Propensity score based data analysis using nonrandom

Susanne Stampf

2011-01-05

Abstract

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

The package `nonrandom` is a tool for a comprehensive data analysis using stratification and matching by the propensity score. Several functions are implemented, starting from the selection of the propensity score model up to estimating propensity score based treatment (exposure) effects. Before estimating the propensity score, function `relative.effect()` permits to investigate the extent to which a covariate is confounding the treatment (exposure) effect on outcome. This measure may support the decision which covariates should be involved to estimate the propensity score. The function `pscore()` estimates the propensity score by fitting a logisitc model. The function `pscore.plot()` visualize the distribution of the estimated propensity score in treatment (exposure) groups. Stratification and matching by the propensity score are implemented in functions `ps.makestrata()` and `ps.match()`, respectively. To check covariate balance between treatment or exposure groups, the function `ps.balance()` applies statistical tests and standardized differences, respectively, to detect covariate differences in groups. In addition, functions dist.plot() and `stdf.plot()` provides a graphical balance check. Finally, propensity score based estimators for the treatment (exposure) effect can be determined by functionn ps.estimate(). It also offers a comparison to regression based estimates alternatively used.

All functions can be applied separately as well as combined. Additionally, it is possible to apply all functions repeatedly to decide which analysis strategy is the most suitable one. Print and summary functions are available for the most implemented functions. There are two 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 data from observational studies. The PS is defined as conditional probability of receiving a certain treatment[1] given covariates (1). In general, the PS is unknown and has to be appropriately estimated, e.g., by fitting a regression model. The selection of an adequate PS model is often the first obstacle. Lunt et al.~proposed a measure estimating the extent to which a covariate is confounding the treatment effect on outcome (2). Covariates with a large extent are potential candidates for the inclusion in the PS model. This proposal is implemented in the `nonrandom`-package.

In observational studies, covariate distributions differ generally between treatment groups and PS methods aim to eliminate such 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 a direct comparison of treatment groups is meaningful. Covariate adjustment by PS is also a favourite approach since it is easily used as traditional regression modeling including PS in the regression model in addition to treatment. 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 stratify individuals with similar or even equal PS such that distributions of measured covariates are sufficiently balanced in treatment groups within each stratum defined by PS (1; 8). It can be supposed that each stratum mimics a randomized situation within which distributions of measured covariates are balanced in expectation. If the assumption of 'SITA' holds (see below), stratum-specific parameters can be unbiasedly estimated (1) and those estimates are then summed up across strata using appropriate weights to estimate the marginal 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 matched sets have similar or even equal PS whereas the similarity is often defined by a 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 total matched sample is often not accounted for the estimation of the interesting parameter (10)-(12). Approaches such as generalized linear mixed models or generalized estimation

[1] In the following, we only use the phrase '... conditional probability of receiving a certain treatment', i.e., we concentrate on the comparison of outcome in treated and untreated individuals. The comparison of two treatments, e.g., new and standard therapy are also possible. The PS can be also be the conditional probability of being exposed given covariates, respectively, such that the comparison of outcome for exposed and unexposed individuals is of interest.

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. The observed outcome is $Y = ZY_1 + (1-Z)Y_0$, and the expected values of counterfactuals $\mathbf{E}[Y_1]$ and $\mathbf{E}[Y_0]$ can be derived if the identifying assumption called 'strongly ignorable treatment assignment' (SITA) holds (1). This assumption states, that, within subgroups 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 $Z = 0$. This means, individuals in treatment groups are comparable and the measured treatment difference between groups can be attributed to treatment itself.

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 estimating the marginal odds ratio of outcome, i.e., the change in odds of outcome, if everybody versus nobody were treated (21)-(23). Therefore, marginal probabilities for potential outcomes $\mathbf{P}[Y_1 = 0]$ and $\mathbf{P}[Y_1 = 1]$ have to be estimated which are used to construct to 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, marginal probabilities of potential outcomes can be estimated by outcome rates from PS strata or derived from standard regression results (21).

In the following, the application of the `nonrandom` package is demonstrated step by step introducing all implemented functions. The usage is illustrated by the exemplary analysis 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, respectively, 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 aged less than three years in the observational study Pri.DE (Pediatric Respiratory Infection, Deutschland) in Germany (26). Here, the impact of the exposure to an infection with the respiratory syncytial virus (RSV) on the severity of LRTI is investigated (23).

2 The estimation of the propensity score

The PS is generally unknown and has to be estimated often done by fitting an appropriate regression model. The selection of such a PS model is mostly a delicate issue (27)-(31). A measure describing the extent to which a covariate is confounding the treatment effect on 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 function relative.effect(). If an appropriate PS model is selected, function pscore() then offers the estimation of PS by fitting a logistic model.

2.1 Selection of the propensity score model: relative.effect()

An important step is to decide which covariates X_k, $k = 1, ..., K$, measured at baseline should be included in the PS model. The measure describing the extent to which a covariate X_k is confounding the effect of treatment Z on outcome Y is defined as relative effect (per cent)

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

with unadjusted treatment effect β_z and treatment effect β_{z,x_k} adjusted for covariate X_k, $k = 1, \ldots, K$. In case of binary outcome, the relative effect (per cent) is defined as

$$\left(\frac{exp\{\beta_{z,x_k}\} - exp\{\beta_z\}}{exp\{\beta_z\}}\right) \times 100.$$

Therefore, $K + 1$ regression models for outcome Y, both unadjusted and adjusted for covariates X_k, $k = 1, ..., K$, are fitted using generalized linear regression models with respect to the measuring scale of outcome (internal use of glm()). There are two options for model specification. On the one hand, it is possbile to set separately outcome, treatment and covariates as strings or numerics using arguments resp, treat and sel.

```
> ## 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")
```

On the other hand, an explicit regression formula using formula can be specified:

```
> ## STU1 data on quality of life
> load(stu1)
> stu1.effect <- relative.effect(data = stu1,
+                                formula = pst~therapie+tgr+age)
```

Information about relative effects as well as corresponding unadjusted and adjusted treatment effects on outcome are available. In the STU1 data example, two covariates are investigated. Both seem to affect the treatment effect on outcome and should be involved in the estimation of PS.

```
> 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
```

2.2 Estimation of the propensity score: `pscore()`

If an appropriate model for PS is selected, it can be estimated using function `pscore()`:

```
> 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")
```

A logistic regression model is internally fitted using function `glm()`. The argument `name.pscore` offers to specify the label for the estimated propensity score with which the estimated propensity score is in turn stored in data. In the PRI.De data example, the default setting *pscore* is modified to *ps*. The output object is of class *pscore* and it contains a list including comprehensive information about the PS model, e.g., names of the estimated PS (`$name.pscore`), treatment (`$name.treat`) and outcome (`$name.resp`) as well as the specified formula of the PS model (`$formula.pscore`). Furthermore, the complete data set (`$data`) extended by the estimated PS, the estimated PS itself (`$pscore`) and treatment variable (`$treat`) are separately available.

In the new released version of `nonrandom` (version 1.1), it is possible to visualize the distribution of the estimated PS in treatment groups (`pscore.plot()`). Therefore, the density of the estimated PS is internally estimated in treatment groups using function `density`. However, the previous use of function `pscore()` is needed since the estimated PS sourced from `$pscore` of the input object is needed:

```
> pscore.plot(object=pride.ps,
+             main="PRI.De study: Density estimation of estimated PS",
+             with.legend=TRUE,
+             cex.main=1.6, cex.axis=1.4, legend.cex=1.25, cex.lab=1.5,
+             par.1=list(lty=1, lwd=2), par.0=list(lty=3, lwd=2),
+             xlab="",
+             ylim=c(0,4.5))
```

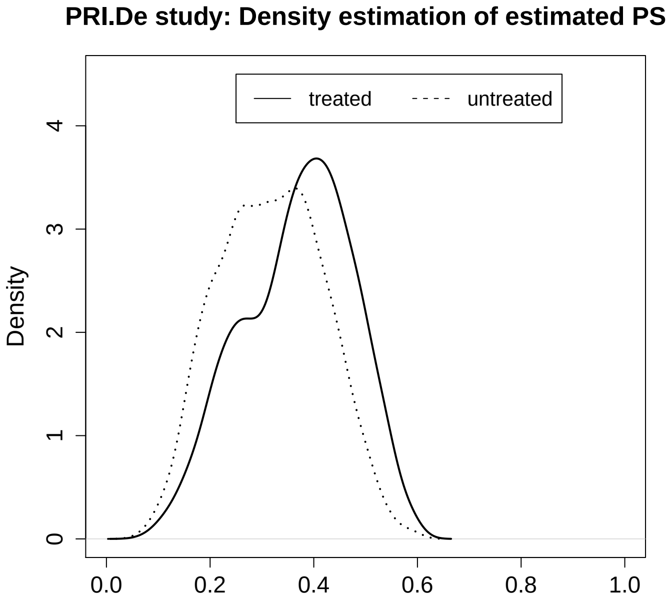

Figure 1: Density estimation of the estimated PS in PRI.De data using function `pscore.plot`

3 Propensity score based methods

Observational studies frequently exhibit imbalances in covariate distributions between treatment groups. Stratification and matching by PS are used to eliminate these imbalances by creating data situation as in randomized experiments.

3.1 Stratification by the propensity score: `ps.makestrata()`

Stratification by PS groups individuals with similar or even identical PS. Several ways for stratification (argument *breaks*) are implemented in function `ps.makestrata()` whereas it has been shown that stratification using quintiles of the PS distribution yields a ninety per cent bias reduction (8; 34).

The usage of `ps.makestrata()` depends on the class of the input object whereas two object classes are allowed: *data.frame* and *pscore* (if `pscore()` is previously used) are permitted. No specification of the stratification variable (argument `stratified.by`) is needed if the input object is of class *pscore*. The estimated PS stored in `$pscore` is automatically sourced. This in contrast to the case where the input object is of class *data frame*.

Strata bounds can be set either by defining a fixed number of strata, using pre-defined strata bounds or applying appropriate `R` functions. However, the default is *NULL*, i.e., the stratification variable is factorized and each factor corresponds to one stratum. This is only meaningful when the stratification variable has only few values.

```
> stu1.str4 <- ps.makestrata(object = stu1.ps)
> stu1.str4

 Stratified by:  pscore

 Strata information:

    Strata bounds     n      n (per cent)
1           0.601   231             35.8
2           0.709    65             10.1
3           0.824   255             39.5
4           0.883    95             14.7
```

In case of the STU1 data, the PS is estimated using function `pscore()` and stored in value `$pscore` of the outpt object `stu1.ps`. It is chosen as stratification variable. Since the estimated PS consists only four values, the arguments `breaks` does not need to be specified. Using print or summary functions, the name of the used stratification variable is autmoatically given, e.g., `'pscore'` in case of STU1 data stratified by PS.

If an integer is given in argument `breaks`, the number of strata with respect to the stratification variable is specified:

```
> pride.str.b5 <- ps.makestrata(object = pride.ps, breaks = 5)
> pride.str.b5
```

```
 Stratified by:  ps

 Strata information:

   Strata bounds     n     n (per cent)
1  [0.0619,0.168]   175             5.7
2   (0.168,0.275]   796            25.9
3   (0.275,0.382]  1015              33
4   (0.382,0.488]   866            28.1
5   (0.488,0.595]   226             7.3
```

If a numeric vector is given or an appropriate R-function is used, e.g., quantile(), these values explicitly indicate the stratum bounds. In case of PRI.De data, quintiles from the distribution of the estimated PS are used for stratification by the estimated PS. The argument name.stratum.index specifies the name of the variable including the generated stratum indices which is in turn stored in data.

```
> pride.str5 <- ps.makestrata(object = pride.ps,
+                             breaks = quantile(pride.ps$pscore,
+                                               seq(0,1,0.2)),
+                             name.stratum.index = "stratum")
> pride.str5

 Stratified by:  ps

 Strata information:

   Strata bounds     n     n (per cent)
1  [0.0624,0.236]   616              20
2   (0.236,0.306]   615              20
3   (0.306,0.369]   616              20
4   (0.369,0.431]   615              20
5   (0.431,0.594]   616              20
```

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 values from the input object. Similar to function pscore(), the complete data set ($data) extended by stratum indices labeled by $name.stratum.index is available for the output object. Furthermore, the name of the stratification variable ($stratified.by), individual stratum indices ($stratum.index) generated at least as well as the corresponding stratum bounds ($intervals) are stored in the output object. Stratification of data can be done repeatedly, but only information from the last application is stored separately in values of 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 similar or identical estimated PS wheres the similarity is determined by a caliper size with a pre-defined maximum width of one-fifth of the standard deviation of the logit of the estimated PS (9).

Similar to the function `ps.makestrata()`, the use of `ps.match()` depends both on classes (*data.frame* and *pscore*) and numbers of input objects (one or two data frames). No specification of the matching variable (`matched.by`) is needed if the input object is of class *pscore*. The estimated PS stored in the input object (`$pscore`) is automatically sourced.

```
> pride.m1 <- ps.match(object = pride.ps,
+                      ratio = 1, x = 0.2, caliper = "logit",
+                      matched.by = "ps", setseed = 38902)
> 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 matched obs:              2062
Number of matched treated obs:      1031
Number of matched untreated obs:    1031
Number of dropped obs:              1016
Number of matching sets:            1031
Number of incomplete matching sets:    0
```

In case of one or two data frames as input objects, the matching (`matched.by`) as well as the treatment variable (argument `treat`) are needed. If the first data frame only contains data from treated (or untreated) individuals, a second input object (argument `object.control`) comprising data from untreated (or treated) individuals is necessary. The indicated matching variable must be the same in both input data frames. If it differs, the matching variable in the second data frame (argument `control.matched.by`) must be specified. Independent of classes and numbers of input objects, the value of treatment indicating *treated* individuals must be given (argument `who.treated`, default is '1').

```
> 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.
> stu1.m2

 Matched by:  pscore

 Matching parameter:

Caliper size:    0.5
Ratio:           2.0
Who is treated?: 1.0

 Matching information:

Untreated to treated?: FALSE
Best match?:            TRUE

 Matching data:

Number of matched obs:              501
Number of matched treated obs:      334
Number of matched untreated obs:    167
Number of dropped obs:              145
Number of matching sets:            167
Number of incomplete matching sets:   0
```

There are further parameters to specify the matching procedure: the matching ratio (`ratio`) indicating how many individuals should be matched and a statement concerning who should be matched to whom (`givenTmatchingC=TRUE`: treated to untreated individuals). The argument `bestmatch.first` indicating whether matching partners should be taken randomly from the pool of potential matching partners (FALSE) or those with the most similar estimated PS (TRUE; default). Furthermore, a random number can be specified (setseed) to make the matching procedure reproducible. The caliper size can be defined using arguments caliper and x. The default setting for the caliper size is one-fifth (x=2) of the standard deviation of the logit of estimated PS (caliper="logit"). However, it is possible to specify numerics in argument caliper (argument x is then disregarded) or to modify argument x in connection with caliper="logit".

As demonstrated in PRI.De data, one untreated individual is matched to a treated individual (ratio=1) and the default caliper size is used. The matching variable is *ps* indicating the estimated PS stored in data of the input object pride.ps. It is here identical to the value $pscore of the input object. The matching algorithm for the STU1 data is switched such that two treated individuals are matched to one untreated individual (ratio=2, givenTmatchingC=FALSE) since fewer untreated than treated individuals are available. Furthermore, the caliper size is set to 0.5.

The function ps.match() may return three different types of classes for output object(s): *matched.pscore*, *matched.data.frame* or *matched.data.frames*. This depends on class(es) and numbers of input object(s) and on the specification of argument combine.output. This argument is reasonable if there are two data frames as input objects. The default is combine.output=TRUE such that both input data frames extended by matching information are combined for output. The complete data set ($data) and the data set limited to matched individuals are stored in the output object ($data.matched). Both data sets are extended by a column including 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 used at last ($match.parameter) are stored. If there are two input objects and argument combine.output is set to FALSE, values $data, $data.matched and $match.index are lists of data frames and vectors corresponding to input objects, respectively. If the class of the input object is *pscore*, the output object also inherits all values from the input object.

4 Balance check for covariates

PS methods are used to eliminate imbalances in covariate distributions between treatment groups. An important, but often neglected issue is the balance check of covariate distributions after stratification or matching by PS. Graphics, statistical tests and standardized differences can be used to examine covariate distributions (35)-(37).

4.1 Grahical balance check: `dist.plot()`

The function `dist.plot()` offers an illustration of covariate distributions in treatment groups. As before, the use of function `dist.plot()` depends on the class of the input object. Therefore, arguments `treat`, `stratum.index` and `match.index` do not be specficied if the input object results from a previous application of `ps.makestrata()` or `ps.match()`. This is in contrast to the case where the input object is a *data frame*.

If input object is of class *stratified.data.frame* or *stratified.pscore*, covariate distributions are plotted separately by treatment and strata. If the class of the input object is either *matched.data.frame*, *matched.data.frames* or *matched.pscore*, covariate distributions are illustrated per treatment group in the matched data. If covariate distributions in strata or matched data should be compared to distributions in original data, i.e., in data before stratification or matching, argument `compare` has to be set to `TRUE`.

```
> ## Figure 2 (left)
> stu1.plot1 <- dist.plot(object = stu1.m2, sel = c("tgr"),
+                         plot.type = 1, compare = TRUE,
+                         bar.cex = 1.2, legend.cex=1.5, sub.cex=1.2,
+                         label.match = c("original data","matched sample"),
+                         col=c("gray65", "gray35"))
>
> ## Figure 2 (right)
> stu1.plot2 <- dist.plot(object = stu1.m2, sel = c("age"),
+                         plot.type = 2, compare = TRUE,
+                         bar.cex = 1.2, legend.cex=1.2, sub.cex=1.2,
+                         legend.title = "Therapy",
+                         col=c("gray65", "gray35"))
```

There are two different plot types (argument `plot.type`) which act depending on the measuring scale of covariates. The selected covariates (argument `sel`) are classified in categorical and non-categorical (i.e., numerical) covariates. This classification is done by argument `cat.level`. The default is 10, i.e., covariates with more than 10 different values are considered as non-categorical. The default of argument `plot.type` is 1, i.e., bar plots are used to show frequencies for categorical and means for numerical covariates.

If argument `plot.type` is set to 2, covariate distributions are illustrated by means of histograms. The argument `plot.levels` specifies here the number of cutpoints needed to define histogram classes for non-categorical covariates. When the covariate is categorical, the number of its categories are used to define cutpoints. However, the use of `plot.levels` still depends on the covariate structure to be plotted such that the used number of classes may differ from its specification.

There are three further useful arguments. Argument `with.legend=TRUE` (default) includes a legend in plots. If `plot.type=1`, category labels (if covariate is categorical) or treatment

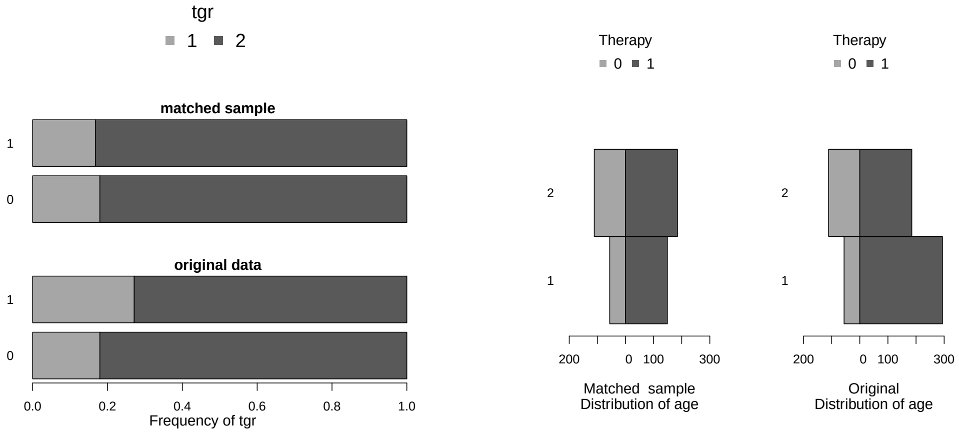

Figure 2: Frequencies of categorical covariate 'tgr' indicating tumor size (1: $\leq$ 10mm, 2: >10mm) of patients in STU1 data before and after matching; different types of plots in function `dist.plot()` are used: `plot.type=1` (left) and `plot.type=2` (right)

labels (in case of non-categorical covariates) are given in the legend. Therefore, users have to be careful to modify this argument when categorical and non-categorical covariates are simultaneously plotted. If `plot.type` is set to 2, treatment labels are always shown in the legend independent of the covariate type. The arguments `label.stratum` and `label.match` permit modification of labels within plots. Defaults are *Original* and *Stratum* and *Matched*, respectively. Further arguments can be used to modify, among others, font sizes, labels, colors and plot margins as illustrated for STU1 data (Figure 2) and PRI.De data (Figure 3).

```
> ## Figure 3 (left)
> pride.plot1 <- dist.plot(object = pride.str5,
+                          sel = c("AGE"), plot.type = 1
+                          bar.cex = 1.2, sub.cex = 1.3, legend.cex = 1.2,
+                          col=c("gray65", "gray35"))
>
> ## Figure 3 (right)
> pride.plot2 <- dist.plot(object = pride.m1, sel = c("AGE"),
+                          plot.type = 2, compare = TRUE,
+                          legend.title = "RSV infection",
+                          legend.cex = 1.2, bar.cex = 1.2, sub.cex = 1.2,
+                          col=c("gray65", "gray35"))
```

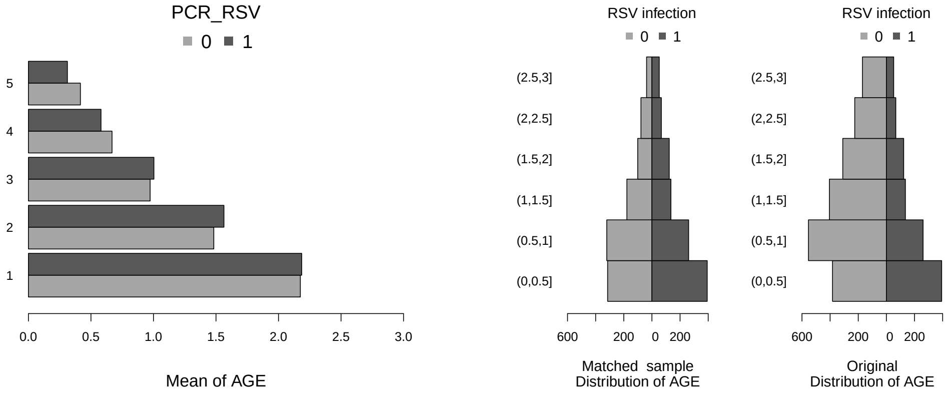

Figure 3: Illustration of function `distplot()` using different `plot.types`: (Left) Means of covariate 'AGE' in treatment groups in five PS strata of PRI.De data; (Right) Histograms of covariate 'AGE' in PRI.De data per treatment group before and after matching

Numerous values from plotted data using function `dist.plot()` are stored as lists in the output object, among others, names of categorical (`$var.cat`) and non-categorical covariates (`$var.noncat`), frequencies (`$frequency`) or means (`$means`) of covariates separated by treatment. The length and the manner of list entries depend on the type of selected covariates and the chosen plot type. For example, the value `frequency` of output object `stu1.plot1` is a list with length equal to the number of plotted categorical covariates. The list entries contain frequencies (scaled to 1) of different values (1 and 2) of covariate `tgr` per treatment group (0 and 1) in original data and in the matched sample.

```
> ## STU1 data matched by PS
> ## Means of covariate 'tgr' in PS strata; plotted in Figure 2 (left)
> stu1.plot1$var.cat
[1] "tgr"
>
> stu1.plot1$frequency
[[1]]
, , index = original data

   treat
            0         1
  1 0.1796407 0.2713987
  2 0.8203593 0.7286013
```

```
, , index = matched sample

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

If argument `plot.type` is set to 2, frequencies of covariates in histogram classes are separately stored for both treatment groups. Furthermore, cutpoints of histograms for non-categorical covariates are stored in the output object (`$breaks.noncat`).

```
> ## STU1 data matched by PS
> ## Frequencies for values of covariate 'age' in original and matched
> ## sample; plotted in Figure 2 (right)
>
> stu1.plot2$var.cat
[1] "age"
>
> ## Frequencies from categories of covariate 'age' in treatment group 0
> ## (Therapy=0) in original data (1st column) and in the matched
> ## sample (2nd column)
> stu1.plot2$x.s.cat
[[1]]
   index
      1   2
  1  56  56
  2 111 111
> ## Frequencies from categories of covariate 'age' in treatment group 1
> ## (Therapy=1) in original data (1st column) and in the matched
> ## sample (2nd column)
> stu1.plot2$y.s.cat
[[1]]
   index
      1   2
  1 294 149
  2 185 185
```

The specification `plot.type=1` in `dist.plot()` for non-categorical covariates results in visualization of means of covariates per treatment group. There are stored in lists of the output object:

```
> ## PRI.De data stratified by quintiles of the
> ## distribution of estimated PS
> ## Means of covariate'AGE' per treatment and strata, Figure 3 (left)
> pride.plot1$var.noncat
[1] "AGE"
>
> ## Means in PS strata (columns) in exposure groups (1st row: no
> ## infection '0', 2nd row: RSV infection '1')
> pride.plot1$mean
[[1]]
         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
```

If histograms are used to plot the distribution of non-categorical covariates (`plot.type=2`), list entries are in turn lists:

```
> ## PRI.De data matched by PS
> ## Figure 3 (right)
> pride.plot2$var.noncat
[1] "AGE"
>
> ## Frequencies per histogram class in group 'RSV infection = 0' for
> ## original data (1st list) and in matched sample (2nd list)
> pride.plot2$x.s.noncat
[[1]]
[[1]]$`1`
[1] 383 554 405 310 225 170

[[1]]$`2`
[1] 315 321 178 101  78  38
>
> ## Frequencies per histogram class in group 'RSV infection = 1' for
> ## original data (1st list) and in matched sample (2nd list)
> pride.plot2$y.s.noncat
[[1]]
[[1]]$`1`
[1] 393 261 135 123  67  52

[[1]]$`2`
[1] 393 261 135 123  67  52
```

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

4.2 Statistical tests and standardized differences: `ps.balance()`

Graphical illustration of covariate distributions in treatment groups may give a first insight in covariate differences between treatment groups. However, statistical tests decide whether differences between groups are significant. The function `ps.balance()` permits the application of statistical tests and the computation of standardized differences. There is an ongoing discussion about the appropriateness of statistical tests for balance decision such that standardized differences are recommended for balance decision especially in matched data (38)-(40).

Similar to functions described above, the usage of function `ps.balance()` depends on the class of the input object. If information about treatment, stratum or matching indices is stored in the input object, it is not necessary to specify them since they are automatically sourced from the input object. If the input object is a *data frame*, the corresponding arguments have to be given. By default, statistical tests (argument `method="classical"`) are used with respect to the measuring scale of selected covariates. That means, the t-test and the χ^2- test for non-categorical and categorical covariates (internal use of `t.test()` and `chisq.test()`) are applied whereas the argument `cat.levels` defines whether a covariate is categorical or non-categorical (see function `dist.plot()`). The tests are applied to data both before and after the balancing procedure (stratification or matching). A table summarizing balance decisions for all tested covariates is given (`Summary of balance check`). Furthermore, the balance decision for each selected covariate is shown as well as more detailed information about test results.

```
> ## Balance check for stratified PRI.De data using statistical tests
> pride.str5.bal <- ps.balance(object = pride.str5, sel = c(2:8),
+                              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 0.766  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  0.757 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
```

In case of the PRI.De data, p-values from statistical tests applied to data before and after stratification are given. Covariates, for which tests are not applicable or standardized differences are not computable, are not contained in the summary balance table, but listed separately. It concerns the covariate 'RSVINF' (describing whether there was a former infection to RSV) since it can be tested in the fourth and fifth stratum.

If the argmuent `method="stand.diff"`, standardized differences are calculated before and after the balancing procedure. The table for detailed balance information then contains standardized differences instead of p-values:

```
> ## Balance check for matched STU1 data using standardized differences
> stu1.m2.bal <- ps.balance(object = stu1.m2,
+                           sel = c("tgr","age"),
+                           method = "stand.diff",
+                           alpha =  20)
> 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
```

The shown information of covariate balance for selected covariates is stored in value `$bal.test` of the output object. When standardized differences are calculated (argument `method="stand.diff"`), means and standard deviations (SD) for each covariate per treatment group are additionally stored:

```
> ## Means for covariates 'tgr' and 'age' before (1st row) and after
> ## matching (2nd row) in treatment group 1
> stu1.balance$bal.test$Means.treat.1
           tgr       age
[1,] 0.7286013 0.3862213
[2,] 0.8323353 0.5538922
>
> ## Standard deviations for covariates 'tgr' and 'age' before (1st row)
> ## and after matching (2nd row) in treatment group 1
> stu1.balance$bal.test$SDs.treat.1
           tgr       age
[1,] 0.4446813 0.4868823
[2,] 0.3735682 0.4970871
```

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.

In the new released version 1.1 of nonrandom it is possible to visualize standardized differences of selected covariates before and after matching (Figure 4).

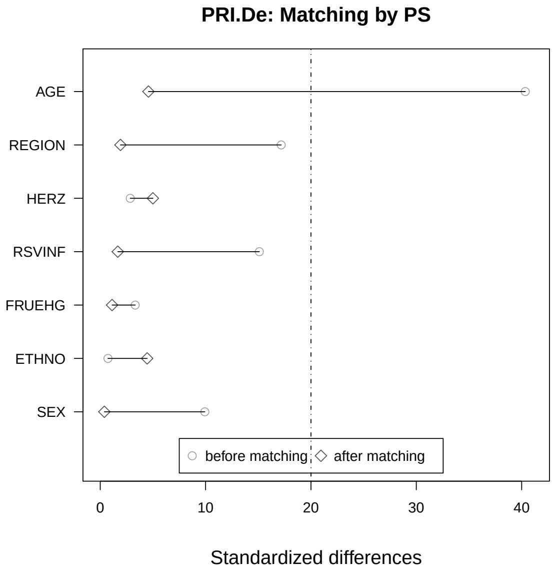

Figure 4: Illustration of standardized differences of several covariates in PRI.De data before and after matching by the estimated PS using function stdf.plot()

```
> ## Plot standardized differences of selected covariates
> ## in (matched) PRI.De data
> pride.m1.bal <- ps.balance(object = pride.m1, sel = c(2:8),
+                            method = "stand.diff", alpha = 20)
>
> ## Figure 4
> stdf.plot(objct = pride.m1.bal,
+           main = "PRI.De: Matching by PS",
+           sub = "Standardized differences",
+           las = 1, cex.main = 1.4, cex.sub = 1.3,
+           legend.label=c("before matching","after matching"),
+           legend.xy=c(7.5,1.5),
+           mymar=c(6,6,4,2), col.p=c("gray65", "gray35"))
```

The check for 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 the stratification or matching. Otherwise remodeling of the PS model should be considered.

5 Propensity score based treatment effects: `ps.estimate()`

The estimation of PS based treatment effect differs in the previous application of PS method and in the measuring scale of outcome. Therefore, the description of function `ps.estimate()` is in the following separated by PS methods directly applicable in the `nonrandom` package.

In general, the use of function `ps.estimate()` depends on the class of the input object. If function `ps.makestrata()` or `ps.match()` are previously used, arguments for treatment (`treat`), stratum (`stratum.index`) or matching indices (`match.index`) are not needed, contrary to the case if the input object is a *data frame*. In addition to PS based effect estimates, it is possible to fit a regression model (argument `regr`) for the issue of comparison. The resulting estimates are those for conditional parameters in regression models. They are automatically projected to estimates for corresponding marginal parameters. Furthermore, additional adjustment for still imbalanced covariates in stratified or matched data can be done (argument `adj`).

5.1 Effect estimator based on stratification by PS

If stratification is applied in data with continuous outcome, the marginal treatment effect based on PS is estimated as a weighted sum of differences of the mean outcomes in treated and untreated individuals across PS strata. Therefore, 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"`).

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

         Summary for effect estimation
```

```
 Treatment/exposure: therapie
 Outcome: pst
 Effect measure: effects

 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
```

The summary function for `ps.estimate()` offers an overview about all estimated effects, standard errors and confidence intervals. In the example of stratified STU1 data, no further adjustment in PS strata is specified such that the respective row in the result table remains empty. Furthermore, stratum-specific parameter estimates and weights are given.

In case of stratified data with binary outcome, the estimator based on outcome rates from PS strata (21) is implemented in function `ps.estimate()` to estimate the marginal odds ratio as effect measure. The marginal odds ratio for outcome describes the change in odds for outcome, if everybody versus nobody were treated. It is different to the conditional odds ratio, e.g., estimated by logistic regression (with the assumption of constant individual odds ratios). The popular stratified MH estimator stratified by PS can fail to estimate both the indivdiual, conditional and the marginal odds ratio (23; 41). However, the stratified MH estimator is implemented since 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 of 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).

```
> ## PRI.De: Effect estimation of exposure to RSV infection on the
> ## severity of LRTI based on data stratified by PS
> pride.estimate <- ps.estimate(object = pride.str5,
+                               family = "binomial",
+                               resp = "SEVERE",
+                               treat = "PCR_RSV",
+                               adj = c("REGION", "ETHNO", "AGE"),
+                               regr = SEVERE˜PCR_RSV+SEX+ETHNO+FRUEHG +
+                                      HERZ+ELTATOP+REGION+AGE+KRANKSUM+
+                                      TOBACCO+VOLLSTIL+EXT+EINZ,
+                               weights = "rr")
> summary(pride.estimate)

         Summary for effect estimation

 Treatment/exposure: PCR_RSV
 Outcome: SEVERE
 Effect measure: odds ratios

 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    1.515       0.0904 [1.269,1.809]
  Marginal       1.399       0.0691 [1.222,1.602]

 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
```

There are two options to specify 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 original data and in each stratum, respectively. The used formulas for those fitted regression models are stored in the output object. Corresponding stratum-specific adjusted treatment effects and weights are also stored in the output object.

Information stored in the output object is managed as follows. Results from regression modeling is 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 outcome (`$name.resp`, `$resp`), treatment (`$name.treat`, `$treat`) and stratum indices (`$name.stratum.index`, `$stratum.index`). The output object inherits all values from the input object as well.

5.2 Effect estimator based on matching by PS

If matching is applied, the dependency structure of the matched sample has to be accounted for in the data analysis (42)-(44). 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 set 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, but only an estimated overall unadjusted and adjusted treatment effects (if argument `adj` is specified) and corresponding standard errors and confidence intervals. If the outcome is binary, the standard error is given for the effect estimate on log-scale.

```
> ## STU1: Effect estimation of therapy on quality of life ('pst')
> ## based on PS matching
> stu1.estimate.m2 <- ps.estimate(object = stu1.m2,
+                                 resp = "pst")
> stu1.estimate.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 print function again provides an overview about effect estimates, corresponding standard errors and confidence intervals. The output object contains the same information as above when the data analysis is based on stratified data unless stratum-specific information.

```
> ## PRI.De: Effect estimation of exposure to RSV infection on the
> ## severity of LRTI based on data matched by PS
> 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 subclassification 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. *Biometrike*, 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*, in press, 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 estimators 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] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

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

[42] 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.

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

[44] 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.