Package 'ipw'

January 27, 2026

Version 1.3.0

Title Estimate Inverse Probability Weights

Author Hung Thai Tran [aut, cre],
Willem M. van der Wal [aut],
Ronald B. Geskus [aut] (maintainer 2011-2022)

Maintainer Hung Thai Tran <hungtt@oucru.org>

Depends R (>= 3.2.0)

Imports MASS, nnet, survival, geepack, graphics, methods, stats

Suggests nlme, survey, boot, testthat (>= 3.0.0)

Description Functions to estimate the probability to receive the observed treatment, based on
individual characteristics. The inverse of these probabilities can be used as weights when
estimating causal effects from observational data via marginal structural models. Both point
treatment situations and longitudinal studies can be analysed. The same functions can be used to
correct for informative censoring.

Encoding UTF-8

License MIT + file LICENSE

NeedsCompilation no

RoxygenNote 7.3.3.9000

LazyData true

Config/testthat/edition 3

URL https://github.com/TranHung93/ipw

BugReports https://github.com/TranHung93/ipw/issues

Repository CRAN

Date/Publication 2026-01-27 07:10:02 UTC

Contents

---

basdat   *HIV: TB and Survival (Baseline Data)*

Description

Simulated dataset. Baseline data of 386 HIV positive individuals, including time of first active tuberculosis, time of death, individual end time. Time varying CD4 measurements of these patients are included in dataset timedat.

Usage

    data(basdat)

Format

A data frame with 386 observations on the following 4 variables:

id Patient ID.

Ttb Time of first active tuberculosis, measured in days since HIV seroconversion.

Tdeath Time of death, measured in days since HIV seroconversion.

Tend Individual end time (either death or censoring), measured in days since HIV seroconversion.

Details

These simulated data are used together with data in timedat in a detailed causal modelling example using inverse probability weighting (IPW). See ipwtm for the example. Data were simulated using the algorithm described in Van der Wal e.a. (2009).#' [Image of a multistate model for HIV, TB, and death]

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, Ronald B. Geskus <rgeskus@oucru.org>

References

Van der Wal W.M. & Geskus R.B. (2011). ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, **43**(13), 1-23. doi:10.18637/jss.v043.i13.

Van der Wal W.M., Prins M., Lumbreras B. & Geskus R.B. (2009). A simple G-computation algorithm to quantify the causal effect of a secondary illness on the progression of a chronic disease. *Statistics in Medicine*, **28**(18), 2325-2337.

See Also

haartdat, ipwplot, ipwpoint, ipwtm, timedat, tstartfun

Examples

```
# Detailed examples can be found in the ipwtm documentation:
# ?ipwtm
```

---

haartdat	*HAART and Survival in HIV Patients*

Description

Survival data measured in 1200 HIV positive patients. Start of follow-up is HIV seroconversion. Each row corresponds to a 100 day interval of follow-up time, using the counting process notation. Patients can initiate HAART therapy. CD4 count is a confounder for the effect of HAART on mortality.

Usage

```
data(haartdat)
```

Format

A data frame with 1200 patients and multiple observations per patient (counting process notation) on the following 8 variables:

patient Patient ID.

tstart Starting time for each interval of follow-up, measured in days since HIV seroconversion.

fuptime End time for each interval of follow-up, measured in days since HIV seroconversion.

haartind Indicator for the initiation of HAART therapy at the end of the interval (0 = HAART not initiated / 1 = HAART initiated).

event Indicator for death at the end of the interval (0 = alive / 1 = died).

sex Sex (0 = male / 1 = female).

age Age at the start of follow-up (years).

cd4.sqrt Square root of CD4 count, measured at fuptime, before haartind.

endtime The final observed time point for the individual.

dropout Indicator for dropout/censoring at the end of the interval (0 = no, 1 = yes).#'

Details

These data were simulated to demonstrate Inverse Probability Weighting (IPW). To allow for models predicting the initiation of HAART at fuptime = 0, the starting time for the first interval of each patient is set to -100.

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, Ronald B. Geskus <rgeskus@oucru.org>

References

Van der Wal W.M. & Geskus R.B. (2011). ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, **43**(13), 1-23. doi:10.18637/jss.v043.i13.

See Also

basdat, ipwplot, ipwpoint, ipwtm, timedat, tstartfun

Examples

```
# For a full example of how to use this data with ipwtm, see:
# ?ipwtm
```

---

healthdat   *IQ, Income and Health*

Description

A simulated dataset containing measurements of IQ, monthly income, and health scores for 1000 individuals.

Usage

```
data(healthdat)
```

Format

A data frame with 1000 rows, with each row corresponding to a separate individual. The following 4 variables are included:

id Individual ID.

iq IQ score.

income Gross monthly income (EUR).

health Health score (0-100).

Details

In these simulated data, IQ acts as a confounder for the causal effect of income on health. This dataset is primarily used to illustrate point-treatment inverse probability weighting.

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, Ronald B. Geskus <rgeskus@oucru.org>

References

Van der Wal W.M. & Geskus R.B. (2011). ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, **43**(13), 1-23. doi:10.18637/jss.v043.i13

See Also

basdat, haartdat, ipwplot, ipwpoint, ipwtm, timedat, tstartfun

Examples

```
# For an example of how to use this data with ipwpoint, see:
# ?ipwpoint
```

---

ipwplot	*Plot Inverse Probability Weights*

Description

For time varying weights: display boxplots within strata of follow-up time. For point treatment weights: display density plot.

Usage

```
ipwplot(
  weights,
  timevar = NULL,
  binwidth = NULL,
  logscale = TRUE,
  xlab = NULL,
  ylab = NULL,
  main = "",
  ref = TRUE,
  ...
)
```

Arguments

weights	numerical vector of inverse probability weights to plot.
timevar	numerical vector representing follow-up time. When specified, boxplots within strata of follow-up time are displayed. When left unspecified, a density plot is displayed.
binwidth	numerical value indicating the width of the intervals of follow-up time; for each interval a boxplot is made. Ignored when `timevar` is not specified.
logscale	logical value. If `TRUE`, weights are plotted on a logarithmic scale.
xlab	label for the horizontal axis.
ylab	label for the vertical axis.

main	main title for the plot.
ref	logical value. If TRUE, a reference line is plotted at y=1.
...	additional arguments passed to [boxplot](#) (when timevar is specified) or [plot](#) (when timevar is not specified).

Value

A plot is displayed.

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, Ronald B. Geskus <rgeskus@oucru.org>

References

Van der Wal W.M. & Geskus R.B. (2011). ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, **43**(13), 1-23. doi:10.18637/jss.v043.i13

See Also

basdat, haartdat, ipwplot, ipwpoint, ipwtm, timedat, tstartfun.

Examples

```
#see ?ipwpoint and ?ipwtm for examples
```

---

ipwpoint   *Estimate Inverse Probability Weights (Point Treatment)*

Description

Estimate inverse probability weights to fit marginal structural models in a point treatment situation. The exposure for which we want to estimate the causal effect can be binomial, multinomial, ordinal or continuous. Both stabilized and unstabilized weights can be estimated.

Usage

```
ipwpoint(
  exposure,
  family,
  link,
  numerator = NULL,
  denominator,
  data,
  trunc = NULL,
  ...
)
```

Arguments

exposure	a vector, representing the exposure variable of interest. Both numerical and categorical variables can be used. A binomial exposure variable should be coded using values 0/1.
family	is used to specify a family of link functions, used to model the relationship between the variables in numerator or denominator and exposure, respectively. Alternatives are "binomial","multinomial", "ordinal" and "gaussian". A specific link function is then chosen using the argument link, as explained below. Regression models are fitted using glm, multinom, polr or glm, respectively.
link	specifies the link function between the variables in numerator or denominator and exposure, respectively. For family = "binomial" (fitted using glm) alternatives are "logit", "probit", "cauchit", "log" and "cloglog". For family = "multinomial" this argument is ignored, and multinomial logistic regression models are always used (fitted using multinom). For family = "ordinal" (fitted using polr) alternatives are "logit", "probit", "cauchit", and "cloglog". For family = "gaussian" this argument is ignored, and a linear regression model with identity link is always used (fitted using glm).
numerator	is a formula, specifying the right-hand side of the model used to estimate the elements in the numerator of the inverse probability weights. When left unspecified, unstabilized weights with a numerator of 1 are estimated.
denominator	is a formula, specifying the right-hand side of the model used to estimate the elements in the denominator of the inverse probability weights. This typically includes the variables specified in the numerator model, as well as confounders for which to correct.
data	is a dataframe containing exposure and the variables used in numerator and denominator.
trunc	optional truncation percentile (0-0.5). E.g. when trunc = 0.01, the left tail is truncated to the 1st percentile, and the right tail is truncated to the 99th percentile.When specified, both un-truncated and truncated weights are returned.
...	are further arguments passed to the function that is used to estimate the numerator and denominator models (the function is chosen using family).

Details

For each unit under observation, this function computes an inverse probability weight, which is the ratio of two probabilities:

- the numerator contains the probability of the observed exposure level given observed values of stabilization factors (usually a set of baseline covariates). These probabilities are estimated using the model regressing exposure on the terms in numerator, using the link function indicated by family and link.

- the denominator contains the probability of the observed exposure level given the observed values of a set of confounders, as well as the stabilization factors in the numerator. These probabilities are estimated using the model regressing exposure on the terms in denominator, using the link function indicated by family and link.

When the models from which the elements in the numerator and denominator are predicted are correctly specified, and there is no unmeasured confounding, weighting the observations by the inverse probability weights adjusts for confounding of the effect of the exposure of interest. On the weighted dataset a marginal structural model can then be fitted, quantifying the causal effect of the exposure on the outcome of interest.

With `numerator` specified, stabilized weights are computed, otherwise unstabilized weighs with a numerator of 1 are computed. With a continuous exposure, using `family = "gaussian"`, weights are computed using the ratio of predicted densities. Therefore, for `family = "gaussian"` only stabilized weights can be used, since unstabilized weights would have infinity variance.

Value

A list containing the following elements:

`ipw.weights`	is a vector containing inverse probability weights for each unit under observation. This vector is returned in the same order as the measurements contained in `data`, to facilitate merging.
`weights.trunc`	is a vector containing truncated inverse probability weights for each unit under observation. This vector is only returned when `trunc` is specified.
`call`	is the original function call to `ipwpoint`.
`num.mod`	is the numerator model, only returned when `numerator` is specified.
`den.mod`	is the denominator model.

Missing values

Currently, the `exposure` variable and the variables used in `numerator` and `denominator` should not contain missing values.

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, Ronald B. Geskus <rgeskus@oucru.org>

References

Cole, S.R. & Hernán, M.A. (2008). Constructing inverse probability weights for marginal structural models. *American Journal of Epidemiology*, **168**(6), 656-664.

Robins, J.M., Hernán, M.A. & Brumback, B.A. (2000). Marginal structural models and causal inference in epidemiology. *Epidemiology*, **11**, 550-560.

Van der Wal W.M. & Geskus R.B. (2011). ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, **43**(13), 1-23. doi:10.18637/jss.v043.i13.

See Also

basdat, haartdat, ipwplot, ipwpoint, ipwtm, timedat, tstartfun.

Examples

```
# Simulate data with continuous confounder and outcome, binomial exposure.
# Marginal causal effect of exposure on outcome: 10.
n <- 1000
simdat <- data.frame(l = rnorm(n, 10, 5))
a.lin <- simdat$l - 10
pa <- exp(a.lin)/(1 + exp(a.lin))
simdat$a <- rbinom(n, 1, prob = pa)
simdat$y <- 10*simdat$a + 0.5*simdat$l + rnorm(n, -10, 5)
simdat[1:5,]

# Estimate ipw weights.
temp <- ipwpoint(
  exposure = a,
  family = "binomial",
  link = "logit",
  numerator = ~ 1,
  denominator = ~ l,
  data = simdat)
summary(temp$ipw.weights)

# Plot inverse probability weights
# ipwplot(weights = temp$ipw.weights, logscale = FALSE,
#         main = "Stabilized weights", xlim = c(0, 8))

#Examine numerator and denominator models.
summary(temp$num.mod)
summary(temp$den.mod)

#Paste inverse probability weights
simdat$sw <- temp$ipw.weights

#Marginal structural model for the causal effect of a on y
#corrected for confounding by l using inverse probability weighting
#with robust standard error from the survey package.
if (requireNamespace("survey", quietly = TRUE)) {
  library(survey)
  msm <- svyglm(y ~ a,
                design = svydesign(~1, weights = ~temp$ipw.weights,
                data = simdat))
  summary(msm)
}
## Not run:
# Compute basic bootstrap confidence interval
# require(boot)
# boot.fun <- function(dat, index){
#   coef(glm(
#       formula = y ~ a,
#       data = dat[index,],
#       weights = ipwpoint(
#           exposure = a,
#           family = "gaussian",
```

```
#            numerator = ~ 1,
#            denominator = ~ l,
#            data = dat[index,])$ipw.weights))[2]
#   }
# bootres <- boot(simdat, boot.fun, 499);bootres
# boot.ci(bootres, type = "basic")

## End(Not run)
```

---

ipwtm   *Estimate Inverse Probability Weights (Time Varying)*

Description

Estimate inverse probability weights to fit marginal structural models, with a time-varying exposure
and time-varying confounders. Within each unit under observation this function computes inverse
probability weights at each time point during follow-up. The exposure can be binomial, multino-
mial, ordinal or continuous. Both stabilized and unstabilized weights can be estimated.

Usage

```
ipwtm(
  exposure,
  family,
  link,
  numerator = NULL,
  denominator,
  id,
  tstart,
  timevar,
  type,
  data,
  corstr = "ar1",
  trunc = NULL,
  ...
)
```

Arguments

exposure   vector, representing the exposure of interest. Both numerical and categorical
                variables can be used. A binomial exposure variable should be coded using
                values 0/1.

family   specifies a family of link functions, used to model the relationship between the
                variables in numerator or denominator and exposure, respectively. Alterna-
                tives are "binomial", "survival", "multinomial", "ordinal" and "gaussian".
                A specific link function is then chosen using the argument link, as explained
                below. Regression models are fitted using stats::glm, survival::coxph,
                nnet::multinom, MASS::polr or geepack::geeglm, respectively.

link	specifies the specific link function between the variables in numerator or denominator and exposure, respectively. For family="binomial" (fitted using stats::glm) alternatives are "logit", "probit", "cauchit", "log" and "cloglog". For family="survival" this argument is ignored, and Cox proportional hazards models are always used (fitted using survival::coxph). For family="multinomial" this argument is ignored, and multinomial logistic regression models are always used (fitted using nnet::multinom). For family="ordinal" (fitted using MASS::polr) alternatives are "logit", "probit", "cauchit", and "cloglog". For family="gaussian" this argument is ignored, and GEE models with an identity link are always used (fitted using geepack::geeglm).
numerator	is a formula, specifying the right-hand side of the model used to estimate the elements in the numerator of the inverse probability weights. When left unspecified, unstabilized weights with a numerator of 1 are estimated.
denominator	is a formula, specifying the right-hand side of the model used to estimate the elements in the denominator of the inverse probability weights.
id	vector, uniquely identifying the units under observation (typically patients) within which the longitudinal measurements are taken.
tstart	numerical vector, representing the starting time of follow-up intervals, using the counting process notation. This argument is only needed when family="survival", otherwise it is ignored. The Cox proportional hazards models are fitted using counting process data. Since a switch in exposure level can occur at the start of follow-up, tstart should be negative for the first interval (with timevar=0) within each patient.
timevar	numerical vector, representing follow-up time, starting at 0. This variable is used as the end time of follow-up intervals, using the counting process notation, when family="survival".
type	specifies the type of exposure. Alternatives are "first", "cens" and "all". With type="first", weights are estimated up to the first switch from the lowest exposure value (typically 0 or the first factor level) to any other value. After this switch, weights will then be constant. Such a weight is e.g. used when estimating the effect of "initiation of HAART" on mortality (see example 1 below). type="first" is currently only implemented for "binomial", "survival", "multinomial" and "ordinal" families. With type="cens" inverse probability of censoring weights (IPCW) are estimated as defined in appendix 1 in Cole & Hernán (2008). IPCW is illustrated in example 1 below. type="cens" is currently only implemented for "binomial" and "survival" families. With type="all", all time points are used to estimate weights. type="all" is implemented only for the "binomial" and "gaussian" family.
data	dataframe containing exposure, variables in numerator and denominator, id, tstart and timevar.
corstr	correlation structure, only needed when using family = "gaussian". Defaults to "ar1". See geepack::geeglm for details.
trunc	optional truncation percentile (0-0.5). E.g. when trunc = 0.01, the left tail is truncated to the 1st percentile, and the right tail is truncated to the 99th percentile. When specified, both un-truncated and truncated weights are returned.
...	are further arguments passed to the function that is used to estimate the numerator and denominator models (the function is chosen using family).

Details

Within each unit under observation i (usually patients), this function computes inverse probability weights at each time point j during follow-up. These weights are the cumulative product over all previous time points up to j of the ratio of two probabilities:

- the numerator contains at each time point the probability of the observed exposure level given observed values of stabilization factors and the observed exposure history up to the time point before j. These probabilities are estimated using the model regressing `exposure` on the terms in `numerator`, using the link function indicated by `family` and `link`.
- the denominator contains at each time point the probability of the observed exposure level given the observed history of time varying confounders up to j, as well as the stabilization factors in the numerator and the observed exposure history up to the time point before j. These probabilities are estimated using the model regressing `exposure` on the terms in `denominator`, using the link function indicated by `family` and `link`.

When the models from which the elements in the numerator and denominator are predicted are correctly specified, and there is no unmeasured confounding, weighting observations ij by the inverse probability weights adjusts for confounding of the effect of the exposure of interest. On the weighted dataset a marginal structural model can then be fitted, quantifying the causal effect of the exposure on the outcome of interest.

With `numerator` specified, stabilized weights are computed, otherwise unstabilized weights with a numerator of 1 are computed. With a continuous exposure, using `family = "gaussian"`, weights are computed using the ratio of predicted densities at each time point. Therefore, for `family = "gaussian"` only stabilized weights can be used, since unstabilized weights would have infinity variance.

Value

A list containing the following elements:

`ipw.weights`	vector containing inverse probability weights for each observation. Returned in the same order as the observations in `data`, to facilitate merging.
`weights.trunc`	vector containing truncated inverse probability weights, only returned when `trunc` is specified.
`call`	the original function call.
`selvar`	selection variable. With `type = "first"`, `selvar = 1` within each unit under observation, up to and including the first time point at which a switch from the lowest value of `exposure` to any other value is made, and `selvar = 0` after the first switch. For `type = "all"`, `selvar = 1` for all measurements. The numerator and denominator models are fitted only on observations with `selvar = 1`. Returned in the same order as observations in `data`, to facilitate merging.
`num.mod`	the numerator model, only returned when `numerator` is specified.
`den.mod`	the denominator model.

Missing values

Currently, the `exposure` variable and the variables used in `numerator` and `denominator`, `id`, `tstart` and `timevar` should not contain missing values.

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, Ronald B. Geskus <rgeskus@oucru.org>

References

Cole, S.R. & Hernán, M.A. (2008). Constructing inverse probability weights for marginal structural models. *American Journal of Epidemiology*, **168**(6), 656-664. https://pubmed.ncbi.nlm.nih.gov:443/18682488/.

Robins, J.M., Hernán, M.A. & Brumback, B.A. (2000). Marginal structural models and causal inference in epidemiology. *Epidemiology*, **11**, 550-560. https://pubmed.ncbi.nlm.nih.gov/10955408/.

Van der Wal W.M. & Geskus R.B. (2011). ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, **43**(13), 1-23. doi:10.18637/jss.v043.i13

See Also

basdat, haartdat, ipwplot, ipwpoint, ipwtm, timedat, tstartfun.

Examples

```
##########################################################################
#EXAMPLE 1

#Load longitudinal data from HIV positive individuals.
data(haartdat)

#CD4 is confounder for the effect of initiation of HAART therapy on mortality.
#Estimate inverse probability weights to correct for confounding.
#Exposure allocation model is Cox proportional hazards model.
temp <- ipwtm(
  exposure = haartind,
  family = "survival",
  numerator = ~ sex + age,
  denominator = ~ sex + age + cd4.sqrt,
  id = patient,
  tstart = tstart,
  timevar = fuptime,
  type = "first",
  data = haartdat)

#plot inverse probability weights
graphics.off()
ipwplot(weights = temp$ipw.weights, timevar = haartdat$fuptime,
      binwidth = 100, ylim = c(-1.5, 1.5), main = "Stabilized inverse probability weights")

#CD4 count has an effect both on dropout and mortality, which causes informative censoring.
#Use inverse probability of censoring weighting to correct for effect of CD4 on dropout.
#Use Cox proportional hazards model for dropout.
temp2 <- ipwtm(
  exposure = dropout,
```

```
    family = "survival",
    numerator = ~ sex + age,
    denominator = ~ sex + age + cd4.sqrt,
    id = patient,
    tstart = tstart,
    timevar = fuptime,
    type = "cens",
    data = haartdat)

#plot inverse probability of censoring weights
graphics.off()
ipwplot(weights = temp2$ipw.weights, timevar = haartdat$fuptime,
        binwidth = 100, ylim = c(-1.5, 1.5),
        main = "Stabilized inverse probability of censoring weights")

#MSM for the causal effect of initiation of HAART on mortality.
#Corrected both for confounding and informative censoring.
#With robust standard error obtained using cluster().
require(survival)
summary(coxph(Surv(tstart, fuptime, event) ~ haartind + cluster(patient),
              data = haartdat, weights = temp$ipw.weights*temp2$ipw.weights))

#uncorrected model
summary(coxph(Surv(tstart, fuptime, event) ~ haartind, data = haartdat))

##########################################################################
#EXAMPLE 2

data(basdat)
data(timedat)

#Aim: to model the causal effect of active tuberculosis (TB) on mortality.
#Longitudinal CD4 is a confounder as well as intermediate for the effect of TB.

#process original measurements
#check for ties (not allowed)
table(duplicated(timedat[,c("id", "fuptime")]))
#take square root of CD4 because of skewness
timedat$cd4.sqrt <- sqrt(timedat$cd4count)
#add TB time to dataframe
timedat <- merge(timedat, basdat[,c("id", "Ttb")], by = "id", all.x = TRUE)
#compute TB status
timedat$tb.lag <- ifelse(with(timedat, !is.na(Ttb) & fuptime > Ttb), 1, 0)
#longitudinal CD4-model
require(nlme)
cd4.lme <- lme(cd4.sqrt ~ fuptime + tb.lag, random = ~ fuptime | id,
               data = timedat)

#build new dataset:
#rows corresponding to TB-status switches, and individual end times
times <- sort(unique(c(basdat$Ttb, basdat$Tend)))
startstop <- data.frame(
  id = rep(basdat$id, each = length(times)),
```

```
      fuptime = rep(times, nrow(basdat)))
#add baseline data to dataframe
startstop <- merge(startstop, basdat, by = "id", all.x = TRUE)
#limit individual follow-up using Tend
startstop <- startstop[with(startstop, fuptime <= Tend),]
startstop$tstart <- tstartfun(id, fuptime, startstop) #compute tstart (?tstartfun)
#indicate TB status
startstop$tb <- ifelse(with(startstop, !is.na(Ttb) & fuptime >= Ttb), 1, 0)
#indicate TB status at previous time point
startstop$tb.lag <- ifelse(with(startstop, !is.na(Ttb) & fuptime > Ttb), 1, 0)
#indicate death
startstop$event <- ifelse(with(startstop, !is.na(Tdeath) & fuptime >= Tdeath),
                          1, 0)
#impute CD4, based on TB status at previous time point.
startstop$cd4.sqrt <- predict(cd4.lme,
                              newdata = data.frame(id = startstop$id,
                                                   fuptime = startstop$fuptime,
                                                   tb.lag = startstop$tb.lag))
#compute inverse probability weights
temp <- ipwtm(
  exposure = tb,
  family = "survival",
  numerator = ~ 1,
  denominator = ~ cd4.sqrt,
  id = id,
  tstart = tstart,
  timevar = fuptime,
  type = "first",
  data = startstop)
summary(temp$ipw.weights)
ipwplot(weights = temp$ipw.weights, timevar = startstop$fuptime, binwidth = 100)

#models
#IPW-fitted MSM, using cluster() to obtain robust standard error estimate
require(survival)
summary(coxph(Surv(tstart, fuptime, event) ~ tb + cluster(id),
              data = startstop, weights = temp$ipw.weights))
#unadjusted
summary(coxph(Surv(tstart, fuptime, event) ~ tb, data = startstop))
#adjusted using conditioning: part of the effect of TB is adjusted away
summary(coxph(Surv(tstart, fuptime, event) ~ tb + cd4.sqrt, data = startstop))

## Not run:
#compute bootstrap CI for TB parameter (takes a few hours)
#taking into account the uncertainty introduced by modelling longitudinal CD4
#taking into account the uncertainty introduced by estimating the inverse probability weights
#robust with regard to weights unequal to 1
#  require(boot)
#  boot.fun <- function(data, index, data.tm){
#     data.samp <- data[index,]
#     data.samp$id.samp <- 1:nrow(data.samp)
#     data.tm.samp <- do.call("rbind", lapply(data.samp$id.samp, function(id.samp) {
#       cbind(data.tm[data.tm$id == data.samp$id[data.samp$id.samp == id.samp],],
```

```
#           id.samp = id.samp)
#        }
#      ))
#   cd4.lme <- lme(cd4.sqrt ~ fuptime + tb.lag, random = ~ fuptime | id.samp, data = data.tm.samp)
#     times <- sort(unique(c(data.samp$Ttb, data.samp$Tend)))
#    startstop.samp <- data.frame(id.samp = rep(data.samp$id.samp, each = length(times)),
#                                 fuptime = rep(times, nrow(data.samp)))
#     startstop.samp <- merge(startstop.samp, data.samp, by = "id.samp", all.x = TRUE)
#     startstop.samp <- startstop.samp[with(startstop.samp, fuptime <= Tend),]
#     startstop.samp$tstart <- tstartfun(id.samp, fuptime, startstop.samp)
#    startstop.samp$tb <- ifelse(with(startstop.samp, !is.na(Ttb) & fuptime >= Ttb), 1, 0)
#   startstop.samp$tb.lag <- ifelse(with(startstop.samp, !is.na(Ttb) & fuptime > Ttb), 1, 0)
#   startstop.samp$event <- ifelse(with(startstop.samp, !is.na(Tdeath) & fuptime >= Tdeath), 1, 0)
#     startstop.samp$cd4.sqrt <- predict(cd4.lme, newdata = data.frame(id.samp =
#    startstop.samp$id.samp, fuptime = startstop.samp$fuptime, tb.lag = startstop.samp$tb.lag))
#
#     return(coef(coxph(Surv(tstart, fuptime, event) ~ tb, data = startstop.samp,
#       weights = ipwtm(
#            exposure = tb,
#            family = "survival",
#            numerator = ~ 1,
#            denominator = ~ cd4.sqrt,
#            id = id.samp,
#            tstart = tstart,
#            timevar = fuptime,
#            type = "first",
#            data = startstop.samp)$ipw.weights))[1])
#    }
# bootres <- boot(data = basdat, statistic = boot.fun, R = 999, data.tm = timedat)
# bootres
# boot.ci(bootres, type = "basic")
#
## End(Not run)
```

---

timedat   *HIV: TB and Survival (Longitudinal Measurements)*

Description

A simulated dataset containing time-varying CD4 measurements for 386 HIV-positive individuals. Corresponding baseline data, including timing of tuberculosis and death, are available in basdat.

Usage

```
data(timedat)
```

Format

A data frame with 6291 observations on the following 3 variables:

id Patient ID.

fuptime Follow-up time (days since HIV seroconversion).

cd4count CD4 count measured at `fuptime`.

Details

These simulated data are used together with [basdat](#) in a detailed causal modeling example using inverse probability weighting (IPW). See [ipwtm](#) for the full example. Data were simulated using the algorithm described in Van der Wal et al. (2009).

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, Ronald B. Geskus <rgeskus@oucru.org>

References

Cole, S.R. & Hernán, M.A. (2008). Constructing inverse probability weights for marginal structural models. *American Journal of Epidemiology*, **168**(6), 656-664.

Robins, J.M., Hernán, M.A. & Brumback, B.A. (2000). Marginal structural models and causal inference in epidemiology. *Epidemiology*, **11**, 550-560.

Van der Wal W.M. & Geskus R.B. (2011). ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, **43**(13), 1-23. doi:10.18637/jss.v043.i13

Van der Wal W.M., Prins M., Lumbreras B. & Geskus R.B. (2009). A simple G-computation algorithm to quantify the causal effect of a secondary illness on the progression of a chronic disease. *Statistics in Medicine*, **28**(18), 2325-2337.

See Also

[basdat](#), [haartdat](#), [ipwplot](#), [ipwpoint](#), [ipwtm](#), [tstartfun](#)

Examples

```
# For an example of how to use these longitudinal measurements with basdat, see:
# ?ipwtm
```

---

tstartfun   *Compute Starting Time For Counting Process Notation*

Description

Function to compute starting time for intervals of follow-up, when using the counting process notation. Within each unit under observation (usually individuals), computes starting time equal to:

- time of previous record when there is a previous record.
- -1 for first record.

Usage

```
tstartfun(id, timevar, data)
```

Arguments

id	numerical vector, uniquely identifying the units under observation, within which the longitudinal measurements are taken.
timevar	numerical vector, representing follow-up time, starting at 0.
data	dataframe containing id and timevar

Value

Numerical vector containing starting time for each record. In the same order as the records in data, to facilitate merging.

Missing values

Currently, id and timevar should not contain missing values.

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, Ronald B. Geskus <rgeskus@oucru.org>

References

Van der Wal W.M. & Geskus R.B. (2011). ipw: An R Package for Inverse Probability Weighting. *Journal of Statistical Software*, **43**(13), 1-23. doi:10.18637/jss.v043.i13.

See Also

basdat, haartdat, ipwplot, ipwpoint, ipwtm, timedat, tstartfun.

Examples

```
#data
mydata1 <- data.frame(
  patient = c(1, 1, 1, 1, 1, 1, 2, 2, 2, 2),
  time.days = c(14, 34, 41, 56, 72, 98, 0, 11, 28, 35))

#compute starting time for each interval
mydata1$tstart <- tstartfun(patient, time.days, mydata1)

#result
mydata1

#see also ?ipwtm for example
```

Index