Journal of Statistical Software

September 2011, Volume 43, Issue 13. *http://www.jstatsoft.org/*

ipw: An R Package for Inverse Probability Weighting

Willem M. van der Wal
University Medical Center Utrecht

Ronald B. Geskus
University of Amsterdam

Abstract

We describe the R package **ipw** for estimating inverse probability weights. We show how to use the package to fit marginal structural models through inverse probability weighting, to estimate causal effects. Our package can be used with data from a point treatment situation as well as with a time-varying exposure and time-varying confounders. It can be used with binomial, categorical, ordinal and continuous exposure variables.

Keywords: inverse probability weighting, marginal structural models, causal inference, R.

1. Introduction

We describe the R (R Development Core Team 2011) package **ipw**, for estimating inverse probability weights. These weights are typically used to perform inverse probability weighting (IPW) to fit a marginal structural model (MSM). The package is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=ipw` . MSMs are used to estimate causal effects from observational data, by correcting for confounding. When using IPW to fit an MSM, there is minimal risk of (1) adjusting away part of the effect (Robins 1997; Robins, Greenland, and Hu 1999), (2) non-collapsibility, (Greenland, Robins, and Pearl 1999), or (3) Berksons bias (Hernán, Hernández-Díaz, and Robins 2004). In contrast, when using conditioning to correct for confounding (1) and (3) can occur in a longitudinal study, and (2) can occur with any statisticical model that does not have a linear or log-linear link function.

The use of IPW to fit an MSM was described in detail, e.g., in Robins, Hernán, and Brumback (2000), Hernán and Robins (2006) and Cole and Hernán (2008). Currently available software to fit MSMs includes **CausalGAM**, an R package for the estimation of causal effects with generalized additive models in a point treatment with a binary exposure (Glynn and Quinn

2010), **cvDSA**, an R package for MSM-based causal inference with point treatment data using data-adaptive estimation with cross-validation and the deletion/substitution/addition (D/S/A) algorithm (Wang, Hartman, and Gruber 2009), **tmleLite**, an R package for targeted maximum likelihood estimation of marginal additive treatment effect of a binary point treatment (Gruber and van der Laan 2010; Van der Laan 2010) and the SAS macro for doubly robust estimation by Jonsson Funk, Westreich, Davidian, and Weisen (2007). Also, Hernán, Brumback, and Robins (2000) described how to program IPW in SAS, and Fewell, Hernán, Wolfe, Tilling, Choi, and Sterne (2004) described how to program IPW in Stata.

This paper is structured as follows. In Section 2 we give a general introduction to IPW. We describe the functions contained in our package **ipw** (version 1.0-10) in Section 3. We demonstrate the use of the package **ipw** in a number of different situations, using simulated example data, in Section 4.

2. Inverse probability weighting

As was shown by Robins (1998), the parameters of MSMs can be estimated using inverse probability weighting (IPW) to correct both for confounding (illustrated in the examples below) and for forms of selection bias such as informative censoring (illustrated in the example in Section 4.2). This amounts to the fitting of a model regressing the outcome of interest on the exposure of interest using observational data, with each observation weighted by the inverse of the probability of the observed exposure level given the observed value of the confounders.

2.1. IPW in a point treatment

In a point treatment situation we can adjust for a set of confounders $\boldsymbol{C}$ when estimating the effect of discrete exposure A by weighting observations i by the inverse probability weights

$$w_i = \frac{1}{P(A_i = a_i | \boldsymbol{C}_i = \boldsymbol{c}_i)}. \tag{1}$$

We indicate the observed exposure and confounder status with a and $\boldsymbol{c}$, respectively. The denominator of (1) contains the probability of the observed exposure level given the observed values of covariates $\boldsymbol{C}$. When $\boldsymbol{C}$ includes all relevant confounders, and we estimate $P(A_i = a_i | \boldsymbol{C}_i = \boldsymbol{c}_i)$ using a correctly specified exposure allocation model, weighting by w_i creates a pseudopopulation in which $\boldsymbol{C}$ no longer predicts A and in which the causal association between A and the outcome of interest is the same as in the original study population[1]. Weighting observations i by w_i, one can fit a causal model, for instance the MSM

$$E(Y_a) = \beta_0 + \beta_1 a, \tag{2}$$

with a continuous outcome Y. The response variable Y_a is the potential outcome that could have been observed in a unit under study, when that unit would have received, perhaps contrary to the fact, a specific treatment level a (Robins *et al.* 2000). The expectation $E(Y_a)$

[1]Note that with unsaturated exposure allocation models, IPW estimators are less efficient than likelihood-based estimators (Clayton, Spiegelhalter, Dunn, and Pickles 1998), and may be unstable when certain strata defined by $\boldsymbol{C}$ have low response probabilities (Little and Rubin 1987; Cole and Hernán 2008; Lefebvre, Delaney, and Platt 2008).

is the mean response, when all units under study would have received a specific treatment level a. Parameter β_1 then quantifies the causal effect of A on Y.

To increase statistical efficiency and attain better coverage of confidence intervals, it is recommended to use stabilized weights (Hernán *et al.* 2000; Cole and Hernán 2008), e.g.,

$$sw_i = \frac{P(A_i = a_i)}{P(A_i = a_i | \boldsymbol{C}_i = \boldsymbol{c}_i)}. \tag{3}$$

The numerator of (3) contains the probability of the observed exposure level, which is just the observed frequency. This introduces an association between the numerator and denominator, which means that on average the difference between the numerator and denominator becomes smaller, as compared to unstabilized weights. This means in turn that stabilized weights will have a narrower distribution than unstabilized weights. To increase the association between the numerator and denominator, further stabilizing the weights, one can condition both in the numerator and denominator of (3) on a set of time-fixed covariates $\boldsymbol{V}$ that are related to A. For instance, when a researcher believes that sex does not influence the outcome of interest, but the distribution of exposure level varies between both sexes, sex could be included in $\boldsymbol{V}$. It must be noted that confounding caused by baseline covariates that are used as stabilization factors, is not adjusted for (Cole and Hernán 2008). Adjustment for such covariates could be made by including them in the MSM, at the cost of possibly inducing non-collapsibility. Also, note that stabilization can be done not only by including a linear term of V, but that more complex functions can also be used, when that would improve the estimation of $P(A = a)$.

With a continuous exposure variable A, one can use the stabilized weights

$$sw_i = \frac{f(a_i)}{f(a_i | \boldsymbol{c}_i)}, \tag{4}$$

where $f(a_i)$ is the marginal density function of A, evaluated at the observed value in unit i, a_i, and $f(a_i|\boldsymbol{c}_i)$ is the conditional density function of A given $\boldsymbol{C}$, evaluated at the observed values in unit i, $\{a_i, \boldsymbol{c}_i\}$. With a continuous exposure variable, unstabilized weights cannot be used, since they would have infinite variance (Robins *et al.* 2000).

The denominators of (1), (3) and (4) can be estimated by using exposure allocation models regressing A on $\boldsymbol{C}$. Similarly, the numerators of (3) and (4) can be estimated by using exposure allocation models regressing A on the constant only. When using additional stabilization variables V, those variables V can be included in the exposure allocation models as well.

2.2. IPW in a longitudinal study

Suppose that a discrete exposure A may change over time, and a decision to allocate a certain exposure level is made and recorded within each unit i at time points t_{ij}. Time-varying confounders for the effect of A_{ij} on the outcome of interest, measured right before each time point t_{ij} in each unit i, are contained in $\boldsymbol{C}_{ij}$. In addition, $\boldsymbol{C}_{ij}$ can also contain time-fixed confounders. Let $\overline{A}_{ij}$ and $\overline{\boldsymbol{C}}_{ij}$ indicate the observed longitudinal history, i.e., all measurements up to time point t_{ij} within unit i, of A and $\boldsymbol{C}$ respectively. $\boldsymbol{V}_i$ are measured time-fixed covariates, that are not confounders but that are associated with the exposure. One can adjust for time-varying confounders $\boldsymbol{C}$ by weighting observations at t_{ij} by the stabilized weights

$$sw_{ij} = \prod_{k=0}^{j} \frac{P(A_{ik} = a_{ik} | \overline{A}_{ik-1} = \overline{a}_{ik-1}, \boldsymbol{V}_i = \boldsymbol{v}_i)}{P(A_{ik} = a_{ik} | \overline{A}_{ik-1} = \overline{a}_{ik-1}, \overline{\boldsymbol{C}}_{ik} = \overline{\boldsymbol{c}}_{ik}, \boldsymbol{V}_i = \boldsymbol{v}_i)}. \tag{5}$$

Equation (5) is a product over all time points from baseline up to time point t_{ij}, within each unit i. The factors in the numerator of (5) contain the probability of the observed exposure status at each time point, a_{ik}, given the observed exposure history up to the previous time point, $\overline{a}_{ik-1}$, and the observed time-fixed covariates, $\boldsymbol{v}_i$. The factors in the denominator of (5) contain the probability of the observed exposure status at each time point, given the observed exposure history up to the previous time point, the observed history of time-varying confounders up to each time point, $\overline{\boldsymbol{c}}_{ik}$, and the observed time-fixed covariates. Note that time-fixed covariates $\boldsymbol{V}$ are included both in the numerator and denominator of (5), to further stabilize the weights. To estimate the causal effect of A on the exposure of interest, one can fit an MSM to the observations made at time points t_{ij}, weighted by sw_{ij}, as was done e.g., by Hernán *et al.* (2000).

With a continuous exposure A, one can use the stabilized weights

$$sw_{ij} = \prod_{k=0}^{j} \frac{f(a_{ik}|\overline{a}_{ik-1}, \boldsymbol{v}_i)}{f(a_{ik}|\overline{a}_{ik-1}, \overline{\boldsymbol{c}}_{ik}, \boldsymbol{v}_i)}, \tag{6}$$

analoguously to (4), as described in Cole and Hernán (2008). The numerator $f(a_{ik}|\overline{a}_{ik-1}, \boldsymbol{v}_i)$ is the conditional density function of A at time point t_{ik} given the history of A up to the previous time point and the time-fixed covariates, evaluated at the observed values in unit i at time point t_{ik}, $\{a_{ik}, \overline{a}_{ik-1}, \boldsymbol{v}_i\}$. The denominator $f(a_{ik}|\overline{a}_{ik-1}, \overline{\boldsymbol{c}}_{ik}, \boldsymbol{v}_i)$ is the conditional density function of A at time point t_{ik} given the history of A up to the previous time point, the history of time-varying confounders $\boldsymbol{C}$ up to time point t_{ik}, and the time-fixed covariates, evaluated at the observed values in unit i at time point t_{ik}, $\{a_{ik}, \overline{a}_{ik-1}, \overline{\boldsymbol{c}}_{ik}, \boldsymbol{v}_i\}$.

The elements in the denominator of (5) and (6) can be estimated by using exposure allocation models regressing time-varying exposure A_{ij} on follow-up time t_{ij}, the history of A up to but not including t_{ij}, $\overline{A}_{i(j-1)}$, the observed history of confounders $\boldsymbol{C}_{ij}$ up to and including t_{ij}, $\overline{\boldsymbol{C}}_{ij}$, and the time-fixed covariates $\boldsymbol{V}_i$. The elements in the numerator of (5) and (6) can be estimated from similar models, not including $\overline{\boldsymbol{C}}$.

Note that when the effects of $\overline{A}_{i(j-1)}$ and $\overline{\boldsymbol{C}}_{ij}$ on A_{ij} are fully expressed through $A_{i(j-1)}$ and $\boldsymbol{C}_{ij}$, only the latter need to be included in the exposure allocation models. Often, A_{ij} will be constant after a certain switch is made, e.g., after a switch from exposure level 0 to exposure level 1 subjects will always remain on exposure level 1. When A_{ij} is deterministically constant after such a switch, the elements in the numerator and denominator of (5) and (6) can be set to 1 after the switch. Time to event models can then be used as exposure allocation models.

Note that a continuously varying exposure $A(t)$ such as disease status can change at any time, not just at certain time points t_{ij}. With such a continuously time-varying exposure $A(t)$, it is necessary to choose the time points that are used to fit an MSM. A logical choice is either to (1) use time points at which changes in exposure status are observed, and time points at which the outcome is observed (e.g., the event times with a survival outcome) or (2) use regularly spaced intervals, with a sufficiently fine discretization. In both cases, the value of time-varying confounders right before each time point may need to be imputed from a longitudinal model that was fitted on the original measurements (e.g., see Section 4.3).

2.3. Inference

When using IPW, observations can have weights unequal to each other, which introduces clustering in the weighted dataset. When this is not taken into account, the standard error

of the causal effect estimate could be underestimated. Therefore, when using IPW to fit an MSM, it is necessary to use a robust standard error estimator for inference (Hernán *et al.* 2000).

3. The R package ipw

The R package **ipw** comes with a namespace. It contains the following functions:

- `ipwpoint`, for estimating inverse probability weights in a point treatment situation.
- `ipwtm`, for estimating inverse probability weights for a time-varying exposure with time-varying confounders.
- `ipwplot`, to plot the distribution of inverse probability weights.
- `tstartfun`, to compute the starting time for intervals of follow-up, when Cox proportional hazards models are used to model the exposure allocation.

We describe these functions below. Package **ipw** also contains the simulated datasets `haartdat`, `basdat` and `timedat`, which are described and analyzed in the examples given in Section 4.

3.1. Function `ipwpoint`

The function `ipwpoint` can be used to estimate inverse probability weights similar to (1), (3) and (4), to fit MSMs in a point treatment situation. The exposure of interest can be binomial, multinomial, ordinal or continuous. Both stabilized and unstabilized weights can be estimated. It is used as:

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

and takes the following arguments:

- `exposure` is 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` and `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 the R functions `glm` (**stats**, see R Development Core Team 2011), `multinom` (**nnet**, see Venables and Ripley 2002), `polr` (**MASS**, see Venables and Ripley 2002) 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` is an optional truncation fraction for the weights (between `0` and `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`).

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 infinite variance (Robins *et al.* 2000). The output returned by `ipwpoint` is 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.

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

3.2. Function `ipwtm`

The function `ipwtm` can be used to estimate inverse probability weights to fit MSMs, with a time-varying exposure and time-varying confounders. Within each unit under observation i

(e.g., patients), this function computes inverse probability weights at each time point t_{ij} during follow-up, similar to (5) and (6). The exposure can be binomial, multinomial, ordinal or continuous. Both stabilized and unstabilized weights can be estimated. It is used as:

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

and takes the following arguments:

- `exposure` is a vector, representing the exposure of interest. As in `ipwpoint`, both numerical and categorical variables can be used. A binomial exposure variable should be coded using values `0` and `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"`, `"survival"`, `"multinomial"`, `"ordinal"` and `"gaussian"`. A specific link function is then chosen using the argument `link`, as explained below. Regression models are fitted using `glm` (**stats**), `coxph` (**survival**, see Therneau and Lumley 2011), `multinom` (**nnet**), `polr` (**MASS**) or `geeglm` (**geepack**, see Halekoh, Højsgaard, and Yan 2005), respectively.

- `link` is the specific 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 = "survival"` this argument is ignored, and Cox proportional hazards models are always used (fitted using `coxph`). 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 GEE models with an identity link are always used (fitted using `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` is a vector, uniquely identifying the units under observation within which the longitudinal measurements are taken.

- `tstart` is a 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 longitudinal data, coded using the counting process notation. When modeling exposure allocation at the start of follow-up (for `timevar = 0`), `tstart` should be negative. For this first interval, the particular value of `tstart` is not important, only that it is smaller than zero.

- `timevar` is a 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"` 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 causal effect of the initiation of highly active anti-retroviral therapy (HAART) on mortality (see Sections 4.2 and 4.3). With `type = "all"`, all time points are used to estimate weights. Currently, only `"first"` is implemented for `"survival"`, `"multinomial"` and `"ordinal"` families. Only `"all"` is implemented for the `"gaussian"` family. Both `type = "first"` and `type = "all"` are implemented for the `"binomial"` family.

- `data` is a dataframe containing `exposure`, the variables used in `numerator` and `denominator`, and variables `id`, `tstart` and `timevar`.

- `corstr` specifies a correlation structure, only needed when using `family = "gaussian"`. Defaults to `"ar1"`. For further details see Halekoh *et al.* (2005).

- `trunc` is an optional truncation fraction for the weights (between `0` and `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`).

With `numerator` specified, stabilized weights are computed, otherwise unstabilized weights with a numerator of 1 are computed. As in `ipwpoint`, 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 variance (Robins *et al.* 2000). The output returned by `ipwtm` is a list containing the following elements:

- `ipw.weights` is a vector containing inverse probability weights for each observation. This vector is returned in the same order as the observations contained in `data`, to facilitate merging.

- `infinityweights.trunc` is a vector containing truncated inverse probability weights for each observation. This vector is only returned when `trunc` is specified.

- `call` is the original function call to `ipwtm`.

- `selvar` is a 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 have been fitted only on observations with `selvar = 1`. This vector is returned in the same order as the observations in `data`, to facilitate merging.

- `num.mod` is the numerator model, only returned when `numerator` is specified.
- `den.mod` is the denominator model.

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

3.3. Function `ipwplot`

The function `ipwplot` can be used to plot inverse probability weights. For time-varying weights (with a time-varying exposure and time-varying confounders) boxplots are made within strata of follow-up time. For inverse probability weights in a point treatment situation, a density plot is displayed. The function is used as:

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

and takes the following arguments:

- `weights` is a numerical vector of inverse probability weights to plot.
- `timevar` is a 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` is a 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` is a logical value. If `TRUE`, weights are plotted on a logarithmic scale.
- `xlab` is the label for the horizontal axis.
- `ylab` is the label for the vertical axis.
- `main` is the main title for the plot.
- `ref` is a logical value. If `TRUE`, a reference line is plotted at $y = 1$.
- `...` are additional arguments passed to `boxplot` (when `timevar` is specified) or `plot` (when `timevar` is not specified).

See the examples below for actual plots (Figures 1, 2 and 3).

3.4. Function `tstartfun`

Function `tstartfun` can be used to compute the starting time for intervals of follow-up, when using the counting process notation. Within each unit under observation, this function computes a starting time equal to:

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

The function is used as:

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

and takes the following arguments:

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

4. Examples

In the following examples we will illustrate the use of functions `ipwpoint`, `ipwtm`, `ipwplot` and `tstartfun`, contained in **ipw**. We also describe the datasets `haartdat`, `basdat`, and `timedat`, contained in **ipw**, which are used in the examples. The following three examples are ordered by increasing complexity.

4.1. Point treatment example

We will illustrate the use of IPW in a point treatment using simulated data. First we will simulate point treatment data with measurements made in 1000 individuals on a continuous confounder L, a dichotomous exposure A and a continuous outcome Y, using:

- $L \sim \mathcal{N}(10, 5)$,
- $\text{logit}P(A = 1) = -10 + L$,
- $Y = 10A + 0.5L + \mathcal{N}(-10, 5)$.

The true parameter for the marginal causal effect of A on Y is 10. We will set the random number seed for reproducibility of this example. The data is simulated as follows:

```
R> set.seed(16)
R> n <- 1000
R> simdat <- data.frame(l = rnorm(n, 10, 5))
R> a.lin <- simdat$l - 10
R> pa <- exp(a.lin)/(1 + exp(a.lin))
R> simdat$a <- rbinom(n, 1, prob = pa)
R> simdat$y <- 10*simdat$a + 0.5*simdat$l + rnorm(n, -10, 5)
R> simdat[1:5,]

          l a          y
1 12.382067 1   6.635898
2  9.373100 0 -11.722042
3 15.481081 1  11.501612
4  2.778855 0  -9.464074
5 15.739146 1  15.085261
```

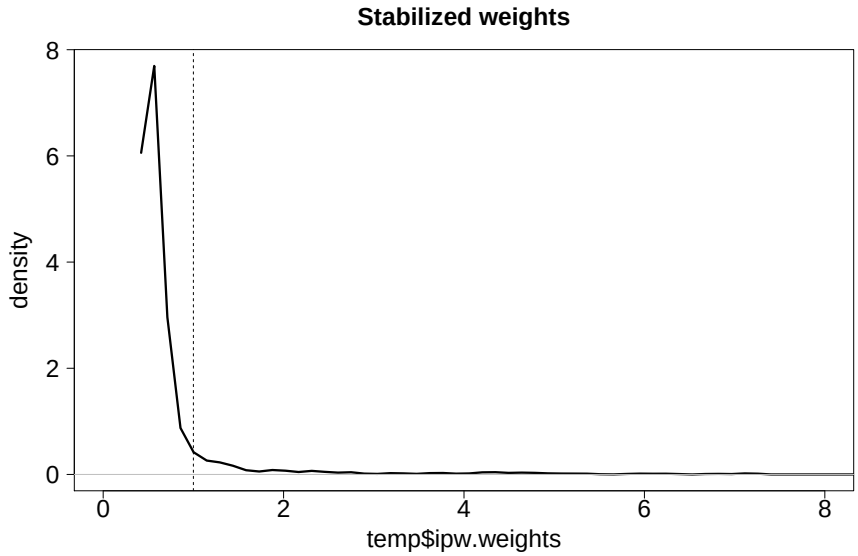

Figure 1: Weights distribution plot for example 1, made using `ipwplot`.

We can estimate inverse probability weights to correct for the confounding. We choose to estimate the stabilized weights

$$sw_i = \frac{P(A_i = a_i)}{P(A_i = a_i|L_i = l_i)}, \tag{7}$$

similar to (4). To estimate the denominator of (7), we use a logistic model regressing A on L. To estimate the numerator of (7), we use a logistic model regressing A on the constant only. Therefore, we estimate the inverse probability weights as follows:

```
R> library("ipw")
R> temp <- ipwpoint(exposure = a, family = "binomial", link = "logit",
+    numerator = ~ 1, denominator = ~ l, data = simdat)
R> summary(temp$ipw.weights)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4810  0.5127  0.5285  0.9095  0.6318 74.7000
```

We can plot the distribution of the weights as follows (see Figure 1):

```
R> ipwplot(weights = temp$ipw.weights, logscale = FALSE,
+    main = "Stabilized weights", xlim = c(0, 8))
```

We can also examine the numerator and denominator models:

```
R> summary(temp$num.mod)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.07604    0.06329   1.201     0.23
```

```
R> summary(temp$den.mod)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.69809    0.66066  -14.68   <2e-16 ***
l            0.97272    0.06525   14.91   <2e-16 ***
```

Pasting the IPW weights to the dataset can be done as:

```
R> simdat$sw <- temp$ipw.weights
```

Weighting the original observations by the stabilized weights (7) to adjust for confounding, we can fit the MSM, estimating the marginal causal effect of A on Y,

$$\mathrm{Y_a} = \beta_0 + \beta_1 \mathrm{a}, \tag{8}$$

which can be done as follows, using a robust standard error estimate from the **survey** package (Lumley 2004):

```
R> msm <- (svyglm(y ~ a, design = svydesign(~ 1, weights = ~ sw,
+    data = simdat)))
R> coef(msm)

(Intercept)           a
  -4.375478   10.646610

R> confint(msm)

                2.5 %    97.5 %
(Intercept) -6.613252 -2.137704
a            8.314527 12.978694
```

Our estimate of the marginal causal effect of A on Y is 10.65 with 95% confidence interval (CI) 8.31–12.98.

4.2. Causal effect of HAART use on mortality in HIV-infected patients

Dataset `haartdat` is a simulated dataset, with survival data measured in 1200 HIV-infected patients. Start of follow-up is HIV seroconversion. Each row corresponds to a 100 day period of follow-up time. Patients can initiate highly active anti-retroviral therapy (HAART) during follow-up. We will estimate the causal effect of HAART on mortality using this dataset, while adjusting both for possible confounding by CD4 count, and for informative censoring due to the effect of CD4 count on dropout, using IPW. In this example, CD4 count is a time-varying covariate. We load the package **ipw** and dataset `haartdat`, and look at the first 10 rows of `haartdat`:

```
R> library("ipw")
R> data("haartdat")
R> haartdat[1:10,]
```

```
   patient tstart fuptime haartind event sex age cd4.sqrt endtime dropout
1        1   -100       0        0     0   1  22 23.83275    2900       0
2        1      0     100        0     0   1  22 25.59297    2900       0
3        1    100     200        0     0   1  22 23.47339    2900       0
4        1    200     300        0     0   1  22 24.16609    2900       0
5        1    300     400        0     0   1  22 23.23790    2900       0
6        1    400     500        0     0   1  22 24.85961    2900       0
7        1    500     600        0     0   1  22 25.94224    2900       0
8        1    600     700        1     0   1  22 26.03843    2900       0
9        1    700     800        1     0   1  22 26.72078    2900       0
10       1    800     900        1     0   1  22 27.47726    2900       0
```

Dataset `haartdat` contains the following variables:

- `patient` is the patient ID,
- `tstart` is the starting time for each interval of follow-up, measured in days since HIV seroconversion; note that the first interval of follow-up is $(-100, 0]$, this is used to allow for the modeling of the initiation of HAART at $t = 0$ (as explained in Section 3.2),
- `fuptime` is the end time for each interval of follow-up measured in days since HIV seroconversion,
- `haartind` is an indicator for the initiation of HAART therapy at the end each interval (`0` = HAART not initiated/`1` = HAART initiated),
- `event` is an indicator for death at the end of the interval (`0` = alive/`1` = died),
- `sex` is sex (`0` = male/`1` = female),
- `age` is age at the start of follow-up (years),
- `cd4.sqrt` is the square root of CD4 count, measured at the end of each interval, but before `haartind`. Note that in each row, corresponding to time point j in individual i, `cd4.sqrt` has an effect on `haartind` in the same row, including at time 0.
- `dropout` is an indicator for dropout of the study, at the end of the interval (`0` = did not drop out/`1` = dropped out).

To adjust for confounding by time-varying CD4 count, we estimate the stabilized inverse probability weights

$$sw_{ij} = \prod_{k=0}^{j} \frac{P(H_{ik} = h_{ik} | \overline{H}_{ik-1} = \overline{h}_{ik-1}, \boldsymbol{V}_i = \boldsymbol{v}_i)}{P(H_{ik} = h_{ik} | \overline{H}_{ik-1} = \overline{h}_{ik-1}, \overline{L}_{ik} = \overline{l}_{ik}, \boldsymbol{V}_i = \boldsymbol{v}_i)}, \tag{9}$$

similar to (5), with H_{ij} and L_{ij} indicating HAART status and the square root of CD4 count in patient i at measurement j, respectively. $\boldsymbol{V}_i$ is the vector of time-fixed covariates, containing sex and age. h_{ij}, l_{ij} and $\boldsymbol{v}_i$ are the observed values of variables H_{ij}, L_{ij} and $\boldsymbol{V}_i$, respectively. For time points after the initiation of HAART within each patient, the elements in the numerator and denominator of (9) are set to 1. For time points up to the time point of

the initiation of HAART within each patient, we estimate the elements in the denominator of (9) using the Cox proportional hazards model

$$\lambda_H[t|L(t), \boldsymbol{V}, H(t^- = 0)] = \lambda_0(t) \exp\{\beta_1 L(t) + \boldsymbol{\beta}_2' \boldsymbol{V}\}, \tag{10}$$

with t follow-up time. We estimate the numerator of (9) using a model similar to (10) but without $L(t)$ as a predictor. We can estimate, and examine sw_{ij} using:

```
R> temp <- ipwtm(exposure = haartind, family = "survival",
+    numerator = ~ sex + age, denominator = ~ cd4.sqrt + sex + age,
+    id = patient, tstart = tstart, timevar = fuptime, type = "first",
+    data = haartdat)
R> summary(temp$ipw.weights)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2459  0.9036  0.9862  1.0390  1.0610  7.1260
```

For comparison, note that similar unstabilized weights can be estimated as:

```
R> temp.unstab <- ipwtm(exposure = haartind, family = "survival",
+    denominator = ~ cd4.sqrt, id = patient, tstart = tstart,
+    timevar = fuptime, type = "first", data = haartdat)
```

As an illustration, note that the unstabilized weights have a much wider distribution than the stabilized weights:

```
R> summary(temp.unstab$ipw.weights)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.002   1.161   1.372  13.140  15.420 401.800
```

We can plot the stabilized inverse probability weights (see Figure 2) using:

```
R> ipwplot(weights = temp$ipw.weights, timevar = haartdat$fuptime,
+    binwidth = 100, ylim = c(-1.5, 1.5), main = "Stabilized weights",
+    xaxt = "n", yaxt = "n")
R> axis(side = 1, at = c(0, 5, 10, 15, 20, 25, 30, 35),
+    labels = as.character(c(0, 5, 10, 15, 20, 25, 30, 35)*100))
R> axis(side = 2, at = c(-1.5, -1, -0.5, 0, 0.5, 1, 1.5),
+    labels = as.character(c(-1.5, -1, -0.5, 0, 0.5, 1, 1.5)))
```

Note that we plot the axes separately, allowing us to specify the positions and labels of the tick-marks.

In this example, CD4 count also has an effect on dropout from the study. Since CD4 count has an effect on mortality, this can cause informative censoring. Note that in this example, censoring for other reasons than dropout is regarded as non-informative. We can estimate inverse probability of censoring weights sw'_{ij}, to correct for the effect of CD4 count on dropout, similarly to (10), but replacing H_{ij} with an indicator for dropout, as:

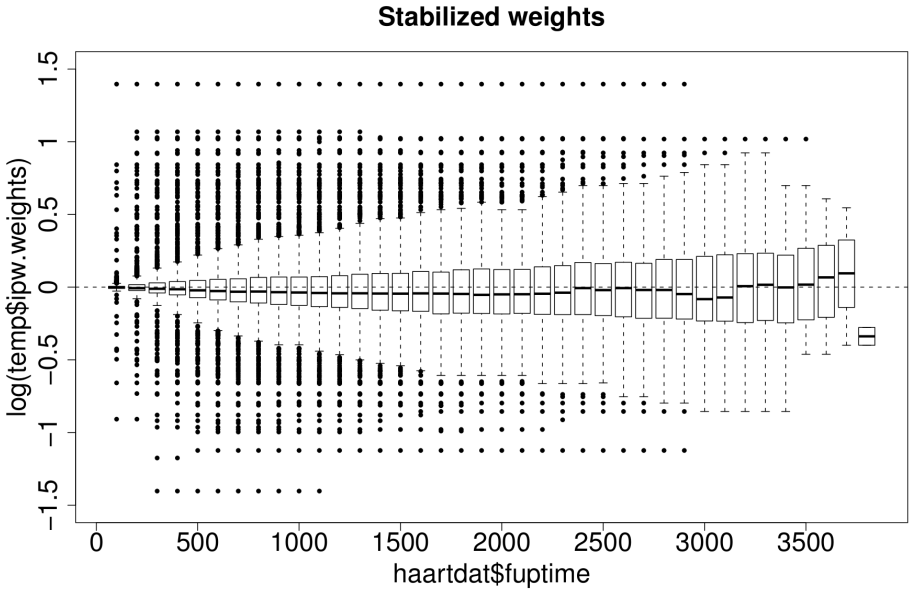

Figure 2: Weights distribution plot for the inverse probability weights that are used to adjust for confounding in example 2, made using `ipwplot`.

```
R> temp2 <- ipwtm(exposure = dropout, family = "survival",
+    numerator = ~ sex + age, denominator = ~ cd4.sqrt + sex + age,
+    id = patient, tstart = tstart, timevar = fuptime, type = "first",
+    data = haartdat)
```

Note that when the exposure also has an effect of dropout, as well as on mortality, it can be added to the model used to estimate the denominator of the weights.

We can now use the inverse probability weights sw_{ij} and inverse probability of censoring weights sw'_{ij} to fit an MSM, quantifying the causal effect of the initiation of HAART on mortality. To combine the adjustment for confounding and for informative censoring, the observations indexed by ij are weighted by the product $sw_{ij} \times sw'_{ij}$. Similarly to Hernán *et al.* (2000), we fit the MSM

$$\lambda_{T_{\bar{h}}}(t) = \lambda_0(t) \exp\{\beta_1 h(t)\}, \tag{11}$$

using a robust variance estimator (through `cluster()`), via:

```
R> summary(coxph(Surv(tstart, fuptime, event) ~ haartind + cluster(patient),
+    data = haartdat, weights = temp$ipw.weights*temp2$ipw.weights))

            coef exp(coef) se(coef) robust se      z Pr(>|z|)
haartind -0.9378    0.3915   0.4300    0.4524 -2.073   0.0382 *

         exp(coef) exp(-coef) lower .95 upper .95
haartind    0.3915      2.554    0.1613    0.9501
```

We estimate a hazard ratio corresponding to the marginal causal effect of HAART on mortality of 0.39 (95% CI 0.16–0.95).

4.3. Causal effect of tuberculosis on mortality in HIV-infected patients

Our third example is similar to example in Section 4.2 but with measurements made at irregular intervals of follow-up time. We estimate the causal effect of active tuberculosis (TB) on mortality in HIV-positive individuals, adjusted for possible confounding by time-varying CD4 count using IPW. We smooth time-varying CD4 using a random effects model, because it is the underlying "true" CD4, separate from short-term fluctuations and measurement error, that is a confounder for the effect of TB. The simulated datasets `basdat` and `timedat` are used in this example. We load package **ipw** and the datasets, and explore the datasets:

```
R> library("ipw")
R> data("basdat")
R> data("timedat")
R> basdat[1:4,]
```

```
  id  Ttb Tdeath Tend
1  1   NA   1846 1846
2  2   NA     NA 1126
3  3 3139   3333 3333
4  4   NA   2253 2253
```

Dataset `basdat` contains the following time-fixed variables, measured in 386 HIV-positive individuals:

- `id` is the patient ID,
- `Ttb` is the time of first active tuberculosis, measured in days since HIV seroconversion,
- `Tdeath` is the time of death, measured in days since HIV seroconversion,
- `Tend` is the individual end time (either death or censoring), measured in days since HIV seroconversion.

```
R> timedat[1:10,]
```

```
   id fuptime cd4count
1   1       4      475
2   1      71      555
3   1     200      456
4   1     280      443
5   1     298      506
6   1     312      431
7   1     517      465
8   1     582      423
9   1     623      388
10  1     642      397
```

Dataset `timedat` contains longitudinal measurements made in the same 386 HIV-positive individuals as `basdat`:

- `id` is the patient ID,
- `fuptime` is follow-up time, in days since HIV seroconversion,
- `cd4count` is CD4 count, measured at `fuptime`.

Note that these data were simulated using the algorithm described in Van der Wal, Prins, Lumbreras, and Geskus (2009). Therefore, CD4 count at a certain time point is affected by the TB status right before that time point. TB status at a certain time point is affected by CD4 count at that specific time point.

Some processing of the original data is necessary. We check if there is more than one CD4 measurement taken on the same day within each patient:

```
R> table(duplicated(timedat[, c("id", "fuptime")]))

FALSE
 6291
```

which is not the case. Because of skewness, we compute the square root of CD4 count:

```
timedat$cd4.sqrt <- sqrt(timedat$cd4count)
```

Add the time of first active TB to `timedat`, and compute `tb.lag`, the time-varying TB status one day before the measurement time (which is necessary for reasons that are explained below):

```
R> timedat <- merge(timedat, basdat[,c("id","Ttb")], by = "id", all.x = TRUE)
R> timedat$tb.lag <- ifelse(with(timedat, !is.na(Ttb) & fuptime > Ttb), 1, 0)
```

To be able to impute CD4 count at time points other than the measurement times, which is necessary when fitting the MSM (see below), and to smooth the original measurements, we fit the random effects model

$$\sqrt{L_i(t)} = \xi_i + \eta_i t + \beta_2 A_i(t-1), \tag{12}$$

with t follow-up time (days since HIV seroconversion), $L(t)$ CD4 count, and $A(t)$ time-varying TB status. Random effects ξ_i and η_i are assumed to be normally distributed with mean $\boldsymbol{\beta}' = (\beta_0, \beta_1)$ and covariance matrix $\Sigma = \left[\begin{smallmatrix} V_1 & V_{12} \\ V_{12} & V_2 \end{smallmatrix}\right]$. The model includes a fixed effect for TB, β_2. Because CD4 is affected by the TB status *right before* t, we include $A(t-1)$, the TB status one day before t in (12). We can fit model (12) using:

```
R> cd4.lme <- lme(cd4.sqrt ~ fuptime + tb.lag, random = ~ fuptime | id,
+    data = timedat)
```

We will construct a new dataframe `startstop`, which will be used to estimate inverse probability weights and to fit an MSM, to quantify the causal effect of TB on mortality. Let $\boldsymbol{T}_{TB}$ be all time points at which the TB-status switches, in any individual. Let $\boldsymbol{T}_{end}$ be all individual end times. Then, to (1) be able to compute inverse probability weights similar to (5) using a Cox proportional hazards model and (2) be able to fit the MSM, the dataframe `startstop` should contain, for each individual, rows corresponding to both $\boldsymbol{T}_{TB}$ and $\boldsymbol{T}_{end}$. For each individual we include these time points only up to his or her individual end time. We also sort the time points chronologically within each individual. The dataframe construction is done as follows:

```
R> times <- sort(unique(c(basdat$Ttb, basdat$Tend)))
R> startstop <- data.frame(
+    id = rep(basdat$id, each = length(times)),
+    fuptime = rep(times, nrow(basdat)))
R> startstop <- merge(startstop, basdat, by = "id", all.x = TRUE)
R> startstop <- startstop[with(startstop, fuptime <= Tend), ]
```

We compute the starting time for each interval of follow-up using `tstartfun`:

```
R> startstop$tstart <- tstartfun(id, fuptime, startstop)
```

Then we compute `tb`, the TB status at each time point for each individual, and `tb.lag`, the time-varying TB status one day before each time point for each individual. We also compute `event`, an indicator for death, and impute time-varying CD4 count `cd4.sqrt`, using (12):

```
R> startstop$tb <- ifelse(with(startstop, !is.na(Ttb) & fuptime >= Ttb),
+    1, 0)
R> startstop$tb.lag <- ifelse(with(startstop, !is.na(Ttb) & fuptime > Ttb),
+    1, 0)
R> startstop$event <- ifelse(with(startstop, !is.na(Tdeath) & fuptime >=
+    Tdeath), 1, 0)
R> startstop$cd4.sqrt <- predict(cd4.lme, newdata = data.frame(id =
+    startstop$id, fuptime = startstop$fuptime, tb.lag = startstop$tb.lag))
```

Note that for each row in `startstop`, `cd4.sqrt` contains imputed CD4 count that predicts `tb` in the same row. To correct for confounding by time-varying CD4 count, we can estimate the stabilized inverse probability weights

$$sw_{ij} = \prod_{k=0}^{j} \frac{P(A_{ik} = a_{ik} | \overline{A}_{ik-1} = \overline{a}_{ik-1})}{P(A_{ik} = a_{ik} | \overline{A}_{ik-1} = \overline{a}_{ik-1}, \overline{L}_{ik} = \overline{l}_{ik})}. \tag{13}$$

For time points up to the time point of the first instance of active TB within each patient, we estimate the elements in the denominator of (13) using the Cox proportional hazards model

$$\lambda_A[t|L(t), A(t^-) = 0] = \lambda_0(t) \exp\{\beta_1 L(t)\}. \tag{14}$$

We estimate the numerator of (13) using a model similar to (14) but only including the constant. Therefore, we can estimate sw_{ij} using:

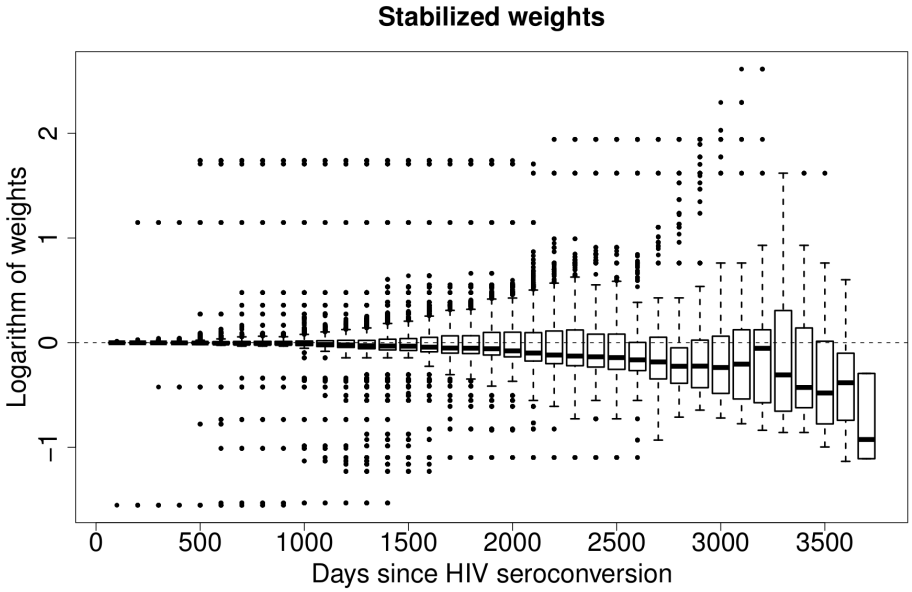

Figure 3: Weights distribution plot for example 3, made using `ipwplot`.

```
R> temp <- ipwtm(exposure = tb, family = "survival",
+    numerator = ~ 1, denominator = ~ cd4.sqrt, id = id,
+    tstart = tstart, timevar = fuptime, type = "first", data = startstop)
```

Since we are using `type = "first"`, the elements in the numerator and denominator of (13) are set to 1 within an individual after the first time point at which that specific individual develops active TB. Summarize and plot (see Figure 3) the inverse probability weights:

```
R> summary(temp$ipw.weights)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2117  0.9409  0.9908  1.0370  1.0120 13.6500

R> ipwplot(weights = temp$ipw.weights, timevar = startstop$fuptime,
+    binwidth = 100, main = "Stabilized weights", xlab = "Days since HIV
+    seroconversion", ylab = "Logarithm of weights", xaxt = "n")
R> axis(side = 1, at = c(0, 5, 10, 15, 20, 25, 30, 35), labels =
+    as.character(c(0, 5, 10, 15, 20, 25, 30, 35)*100))
```

To estimate the marginal causal effect of TB on mortality, we fit the MSM

$$\lambda_{T_{\bar{a}}}(t) = \lambda_0(t) \exp\{\beta_1 a(t)\}, \tag{15}$$

adjusted for confounding by CD4 count using IPW, and using a using a robust variance estimator, as follows:

```
R> summary(coxph(Surv(tstart, fuptime, event) ~ tb + cluster(id),
+    data = startstop, weights = temp$ipw.weights))
```

```
     coef exp(coef) se(coef) robust se     z Pr(>|z|)
tb 0.8127    2.2541   0.1901    0.2599 3.127  0.00177 **

   exp(coef) exp(-coef) lower .95 upper .95
tb     2.254     0.4436     1.354     3.751
```

We can compare the MSM to an unadjusted model:

```
R> summary(coxph(Surv(tstart, fuptime, event) ~ tb, data = startstop))

     coef exp(coef) se(coef)     z Pr(>|z|)
tb 1.4954    4.4612   0.1811 8.257   <2e-16 ***

   exp(coef) exp(-coef) lower .95 upper .95
tb     4.461     0.2242     3.128     6.362
```

We can also compare the MSM to a standard model, using conditioning to adjust for confounding:

```
R> summary(coxph(Surv(tstart, fuptime, event) ~ tb + cd4.sqrt,
+    data = startstop))

              coef exp(coef) se(coef)      z Pr(>|z|)
tb         0.24618   1.27913  0.24288  1.014    0.311
cd4.sqrt -0.24444   0.78314  0.03313 -7.378 1.61e-13 ***

          exp(coef) exp(-coef) lower .95 upper .95
tb           1.2791     0.7818    0.7947    2.0590
cd4.sqrt     0.7831     1.2769    0.7339    0.8357
```

The estimated hazard ratio corresponding to the causal effect of TB on mortality is 2.25 (95% CI 1.35–3.75). Note that the estimate from an unadjusted model of 4.46 (95% CI 3.13–6.36) is an overestimate, since both TB and death are more likely at lower CD4 counts. The estimate from the conditional model of 1.28 (95% CI 0.79–2.06) is an underestimate, since the indirect effect of TB through CD4 count is "conditioned away", as explained e.g., in Robins (1997) and Robins *et al.* (1999).

5. Conclusion

We have demonstrated how IPW can be performed to fit MSMs using our R package **ipw**, both in point treatment studies and in longitudinal studies, correcting for confounding and informative censoring. The package can accomodate for a wide range of exposure allocation models. Our package is easily used and does not involve extensive programming. We have also demonstrated how robust standard errors can be used for inference when fitting an MSM using IPW. Our contribution of the package **ipw** will make the MSM methodology more accessible to applied researchers. The package will also be useful to those using inverse probability

weighting for other purposes such as missing data problems (see e.g., Rao, Sigurdson, Doody, and Graubard 2005) or correcting for informative censoring (see e.g., Hernán *et al.* 2000 and Cole and Hernán 2004). In future updates of the package, the functions will also be applicable to other situations in which IPW is used, such as estimation and inference in competing risks survival analysis (Geskus 2011).

References

Clayton D, Spiegelhalter D, Dunn G, Pickles A (1998). "Analysis of Longitudinal Binary Data from Multiphase Sampling." *Journal of the Royal Statistical Society B*, **60**, 71–87.

Cole SR, Hernán MA (2004). "Adjusted Survival Curves with Inverse Probability Weights." *Computer Methods and Programs in Biomedicine*, **75**, 45–49.

Cole SR, Hernán MA (2008). "Constructing Inverse Probability Weights for Marginal Structural Models." *American Journal of Epidemiology*, **168**(6), 656–664.

Fewell Z, Hernán MA, Wolfe F, Tilling K, Choi H, Sterne JAC (2004). "Controlling for Time-Dependent Confounding Using Marginal Structural Models." *The Stata Journal*, **4**(4), 402–420.

Geskus RB (2011). "Cause-Specific Cumulative Incidence Estimation and the Fine and Gray Model Under both Left Truncation and Right Censoring." *Biometrics*, **67**(1), 39–49.

Glynn A, Quinn K (2010). ***CausalGAM**: Estimation of Causal Effects with Generalized Additive Models*. R package version 0.1-3, URL http://CRAN.R-project.org/package=CausalGAM.

Greenland S, Robins JM, Pearl J (1999). "Confounding and Collapsibility in Causal Inference." *Statistical Science*, **14**(1), 29–46.

Gruber S, van der Laan M (2010). ***tmleLite**: Targeted Maximum Likelihood Estimation of Additive Treatment Effect*. R package version 1.0-2, URL http://www.stat.berkeley.edu/~laan/Software/.

Halekoh U, Højsgaard S, Yan J (2005). "The R Package **geepack** for Generalized Estimating Equations." *Journal of Statistical Software*, **15**(2), 1–11. URL http://www.jstatsoft.org/v15/i02/.

Hernán MA, Brumback BA, Robins JM (2000). "Marginal Structural Models to Estimate the Causal Effect of Zidovudine on the Survival of HIV-Positive Men." *Epidemiology*, **11**(5), 561–570.

Hernán MA, Hernández-Díaz S, Robins JM (2004). "A Structural Approach to Selection Bias." *Epidemiology*, **15**(5), 615–625.

Hernán MA, Robins JM (2006). "Estimating Causal Effects from Epidemiological Data." *Journal of Epidemiology and Community Health*, **60**, 578–586.

Jonsson Funk M, Westreich D, Davidian M, Weisen C (2007). "Introducing a SAS Macro for Doubly Robust Estimation." In *SAS Global Forum 2007*. Paper 189-2007.

Lefebvre G, Delaney JAC, Platt RW (2008). "Impact of Mis-Specification of the Treatment Model on Estimates from a Marginal Structural Model." *Statistics in Medicine*, **27**, 3629–3642.

Little RJA, Rubin DB (1987). *Statistical Analysis with Missing Data*. John Wiley & Sons, Chichester.

Lumley T (2004). "Analysis of Complex Survey Samples." *Journal of Statistical Software*, **9**(8), 1–19. URL http://www.jstatsoft.org/v09/i08/.

Rao RS, Sigurdson AJ, Doody MM, Graubard BI (2005). "An Application of a Weighting Method to Adjust for Nonresponse in Standardized Incidence Ratio Analysis of Cohort Studies." *Annals of Epidemiology*, **15**(2), 129–136.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Robins JM (1997). "Causal Inference from Complex Longitudinal Data." In M Berkane (ed.), *Latent Variable Modeling and Applications to Causality: Lecture Notes in Statistics 120.*, pp. 69–117. Springer-Verlag, New York.

Robins JM (1998). "Marginal Structural Models." In *1997 Proceedings of the American Statistical Association, Section on Bayesian Statistical Science*, pp. 1–10. American Statistical Association, Alexandria. URL http://biosun1.harvard.edu/~robins/msm-web.pdf.

Robins JM, Greenland S, Hu FC (1999). "Estimation of the Causal Effect of a Time-Varying Exposure on the Marginal Mean of a Repeated Binary Outcome." *Journal of the American Statistical Association*, **94**(447), 687–700.

Robins JM, Hernán MA, Brumback BA (2000). "Marginal Structural Models and Causal Inference in Epidemiology." *Epidemiology*, **11**, 550–560.

Therneau T, Lumley T (2011). ***survival**: Survival Analysis, Including Penalised Likelihood*. R package version 2.36-8, URL http://CRAN.R-project.org/package=survival.

Van der Laan MJ (2010). "Targeted Maximum Likelihood Based Causal Inference: Part I." *The International Journal of Biostatistics*, **6**(2), 2.

Van der Wal WM, Prins M, Lumbreras B, Geskus RB (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**, 2325–2337.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.

Wang Y, Hartman E, Gruber S (2009). ***cvDSA**: Selecting MSM Using Cross-Validation Deletion/Substitution/Addition Algorithm*. R package version 0.5-3-2, URL http://www.stat.berkeley.edu/~laan/Software/.

Affiliation:

Willem M. van der Wal
Department of Biostatistics
Julius Center, University Medical Center Utrecht
Room 7.125
P.O. Box 85500
3508 GA Utrecht, The Netherlands
E-mail: w.m.vd.wal@umcutrecht.nl

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 43, Issue 13 *Submitted:* 2010-04-07
September 2011 *Accepted:* 2011-07-23