# Propensity score based data analysis

Susanne Stampf

October 29, 2012

**Abstract**

For some time, propensity score (PS) based methods have been frequently applied in the analysis of observational and registry data. The PS is the conditional probability of a certain treatment given patient's covariates. PS methods are used to eliminate imbalances in baseline covariate distributions between treatment groups and permit to estimate marginal effects.

The package `nonrandom` is a tool for a comprehensive data analysis using stratification, matching and covariate adjustment by PS. Several functions are implemented, starting from the selection of the PS model up to estimating the PS based treatment effect. Before estimating the PS, knowledge about which covariates should be included in the PS model is needed. `relative.effect()` provides the opportunity to investigate the extent to which a covariate confounds the treatment-outcome relationship. `pscore()` estimates the PS and `plot.pscore()` offers a graphical presentation of the PS distribution. Stratification and matching by PS are directly implemented in `ps.makestrata()` and `ps.match()`. To check covariate balance between treatment groups, statistical tests or standardized differences are given in `ps.balance()`. In addition, `dist.plot()` and `plot.stdf()` provide a graphical balance check. Finally, the PS based treatment effect can be estimated in `ps.estimate()`. This function also offers a comparison to regression based estimates. A special case of regression based analysis may be covariate adjustment by PS.

All functions can be applied separately and combined. Additionally, it is possible to apply all functions repeatedly to decide which analysis strategy is most suitable. Print and summary functions are available for the most implemented functions.

There are two real data examples to illustrate the application of `nonrandom`. In the first data example, quality of life is investigated in breast cancer patients in an observational treatment study of the German Breast Cancer Study Group (GBSG). The second data example deals with lower respiratory tract infections (LRTI) in infants and children in the observational study Pri.DE (Pediatric Respiratory Infection, Deutschland) in Germany.

# Contents

# 1 Introduction

For some time, propensity score (PS) based methods have been frequently applied in the analysis of observational and registry data. In the medical field, the PS is often defined as the conditional probability of receiving a certain treatment given covariates [1]. In general, it is unknown and has to be estimated, e.g.~via logistic regression. The selection of an appropriate PS model may be the first obstacle. Lunt et al.~[2] proposed a measure estimating the extent to which a covariate confounds the treatment effect on outcome. Covariates with a large extent are potential candidates for the inclusion in the PS model.

In observational studies, covariate distributions differ generally between treatment groups and PS based methods aim to eliminate these imbalances. There are four PS based methods: stratification, matching, covariate adjustment and inverse probability weighting by PS. The first both methods are commonly used and intend to create data situations as in randomized controlled trials (RCTs) within which direct comparison of treatment groups are meaningful. Covariate adjustment by PS is also a favourite approach since it is easy to apply. The PS is included as a covariate in addition to treatment in a usual regression model. The fourth approach, namely the inverse probability weighting by PS, is rather rarely used [3, 4]. The PS is here used to weight each individual and it is often applied as weighted regression [5]. Stratification and matching by PS are more popular methods since they are easy to understand. However, matching by PS is applied at most

in medical research [6, 7].

Stratification by PS is used to group individuals with similar or equal PS such that the distributions of measured covariates are sufficiently balanced in the treatment groups within each stratum [1, 8]. It can be supposed that each stratum mimics a randomized situation within which the distributions of the measured covariates are balanced in expectation. If the assumption of 'strongly ignorable treatment assignment' (SITA) holds, stratum-specific parameters can be unbiasedly estimated [1] and those estimates are in turn summed up using appropriate weights to estimate the parameter of interest.

If matching by PS is used, one or more untreated individuals are matched to one treated individual or vice versa. Individuals within these matched pairs or sets have similar or equal PS whereas the similarity is often defined by a so-called caliper, generally used as one-fifth of the standard deviation of the logit of the estimated PS [9]. Although matching by PS has been frequently applied [6, 7], it has been shown that the dependence structure in the matched sample is often not accounted for in the estimation of the interesting parameter [10]-[12]. Approaches such as generalized linear mixed models and generalized estimation equations are appropriate to analyze data with correlation structure [13]-[17].

In general, PS methods are embedded in the framework of causal modeling dealing with potential outcomes [18]-[20]. Consider a pair of random variables $(Y_0, Y_1)$, where $Y_1$ denotes the outcome of an individual if treated, and $Y_0$ represents the outcome of the same individual if not treated[1]. The observed outcome is $Y = ZY_1 + (1 - Z)Y_0$ whereas $Z$ is the treatment indicator with $Z = 1$ if the individual is treated and $Z = 0$ if not. The expected values of the potential outcomes $\mathbf{E}[Y_1]$ and $\mathbf{E}[Y_0]$ can be derived if the identifying assumption (SITA) holds [1]. This assumption states, that, within groups defined by PS, the observed outcome of individuals assigned to treatment $Z = 0$ has the same distribution as the unobserved outcome of individuals assigned to treatment $Z = 1$, if the latter had been assigned to treatment $Z = 0$.

The idea of PS was initiated to estimate marginal linear treatment effects defined as $\Delta = \mathbf{E}[Y_1 - Y_0]$ [1]. By now, the idea has been transferred to estimate non-linear effects, e.g.~the marginal odds ratio which is the change in odds of outcome, if everybody versus nobody were treated [21]-[23]. Therefore, the marginal probabilities for the potential outcomes $\mathbf{P}[Y_1 = 0]$ and $\mathbf{P}[Y_1 = 1]$ have to be estimated which are used to construct an appropriate estimator for the marginal odds ratio defined as $\delta = \frac{p_1/(1-p_1)}{p_0/(1-p_0)}$ with $p_z = \mathbf{P}[Y_z = 1]$, $z = 0, 1$. In case of stratified data, the marginal probabilities of the potential outcomes can be estimated by the outcome rates from the PS strata or derived from the standard regression results [21].

---

[1]For simplicity, indices for individiuals are here ignored.

In the following, the application of the `nonrandom` package is demonstrated step by step introducing all implemented functions. The usage of all functions is illustrated by exemplary analyses of two data sets. First, there are data on quality of life in breast cancer patients in an observational treatment study of the German Breast Cancer Study Group (GBSG) [24, 25]. Patients with mastectomy and lumpectomy are compared with each other regarding the quality of life measured as a linear sum score. The second data example deals with lower respiratory tract infections (LRTI) in a population of infants and children years in the observational study Pri.DE (Pediatric Respiratory Infection, Deutschland) in Germany [26]. Here, the impact of a current infection with the respiratory syncytial virus (RSV) on the severity of LRTI is investigated [23].

## 2 The estimation of PS

The PS is generally unknown and has to be estimated whereas the selection of the PS model is often a delicate issue [27]-[31]. A measure describing the extent to which a covariate is confounding the treatment effect on the outcome is proposed by Lunt et al.~[2]. Covariates with a large impact are potential candidates for the inclusion in the PS model. This proposal is implemented in the function `relative.effect()`.

### 2.1 Selection of PS model ... relative.effect()

An important step is to decide which baseline covariates $X_k$, $k = 1, ..., K$, are necessary to include in the PS model. A measure describing the extent to which a covariate $X_k$ confounds the effect of treatment $Z$ on the outcome $Y$ may be helpful. In linear case, it is defined as the relative effect (per cent)

$$\left| \frac{\beta_{z,x_k} - \beta_z}{\beta_z} \right| \times 100$$

and for binary outcome

$$\left| \frac{e^{\{\beta_{z,x_k}\}} - e^{\{\beta_z\}}}{e^{\{\beta_z\}}} \right| \times 100$$

with regression coefficients for treatment $\beta_z$ and $\beta_{z,x_k}$ from the corresponding regression models. That means, $K$ adjusted (in abstract formula: $Y \sim Z + X_k$) and one unadjusted regression models ($Y \sim Z$) are fitted using generalized linear regression with respect to the measuring scale of the outcome (internal use of `glm()`).

To define the models, two specification options are available in `relative.effect()`. First, outcome, treatment and covariates can be defined separately as strings or numerics

using the arguments `resp`, `treat` and `sel` or, secondly, an explicit regression formula using the argument `formula` can be specified:

```
> ## (1) PRI.De data dealing with LRTI
> load(pride)
> pride.effect <- relative.effect(data = pride,
+                                  sel = c(2:14),
+                                  family = "binomial",
+                                  resp = 15,
+                                  treat = "PCR_RSV")
>
>
> ## (2) STU1 data on quality of life
> load(stu1)
> stu1.effect <- relative.effect(data = stu1,
+                                  formula = pst~therapie+tgr+age)
```

Information about the unadjusted and the adjusted treatment effects on outcome and the corresponding relative effects are available, e.g.~in the STU1 example:

```
> stu1.effect

 Treatment: therapie
 Outcome: pst
 Covariates:  tgr age

 Unadjusted treatment effect: 1.5894

 Adjusted and relative effects:

    adj. treatment effect rel. effect
age           0.7880392   50.420198
tgr           1.7004732    6.985956
```

Two covariates 'tgr' (tumor size, binary) and 'age' (age groups) are investigated and they seem to affect the 'therapie' effect on quality of life ('pst') in two different directions.

## 2.2  Estimation of PS ... pscore() and plot.pscore()

If covariates needed for the PS estimation are determined, a logistic regression model is fitted (internal use of `glm()`) assuming a binary treatment variable:

```
> stu1.ps <- pscore(data = stu1,
+                   formula = therapie~tgr+age)
>
> pride.ps <- pscore(data = pride,
+                   formula = PCR_RSV~SEX+RSVINF+REGION+
+                                     AGE+ELTATOP+EINZ+EXT,
+                   name.pscore = "ps")
```

It is possible to specify a name for the estimated PS which in turn is used to save it in the data set (argument `name.pscore`, default is *'pscore'*).

**PRI.De study**



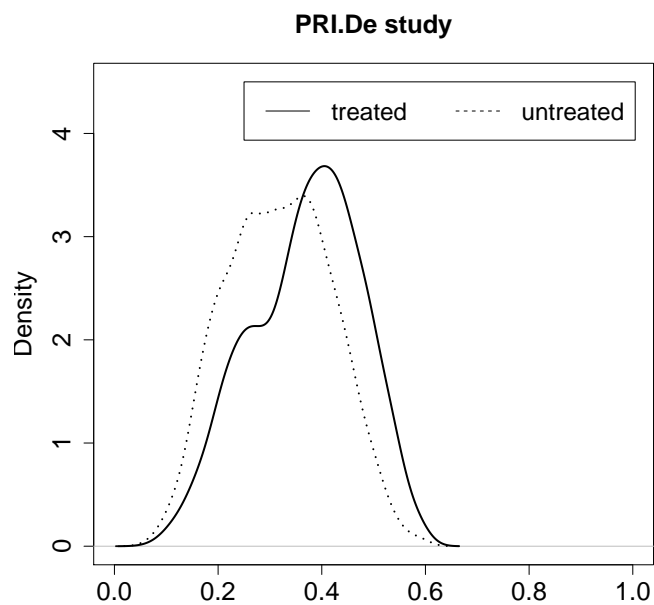Figure 1: Density estimation of the estimated PS in the PRI.De data using `plot.pscore()`

In the current version 1.3, a plot function `plot.pscore()` is available to illustrate the estimated density of the estimated PS in the treatment groups. However, the function `pscore()` has to be previously used.

```
> ## Figure 1
> plot.pscore(pride.ps,
+             main = "PRI.De study",
+             with.legend = TRUE,
```

```
+              par.1 = list(lty=1, lwd=2), ## line type and size
+                                          ## for PS density in
+                                          ## the group '1'
+              par.0 = list(lty=3, lwd=2), ## ... in the group '0'
+              xlab = "",
+              ylim = c(0,4.5))
```

The output object is of class *'pscore'* and contains comprehensive information about the PS model, e.g.~the specified PS model formula (`$formula.pscore`) and the name of the PS estimated at last (`$name.pscore`), the treatment (`$name.treat`) and the outcome (`$name.resp`). Furthermore, the complete data set (`$data`) extended by the estimated PS, a vector with values of the PS estimated at last (`$pscore`) and the treatment variable (`$treat`) are separately available.

# 3   PS based methods

Observational and registry data frequently present imbalances in covariate distributions between treatment groups. Stratification and matching by PS can be used to eliminate these imbalances by creating data situation as in RCTs.

## 3.1   Stratification by PS ... ps.makestrata()

In general, stratification groups individuals who have similar or equal values for the stratification variable. If the (estimated) PS is the stratification variable, its distribution is often used to define the strata [8, 34].

The use of the function `ps.makestrata()` depends on the class of the input object whereas *'data.frame'* and *'pscore'* (if `pscore()` is previously used) are permitted. The arguments `stratified.by` and `breaks` are crucial for the stratification procedure: `stratified.by` defines the stratification variable in the data (i.e.~defined by argument `object`) and the way to stratify is specified by `breaks`. The former argument is not necessarily needed if the input object is of class *'pscore'*. The estimated PS stored in `$pscore` is then automatically sourced. Contrary, it is needed if the input object is of class *'data frame'*. The argument `name.stratum.index` specifies the name of the variable containing the stratum indices which is stored in the data.

There are three possibilities to use the argument `breaks`. First, the stratification variable is simply factorized and each factor corresponds to one stratum (`breaks` is set to *'NULL'*, default setting). This is useful when the stratification variable has only few values.

```
> ## (1) stratification (factorization)
> stu1.str4 <- ps.makestrata(object = stu1.ps)
> stu1.str4

 Stratified by:  pscore

 Strata information:

    Strata bounds      n     n (per cent)
1           0.601    231           0.358
2           0.709     65           0.101
3           0.824    255           0.395
4           0.883     95           0.147
```

Secondly, the number of strata can be specified. Therefore, an integer has to be specified.

```
> ## (2) stratification (specification of the number of strata)
> pride.str.b3 <- ps.makestrata(object = pride.ps, breaks = 3)
> pride.str.b3

  Stratified by:  ps

  Strata information:

     Strata bounds      n     n (per cent)
1    [0.0619,0.24]    634           0.206
2     (0.24,0.417]   1713           0.557
3    (0.417,0.595]    731           0.237
```

The third method permits the definition of specific stratum limits. For it, appropriate R-functions as e.g.~`quantile()` can be used as well as pre-specified interval limits. In case of the PRI.De example, quintiles from the distribution of the estimated PS are chosen:

```
> ## (3) stratification (definition of specific stratum limits)
> pride.str5 <- ps.makestrata(object = pride.ps,
+                             breaks = quantile(pride.ps$pscore,
```

```
+                                              seq(0,1,0.2)),
+                           name.stratum.index = "stratum")
```

Depending on the class of the input object, `ps.makestrata()` returns an object of class *'stratified.pscore'* or *'stratified.data.frame'*. If the class of the input object is *'pscore'*, the output object inherits all information from the input object. The complete data set (`$data`) extended by the generated stratum indices labeled by `$name.stratum.index` is available. Furthermore, the name of the stratification variable (`$stratified.by`), the variable with the stratum indices (`$stratum.index`) and the corresponding stratum limits (`$intervals`) generated at last are stored in the output object.

Stratification of data can be done repeatedly, but only the information from the last application is stored separately in the output object. However, stratum indices from all stratification procedures previously done are included in the data set stored in the output object.

## 3.2   Matching by PS ... ps.match()

The most popular PS method to cope with covariate imbalances is matching by PS. One or more untreated individuals are matched to treated individuals (or vice versa) according to the estimated PS. Individuals matched to each other have a similar or an identical estimated PS wheres the similarity is determined by a caliper size defined by a pre-specified maximum distance between the estimated PS from treated and untreated individuals.

Similar to `ps.makestrata()`, the use of `ps.match()` depends both on the class and the numbers of the input objects. The classes *'data.frame'* and *'pscore'* (if `pscore()` is previously used) are supported and one or two input objects of class *'data.frame'* are allowed. The argument `matched.by` defining the matching variable in the data has not to be specified when the input object is of class *'pscore'*. The estimated PS stored in `$pscore` is then automatically used. Additionally, the treatment variable is also not necessarily needed since it is already stored in `$treat`. Contrary, both arguments are needed when the input object(s) are of class *'data.frame'*. The argument `name.match.index` specifies the name of the variable that contains the matching indices (matching sets of individuals get a consecutive matching index). This variable is stored in the data.

In `ps.match()` it is possible to pass two data frames by using the argument `object.control` in addition to the argument `object`. This can be useful if one data set only includes treated individuals and a second data set contains information about

the untreated individuals. If the matching variable (`matching.by`) differs in both data sets, the matching variable in the second data set must also be specified in argument `control.matched.by`. Independent of the classes and the number of input objects, the treatment value defining *'treated'* individuals must be given in argument `who.treated` (default is *'1'* and means *'treated'*).

There are further parameters specifying the matching procedure. The matching ratio (argument `ratio`) indicates how many individuals should be matched and the argument `givenTmatchingC` concerns the direction of the matching procedure, i.e.~who is matched to whom. The default is *'TRUE'*, i.e.~treated individuals are matched to untreated individuals as long as `who.treated` is set to *'1'* describing *'treated'*). The argument `bestmatch.first` means that matching partners with the most similar estimated PS are selected to be matched (*'TRUE'*, default) or they are taken randomly from the pool of potential matching partners (*'FALSE'*). The caliper size can be defined by the arguments `caliper` and `x`. The default setting for the caliper size is one-fifth (`x`=*'0.2'*) of the standard deviation of the logit of estimated PS (`caliper`=*'logit'*) [9, 35]. However, it is possible to specify numerics in the argument `caliper` (argument `x` is then disregarded) or to modify the argument `x` in connection with `caliper="logit"`. Furthermore, a random number can be specified (`setseed`) to make the matching procedure reproducible.

In PRI.De data, untreated individuals are matched to treated individuals in a ratio of 1 : 1 (default) and the default caliper size is used. The matching variable is set to *'ps'*. Comprehensive information are available via the `summary` function:

```
> pride.m1 <- ps.match(object = pride.ps,
+                      ratio = 1, ## default
+                      x = 0.2, caliper = "logit", ## default
+                      matched.by = "ps",
+                      setseed = 38902)
> summary(pride.m1)

 Matched by:  ps

 Matching parameter:

Caliper size:    0.102
Ratio:           1.000
Who is treated?: 1.000

 Matching information:
```

```
Untreated to treated?: TRUE
Best match?:           TRUE

 Matching data:

Number of treated obs.:            1031
Number of matched treated obs.:    1031
Number of untreated obs.:          2047
Number of matched untreated obs.:  1031
Number of total matched obs.:      2062
Number of not matched obs.:        1016
Number of matching sets:           1031
Number of incomplete matching sets:   0
```

The matching algorithm for the STU1 example is switched, i.e.~two treated individuals should be matched to one untreated individual (`givenTmatchingC=FALSE`) since there are fewer untreated than treated individuals in the data. Furthermore, the caliper size is set to 0.5.

```
> stu1.m2 <- ps.match(object = stu1.ps,
+                     ratio = 2, caliper = 0.5,
+                     givenTmatchingC = FALSE,
+                     setseed = 39062)
Argument 'givenTmatchingC'=FALSE: Treated elements were matched to
each untreated element.
```

The output object(s) are either of class *'matched.pscore'*, *'matched.data.frame'* or *'matched.data.frames'* which depends on the class and the numbers of the input objects and on the specification of the argument `combine.output`. This is useful when two data frames are specified as input objects. The default is *'TRUE'*, i.e.~the two input data frames extended by the matching information are combined for the output and only one output data frame of class *'matched.data.frame'* is given. Otherwise, the output object is of class *'matched.data.frames'*.

The data set and the data set limited to the matched individuals are stored in the output object (`$data`, `$data.matched`). Both are extended by column(s) including the matching indices labeled by `name.match.index`. Furthermore, individual matching indices generated at last (`$match.index`, `$name.match.index`), the name of the matching variable (`$matched.by`) and several matching parameters (`$match.parameter`) used at last are stored. If there are two input objects and argument `combine.output` is set to *'FALSE'*, the values `$data`, `$data.matched` and `$match.index` are lists of data

frames and vectors corresponding to the input objects. If the class of the input object is *pscore*, the output object also inherits all information from the input object.

# 4   The balance check for covariate distributions

PS methods are used to cope with imbalances in covariate distributions between the treatment groups. An important, but often neglected issue is the check for covariate balance between treatment groups after stratification or matching by PS. Graphics, statistical tests and standardized differences can therefore be applied [36]-[38].

## 4.1   Graphical balance check ... dist.plot()

The function `dist.plot()` offers an illustration of the covariate distributions in the treatment groups. Its usage depends on the class of the input object. As before, arguments `treat`, `stratum.index` and `match.index`, respectively, are not necessarily needed if the input object results from a previous application of `ps.makestrata()` or `ps.match()`. This is in contrast to the situation when the input object is of class *'data frame'*.

If the input object is of class *'stratified.data.frame'* or *'stratified.pscore'*, covariate distributions are plotted separately by the treatment in each stratum. If the class of the input object is either *'matched.data.frame'*, *'matched.data.frames'* or *'matched.pscore'*, the covariate distributions are illustrated per treatment group in the matched data. Of course, a comparison to the covariate distribution in the original data is possible (argument `compare`, default is *'FALSE'*).

```
> ## Figure 2 (left)
> stu1.dplot1 <- dist.plot(object      = stu1.m2,
+                          sel         = c("tgr"),
+                          compare     = TRUE,
+                          label.match = c("original data",
+                                          "matched sample"))
>
> ## Figure 2 (right)
> stu1.dplot2 <- dist.plot(object      = stu1.m2,
+                          sel         = c("age"),
+                          plot.type   = 2,
+                          compare     = TRUE,
```
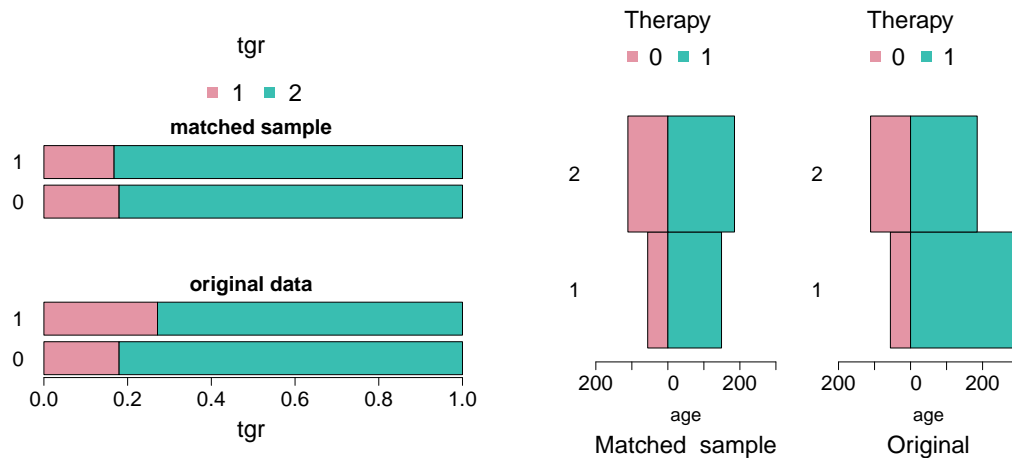
Figure 2: Balance check for the categorical covariates 'tgr' and 'age' in the STU1 data before and after matching using different plot types in `dist.plot()`: `plot.type=1` (left) and `plot.type=2` (right)

```
+                                    legend.title = "Therapy")
```

There are two different plot types (argument `plot.type`) which act differently depending on the measuring scale of covariates. The default is *'1'*, i.e.~stacked bar plots are used to show frequencies (or rather proportions) for categorical and means for continuous covariates. The decision about the categorical or the continuous character of the selected covariates (argument `sel`) is controlled by the argument `cat.level` (default is *'2'*, i.e.~covariates with more than 2 different values are considered as continuous). The covariate distributions are illustrated by histograms if `plot.type` is set to *'2'*. Therefore, argument `plot.levels` specifies the number of cutpoints needed to define histogram classes for continuous covariates. However, the definition of the number of histogram classes (`plot.levels`, default is *'5'*) in case of continuous covariates still depends on the covariate structure to be plotted such that class number finally used may differ from the specified number. If the covariate is categorical, the number of its categories are used to define cutpoints.

```
> ## Figure 3 (left)
> pride.dplot1 <- dist.plot(object       = pride.str5,
+                           sel          = "AGE",
+                           label.stratum = c("Str.","Original"))
>
> ## Figure 3 (right)
```

```
> pride.dplot2 <- dist.plot(object       = pride.m1,
+                           sel          = c("AGE"),
+                           plot.type    = 2,
+                           compare      = TRUE,
+                           legend.title = "RSV infection")
```

If `plot.type=1` and the covariates are categorical/continuous, the category/treatment labels are given in the legend. Therefore, users have to be careful to modify the argument `legend.title` if both covariate types are simultaneously plotted. If `plot.type=2`, the treatment labels are shown in the legend independent of the covariate type.
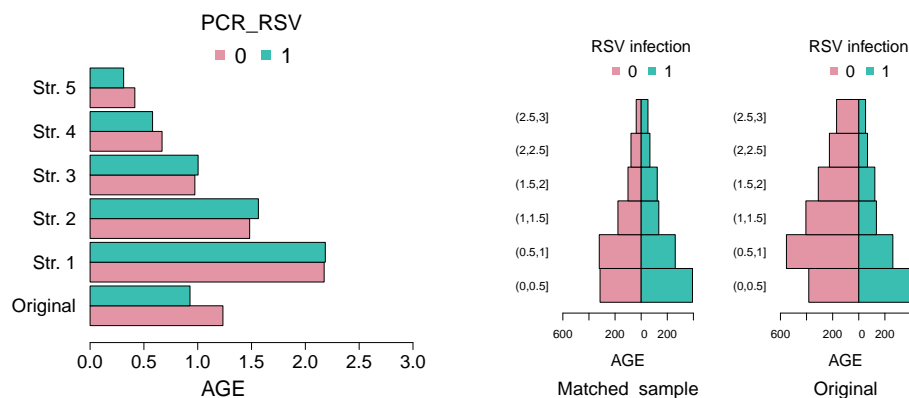


Figure 3: Balance check for 'Age' (in years) in the PRI.De data before and after stratification (left) and matching (right) using different plot types in `distplot()`: mean bars (left, `plot.type=1`) and histograms (right, `plot.type=2`)

There are further useful arguments. For example, the arguments `label.stratum` and `label.match` provide a modification of the label names within the graphic. Defaults are *'Original'* and *'Stratum'* and *'Matched'* for `label.stratum` and `label.match`, respectively. Further arguments can be used to modify, among others, font sizes, labels, colors and plot margins.

The function `dist.plot()` returns numerous values which are stored in the output object, e.g.~names of categorical (`$var.cat`) and continuous covariates (`$var.noncat`). The length and the structure of the output (stored as lists) depend on the type of the selected covariates and the chosen plot type.

For example, frequencies (scaled to one, therefore it should be called rather proportions than frequencies) of the categorical covariate 'tgr' in the treatment groups before and

after matching are shown in Figure 2 on the left hand side. They are stored in the value `$frequency` of the output object `stu1.dplot1` containing a list for each selected categorical covariate which in turn includes cross tables for treatment (columns) and covariate (rows), one for the 'original data' and for the 'matched sample'. There is no proportion difference in the group of untreated individuals (labeled by treatment '0'), but there is a difference in the group of treated individuals (since treated to untreated individuals were matched). After matching ('matched sample'), the proportions of 'tgr' are more similar in both treatments compared to the original data.

```
> ## Return values of 'stu1.dplot1' presenting in Figure 2 (left)
> stu1.dplot1$var.cat
[1] "tgr"
>
> stu1.plot1$frequency
[[1]]
, , index = original data

   treat
           0         1
  1 0.1796407 0.2713987       ## rows: 'tgr' categories
  2 0.8203593 0.7286013       ## columns : treatment

, , index = matched sample

   treat
           0         1
  1 0.1796407 0.1676647
  2 0.8203593 0.8323353
```

If the argument `plot.type` is set to 2 as done in Figure 2 on the right hand side, frequencies shown in histograms are separately saved for both treatment groups before and after matching.

```
> ## Return values of 'stu1.dplot2' presenting in Figure 2 (right)
> stu1.dplot2$var.cat
[1] "age"
>
> stu1.dplot2$x.s.cat
[[1]]                       ## x.s.cat: category frequencies for
   index                    ## covariate 'age' (rows) in treatment
      1    2                 ## group (Therapy=0) in original data
  1  56   56                 ## (index=1) and in the matched sample
```

```
  2 111 111                 ## (index=2) (columns)


> stu1.dplot2$y.s.cat
[[1]]                       ## y.s.cat: category frequencies for
   index                    ## covariate 'age' (rows) in treatment
      1   2                 ## group (Therapy=1) in original data
  1 294 149                 ## (index=1) and in the matched sample
  2 185 185                 ## (index=2) (columns)
```

In Figure 3, the distribution of the continuous covariate 'AGE' in the PRI.De data set are illustrated in both treatment groups before and after stratification (left) and matching (right). On the left hand side, means of 'age' in both treatment groups in original data and in each stratum are given. The corresponding means are stored as follows:

```
> ## Return values of 'pride.dplot1' presenting in Figure 3 (left)
> pride.dplot1$var.noncat
[1] "AGE"
>
> pride.plot1$mean
[[1]]                       ## 1st list element for original data
[[1]][[1]]                  ## one list entry for each 'var.noncat'
        0         1
1.2342170 0.9283253         ## means per treatment groups 0/1


[[2]]                       ## 2nd list element for stratified data
[[2]][[1]]                  ## one list entry for each 'var.noncat'
         1        2         3         4         5
0 2.174609 1.482517 0.9733666 0.6686585 0.4156446
1 2.185500 1.563706 1.0035175 0.5804821 0.3117728
```

The first list element of `...$mean` contains the 'age' means in both treatment group in the original data. There is in turn one entry for each selected covariate (the sequence of covariates is identical to that defined in argument `sel`; since only 'age' is here selected, there is only one entry). Stratum-specific means are given in the second list elements (rows: treatment, columns: strata) which in turn have so many entries as covariates were selected in argument `sel`.

The return values of the object `pride.m1` plotted on the right hand side in Figure 3 are stored in `pride.dplot2`:

```
> pride.dplot1$var.noncat
[1] "AGE"
>
pride.plot2$breaks.noncat        ## breaks points for
[[1]]                            ## histogram classes
[1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0
>
> pride.dplot2$x.s.noncat        ## frequencies for
[[1]]                            ## 'RSV infection = 0'
[[1]]$`1`                        ## in original data('1') and
[1] 383 554 405 310 225 170      ##
                                 ##
[[1]]$`2`                        ## in matched data ('2')
[1] 315 321 178 101  78  38      ##

> pride.dplot2$y.s.noncat        ## frequencies for
[[1]]                            ## 'RSV infection = 1'
[[1]]$`1`                        ## in original data('1') and
[1] 393 261 135 123  67  52      ##
                                 ##
[[1]]$`2`                        ## in matched data ('2')
[1] 393 261 135 123  67  52      ##
```

In addition to the data information that is plotted, treatment ($treat), individual stratum indices ($stratum.index) or matching indices ($match.index) and selected covariates ($name.sel, $sel) are saved in the output object.

## 4.2 Statistical tests and standardized differences ... ps.balance() and plot.stdf()

The graphical illustration of the covariate distributions in the treatment groups gives a first insight, however, statistical tests decide whether differences potentially visible in the graphics are significant. The function ps.balance() permits the application of statistical tests and results can be used to decide about covariate balance. Since there is an ongoing discussion about the appropriateness of statistical tests for balance decision, standardized differences are proposed to decide about covariate balance, especially in matched data [39]-[41].

The use of the function ps.balance() depends on the class of the input object. If information about treatment, stratum or matching indices are stored in the input object, they

are automatically sourced from the input object if not specified in the corresponding argu-
ments. If the input object is a *'data frame'*, those arguments have to be given. By default,
statistical tests (argument `method="classical"`) are used with respect to the measur-
ing scale of the selected covariates, i.e.~$t/\chi^2$-test for continuous/categorical covariates
(internal use of `t.test()` and `chisq.test()`). As before, the argument `cat.levels`
defines whether a covariate is categorical or continuous. The tests are applied to data
both before and after stratification or matching.

```
> ## PRI.De, stratification
> ## Balance check using statistical tests
>
> pride.str5.bal <- ps.balance(object    = pride.str5,
+                              sel       = c(2:8),
+                              cat.levels = 4,
+                              method     = "classical",
+                              alpha      = 5)
> pride.str5.bal

 Summary of balance check:

               Before: no bal (0) Before: bal (1)
After: no bal (0)                2               1
After: bal (1)                   1               2

 Covariates not completely tested:
RSVINF

 Detailed balance check (overall):

       SEX ETHNO FRUEHG RSVINF HERZ REGION AGE
Before   0     1      1      0    1      0   0
After    1     0      1     NA    1      0   0

 Detailed balance check (per stratum):
 [p-values from tests (significance level: 0.05)]

            SEX ETHNO FRUEHG RSVINF  HERZ REGION   AGE
Before     0.01 0.907  0.413      0 0.518      0     0
------     ----- ----- -----  ----- ----- ----- -----
Stratum 1 0.296 0.632  0.223  0.647     1  0.058  0.83
Stratum 2  0.16 0.003   0.98  0.422 0.642  0.133 0.084
Stratum 3 0.798 0.169  0.678      1 0.484  0.038 0.429
```

```
Stratum 4 0.124 0.212  0.724     NA 0.843  0.542 0.002
Stratum 5  0.96 0.882  0.404     NA 0.523  0.415     0
--------- ---- ----   ----   ---- ----   ---- ----
Test      chi^2 chi^2 chi^2  chi^2 chi^2  chi^2    t
```

For example, seven covariates in the stratified PRI.De data are checked by statistical tests for the balance between the treatment groups. Using the `print` function, four different information are available. First, a summary table for the decision of 'balance' or 'no balance' is printed. There are each three balanced and three not balanced covariates before stratification (columns of table 'Summary of balance check'). Stratification then results in three balanced and three not balanced covariates (rows). However, there is one of the three covariates not balanced before that becomes balanced after stratification. Covariate-specific information are not given in this table. The next information is about covariates for which tests in the original data or in at least one stratum could not be performed correctly. This is the case for the covariate 'RSVINF' describing a previous RSV infection (yes/no). The table 'Detailed balance check (overall)' presents covariate-specific balance information where '0' indicates 'not balanced' and '1' means 'balanced'. It is noticeable that covariate 'SEX' becomes balanced (indicated by '1') after stratification and 'REGION' and 'AGE' remain unbalanced ('0'). The fourth and last information shows the test results, i.e.~p-values from statistical tests applied in the original data and in each stratum if possible. Additionally, the used tests are given. 'NA' means that no test is performed, i.e.~covariate 'RSVINF' is not checked in stratum 4 and 5 since there are no individuals with 'RSVINF=1'.

In the STU1 data example, the covariate balance of 'tgr' and 'age' between the treatment groups is of interest before and after matching. Therefore, standardized differences are calculated based on the formula described in [42]. Tables are similar to those described above. Standardized differences are now listed in the last information part 'Detailed balance check'. The cutpoint at which a decision about balance is made is set to *'20'* (argument `alpha`). Both covariates are classified as unbalanced before matching since the standardized differences are larger than 20. The covariate describing tumor size ('tgr') becomes balanced after the matching procdure. All these information are stored in the return value `$bal.test` of the output object `stu1.m2.bal`. In Figure 4, standardized differences are plotted.

```
> ## STU1, Matching
> ## Balance check using standardized differences
>
> stu1.m2.bal <- ps.balance(object  = stu1.m2,
+                           sel     = c("tgr","age"),
+                           method  = "stand.diff", alpha   =  20)
```
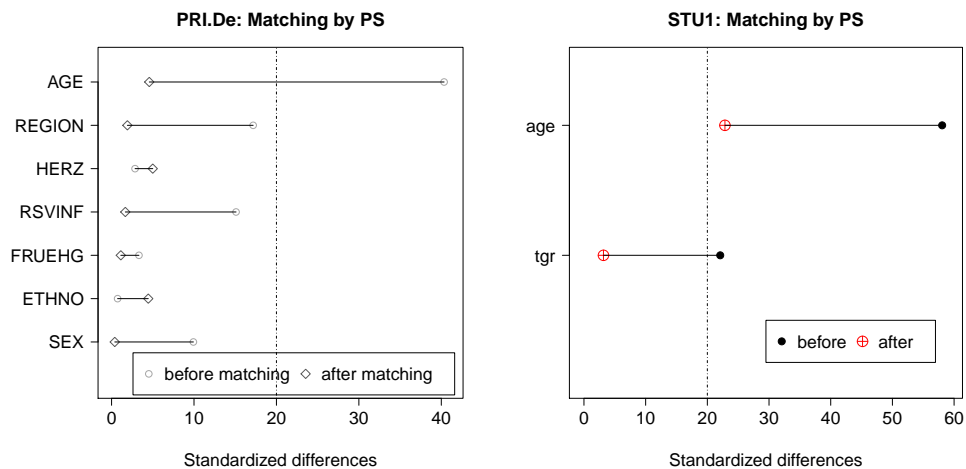
Figure 4: Illustration of standardized differences in original data and after matching in PRI.De (left) and STU1 data (right) using in `plot.stdf()` (program code for the left graphic is not shown)

```
> stu1.m2.bal

 Summary of balance check:

                   before: no balance (0) before: balance (1)
after: no balance (0)                   1                   0
after: balance (1)                      1                   0

 Covariates not completely tested: ---

 Detailed balance check (overall):

          tgr age
table.before   0   0
table.after    1   0

 Detailed balance check:
 [Standardized differences (cut point: 20)]

          tgr    age
Before   22.089 58.065
------      -----  -----
```

```
After       3.162 22.852


---------   ----   ----
Scale       bin    bin


> ## Illustration of standardized differences, Figure 4 (right)
>
> stu1.distplot <- plot.stdf(x     = stu1.m2.bal,
                             sel   = c("tgr", "age"),
                             main  = "STU1: Matching by PS",
                             pch.p = c(20,10))
```

Information about statistical tests applied or covariate types needed for correct calculation of standardized differences is also stored (`$bal.test$method`) in the output object. In addition, the significance level is available (`$bal.test$alpha`). It has to be interpreted as cutpoint at which the decision about the balance of a covariate distribution is made if standardized differences are calculated.

The check for the balance of covariate distributions entails the knowledge about the correctness of the PS model. If the PS model is correctly fitted, at least covariates included in PS model should be sufficiently balanced after stratification or matching. Otherwise re-modeling of the PS model should be considered.

# 5   Propensity score based treatment effects ... ps.estimate()

The use of the function `ps.estimate()` depends on the class of the input object. If functions `ps.makestrata()` or `ps.match()` are previously used, the arguments for the treatment (`treat`), the stratum (`stratum.index`) or the matching indices (`match.index`) are not needed, contrary to the case if the input object is of class *'data frame'*. In addition to PS based effect estimates, a regression model can be simultaneously fitted (argument `regr`). Results from regression models are estimates for conditional parameters which are automatically projected to estimates for corresponding marginal parameters. Both conditional and marginal parameter estimates are presented. Furthermore, additional adjustment for covariates which yet unbalanced in strata or in matched data can be done using argument `adj`.

There are two options to specify the arguments `adj` and `regr`. On the one hand, they can be specified as typical formulas whereas treatment has to be the first independent

variable. Furthermore, outcome and treatment must be the same as specified in arguments `resp` and `treat`. On the other hand, a vector of strings or numerics can be given which indicate covariates to be adjusted for in strata or in the matched sample.

If stratification based on PS is applied in data with continuous outcome, the marginal and the conditional treatment effect based on PS stratification coincide. It is estimated as a weighted sum of differences of the mean outcomes in treated and untreated individuals across PS strata [25]. Two different weighting schemes are available: on the one hand weights are equal to the proportion of individuals per stratum (`weights="rr"`) and on the other hand weights are related to the inverse variance of stratum-specific treatment effects (`weights="opt"`).

In a first example, the treatment effect on the quality of life (argument `resp='pst'` in `ps.estimate()`) in the STU1 data is estimated. Using the summary function, comprehensive result information are available. First, the crude, the stratification based and the regression based treatment effect is given as well as the corresponding standard errors and 95% confidence intervals. To estimate the regression based treatment effect, the regression formula is defined as 'pst~therapie+tgr+age' (argument `resp`) whereas only the covariates additional to treatment ('therapie') have to be specified.

```
> ## STU1: Effect estimation of therapy ('ther') on quality
> ## of life ('pst') based on PS stratification
>
> stu1.est <- ps.estimate(object  = stu1.str4,
+                         resp    = "pst",
+                         weights = "opt",
+                         regr    =pst~therapie+tgr+age)
```

Since the argument `adj` is not used in `ps.estimate()`, no further covariate adjustment is done within the PS strata. Therefore, the respective row in section 'Effect estimates' is empty and the information about 'Stratum-specific adjusted parameter estimates' is missing. Additional to the effect estimates, the stratum-specific treatment effects are listed as well as the stratum-specific weights used to estimate the (overall) treatment effect. As conditional and marginal parameter coincide in case of continuous outcome, parameter estimates are not explicitly labeled as marginal.

```
> ## PS stratification in STU1 data
> ## Summary for treatment effect estimation
>
> summary(stu1.est)
```

```
          Summary for effect estimation

 Treatment/exposure: therapie
 Outcome: pst
 Effect measure: differences ('effect')

 Effect estimates:

               effect  SE[effect]  [95%-CI[effect]]
               ------  ----------  ----------------
Crude           1.589       1.261   [-0.883,4.061]
Stratification
 Unadjusted     0.793      1.3068   [-1.768,3.354]
 Adjusted                                      [,]
Regression      0.788      1.2951    [-1.75,3.326]


 Stratum-specific parameter estimates:

                 S1      S2    S3     S4
               -----   ----- -----  -----
Unadjusted effect 3.453 -6.703 1.223 -7.001

 Stratum-specific adjusted parameter estimates:

 Stratum-specific weights:
0.47 0.13 0.34 0.06
```

In case of stratified data with binary outcome, both the stratified Mantel-Haenszel (MH) estimator and the estimator based on outcome rates from PS strata [21] are implemented. The marginal odds ratio for outcome describes the change in odds for the outcome, if everybody versus nobody were treated. It is different to the conditional odds ratio, e.g.~estimated by logistic regression (in combination with the assumption of constant individual odds ratios).

The popular stratified MH estimator (stratified by PS) can fail to estimate both the indivudual, conditional and the marginal odds ratio [23, 43, 44]. However, the stratified MH estimator is implemented since it is often used in the analysis of stratified data. The approach of outcome rates from PS strata is proposed by Graf et al.~[21]. PS methods are used to estimate marginal treatment effects, but only the outcome rates based estimator fulfills the criteria for an estimator for the marginal odds ratio. It is defined as an odds ratio of marginal outcome probabilities, contrary to the stratified MH estimator which is a

weighted sum of stratum-specific odds ratios [21, 23].

```
> ## PS stratification in PRI.De
> ## Effect estimation of exposure to RSV infection on the
> ## severity of LRTI
> pride.est <-
+     ps.estimate(object  = pride.str5,
+                 family  = "binomial",
+                 resp    = "SEVERE",
+                 treat   = "PCR_RSV",
+                 adj     = c("REGION", "ETHNO", "AGE"),
+                 weights = "rr")
```

The analysis of the effect of exposure to an RSV infection on the severity of LRTI in the PRI.De data is extended by additional covariate adjustment in the PS based strata (argument `adj`) since variables 'AGE', 'REGION' and 'ETHNO' are still unbalanced after stratification in at least one stratum. The summary and print functions give a slightly different and more comprehensive output in case of a binary outcome compared to a continuous outcome as shown for the STU1 data. If the outcome is binary, the standard error is given for the effect estimate on log-scale.

```
> summary(pride.est)

        Summary for effect estimation

 Treatment/exposure: PCR_RSV
 Outcome: SEVERE
 Effect measure: odds ratios ('or')

 Effect estimates:

                  or  SE[log[or]]  [95%-CI[or]]
                ----- -----------  ------------
Crude           1.677      0.0796  [1.435,1.96]
Stratification
  Outcome rates 1.362      0.0805 [1.163,1.595]
  MH            1.419      0.0823 [1.208,1.667]
  Adjusted      1.565      0.2013 [1.055,2.322]
Regression
  Conditional                              [,]
  Marginal                                 [,]
```

```
Stratum-specific parameter estimates:

                          S1     S2     S3     S4     S5
                         _____  _____  _____  _____  _____
  outcome rates 'p0'     0.44   0.53   0.55    0.6   0.66
  outcome rates 'p1'     0.48   0.57   0.58   0.71   0.81
  odds ratio             1.16   1.15   1.16   1.67   2.17

 Stratum-specific adjusted parameter estimates:
1.407 1.398 1.245 1.62 2.154
 Stratum-specific weights:
0.2 0.2 0.2 0.2 0.2
```

In the section 'Effect estimates', the crude and stratification based odds ratio estimators (plus standard errors and 95% confidence intervals) are given. Since argument `adj` is used in `ps.estimate`, an adjusted stratification based estimator is presented. It is a weighted mean of adjusted stratum-specific regression coefficients across the PS strata. Additionally, the stratified MH estimator is shown. Users have to be careful to interpret the adjusted stratified estimate as well as the stratified MH estimate. No regression based estimates are given since the argument `regr` is not specified in `ps.estimate()`. In general, both conditional and marginal parameter estimates are given since they may differ in case of binary outcome. Furthermore, estimated stratum-specific parameters are given: stratum-specific outcome rates per treatment group and stratum-specific odds ratios are presented in section 'Stratum-specific parameter estimates'. If `adj` is specified, estimated stratum-specific parameters are also given (section 'Stratum-specific adjusted parameter estimates').

All these comprehensive information is stored in the output object and this is managed as follows. Results from regression models are stored in value `$lr.estimation`. PS based results can be found in value `$ps.estimation`. This value is in turn divided in `$ps.estimation$unadj` and `$ps.estimation$adj` for unadjusted and adjusted analyses, respectively. In both values stratum-specific information are stored. Further values of the output object contain information about the outcome (`$name.resp`, `$resp`), the treatment (`$name.treat`, `$treat`) and the stratum indices (`$stratum.index`, `$name.stratum.index`). The output object inherits all values from the input object as well.

If matching is applied, the dependency structure of the matched sample has to be accounted for in the data analysis [45]-[47]. For it, generalized linear mixed models are appropriate and involved in the `nonrandom` package. Therefore, the function `lmer` implemented in

the `lme4` package is internally used whereas random intercepts for matching sets are modeled.

The data analysis of matched data can be done in the same way as for stratified data. The values of the output object in case of matched data differ slightly from the those based on the analysis of stratified data. There are naturally no stratum-specific information and no corresponding weights available, only an estimated overall unadjusted and adjusted treatment effect (if argument `adj` is specified) and corresponding standard errors and 95% confidence intervals. If the outcome is binary, the standard error is given for the effect estimate on log-scale.

```
> ## STU1, matching by PS, continuous outcome 'pst'
> ## Effect estimation of therapy on quality of life ('pst')
>
> stu1.est.m2 <- ps.estimate(object = stu1.m2,
+                            resp   = "pst")
> stu1.est.m2

 Effect estimation for treatment/exposure on outcome

 Treatment/exposure: therapie
 Outcome: pst
 Effect measure: difference ('effect')


 Table of effect estimates:

               effect    SE[effect]   [95%-CI[effect]]
               -----     -----------  ----------------
Crude          1.589         1.261    [-0.883,4.061]
Matching
  Unadjusted   0.873         1.3176   [-1.709,3.455]
  Adjusted                                     [,]
Regression                                     [,]
```

The output object contains information as described above for the data analysis based on stratified data except for stratum-specific information.

```
> ## PRI.De, matching on PS, binary outcome 'SEVERE'
> ## Effect estimation of exposure to RSV infection on the
> ## severity of LRTI
>
> pride.estimate.m <- ps.estimate(object = pride.m1,
```

```
+                                        resp = "SEVERE",
+                                        family = "binomial")
> pride.estimate.m

 Effect estimation for treatment/exposure on outcome

 Treatment/exposure: PCR_RSV
 Outcome: SEVERE
 Effect measure: odds ratio ('or')

 Table of effect estimates:

               or   SE[log[or]]   [95%-CI[or]]
               -----  -----------   ------------
Crude          1.677       0.0796  [1.435,1.96]
Matching
  Unadjusted   1.379       0.0916  [1.152,1.65]
  Adjusted                               [,]
Regression
  Conditional                            [,]
  Marginal                               [,]
```

Altogether, the `nonrandom` package offers PS based analyses in an easy way, however, suitable knowledge for adequaete interpretation of results is still needed. The estimation of treatment effects on linear and binary outcome is implemented, limited to the situation considering a binary treatment. It provides the experienced user a set of functions for an easy and flexible implementation of PS based analyses. Users who are not familiar with the application of such methods and the underlying theory are enabled to conduct an adequate PS based analysis guided by the package.

## References

[1] PR~Rosenbaum and DB~Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[2] M~Lunt, D~Solomon, K~Rothman, R~Glynn, K~Hyrich, DPM Symmons, and T~Stürmer. Different methods of balancing covariates leading to different effect estimates in the presence of effect modification. *American Journal of Epidemiology*, 169(7):909–917, 2009.

[3] PR~Rosenbaum. Model-based direct adjustment. *Journal of American Statistical Association*, 82:387–394, 1987.

[4] K~Hirano and GW~Imbens. Estimation of causal effects using propensity score weighting: An application to data on right heart catheterization. *Health Services & Outcomes Research Methodology*, 2:259–278, 2001.

[5] DA~Freedman and RA~Berk. Weighting regressions by propensity scores. *Evaluation Review*, 32:392–409, 2008.

[6] BR~Shah, A~Laupacis, JE~Hux, and PC~Austin. Propensity score methods gave similar results to traditional regression modeling in observational studies: A systematic review. *Journal of Clinical Epidemiology*, 58(6):550–559, 2005.

[7] T~Stürmer, M~Joshi, RJ~Glynn, J~Avorn, KJ~Rothman, and S~Schneeweiss. A review of the application of propensity score methods yielded increasing use, advantages in specific settings, but not substantially different estimates compared with conventional multivariable methods. *Journal of Clinical Epidemiology*, 59(5):437–461, 2006.

[8] PR~Rosenbaum and DB~Rubin. Reducing bias in observational studies using sub-classification on the propensity score. *Journal of American Statistical Association*, 79(387):516–524, 1984.

[9] WG~Cochran and DB~Rubin. Controlling bias in observational studies: a review. *Sankhya Series A*, (35):516–524, 1973.

[10] DB~Rubin and N~Thomas. Characterizing the effect of matching using linear propensity score methods with normal distributions. *Biometrika*, 79(4):797–809, 1992.

[11] P~Austin. Propensity-score matching in the cardiovascular surgery literature from 2004 to 2006: a systematic review and suggestions for improvement. *Journal of Thoracic and Cardiovascular Surgery*, 134:1128–1135, 2007.

[12] PC~Austin. A critical appraisal of propensity-score matching in the medical literature between 1996 and 2003. *Statistics in Medicine*, 27(12):2037–2049, 2008.

[13] JA~Nelder and RWM Wedderburn. Generalized linear models. *Jornal of Royal Statistical Society A*, 135(3):370–384, 1972.

[14] DR~Cox and EJ~Snell. *Analysis of binary data*. Chapman and Hall, London, second edition, 1989.

[15] PJ~Diggle, KY~Liang, and SL~Zeger. *Analysis of Longitudinal Data*. Oxford University Press, Oxford, 1994.

[16] JA~Hanley, A~Negassa, MD~deB. Edwardes, and JE~Forrester. Statistical analysis of correlated data using generalized estimating equations: An orientation. *American Journal of Epidemiology*, 157(4):364–375, 2003.

[17] AJ~Dobson and AG~Barnett. *Introduction to Generalized Linear Models*. Chapman and Hall, London, third edition, 2008.

[18] J~Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

[19] MA~Hernan. A definition of causal effect for epidemiological research. *Journal of American Statistical Association*, 58:265–271, 2003.

[20] MA~Hernan and JM~Robins. Estimating causal effects from epidemiological data. *Journal of Epidemiology and Community Health*, 60:578–586, 2006.

[21] E~Graf and M~Schumacher. Letter to the editor: Comments on ~the performance of different propensity score methods for estimating marginal odds ratios. *Statistics in Medicine*, 27(19):3915–3917, 2008.

[22] A~Forbes and S~Shortreed. Letter to the editor: Inverse probability weighted estimation of the marginal odds ratio: Correspondence regarding 'The performance of different propensity score methods for estimating marginal odds ratios'. *Statistics in Medicine*, 27(26):5556–5559, 2008.

[23] S~Stampf, E~Graf, C~Schmoor, and M~Schumacher. Estimators and confidence intervals for the marginal odds ratio using logistic regression and propensity score stratification. *Statistics in Medicine*, 29(7-8):760–769, 2010.

[24] H.~F. Rauschecker, R.~Sauer, A.~Schauer, M.~Schumacher, M.~Olschewski, W.~Sauerbrei, M.~H. Seegenschmiedt, and C.~Schmoor. Therapy of small breast cancer – four–year results of a prospective non–randomized study. *Breast Cancer Research and Treatment*, 34:1–13, 1995.

[25] Stephen Senn, Erika Graf, and Angelika Caputo. Stratification for the propensity score compared with linear regression techniques to assess the effect of treatment or exposure. *Statistics in Medicine*, 26(30):5529–5544, 2007.

[26] J~Forster, G~Ihorst, CH~Rieger, V~Stephan, HD~Frank, H~Gurth, R~Berner, A~Rohwedder, H~Werchau, M~Schumacher, T~Tsai, and G~Petersen. Prospective population-based study of viral lower respiratory tract infections in children under 3 years of age (the pri.de study). *European Journal of Pediatrics*, 163(12):709–716, 2004.

[27] C~Drake. Effects of misspecification on the propensity score on estimatiors of treatment effects. *Biometrics*, 49(4):1231–1236, 1993.

[28] Katherine Huppler~Hullsiek and Thomas~A. Louis. Propensity score modeling strategies for the causal analysis of observational data. *Biostatistics*, 2(4):179–193, 2002.

[29] S~Weitzen, KL~Lapane, AY~Toledano, AL~Hume, and V~Mor. Principles for modelling propensity scores in medical research. *Pharmacoepidemiology and Drug Safety*, 13(12):841–853, 2004.

[30] Alan~M. Brookhart, Sebastian Schneeweiss, Kenneth~J. Rothman, Robert~J. Glynn, Jer~ry Avorn, and Til Stürmer. Variable selection for propensity score models. *American Journal of Epidemiology*, 141(12):1–8, 2006.

[31] DB~Rubin. The design versus the analysis of observational studies for causal effects: Parallels with the design of randomized trials. *Statistics in Medicine*, 26:20–36, 2007.

[32] PR~Rosenbaum. Observational studies. *Springer Verlag*, New York, 1995.

[33] MM~Joffe and PR~Rosenbaum. Invited commentary: Propensity scores. *American Journal of Epidemiology*, 150:327–333, 1999.

[34] WG~Cochran. The effectiveness of adjustment by subclassification in removing bias in observational studies. *Biometrics*, 24:295–313, 1968.

[35] P~Austin. Some methods of propensity-score matching had superior performance to others: Results of an empirical investigation and monte carlo simulations. *Biometrical Journal*, 51(1):1–14, 2009.

[36] PC~Austin. Assessing balance in measured baseline covariates when using many-to-one matching on the propensity score. *Pharmacoepidemiology and drug safety*, 17(12):1218–1225, 2008.

[37] BB~Hansen. Commentary: The essential role of balance tests in propensity-matched observational studies: Comments on 'a critical appraisal of propensity-score matching in the medical literature between 1996 and 2003' by peter austin, statistics in medicine. *Statistics in Medicine*, 27(12):2050–2054, 2008.

[38] PC~Austin. The realibility of different propensity score methods to balance measured covariates between treated and untreated subjects in observational studies. *Medical decision making*, doi:10.1177/0272989X09341755, 2008.

[39] J~Hill. Commentary: Discussion of research using propensity-score matching: Comments on 'a critical appraisal of propensity-score matching in the medical literature between 1996 and 2003' by peter austin, statistics in medicine. *Statistics in Medicine*, 27(12):2055–2061, 2008.

[40] EA~Stuart. Commentary: Developing practical recommendations for the use of propensity scores: Discussion of 'a critical appraisal of propensity score matching in the medical literature between 1996 and 2003' by peter austin, statistics in medicine. *Statistics in Medicine*, 27(12):2062–2065, 2008.

[41] PC~Austin. Rejoinder: Discussion of 'a critical appraisal of propensity-score matching in the medical literature between 1996 and 2003'. *Statistics in Medicine*, 27(12):2066–2069, 2008.

[42] D~Yang and JE~Dalton. A unified approach to measuring the effect size between two groups using sas. *SAS Global Forum 2012: Statistics and Data Analysis*, (Paper 335-2012), 2012.

[43] C~Schmoor, C~Gall, S~Stampf, and E~Graf. Correction for confounding bias in non-randomized studies by appropriate weighting. *Biometrical Journal*, 53:1–19, 2011.

[44] S~Greenland. Interpretation and choice of effect measures in epidemiologic analyses. *American Journal of Epidemiology*, 125:761–768, 1987.

[45] NE~Breslow and NE~Day. *Statistical Methods in Cancer Research, Volume 1 - The Analysis of Case-Control Studies*. International Agency for Research on Cancer (IARC Scientific Publications No. 32), Lyon, 1980.

[46] KJ~Rothman, S~Greenalnd, and TL~Lash. *Modern epidemiology*. Lippincott Williams & Wilkins, Philadelphia, third edition, 2008.

[47] A~Agresti and Y~Min. Effects and non-effects of paired identical observations in comparing proportions with binary matched-pairs data. *Statistics in Medicine*, 23:65–75, 2004.