one

Function to create exposure history for a single time varying exposure.

Description

Function to create exposure history for a single time varying exposure.

Usage

```
makehistory.one(input, id, times, group = NULL, exposure,
               name.history = "h")
```

Arguments

input	dataframe in wide format (e.g., indexed by person)
id	unique observation identifier e.g. "id"

times	a vector of measurement times e.g. c(0,1,2)
group	an optional baseline variable upon which to aggregate the exposure history. This argument provides a way to adjust the metrics for a baseline covariate. For example, in the context of a trial, the grouping variable could be treatment assignment. In the context of a cohort study, this could be site e.g. "v".
exposure	the root name for exposure e.g. "a"
name.history	desired root name for time-indexed history variables e.g. "h"

Value

A "wide" dataframe with an added set of exposure history variables for a time-varying exposure. The new history variables will use the time-indices in the exposure vectors you supply.

Examples

```
# Simulate wide data set for two subjects
id <- as.numeric(c(1, 2))
a_0 <- as.numeric(c(0, 1))
a_1 <- as.numeric(c(1, 1))
a_2 <- as.numeric(c(1, 0))
l_0 <- as.numeric(rbinom(2, 1, 0.5))
l_1 <- as.numeric(rbinom(2, 1, 0.5))
l_2 <- as.numeric(rbinom(2, 1, 0.5))
m_0 <- as.numeric(rbinom(2, 1, 0.5))
m_1 <- as.numeric(rbinom(2, 1, 0.5))
m_2 <- as.numeric(rbinom(2, 1, 0.5))
n_0 <- as.numeric(rbinom(2, 1, 0.5))
n_1 <- as.numeric(rbinom(2, 1, 0.5))
n_2 <- as.numeric(rbinom(2, 1, 0.5))

mydata.wide <- data.frame(id, a_0, a_1, a_2,
                        l_0, l_1, l_2,
                        m_0, m_1, m_2,
                        n_0, n_1, n_2)

# Run the makehistory.one() function
mydata.history <- makehistory.one(input=mydata.wide,
                                id="id",
                                times=c(0,1,2),
                                exposure="a",
                                name.history="h"
                                )
```

makehistory.two

Function to create joint exposure history for two distinct time-varying exposures

Description

Function to create joint exposure history for two distinct time-varying exposures

Usage

```
makehistory.two(input, id, group = NULL, exposure.a, exposure.b,
  name.history.a = "ha", name.history.b = "hb", times)
```

Arguments

input	dataframe in wide format (e.g., indexed by person)
id	unique observation identifier e.g. "id"
group	an optional baseline variable upon which to aggregate the exposure history. This argument provides a way to adjust the metrics for a baseline covariate. For example, in the context of a trial, the grouping variable could be treatment assignment. In the context of a cohort study, this could be site e.g. "v".
exposure.a	the root name for the first exposure e.g. "a"
exposure.b	the root name for the second exposure e.g. "z"
name.history.a	desired root name for the first time-indexed history variables e.g. "ha"
name.history.b	desired root name for the second time-indexed history variables e.g. "hb"
times	a vector of measurement times e.g. c(0,1,2)

Details

When the exposure is multivariate, the idea is to diagnose each exposure separately (see eAppendix of Jackson 2016). From the perspective of using the R-functions, the only difference is to use exposure history based on all exposures that comprise the multivariate exposure. It is important that such joint exposure history accurately reflect the ordering of each component exposure. The function `makehistory.two()` creates an appropriate joint exposure history for each of two exposures, assuming that exposures in its argument `list.exposure.a` (e.g. A) precede those in `list.exposure.b` (e.g. Z) at any given index as described in the eAppendix of Jackson 2016. In that example, exposure A(t) always precedes exposure Z(t) such that the joint history of A(2) is A(1),A(0),Z(0) while the joint history of Z(2) is A(1),A(0),Z(1),Z(0). If one exposure does not precede the other, investigators will still need to use an appropriate joint exposure history and can specify either order as desired. Note that the exposure history produced by the function `makehistory.two()` will be inappropriate if the relative ordering of A(t) and Z(t) varies over time.

Value

A "wide" dataframe with an added set of exposure history variables for each of the two time-varying exposures, properly accounting for their temporal ordering (i.e. exposure "a" precedes exposure "b" at any time t). The new history variables will use the time-indices in the exposure vectors you supply.

Examples

```

# Simulate wide data set for two subjects
id <- as.numeric(c(1, 2))
a_0 <- as.numeric(c(0, 1))
a_1 <- as.numeric(c(1, 1))
a_2 <- as.numeric(c(1, 0))
z_0 <- as.numeric(c(1, 0))
z_1 <- as.numeric(c(0, 0))
z_2 <- as.numeric(c(0, 1))
l_0 <- as.numeric(rbinom(2, 1, 0.5))
l_1 <- as.numeric(rbinom(2, 1, 0.5))
l_2 <- as.numeric(rbinom(2, 1, 0.5))
m_0 <- as.numeric(rbinom(2, 1, 0.5))
m_1 <- as.numeric(rbinom(2, 1, 0.5))
m_2 <- as.numeric(rbinom(2, 1, 0.5))
n_0 <- as.numeric(rbinom(2, 1, 0.5))
n_1 <- as.numeric(rbinom(2, 1, 0.5))
n_2 <- as.numeric(rbinom(2, 1, 0.5))

mydata.wide <- data.frame(id, a_0, a_1, a_2,
                        z_0, z_1, z_2,
                        l_0, l_1, l_2,
                        m_0, m_1, m_2,
                        n_0, n_1, n_2)

# Run the makehistory.two() function
mydata.history <- makehistory.two(input=mydata.wide,
                                id="id",
                                times=c(0,1,2),
                                exposure.a="a",
                                exposure.b="z",
                                name.history.a="ha",
                                name.history.b="hb"
                                )

```

makeplot

Function to create balance plot for a specified diagnostic. Takes input from balance() or apply.scope() or diagnose().

Description

Function to create balance plot for a specified diagnostic. Takes input from balance() or apply.scope() or diagnose().

Usage

```

makeplot(input, diagnostic, approach, metric = "SMD", censoring, scope,
         average.over = NULL, stratum = NULL, label.exposure = "A",
         label.covariate = "C", lbound = -1, ubound = 1, ratio = 2,

```

```

text.axis.title = 8, text.axis.y = 6.5, text.axis.x = 6.5,
text.strip.y = 10, text.strip.x = 10, point.size = 0.75,
zeroline.size = 0.1, refline.size = 0.1, refline.limit.a = -0.25,
refline.limit.b = 0.25, panel.spacing.size = 0.75,
axis.title = NULL, label.width = 15, grouptype = "none",
shapetype = NULL, colourtype = NULL, colour_palette = "Set1",
legend.title = "", legend.position = "bottom", text.legend = NULL)

```

Arguments

input	output from <code>balance()</code> or <code>diagnose()</code> or <code>apply.scope()</code>
diagnostic	diagnostic of interest e.g. 1, 2, or 3
approach	adjustment method e.g. "none" or "weight" or "stratify"
metric	scale e.g. "D" for mean difference, "SMD" for standardized mean difference
censoring	use censoring indicators/weights e.g. "yes" or "no"
scope	report the entire trellis e.g. "all", the diagonal e.g. "recent", or a summary e.g. "average"
average.over	level of summary for average e.g. "values" or "history" or "time" or "distance"
stratum	the propensity-score stratum to plot
label.exposure	common label used for exposure axis in plot (default = "A")
label.covariate	common label used for covariate axis in plot (default = "C")
lbound	lower bound for mean difference or standardized mean difference (default = -1)
ubound	upper bound for mean difference or standardized mean difference (default = 1)
ratio	aspect ratio of plot (default = 2)
text.axis.title	font size of axis title (default = 8)
text.axis.y	font size of y-axis values (default = 6.5)
text.axis.x	font size of x-axis values (default = 6.5)
text.strip.y	font size of y-axis label (default = 10)
text.strip.x	font size of x-axis label (default = 10)
point.size	size of data points (default = 0.75)
zeroline.size	width of the line plotted at mean difference = 0 or standardized mean difference = 0 (default = 0.1)
refline.size	width of the lines plotted at the specified fraction of the mean difference or standardized mean difference (default = 0.1)
refline.limit.a	position of the lower reference line, specified as a fraction of the mean difference or standardized mean difference (default = -0.25)
refline.limit.b	position of the upper reference line, specified as a fraction of the mean difference or standardized mean difference (default = 0.25)

omit.history	<i>Function to remove irrelevant covariate history from a tidy dataframe used to construct balance tables and plots. Takes input from lengthen(), balance() or diagnose().</i>
--------------	--

Description

Function to remove irrelevant covariate history from a tidy dataframe used to construct balance tables and plots. Takes input from lengthen(), balance() or diagnose().

Usage

```
omit.history(input, omission, covariate.name, distance = NULL,
            times = NULL)
```

Arguments

input	restructured tidy dataframe from lengthen() or a dataframe from balance() or diagnose()
omission	type of omission e.g. "fixed" or "relative" or "same.time"
covariate.name	root name of the covariate e.g. "m"
distance	the distance between exposure and covariate measurements e.g. 2
times	a vector of measurement times for the covariate e.g. c(1,2,3)

Details

Intended for use with Diagnostics 1 and 3. omit.history() will take the dataframe produced by lengthen() and remove covariate measurements based on their fixed measurement time or relative distance from exposure measurements (at time t) i.e. ones that do not support exchangeability assumptions at time t. The covariate.name argument is used to name the covariate whose history you wish to modify. To process the same manipulation for a set of covariates, simply supply a vector of covariate names to covariate.name. The omission argument determines whether the covariate history is (i) set to missing for certain covariate measurement times (omission = "fixed" with times = a vector of integers) or (ii) set to missing only for covariate measurement times at or before a certain distance k from exposure measurement times (omission = "relative" with distance = some integer) or (iii) set to missing only for covariate measurements that share the same timing as exposure measurements (omission = "same.time"). The removed values are set to missing. For example, using the "fixed" omission option for covariate "I" at time 2 will set all data on "I" at time 2 to missing, regardless of the exposure measurement time. In contrast, using the "relative" omission option for covariate "I" with distance 2 will only set to missing data on "I" that is measured two units or more before the exposure measurement time (i.e. t-2, t-3, t-4 and so on). Last, using the "same.time" omission option for covariate "I" will set to missing all data on "I" that is measured at the same time as the exposure. Missing data will be ignored when this dataframe is supplied to the balance() function. They will not contribute to the resulting covariate balance table, nor to plots produced by makeplot(), nor will they contribute to any summary metrics are estimated by averaging over person-time. Note that omit.history also accepts input from balance() and diagnose()

when their scope argument has been set to "all" (i.e., not averaging over time or distance or selecting times based on recency of measurements).

Value

A "tidy" dataframe where covariate measurements have been removed based on their fixed measurement time or relative distance from exposure measurements (at time t). The removed covariate measurements are typically ones chosen to be ones that do not support exchangeability assumptions at time t.

Examples

```
# Simulate the output of lengthen()
id <- as.numeric(rep(c(1,1,1,2,2,2), 7))
time.exposure <- as.numeric(rep(c(0,1,2), 14))
a <- as.character(rep(c(0,1,1,1,1,0), 7))
h <- as.character(rep(c("H", "H0", "H01", "H", "H1", "H11"), 7))

name.cov <- as.character(c(rep("n",6), rep("l",18), rep("m",18)))

time.covariate <- as.numeric(c(rep(0,6), rep(c(rep(0,6),
      rep(1,6),rep(2,6)), 2)))

value.cov <- as.numeric(c(rep(1,9), rep(0,3), rep(1,6),
      rep(0,3), rep(1,3), rep(0,12),
      rep(1,3), rep(0,3)))

mydata.long <- data.frame(id, time.exposure, a, h,
      name.cov, time.covariate, value.cov)

# Run the omit.history() function
mydata.long.omit <- omit.history(input=mydata.long,
      omission="relative",
      covariate.name=c("l", "m"),
      distance=1)
```

toy_long

Artificial data set used to test the functionality of confoundr.

Description

The toy_long data set contains 30,000 records and 15 variables. These variables include time-varying exposures, outcomes, and covariates, along with strata and censoring indicators. Time-varying inverse-probability-of-exposure weights and censoring weights are available as well.

Usage

```
data(toy_long)
```

Format

A data frame with 3,000 rows and 13 variables:

uid subject ID

time time of observation

a exposure measurement at time t

l covariate measurement at time t

m covariate measurement at time t

n covariate measurement at time t

o covariate measurement at time t

p covariate measurement at time t

s censoring indicator at time t

h exposure history at time t

hx grouped exposure history, by p_0, at time t

wa inverse probability of exposure and censoring weight at time t

wax cumulative inverse probability of exposure weight at time t

wsx cumulative inverse probability of censoring weight at time t

e5 propensity score strata at time t

toy_long_dropoutN *Artificial data set used to test the functionality of confoundr.*

Description

The toy_wide_dropoutN data set contains 10,000 records and 52 variables. These variables include time-varying exposures, outcomes, and covariates, along with strata and censoring indicators. Time-varying inverse-probability-of-exposure weights and censoring weights are available as well.

Usage

```
data(toy_long_dropoutN)
```

Format

A data frame with 3,000 rows and 13 variables:

uid subject ID
time time of observation
a exposure measurement at time t
l covariate measurement at time t
m covariate measurement at time t
n covariate measurement at time t
o covariate measurement at time t
p covariate measurement at time t
s censoring indicator at time t
h exposure history at time t
hx grouped exposure history, by p_0, at time t
wa inverse probability of exposure and censoring weight at time t
wax cumulative inverse probability of exposure weight at time t
wsx cumulative inverse probability of censoring weight at time t
e5 propensity score strata at time t

toy_long_dropoutY	<i>Artificial data set used to test the functionality of confoundr.</i>
-------------------	---

Description

The toy_long_dropoutY data set contains 28,410 records and 16 variables. These variables include time-varying exposures, outcomes, and covariates, along with strata and censoring indicators. Toy data are removed after s equals one. Time-varying inverse-probability-of-exposure weights and censoring weights are available as well.

Usage

```
data(toy_long_dropoutY)
```

Format

A data frame with 2,847 rows and 13 variables:

uid subject ID
time time of observation
a exposure measurement at time t
l covariate measurement at time t
m covariate measurement at time t

n covariate measurement at time t
o covariate measurement at time t
p covariate measurement at time t
s censoring indicator at time t
h exposure history at time t
hx grouped exposure history, by p_0, at time t
wa inverse probability of exposure and censoring weight at time t
wax cumulative inverse probability of exposure weight at time t
wsx cumulative inverse probability of censoring weight at time t
e5 propensity score strata at time t

toy_wide_censN	<i>Artifical data set used to test the functionality of confoundr.</i>
----------------	--

Description

The toy_wide_censN data set contains 1,000 records and 52 variables. These variables include time-varying exposures, outcomes, and covariates, along with strata and censoring indicators. Time-varying inverse-probability-of-exposure weights and censoring weights are available as well.

Usage

```
data(toy_wide_censN)
```

Format

A data frame with 1,000 rows and 52 variables:

uid subject ID
a_0 exposure measurement at time 0
a_1 exposure measurement at time 1
a_2 exposure measurement at time 2
l_0 covariate measurement at time 0
l_1 covariate measurement at time 1
l_2 covariate measurement at time 2
m_0 covariate measurement at time 0
m_1 covariate measurement at time 1
m_2 covariate measurement at time 2
n_0 covariate measurement at time 0
n_1 covariate measurement at time 1
n_2 covariate measurement at time 2

o_0 covariate measurement at time 0
o_1 covariate measurement at time 1
o_2 covariate measurement at time 2
p_0 covariate measurement at time 0
p_1 covariate measurement at time 1
p_2 covariate measurement at time 2
s_0 censoring indicator at time 0
s_1 censoring indicator at time 1
s_2 censoring indicator at time 2
hsone_0 censor history at time 0
hsone_1 censor history at time 1
hsone_2 censor history at time 2
hsoneg_0 grouped by p_0, censor history at time 0
hsoneg_1 grouped by p_0, censor history at time 1
hsoneg_2 grouped by p_0, censor history at time 2
hatwo_0 a joint history given a,s at time 0
hatwo_1 a joint history given a,s at time 1
hatwo_2 a joint history given a,s at time 2
hatwog_0 grouped by p_0, a joint history given a,s at time 0
hatwog_1 grouped by p_0, a joint history given a,s at time 1
hatwog_2 grouped by p_0, a joint history given a,s at time 2
hstwo_0 s joint history given a,s at time 0
hstwo_1 s joint history given a,s at time 1
hstwo_2 s joint history given a,s at time 2
hstwog_0 grouped by p_0, s joint history given a,s at time 0
hstwog_1 grouped by p_0, s joint history given a,s at time 1
hstwog_2 grouped by p_0, s joint history given a,s at time 2
wa_0 inverse probability of exposure weight at time 0
wa_1 inverse probability of exposure weight at time 1
wa_2 inverse probability of exposure weight at time 2
wax_0 cumulative inverse probability weight of exposure at time 0
wax_1 cumulative inverse probability weight of exposure at time 1
wax_2 cumulative inverse probability weight of exposure at time 2
wsx_0 cumulative inverse probability of censoring weight at time 0
wsx_1 cumulative inverse probability of censoring weight at time 1
wsx_2 cumulative inverse probability of censoring weight at time 2
e5_0 propensity score strata at time 0
e5_1 propensity score strata at time 1
e5_2 propensity score strata at time 2

toy_wide_censY	<i>Artificial data set used to test the functionality of confoundr.</i>
----------------	---

Description

The toy_wide_censY data set contains 1,000 records and 52 variables. These variables include time-varying exposures, outcomes, and covariates, along with strata and censoring indicators. Time-varying inverse-probability-of-exposure weights and censoring weights are available as well.

Usage

```
data(toy_wide_censY)
```

Format

A data frame with 1,000 rows and 52 variables:

- uid** subject ID
- a_0** exposure measurement at time 0
- a_1** exposure measurement at time 1
- a_2** exposure measurement at time 2
- l_0** covariate measurement at time 0
- l_1** covariate measurement at time 1
- l_2** covariate measurement at time 2
- m_0** covariate measurement at time 0
- m_1** covariate measurement at time 1
- m_2** covariate measurement at time 2
- n_0** covariate measurement at time 0
- n_1** covariate measurement at time 1
- n_2** covariate measurement at time 2
- o_0** covariate measurement at time 0
- o_1** covariate measurement at time 1
- o_2** covariate measurement at time 2
- p_0** covariate measurement at time 0
- p_1** covariate measurement at time 1
- p_2** covariate measurement at time 2
- s_0** censoring indicator at time 0
- s_1** censoring indicator at time 1
- s_2** censoring indicator at time 2
- hson_0** censor history at time 0

hsone_1 censor history at time 1
hsone_2 censor history at time 2
hsoneg_0 grouped by p_0, censor history at time 0
hsoneg_1 grouped by p_0, censor history at time 1
hsoneg_2 grouped by p_0, censor history at time 2
hatwo_0 a joint history given a,s at time 0
hatwo_1 a joint history given a,s at time 1
hatwo_2 a joint history given a,s at time 2
hatwog_0 grouped by p_0, a joint history given a,s at time 0
hatwog_1 grouped by p_0, a joint history given a,s at time 1
hatwog_2 grouped by p_0, a joint history given a,s at time 2
hstwo_0 s joint history given a,s at time 0
hstwo_1 s joint history given a,s at time 1
hstwo_2 s joint history given a,s at time 2
hstwog_0 grouped by p_0, s joint history given a,s at time 0
hstwog_1 grouped by p_0, s joint history given a,s at time 1
hstwog_2 grouped by p_0, s joint history given a,s at time 2
wa_0 inverse probability of exposure weight at time 0
wa_1 inverse probability of exposure weight at time 1
wa_2 inverse probability of exposure weight at time 2
wax_0 cumulative inverse probability weight of exposure at time 0
wax_1 cumulative inverse probability weight of exposure at time 1
wax_2 cumulative inverse probability weight of exposure at time 2
wsx_0 cumulative inverse probability of censoring weight at time 0
wsx_1 cumulative inverse probability of censoring weight at time 1
wsx_2 cumulative inverse probability of censoring weight at time 2
e5_0 propensity score strata at time 0
e5_1 propensity score strata at time 1
e5_2 propensity score strata at time 2

toy_wide_dropoutN *Artificial data set used to test the functionality of confoundr.*

Description

The toy_wide_censN data set contains 1,000 records and 52 variables. These variables include time-varying exposures, outcomes, and covariates, along with strata and censoring indicators. Time-varying inverse-probability-of-exposure weights and censoring weights are available as well.

Usage

```
data(toy_wide_dropoutN)
```

Format

A data frame with 1,000 rows and 52 variables:

- uid** subject ID
- a_0** exposure measurement at time 0
- a_1** exposure measurement at time 1
- a_2** exposure measurement at time 2
- l_0** covariate measurement at time 0
- l_1** covariate measurement at time 1
- l_2** covariate measurement at time 2
- m_0** covariate measurement at time 0
- m_1** covariate measurement at time 1
- m_2** covariate measurement at time 2
- n_0** covariate measurement at time 0
- n_1** covariate measurement at time 1
- n_2** covariate measurement at time 2
- o_0** covariate measurement at time 0
- o_1** covariate measurement at time 1
- o_2** covariate measurement at time 2
- p_0** covariate measurement at time 0
- p_1** covariate measurement at time 1
- p_2** covariate measurement at time 2
- s_0** censoring indicator at time 0
- s_1** censoring indicator at time 1
- s_2** censoring indicator at time 2
- haone_0** exposure history at time 0

haone_1 exposure history at time 1
haone_2 exposure history at time 2
haoneg_0 grouped by p_0, exposure history at time 0
haoneg_1 grouped by p_0, exposure history at time 1
haoneg_2 grouped by p_0, exposure history at time 2
hatwo_0 a joint history given a,s at time 0
hatwo_1 a joint history given a,s at time 1
hatwo_2 a joint history given a,s at time 2
hatwog_0 grouped by p_0, a joint history given a,s at time 0
hatwog_1 grouped by p_0, a joint history given a,s at time 1
hatwog_2 grouped by p_0, a joint history given a,s at time 2
hstwo_0 s joint history given a,s at time 0
hstwo_1 s joint history given a,s at time 1
hstwo_2 s joint history given a,s at time 2
hstwog_0 grouped by p_0, s joint history given a,s at time 0
hstwog_1 grouped by p_0, s joint history given a,s at time 1
hstwog_2 grouped by p_0, s joint history given a,s at time 2
wa_0 inverse probability of exposure weight at time 0
wa_1 inverse probability of exposure weight at time 1
wa_2 inverse probability of exposure weight at time 2
wax_0 cumulative inverse probability weight of exposure at time 0
wax_1 cumulative inverse probability weight of exposure at time 1
wax_2 cumulative inverse probability weight of exposure at time 2
wsx_0 cumulative inverse probability of censoring weight at time 0
wsx_1 cumulative inverse probability of censoring weight at time 1
wsx_2 cumulative inverse probability of censoring weight at time 2
e5_0 propensity score strata at time 0
e5_1 propensity score strata at time 1
e5_2 propensity score strata at time 2

toy_wide_dropoutY *Artificial data set used to test the functionality of confoundr.*

Description

The toy_wide_dropoutY data set contains 1,000 records and 52 variables. These variables include time-varying exposures, outcomes, and covariates, along with strata and censoring indicators. Time-varying inverse-probability-of-exposure weights and censoring weights are available as well.

Usage

```
data(toy_wide_dropoutY)
```

Format

A data frame with 1,000 rows and 52 variables:

- uid** subject ID
- a_0** exposure measurement at time 0
- a_1** exposure measurement at time 1
- a_2** exposure measurement at time 2
- l_0** covariate measurement at time 0
- l_1** covariate measurement at time 1
- l_2** covariate measurement at time 2
- m_0** covariate measurement at time 0
- m_1** covariate measurement at time 1
- m_2** covariate measurement at time 2
- n_0** covariate measurement at time 0
- n_1** covariate measurement at time 1
- n_2** covariate measurement at time 2
- o_0** covariate measurement at time 0
- o_1** covariate measurement at time 1
- o_2** covariate measurement at time 2
- p_0** covariate measurement at time 0
- p_1** covariate measurement at time 1
- p_2** covariate measurement at time 2
- s_0** censoring indicator at time 0
- s_1** censoring indicator at time 1
- s_2** censoring indicator at time 2
- haone_0** exposure history at time 0

haone_1 exposure history at time 1
haone_2 exposure history at time 2
haoneg_0 grouped by p_0, exposure history at time 0
haoneg_1 grouped by p_0, exposure history at time 1
haoneg_2 grouped by p_0, exposure history at time 2
hatwo_0 a joint history given a,s at time 0
hatwo_1 a joint history given a,s at time 1
hatwo_2 a joint history given a,s at time 2
hatwog_0 grouped by p_0, a joint history given a,s at time 0
hatwog_1 grouped by p_0, a joint history given a,s at time 1
hatwog_2 grouped by p_0, a joint history given a,s at time 2
hstwo_0 s joint history given a,s at time 0
hstwo_1 s joint history given a,s at time 1
hstwo_2 s joint history given a,s at time 2
hstwog_0 grouped by p_0, s joint history given a,s at time 0
hstwog_1 grouped by p_0, s joint history given a,s at time 1
hstwog_2 grouped by p_0, s joint history given a,s at time 2
wa_0 inverse probability of exposure weight at time 0
wa_1 inverse probability of exposure weight at time 1
wa_2 inverse probability of exposure weight at time 2
wax_0 cumulative inverse probability weight of exposure at time 0
wax_1 cumulative inverse probability weight of exposure at time 1
wax_2 cumulative inverse probability weight of exposure at time 2
wsx_0 cumulative inverse probability of censoring weight at time 0
wsx_1 cumulative inverse probability of censoring weight at time 1
wsx_2 cumulative inverse probability of censoring weight at time 2
e5_0 propensity score strata at time 0
e5_1 propensity score strata at time 1
e5_2 propensity score strata at time 2

widen	<i>Function to transform data from person-time format to person format suitable for lengthen()</i>
-------	--

Description

Function to transform data from person-time format to person format suitable for lengthen()

Usage

```
widen(input, id, time, exposure, covariate, history = NULL,
      weight.exposure = NULL, weight.censor = NULL, strata = NULL,
      censor = NULL)
```

Arguments

input	dataframe in long format e.g., a person-time format
id	unique identifier at the unit (person) level
time	unique index for each observation within each unit
exposure	the exposure of interest at time t
covariate	a vector of covariates at time t
history	variable describing exposure history through time t
weight.exposure	inverse probability weight for exposure, at or through time t
weight.censor	cumulative inverse probability weight for censoring through time t
strata	propensity score strata at time t
censor	censoring indicators at time t

Details

Numeric formats are preserved, factors are coerced into character.

Value

A "wide" dataframe where each row uniquely indexes a single subject's data, so that columns index measurement of each variable at each time. The indices should be indicated with an underscore suffix followed by the time, e.g. `variable_1`, `variable_2`.

Examples

```
# Simulate long data set for two subjects
id <- as.numeric(c(1, 1, 1, 2, 2, 2))
time <- as.numeric(c(0, 1, 2, 0, 1, 2))
a <- as.numeric(c(0, 1, 1, 1, 1, 0))
l <- as.numeric(rbinom(6, 1, 0.5))
```

```
m <- as.numeric(rbinom(6, 1, 0.5))
n <- as.numeric(rbinom(6, 1, 0.5))

mydata.long <- data.frame(id, time, a, l, m, n)

# Run the widen() function
mydata.wide <- widen(input=mydata.long,
                    id="id", time="time",
                    exposure="a",
                    covariate=c("l", "m", "n")
                    )
```

Index

* datasets

- catie_sim, 6
- example_sml, 11
- toy_long, 21
- toy_long_dropoutN, 22
- toy_long_dropoutY, 23
- toy_wide_censN, 24
- toy_wide_censY, 26
- toy_wide_dropoutN, 28
- toy_wide_dropoutY, 30

apply.scope, 2

balance, 4

catie_sim, 6

diagnose, 8

example_sml, 11

lengthen, 12

makehistory.one, 14

makehistory.two, 15

makeplot, 17

omit.history, 20

toy_long, 21

toy_long_dropoutN, 22

toy_long_dropoutY, 23

toy_wide_censN, 24

toy_wide_censY, 26

toy_wide_dropoutN, 28

toy_wide_dropoutY, 30

widen, 32