stratamatch: Prognostic Score Stratification Using a Pilot Design

by Rachael C. Aikens, Joseph Rigdon, Justin Lee, Michael Baiocchi, Andrew B. Goldstone, Peter Chiu, Y. Joseph Woo, and Jonathan H. Chen

Abstract Optimal propensity score matching has emerged as one of the most ubiquitous approaches for causal inference studies on observational data. However, outstanding critiques of the statistical properties of propensity score matching have cast doubt on the statistical efficiency of this technique, and the poor scalability of optimal matching to large data sets makes this approach inconvenient if not infeasible for sample sizes that are increasingly commonplace in modern observational data. The **stratamatch** package provides implementation support and diagnostics for 'stratified matching designs,' an approach that addresses both of these issues with optimal propensity score matching for large-sample observational studies. First, stratifying the data enables more computationally efficient matching of large data sets. Second, **stratamatch** implements a 'pilot design' approach in order to stratify by a prognostic score, which may increase the precision of the effect estimate and increase power in sensitivity analyses of unmeasured confounding.

1 Introduction

To make causal inference from observational data, researchers must address concerns that effect estimates may be biased due to confounding factors – baseline characteristics of the individuals in the study that influence both their selection of treatment and their probable outcome. Matching methods seek to account for this self-selection by grouping treated and control individuals with similar baseline characteristics. One of the most common such methods, propensity score matching, pairs individuals who appear to have had similar probabilities of receiving the treatment according to their baseline characteristics (Rosenbaum and Rubin, 1983), with the goal of coercing the data set into a form that resembles a fully-randomized controlled trial (King and Nielsen, 2019; Rosenbaum et al., 2010; Hernán and Robins, 2016). However, propensity score matching can only address bias due to *measured* baseline covariates, necessitating sensitivity analyses to interrogate the potential of bias due to *unmeasured* confounding (Rosenbaum, 2005b; Rosenbaum et al., 2010).

In their provocative article "Why Propensity Should Not Be Used for Matching," King and Nielson argue that the fully randomized controlled trial – the design emulated by propensity score matching – is less statistically efficient than the block-randomized controlled experiment (King and Nielsen, 2019). In block-randomized designs, individuals are stratified by prognostically important covariates (e.g., for a clinical trial: sex, age group, smoking status) *prior* to randomization in order to reduce the heterogeneity between the treatment and control groups. In the experimental context, these efforts to reduce heterogeneity between compared groups help to increase the precision of the treatment effect estimate. In observational settings, reducing this type of heterogeneity not only improves precision but increases the robustness of the study's conclusions to being explained away by the possibility of unobserved confounding (Rosenbaum, 2005a; Aikens et al., 2020). The stratified matching design – in which observations are stratified prior to matching within strata – attempts to emulate the block-randomized controlled trial design in the observational context in order to secure these statistical benefits over pure propensity score matching. In addition, since the computation required for optimal matching can be quite time-consuming for studies of more than a few thousand observations, stratified matching designs could greatly improve the scalability of optimal matching. While the worst-case computational complexity of optimal matching is unfavorable, the process of matching a stratified data set with a constant stratum size scales much more effectively with sample size (for a summary of empirical run-times, see section 2.5.3).

While a variety of packages in R (R Core Team, 2019) exist for matching subjects in observational studies, limited support exists for researchers seeking to implement a stratified matching design. The popular **MatchIt** package (Ho et al., 2011) is a user-friendly option for common propensity score matching designs and related approaches, **optmatch** (Hansen and Klopfer, 2006) and **DOS2** (Rosenbaum, 2019) are a powerful combination for implementing a variety of more complicated optimal matching schemes, and **nearfar** (Rigdon et al., 2018) implements a different form of matching for the instrumental variable study. The primary goal of **stratamatch** is to make stratified matching and prognostic score designs accessible to a wider variety of applied researchers and to suggest a suite of diagnostic tools for the stratified observational study. In favorable settings, these designs could not only increase the precision and robustness of inference but could facilitate the optimal matching of sample sizes for which this technique was previously computationally impractical.

This paper discusses the methodological contributions of **stratamatch** – in particular, the implementation of a novel pilot design approach suggested by Aikens et al. (2020) (section 2.2) – and summarizes the package implementation (Section 2.3) with illustrative examples (Section 2.4). While **stratamatch** may substantially improve the scalability of optimal matching for some large data sets, the main objective of the package is not to implement a computationally complex task but to make sophisticated study design tools and concepts accessible to a wide variety of researchers.

2 Study design

A prognostic score stratification pilot design

Stratifying a data set based on baseline variation prior to matching reduces the heterogeneity between matched sets with respect to that baseline variation. But what baseline characteristics should be used? One option is to select prognostically important covariates by hand, based on expert knowledge. However, in practice, this "manual" stratification process often produces strata that vary wildly in size and composition. Some strata may be so small or so imbalanced in their composition of treated and control individuals that it is difficult to find high-quality matches, or many observations are thrown away. Other strata may be so large that matching within them is still computationally infeasible.

The `auto_stratify` function in **stratamatch** divides subjects into strata using a *prognostic score* (see Hansen (2008)), which summarizes the baseline variation most important to the outcome. In addition to producing strata of more regular size and composition, balancing treatment and control groups based on the prognostic score may confer several statistical benefits: increasing precision (Aikens et al., 2020; Leacy and Stuart, 2014), providing some protection against mis-specification of the propensity score (Leacy and Stuart, 2014; Antonelli et al., 2018), and decreasing the susceptibility of an observed effect to being explained away by unobserved confounding (Rosenbaum and Rubin, 1983; Aikens et al., 2020). However, fitting the prognostic score on the same data set raises concerns of overfitting and may lead to biased effect estimates (Hansen, 2008; Abadie et al., 2018). For this reason, (Aikens et al., 2020) suggest using a *pilot design* for estimating the prognostic score.

Central to the pilot design concept is maintaining separation between the design and analysis phases of a study (see table 1, or for more information Goodman et al. (2017) and Rubin (2008)). Using an observational pilot design, the researchers partition their data set into an *analysis set* and a held-aside *pilot set*. Outcome information in the pilot set can be observed (e.g. to fit a prognostic score), and the information gained can be used to inform the study design. Subsequently, in order to preserve the separation of the study design from the study analysis, the individuals from the pilot set are omitted from the main analysis (i.e., they are not reused in the analysis set). The primary insight of the pilot design is that reserving all of the observations in a study for the analysis phase (i.e., in the analysis set) is not always better. Rather, clever use of data in the design phase (i.e., in the pilot set) may facilitate the design of stronger studies.

In the **stratamatch** approach, a random subsample of controls is extracted as a pilot set to fit a prognostic model, and that model is then used to estimate prognostic scores on the mix of control and treated individuals in the analysis set. The observations in the analysis set can then be stratified based on the quantiles of the estimated prognostic score and matched by propensity score or Mahalanobis distance within strata (see section 2.3).

When to use this approach

Aikens et al. (2020) describe the scenarios in which a prognostic score matching pilot design is most useful. Briefly, the **stratamatch** approach is best for large data sets (i.e., thousands to millions of observations), especially when the number of control observations is plentiful. This technique may be particularly useful when modeling a prognostic score with the measured covariates is straightforward, and when propensity score alone is likely to exclude certain aspects of variation highly associated with outcome but unassociated with treatment assignment. While computational gains vary, stratification tends to noticeably accelerate matching for sample sizes of 5,000 or more (see section 2.5.3).

Conversely, this technique is not recommended for small data sets in which each control observation is precious, especially when prognostic scores are likely to be difficult to estimate from the measured covariates (see Aikens et al. (2020) for a lengthy discussion). Ideally, there should be ample control observations available to fit a usable prognostic model and still leave sufficient controls remaining to select high-quality matches for the treated individuals in the data set. While some **stratamatch** designs may be useful for the estimation of other causal estimands, the statistical properties of prognostic pilot designs for estimands other than the average treatment effect among the treated have not yet been characterized (Aikens et al., 2020).

Term	Description
Design phase	Phase of a study in which the researcher considers what kinds of data will provide the strongest information to address the question at hand (e.g., randomization, sampling, matching, inverse probability weighting). The goal of the design phase is to obtain data that will provide strong inference.
Analysis phase	Phase of a study in which the data that comes from the design phase are summarized into statistics. Inference and sensitivity analyses are performed.
Pilot Design	An observational study approach in which some data is spent in the design phase to improve the study design/preprocessing.
Pilot Set	A subset of data extracted to be used in the design phase.
Analysis set	The set of data reserved for inference in the analysis phase.
Propensity score	Probability of assignment to the treatment group based on measured baseline characteristics.
Prognostic score	Expectation of the outcome in the absence of treatment based on measured baseline characteristics.
Prognostic model	A model (e.g., logistic regression) used to estimate prognostic scores.
Stratum	A subset of observations in the analysis set to be matched together.

Table 1: Summary of relevant methodological terms as they apply to **stratamatch**.

3 Software

The **stratamatch** function, `auto_stratify`, implements the prognostic score stratification in the pilot design described above. The most basic procedure does the following:

1. Partition the data set into a pilot data set and an analysis data set
2. Fit a model for the prognostic score from the observations in the pilot set
3. Estimate prognostic scores for the analysis set using the prognostic model
4. Stratify the analysis set based on prognostic score quantiles.

A call to `auto_stratify` produces an `auto_strata` object, which contains the analysis set, the pilot set, and other information about the strata and prognostic scores. The **stratamatch** package implements a set of diagnostic plots and tables that can be used to assess the quality of a stratification. Example code, output, and diagnostics are provided in section 2.4. If the strata are satisfactory, the treatment and control individuals within each stratum can then be matched. By default, the `strata_match` function performs $1 : 1$ propensity score matching within each stratum. Other matching scheme possibilities are discussed in section 2.5.3).

4 Illustrations

Simulated example

This section demonstrates the basic functionality of **stratamatch** in simulated example. The function `make_sample_data` generates a simple simulated data set so that users can explore the design options implemented by **stratamatch**. Below, we generate a sample of 10,000 observations and print the first few rows as an illustration.

```
library("stratamatch")
library("dplyr")
dat <- make_sample_data(n = 10000)
head(dat)
```

```
           X1        X2 B1 B2 C1 treat outcome
1  0.93332697 1.0728339  1  0  a     1       0
2 -0.52503178 0.3449057  1  1  b     0       1
3  1.81443979 1.0361942  1  1  a     0       0
4  0.08304562 0.3017060  1  1  a     0       1
```

```
5  0.39571880 0.5397257  0  0  c     0       0
6 -2.19366962 1.4523274  1  1  b     0       1
```

The user should suppose that the rows of dat are individuals in an observational study, and the objective of the study is to estimate the effect of a binary treatment assignment (treat) on a binary outcome (outcome). Columns 1-5 represent three types of measured baseline covariates: continuous (X1 and X2), binary (B1 and B2), and categorical (C1). For this example, we assume strongly ignorable treatment assignment - that is, roughly, there are no unmeasured confounding factors (Rosenbaum and Rubin, 1983). (For sensitivity analyses for this assumption, see, for example, Rosenbaum (2005b)).

Automatic stratification

The command below uses auto_stratify to (1) partition 10% of the controls in dat into the pilot set (2) fit a prognostic score model for outcome based on X1 and X2, (3) estimate prognostic scores on the analysis set, and (4) return to us the analysis set, divided into strata of approximately 500 individuals, based on prognostic score quantiles. All of these steps are completed automatically with this function call, and the results are returned to us as a.strat.

```
a.strat <- auto_stratify(dat, treat = "treat", prognosis = outcome ~ X1 + X2,
+    pilot_fraction = 0.1, size = 500)
```

```
Constructing a pilot set by subsampling 10% of controls.
Fitting prognostic model via logistic regression: outcome ~ X1 + X2
```

The result returned by auto_stratify is an auto_strata object. Running print on this object supplies basic information about how the stratification process has been completed.

```
print(a.strat)
```

```
auto_strata object from package stratamatch.

Function call:
auto_stratify(data = dat, treat = "treat", prognosis = outcome ~
    X1 + X2, size = 500, pilot_fraction = 0.1)

Analysis set dimensions: 9234 X 8

Pilot set dimensions: 766 X 7

Prognostic Score Formula:
outcome ~ X1 + X2
```

Here, auto_stratify has partitioned away a pilot set of 766 control individuals to fit our desired prognostic model, leaving 9,234 individuals in the analysis set. Using the prognostic model, prognostic scores were estimated on the individuals in the analysis set, and these individuals were divided into strata with a target size of 500. In order to record these stratification assignments, an eighth column, stratum, has been appended to the analysis set. The number strata and range of strata sizes can be obtained from summary(a.strat).

The analysis set and pilot set are accessible via a.strat$analysis_set and a.strat$pilot_set, respectively. The strata_table (accessed via a.strat$strata_table) reports the strata sizes and the prognostic score quantile bins that define each stratum.

Diagnostics

A major focus of the **stratamatch** package is suggesting diagnostics for the quality of stratification in observational studies. The issue_table reports the total size and composition of each stratum.

```
a.strat$issue_table
```

```
# A tibble: 19 x 6
   Stratum Treat Control Total Control_Proportion Potential_Issues
     <int> <int>   <int> <int>              <dbl> <chr>
 1       1   167     319   486              0.656 none
 2       2   149     337   486              0.693 none
 3       3  160     326   486              0.671 none
 4       4  132     354   486              0.728 none
 5       5  123     363   486              0.747 none
 6       6  122     364   486              0.749 none
 7       7  146     340   486              0.700 none
 8       8  109     377   486              0.776 none
 9       9  131     355   486              0.730 none
10      10  132     354   486              0.728 none
11      11  111     375   486              0.772 none
12      12  108     378   486              0.778 none
13      13  112     374   486              0.770 none
14      14  122     364   486              0.749 none
15      15  100     386   486              0.794 none
16      16  109     377   486              0.776 none
17      17  114     372   486              0.765 none
18      18  107     379   486              0.780 none
19      19   85     401   486              0.825 Small treat:control ratio
```

The 'Potential_Issues' column is meant to quickly flag strata that may be problematically large, small, or imbalanced in the ratio of treated and control samples. The "small treat:control ratio" flag for stratum 19 indicates that the proportion of treated individuals is 0.2 or lower[1]. This is a relatively common issue, which is often easily addressed (see section 2.6).

The **stratamatch** package implements four diagnostic plotting options:

1. **Size-Ratio Plot:** (Figure 1) Displays each stratum in the analysis set based on its size and the percentage of control observations in order to identify potentially problematic strata.
2. **Propensity Score Histogram:** (Figure 1) Displays the distribution of estimated propensity scores across the treatment and control groups within a single stratum or the entire analysis set. These plots are used for assessing propensity score overlap.
3. **Assignment-Control Plot:** (Figure 2) Displays each individual based on estimated propensity score and estimated prognostic score, based on visualizations from Aikens et al. (2020). As above, these plots can display a single stratum or the entire analysis set. Assignment-control plots are useful for checking the overlap and correlation of prognostic and propensity scores.
4. **Residual Plots:** (Not shown) Show the diagnostic plots for the prognostic model used to estimate the prognostic scores. It is essentially a wrapper for `plot.lm` (see the documentation for `plot.lm` in the base R package, **stats**). Note that since the pilot set alone is used to fit the prognostic model, only the pilot set is used for these diagnostic plots.

The code below makes each of the plot types listed above, including two assignment-control plots: one for the entire analysis set and one for a single stratum. The results are shown in figures 1 and 2, with interpretation in the figure captions. For propensity score histograms and assignment-control plots, the 'propensity' argument is required, specifying how the propensity scores should be estimated. Below, the propensity score is fit on the analysis set based on a regression of treatment assignment on 'X1', 'X2', 'B1', and 'B2' (for other input options, run `help(plot.auto_strata)` or `help(plot.manual_strata)`).

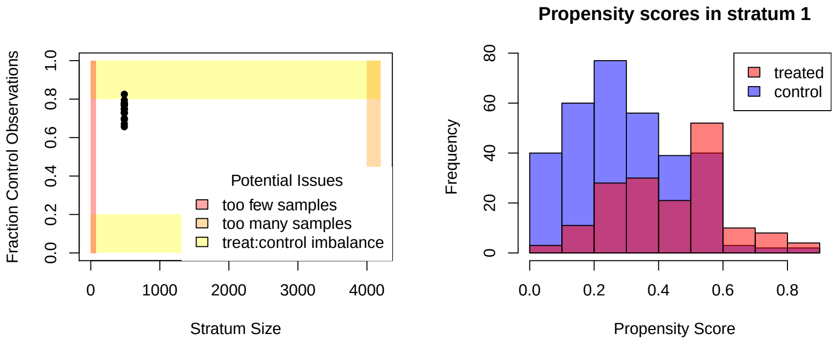
```
plot(a.strat, type = "SR")
plot(a.strat, type = "hist", propensity = treat ~ X2 + X1 + B1 + B2, stratum = 1)
plot(a.strat, type = "AC", propensity = treat ~ X2 + X1 + B1 + B2)
plot(a.strat, type = "AC", propensity = treat ~ X2 + X1 + B1 + B2, stratum = 1)
plot(a.strat, type = "residual")
```

In this example, the command `a.strat$prognostic_model` would supply the prognostic model (an `lm` or `glm` object) for further diagnostics (e.g., with `summary(a.strat$prognostic_model)`). Assessment of the prognostic model can indicate whether a sufficient number of observations has been partitioned into the pilot set (see section 2.6). However, one benefit of a stratified matching design is that even an imperfect prognostic model may yield robust inference if the resulting strata are of sufficient quality to allow for a strong propensity match (see, for example, theory on stratified sampling (Lohr, 2019) or commentary on doubly robust matching (Leacy and Stuart, 2014; Antonelli et al., 2018))

Matching

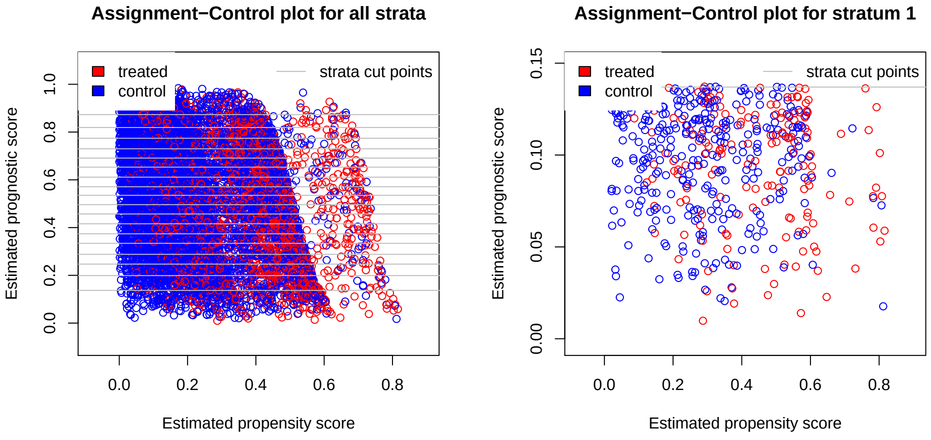
Once the data have been stratified, the user can optimally match individuals within each stratum. The `strata_match` function supports optimal 1:1, 1:k, or full matching (Rosenbaum, 1991; Hansen and

[1] Note that the specific thresholds defining the potential issue flags (e.g., 20% treated individuals or fewer) are not universal cutoffs but guidelines meant to draw researchers' attention to possible irregularities.



(a) A size ratio plot **(b)** A propensity score histogram

Figure 1: (A) A size-ratio plot, with each point representing a stratum. Yellow regions: treated to control ratio is imbalanced. Orange: strata size is large enough that matching may be computationally time-consuming. Red: strata are small enough that match quality may be poor. In a perfectly ideal stratification, all strata would fall within the white rectangle. In practice, some stratification issues are common and easily addressed; see section 2.6. (B) A histogram of estimated propensity scores for a selected stratum. In an ideal scenario, there is ample overlap between treated and control individuals within each stratum.



(a) An assignment-control plot across all strata **(b)** An assignment-control plot for stratum 1

Figure 2: Assignment-control plots (Aikens et al., 2020) showing estimated propensity score versus estimated prognostic score for each subject in the analysis set (A) or a selected stratum (B). In an ideal scenario, there is ample overlap between treated and control individuals in terms of both prognosis and propensity (for other cases, see section 2.6). Grey lines denote the prognostic score thresholds defining the strata.

Klopfer, 2006), based on a propensity score or Mahalanobis distance. The sample code below performs 1:1 propensity score matching. This function makes essential use of the **optmatch** package (Hansen and Klopfer, 2006; Bertsekas and Tseng, 1988) to perform the matching within strata.

```
mymatch <- strata_match(a.strat, model = treat ~ X1 + X2 + B1 + B2)

Fitting propensity model: treat ~ X1 + X2 + B1 + B2
```

The result is an optimal 1 to 1 matching within prognostic score strata. Above, `mymatch` is an `optmatch` class object, as described by the **optmatch** package (Hansen and Klopfer, 2006). For the most part, `mymatch` can be treated as a factor giving match assignments for each row of the data set. The command `summary(mymatch)` would display the number of pairs, the number of unmatched individuals, and the effective sample size. For suggestions regarding other matching schemes for stratified data, see 2.5.3.

A brief comment on estimation

The procedure for performing inference after matching – in particular the estimation of the standard error of the effect estimate for the purposes of hypothesis tests and confidence intervals – is a topic of some debate in the literature. We will not attempt to resolve this debate here, although interested readers may find the commentary by Stuart (Stuart, 2010) to be an accessible starting place, and note more recent work by Abadie and Speiss (Abadie and Spiess, 2021), and numerous other authors (for example Abadie and Imbens (2006, 2011); Austin and Small (2014); Austin and Cafri (2020)). Below, we describe two contrasting approaches that are most familiar to epidemiology and statistics, with references to coding resources.

First, Rosenbuam (Rosenbaum, 2005b; Rosenbaum et al., 2010) motivates the use of permutation-based tests followed by sensitivity analyses for unobserved confounding. Both of these are implemented in **sensitivitymw** (Rosenbaum, 2014, 2015) for pairmatching and **sensitivityfull** (Rosenbaum, 2017, 2007) for full matching. In keeping with a randomization inference framework, these techniques generally consider inference conditional on the matched sample and focus on uncertainty derived from the randomization process emulated by the matching. Researchers with further information on the sampling process that generated the observational data may thereafter combine this approach with a sampling variation framework to estimate parameters and standard errors for a more general target population (see the framework outlined by Tipton (2013) for the experimental setting).

A second common approach uses covariate adjustment. This framework is motivated importantly by Ho et al. (2007), who make the case for matching as a preprocessing step to reduce the dependence of parametric analyses on model selection. In keeping with the regression literature from the social sciences, these approaches often begin by supposing that the complete observational data set is an independent and identically distributed set of observations from some larger population, perhaps according to some parametric data-generating model. Within this framework, there is still considerable debate regarding correct standard error estimation. A thorough practical tutorial for the covariate adjustment approach with code examples and some suggestions for standard error estimation is featured in the recent **MatchIt** vignette, "Estimating Effects after Matching" (Greifer, 2020). Note that the pilot design implemented by **stratamatch** removes control individuals at random from the data set while retaining all treated individuals. Thus, while we recommend stratamatch for the estimation of the average treatment effect among the treated, the characteristics of stratamatch designs for estimation of other causal estimands (e.g., average treatment effect) have not yet been well characterized.

Real-data example: Life sustaining treatments for critical care patients

As an applied example, the **stratamatch** package contains a re-processed version of de-identified medical data from Chavez et al. (2018). Briefly, the authors extracted demographic information, common laboratory test results, comorbidity information, and treatment team assignments for 10,157 ICU patients from the Stanford University Hospital who met their inclusion criteria. During their stay, each patient's critical care preferences are summarized with a code status. The default – Full Code status – indicates no limitations on resuscitative measures, while other codes (e.g. 'Do not resuscitate', or 'DNR') indicate different limitations on the intensity and type of resuscitation the patient should receive if they become pulseless or apneic (i.e., their heart stops or they stop breathing). This code status is a product of complex dynamics between patient and provider. When a patient's code status does not reflect their goals of care, patients may have life-sustaining care inappropriately withheld, or they may receive aggressive treatment that does not effectively increase their quality or quantity of remaining life.

In this example, suppose a researcher wants to study whether comparable patients under the care of surgical teams vs. non-surgical teams are more likely to have their code status set to limit resuscitation (i.e., any form of 'DNR'). From this, we could infer tendencies that different treatment teams have in counseling and decision-making about life-sustaining treatments for the critically ill. However, the patient groups seen by surgical vs. non-surgical teams are necessarily different, because patients are assigned to treatment teams based on their reason for being in the hospital and their treatment history. Thus, a naive comparison of DNR order frequency between care team types would be misleading. To better account for these potential differences, we employ a stratified pilot matching design to compare "treated" (assigned to a surgical care team) individuals with "control" (assigned to a non-surgical care team) ones that are similar in terms of their prognostic and propensity scores.

Automatic stratification

Patients must be first stratified by a prognostic score (i.e., their estimated probability of receiving a DNR order if they are not assigned to a surgical care team) before being matched on a propensity score (i.e., their estimated probability of assignment to a surgical care team). In the example below, we use `auto_stratify` on the `ICU_data` to (1) partition 10% of controls into a pilot set, (2) build a prognostic score model on that pilot set based on age ('`Birth.preTimeDays`'), sex, and race/ethnicity (3) estimate prognostic scores on the analysis set and (4) return a stratified data set with approximately 500 individuals per stratum.

```
ICU_astrat <- auto_stratify(data = ICU_data, treat = "surgicalTeam",
    prognosis = DNR ~ Birth.preTimeDays + Female.pre + RaceAsian.pre +
      RaceUnknown.pre + RaceOther.pre + RacePacificIslander.pre +
      RaceBlack.pre + RaceNativeAmerican.pre + all_latinos,
    pilot_fraction = 0.1, size = 500)
```

```
Constructing a pilot set by subsampling 10% of controls.
Fitting prognostic model via logistic regression: DNR ~ Birth.preTimeDays +
    Female.pre + RaceAsian.pre + RaceUnknown.pre +  RaceOther.pre +
    RaceBlack.pre + RacePacificIslander.pre + RaceNativeAmerican.pre +
    all_latinos
```

Next, we print the results.

```
print(ICU_astrat)
```

```
auto_strata object from package stratamatch.

Function call:
auto_stratify(data = ICU_data, treat = "surgicalTeam",
    prognosis = DNR ~ Birth.preTimeDays + Female.pre + RaceAsian.pre +
        RaceUnknown.pre + RaceOther.pre + RaceBlack.pre +
        RacePacificIslander.pre + RaceNativeAmerican.pre + all_latinos,
        size = 500, pilot_fraction = 0.1)

Analysis set dimensions: 9364 X 14

Pilot set dimensions: 793 X 13

Prognostic Score Formula:
DNR ~ Birth.preTimeDays + Female.pre + RaceAsian.pre + RaceUnknown.pre +
    RaceOther.pre + RaceBlack.pre + RacePacificIslander.pre +
    RaceNativeAmerican.pre + all_latinos
```

```
summary(ICU_astrat)
```

```
Number of strata: 19

        Min size: 492        Max size: 494

Strata with Potential Issues: 2
```

We see here that `auto_stratify` partitioned the data into a pilot set of 793 "controls" (i.e., patients not assigned to a surgical treatment team) and an analysis set of the 9,364 remaining individuals. The prognostic model was fit on the pilot set according to the formula we provided, regressing DNR code assignment on age, sex, and race. This model was used to estimate the prognostic score (probability of DNR code assignment based on demographics) for each of the 9,364 individuals in the analysis set. Finally, each individual in the analysis set was assigned to a stratum based on this score. 19 strata, each containing between 492 and 494 patients, were created. This stratum assignment information was appended to the analysis set by adding a new 14th column, stratum.

Manual stratification

Rather than using a pilot design to build a prognostic score, researchers may wish to stratify the data set based on discrete covariates (e.g., chosen by a domain expert). The manual_stratify function supports these study designs. For example, the code below bins the 10,157 patients in the data set purely based on race/ethnicity and sex. In contrast, the size-ratio plots for the automatic stratification show a much smaller range of sizes and control proportions, with fewer – and more easily addressed – potential issues.

```
ICU_mstrat <- manual_stratify(data = ICU_data,
    strata_formula = surgicalTeam ~ Female.pre + RaceAsian.pre +
      RaceUnknown.pre + RaceOther.pre + RaceBlack.pre +
      RacePacificIslander.pre + RaceNativeAmerican.pre + all_latinos)
summary(ICU_mstrat)
```

```
Number of strata: 16

        Min size: 17         Max size: 3314

Strata with Potential Issues: 9
```

The resulting manual_strata object has many of the same properties as an auto_strata object from auto_stratify and can be matched in the same way with strata_match. However, manual_strata objects do not have a pilot set prognostic score information, and accordingly assignment-control and residual plots are not supported for these inputs.

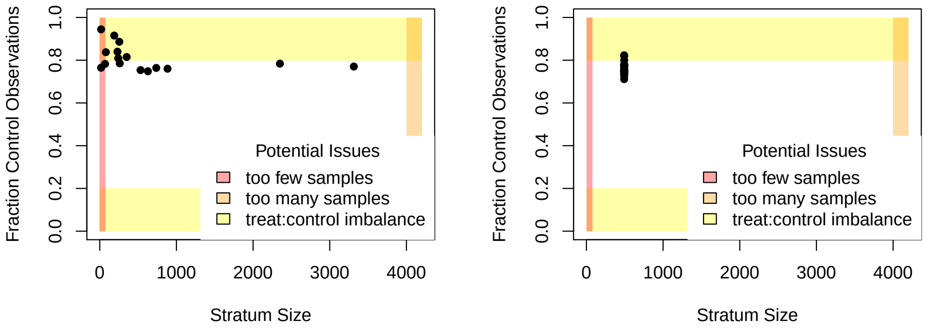
This more traditional manual approach may be preferred in some cases for its simplicity and because it obviates the need to sacrifice observations to fit a prognostic model. However, selecting a binning scheme that results in favorable strata may be a time-consuming iterative process, as highlighted by the diagnostics in the following section. These issues underscore the potential usefulness of the prognostic score stratification implemented by auto_stratify.

Diagnostics

Size-ratio plots for the manual and automatic stratification illustrate a common issue with manual stratification: it is often difficult to select discrete covariates that result in appropriately sized and balanced strata (Figure 3). This also is reflected by the number of strata with potential issues in the manual stratification issue table below. For example, stratum 1 below (white males) contains 3,314 patients, while stratum 3 (Native American males) contains only 18 patients, only one of whom was assigned to a surgical team. In exceedingly large strata, matching becomes increasingly computationally intensive, while in exceedingly small and/or highly imbalanced strata, finding high-quality matches can be difficult or infeasible (see section 2.5.3).

```
 ICU_mstrat$issue_table
```

```
# A tibble: 16 x 6
   Stratum Treat Control Total Control_Proportion Potential_Issues
     <int> <int>   <int> <int>              <dbl> <chr>
 1       1   761    2553  3314              0.770 none
 2       2   212     672   884              0.760 none
 3       3     1      17    18              0.944 Too few samples; Small treat:con...
 4       4    13      67    80              0.838 Small treat:control ratio
 5       5    56     205   261              0.785 none
 6       6    65     286   351              0.815 Small treat:control ratio
 7       7    29     226   255              0.886 Small treat:control ratio
 8       8   174     563   737              0.764 none
 9       9   508    1842  2350              0.784 none
10      10   158     470   628              0.748 none
```



(a) Size-ratio plot for manual stratification **(b)** Size-ratio plot for automatic stratification

Figure 3: Size-ratio plots for (A) manual stratification on sex and race/ethnicity and (B) automatic stratifications of the same data set of ICU patients. Manual stratification often results in highly variable size and treat:control balance between strata, as reflected by the number of strata points in the shaded zones.

```
11       11     4      13    17              0.765 Too few samples
12       12    15      54    69              0.783 Too few samples
13       13    37     194   231              0.840 Small treat:control ratio
14       14    46     195   241              0.809 Small treat:control ratio
15       15    16     173   189              0.915 Small treat:control ratio
16       16   131     401   532              0.754 none
```

The code below displays the assignment-control plot for one of the strata in the automatically stratified data set (Figure 4).

```
plot(ICU_astrat, type = "AC",
    propensity = surgicalTeam ~ Female.pre + Birth.preTimeDays +
      RaceAsian.pre + RaceUnknown.pre + RaceOther.pre + RaceBlack.pre +
      RacePacificIslander.pre + RaceNativeAmerican.pre + all_latinos,
    stratum = 2)
```

The striae in this assignment-control plot appear when discrete characteristics (e.g. sex and race/ethnicity) are highly weighted in the propensity or prognostic score, causing observations to cluster together. Since this is relatively common, 'jitter' arguments can be used to add small amounts of random noise to the coordinates of each point in order to avoid stacking.

Matching

After a suitable stratification is selected, observations can be matched within strata using strata_match. Since every stratum from the automatic stratification in this example contains at least a 1:2 ratio of patients who were assigned to surgical teams and those who were not, we can match 2 "control" (i.e., non-surgical team) patients to each "treated" (i.e., surgical team) subject in each stratum. In this step, we match individuals who, based on their baseline covariates, appear equally likely to have been assigned to a surgical team vs. not. The following performs the matching.

```
ICU_match <- strata_match(ICU_astrat,
     model = surgicalTeam ~ Birth.preTimeDays + Female.pre +
      RaceAsian.pre + RaceUnknown.pre + RaceOther.pre + RaceBlack.pre +
      RacePacificIslander.pre + RaceNativeAmerican.pre + all_latinos,
    k = 2)
```

```
Fitting propensity model: surgicalTeam ~ Birth.preTimeDays + Female.pre +
RaceAsian.pre + RaceUnknown.pre + RaceOther.pre + RaceBlack.pre +
RacePacificIslander.pre + RaceNativeAmerican.pre + all_latinos
```

Below, we print a summary.

```
summary(ICU_match)
```

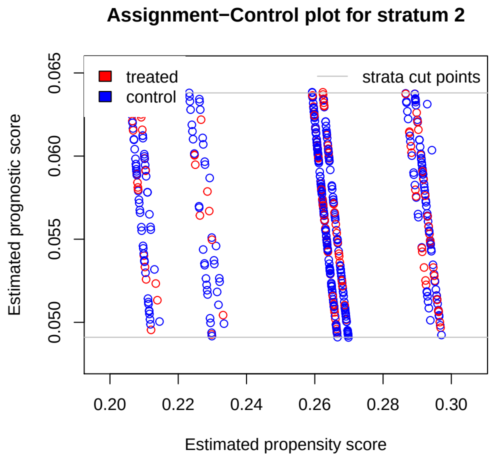


Figure 4: Assignment-control plot for automatic stratification of ICU data. The vertical striations are caused by heavily weighted discrete features in the propensity model, which cause points to align together.

```
Structure of matched sets:
 1:2  0:1
2226 2686
Effective Sample Size:  2968
(equivalent number of matched pairs).
```

At this point, the researcher can compare matched treated and control individuals to infer whether patients assigned to surgical treatment teams are more or less likely to be assigned a DNR code status, following up with sensitivity analyses (see, for example, **sensitivitymw** (Rosenbaum, 2014))

5 Key design choices and advanced functionality

The selection of the pilot set

The previous illustrations demonstrated the simplest method of extracting the pilot set: a random subsampling of all controls. Prior work by Aikens et al. (2020) contains a more thorough discussion of the considerations that might inform the selection of a pilot set.

A first consideration is the pilot set size. In general, the researcher should create a pilot set large enough to build a reliable prognostic model and retain enough remaining controls to select high-quality matches to the treatment group. This depends on the quality and number of available controls and the relative difficultly of fitting a prognostic model on the measured covariates. When high-quality controls (i.e., those resembling the treatment group) are scarce, the researcher should consider a smaller pilot set or a different study design altogether.

Another consideration is composition. Ideally, the individuals in the pilot set should be similar to the individuals in the treatment group, so a prognostic model built on this pilot set will not be extrapolating heavily when estimating prognostic scores on the analysis set. This approach can be especially important when there is some category of observations in the data that is relatively rare, and the researcher would like to ensure that some observations in this category end up in both the pilot and analysis sets. When discrete covariates are specified with the 'group_by_covariates' argument to auto_stratify the pilot set will be split proportionally based on these covariates so that the pilot set will be representative of the total control sample in terms of these covariates. This option can be used directly with auto_stratify. However, the split_pilot_set function is supplied as a convenience for users who prefer to split the pilot set themselves before stratification, as demonstrated below.

```
ICU_split <- split_pilot_set(ICU_data, treat = "surgicalTeam",
    pilot_fraction = 0.1, group_by_covariates = c("Female.pre", "self_pay"))
```

```
Constructing a pilot set by subsampling 10% of controls.
Subsampling while balancing on:
Female.pre self_pay
```

ICU_split, above, is a list containing a pilot_set and an analysis_set, partitioned while balancing sex and payment method (i.e., insurance or self-pay). Once this is done, the results can be passed to auto_stratify such as with the code below.

```
ICU_astrat2 <- auto_stratify(data = ICU_split$analysis_set,
    treat = "surgicalTeam",
    prognosis = DNR ~ Birth.preTimeDays + Female.pre + RaceAsian.pre +
      RaceUnknown.pre + RaceOther.pre + RacePacificIslander.pre +
      RaceBlack.pre + RaceNativeAmerican.pre + all_latinos,
    pilot_sample = ICU_split$pilot_set, size = 500)
```

Fitting the prognostic model

To fit the prognostic model, auto_stratify uses either linear (continuous outcome) or logistic regression (binary outcome). To accommodate a wider variety of modeling choices, auto_stratify can also be run using a vector of analysis set prognostic scores or prognostic model object[2].

The example below uses the **glmnet** package (Friedman et al., 2010) to fit a cross-validated lasso on the pilot set that was extracted in the previous section.

```
library("glmnet")
x_pilot <- ICU_split$pilot_set %>%
    dplyr::select(Birth.preTimeDays, Female.pre, RaceAsian.pre,
      RaceUnknown.pre, RaceOther.pre, RaceBlack.pre,
      RacePacificIslander.pre, RaceNativeAmerican.pre, all_latinos) %>%
    as.matrix()
y_pilot <- ICU_split$pilot_set %>%
    dplyr::select(DNR) %>%
    as.matrix()

cvfit <- cv.glmnet(x_pilot, y_pilot, family = "binomial")
```

The prognostic scores can then be estimated on the analysis set.

```
x_analysis <- ICU_split$analysis_set %>%
    dplyr::select(Birth.preTimeDays, Female.pre, RaceAsian.pre,
      RaceUnknown.pre, RaceOther.pre, RaceBlack.pre,
      RacePacificIslander.pre, RaceNativeAmerican.pre, all_latinos) %>%
    as.matrix()

lasso_scores <- predict(cvfit, newx = x_analysis, s = "lambda.min",
    type = "response")
```

Finally, these scores can be passed to auto_stratify with the 'prognosis' argument, producing a stratified data set that can be examined further with **stratamatch** diagnostic tools.

```
ICU_astrat3 <- auto_stratify(data = ICU_split$analysis_set,
    treat = "surgicalTeam", outcome = "DNR", prognosis = lasso_scores,
    pilot_sample = ICU_split$pilot_set, size = 500)
```

Other examples of prognostic score modeling options can be found in the **stratamatch** "Advanced Functionality" vignette.

Matching

Section 2.4 demonstrates how the **stratamatch** package can be used for optimal $1 : k$ matching on a propensity score. The strata_match function also supports full matching (Hansen and Klopfer, 2006; Rosenbaum, 1991), and the use of Mahalanobis distance instead of a propensity score. If desired, a data set stratified with **stratamatch** can instead be matched within strata using other matching software (e.g., **optmatch** (Hansen and Klopfer, 2006) or **MatchIt** (Ho et al., 2011)). For example, users proficient with **optmatch** will note that adding + strata(stratum) to the matching formula supplied to optmatch::pairmatch and other matching functions will match within stratum assignments in the analysis set.

[2] Model objects must have a method associated with the predict generic function

More nuanced matching schemes may also help address imbalances in the number of treated and control units within strata. For example, the researcher could perform $1 : k$ matching within each stratum, but allow k to vary between strata - matching more controls to each treated individual in strata where controls are plentiful and performing $1 : 1$ or $1 : 2$ matching where controls are less abundant. Another solution is to use a matching scheme within strata that naturally allows for variation in the ratio of treated and control individuals in matched sets, such as full matching (Rosenbaum, 1991; Hansen and Klopfer, 2006) or variable k matching (Pimentel et al., 2015).

As shown in figure 5, stratification is expected to substantially accelerate the matching process, especially for large sample sizes (several thousand or more). Hansen and Klopfer articulate a worst-case run-time for various forms of optimal matching with **optmatch** as $O(n^3 \log(nM))$, where M represents the maximum matching discrepancy between treated and control observations (Hansen and Klopfer, 2006). For context, this scales slightly less favorably than matrix inversion, which quickly becomes time-consuming for large inputs. By comparison, matching within strata of a fixed size tends to scale much more favorably for large n (figure 5). To further accelerate computation, a researcher might distribute matching the stratified data set over several computing nodes.

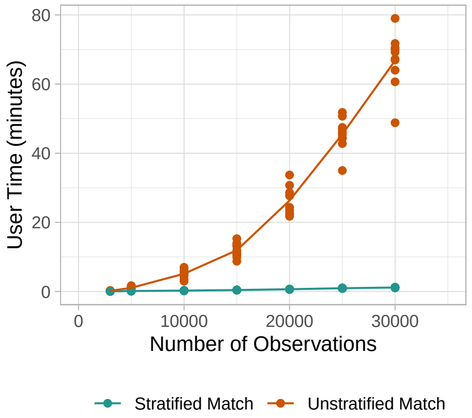


Figure 5: Measured computation times for stratified and unstratified matching on a modern laptop. Unstratified matching scales in a supra-linear manner with sample size (Hansen and Klopfer, 2006), while stratified matching with a set strata size tends to scale more favorably with n. At sample sizes of 30,000, optimally matching a whole data set may take over an hour, and much larger sample sizes may quickly become infeasible.

6 Trouble-shooting a stratification scheme

This section summarizes some common pitfalls and workarounds while stratifying a data set. Importantly, in order to preserve the separation of the design and analysis set, individuals partitioned into the pilot set must not be recombined with the analysis set. For instance, simply running `auto_stratify` repeatedly with different seeds to sample new pilot sets from the data and fit new prognostic score models may lead to overfitting of the prognostic model, raising concerns of bias in the study results (see Hansen (2008); Abadie et al. (2018)).

The following issues are common:

1. **Some strata are too small or too large:** This problem can often be solved simply by rerunning `auto_stratify` with a different 'size' parameter. When this is done, the researcher should be sure to use the same pilot and analysis set as they received when they first ran `auto_stratify` (i.e., do not partition a new pilot set).
2. **The strata have a poor balance of treated and control individuals:** This situation is relatively common but often straightforward to address with matching schemes that match more controls to each treated observation or allow for variable treat:control ratios. See section 2.5.3 for some suggestions.
3. **The prognostic model is poor:** In some cases, the user may encounter an error fitting the prognostic model, or they may suspect from prognostic model diagnostics that the model does a poor job of capturing variation predictive of the outcome. There are a few reasons the prognostic model may be problematic.

(a) *The prognostic model was mis-specified.* In this case, the user should fit a revised prognostic model on the same pilot set as was previously used. However, refitting repeatedly can lead to overfitting, so this should be done in moderation.

(b) *The pilot set was too small to get a reliable fit.* In this case, the user can add more samples from the analysis set to the existing pilot set. Samples that are moved into the pilot set must stay in the pilot set and should not be re-pooled with the analysis set.

(c) *Pilot set size is sufficient, but prognostic model perfectly separates treated individuals from control individuals:* If this occurs in either the pilot set or analysis set, it may be a sign that overlap is poor. See below.

4. **The treated and control individuals within strata have poor overlap in propensity and/or prognostic scores:** This problem is best diagnosed with assignment-control plots (see Aikens et al. (2020) for a deeper description). Propensity and prognostic score based subclassification methods both depend on some form of overlap in the baseline characteristics of treated and control individuals in order to make a valid estimate of causal effect (for a summary, see Leacy and Stuart (2014)). Treatment and control groups that are clearly separated in terms of either their propensity scores or prognostic scores can indicate that these two groups should not be compared because the resulting inference on treatment effect would be misleading. A researcher facing this situation might consider trimming the score space (Glynn et al., 2019) in some cases or seeking out another data set if the overlap problems are severe. While this may seem to be a disappointing result, the ability to identify these data issues before proceeding is one of the most important strengths of design-based causal inference (see, for example, Austin (2011)).

7 Summary and discussion

Stratifying a data set prior to matching may make optimal and full matching designs scale more practically for modern observational sample sizes (Figure 5). However, the primary objective of **stratamatch** is not to directly implement a computationally taxing task, but to expand access to sophisticated study design tools for a wide range of researchers with varying levels of technical and statistical sophistication. Indeed, the computational steps of stratification are relatively straightforward; however, the statistical concept of the pilot design is nuanced, and the process of stratifying a data set and interrogating the quality of that stratification can be thought-intensive and isn't well-supported by other resources. The **stratamatch** package is intended to make prognostic score stratification pilot designs – and stratified matching designs in general – easily implementable, with helpful diagnostic tools and documentation. The overall goal of this effort is to push researchers toward approaches and diagnostics that emphasize stronger study design in the observational setting. In modern observational studies, designs, such as the **stratamatch** approach, that are *tailored* to large-sample studies can offer increased precision and other statistical benefits that might otherwise be left on the table by more traditional approaches.

Bibliography

A. Abadie and G. W. Imbens. Large sample properties of matching estimators for average treatment effects. *econometrica*, 74(1):235–267, 2006. [p620]

A. Abadie and G. W. Imbens. Bias-corrected matching estimators for average treatment effects. *Journal of Business & Economic Statistics*, 29(1):1–11, 2011. [p620]

A. Abadie and J. Spiess. Robust post-matching inference. *Journal of the American Statistical Association*, pages 1–13, 2021. [p620]

A. Abadie, M. M. Chingos, and M. R. West. Endogenous stratification in randomized experiments. *Review of Economics and Statistics*, 100(4):567–580, 2018. [p615, 626]

R. C. Aikens, D. Greaves, and M. Baiocchi. A pilot design for observational studies: Using abundant data thoughtfully. *Statistics in Medicine*, 39(30):4821–4840, 2020. [p614, 615, 618, 619, 624, 627]

J. Antonelli, M. Cefalu, N. Palmer, and D. Agniel. Doubly robust matching estimators for high dimensional confounding adjustment. *Biometrics*, 74(4):1171–1179, 2018. [p615, 618]

P. C. Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, 46(3):399–424, 2011. [p627]

P. C. Austin and G. Cafri. Variance estimation when using propensity-score matching with replacement with survival or time-to-event outcomes. *Statistics in medicine*, 39(11):1623–1640, 2020. [p620]

P. C. Austin and D. S. Small. The use of bootstrapping when using propensity-score matching without replacement: a simulation study. *Statistics in medicine*, 33(24):4306–4319, 2014. [p620]

D. P. Bertsekas and P. Tseng. Relaxation methods for minimum cost ordinary and generalized network flow problems. *Operations Research*, 36(1):93–114, 1988. [p620]

G. Chavez, I. B. Richman, R. Kaimal, J. Bentley, L. A. Yasukawa, R. B. Altman, V. S. Periyakoil, and J. H. Chen. Reversals and limitations on high-intensity, life-sustaining treatments. *PloS one*, 13(2): e0190569, 2018. [p620]

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`. [p625]

R. J. Glynn, M. Lunt, K. J. Rothman, C. Poole, S. Schneeweiss, and T. Stürmer. Comparison of alternative approaches to trim subjects in the tails of the propensity score distribution. *Pharmacoepidemiology and drug safety*, 28(10):1290–1298, 2019. [p627]

S. N. Goodman, S. Schneeweiss, and M. Baiocchi. Using design thinking to differentiate useful from misleading evidence in observational research. *Jama*, 317(7):705–707, 2017. [p615]

N. Greifer. Estimating effects after matching. `https://cran.r-project.org/web/packages/MatchIt/vignettes/estimating-effects.html`, Dec. 2020. Accessed: 2021-5-3. [p620]

B. B. Hansen. The prognostic analogue of the propensity score. *Biometrika*, 95(2):481–488, 2008. [p615, 626]

B. B. Hansen and S. O. Klopfer. Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15(3):609–627, 2006. [p614, 618, 620, 625, 626]

M. A. Hernán and J. M. Robins. Using big data to emulate a target trial when a randomized trial is not available. *American journal of epidemiology*, 183(8):758–764, 2016. [p614]

D. E. Ho, K. Imai, G. King, and E. A. Stuart. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political analysis*, 15(3):199–236, 2007. [p620]

D. E. Ho, K. Imai, G. King, E. A. Stuart, et al. **MatchIt**: Nonparametric preprocessing for parametric causal inference. *Journal of Statistical Software, http://gking. harvard. edu/matchit*, 2011. [p614, 625]

G. King and R. Nielsen. Why propensity scores should not be used for matching. *Political Analysis*, 27 (4):435–454, Oct. 2019. [p614]

F. P. Leacy and E. A. Stuart. On the joint use of propensity and prognostic scores in estimation of the average treatment effect on the treated: A simulation study. *Statistics in medicine*, 33(20):3488–3508, 2014. [p615, 618, 627]

S. L. Lohr. *Sampling: Design and analysis: Design and analysis*. CRC Press, 2019. [p618]

S. D. Pimentel, F. Yoon, and L. Keele. Variable-ratio matching with fine balance in a study of the peer health exchange. *Statistics in medicine*, 34(30):4070–4082, 2015. [p626]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL `https://www.R-project.org/`. [p614]

J. Rigdon, M. Baiocchi, and S. Basu. Near-far matching in R: The nearfar package. *Journal of Statistical Software, Code Snippets*, 86(5):1–21, 2018. [p614]

P. R. Rosenbaum. A characterization of optimal designs for observational studies. *Journal of the Royal Statistical Society B*, 53(3):597–610, 1991. [p618, 625, 626]

P. R. Rosenbaum. Heterogeneity and causality: Unit heterogeneity and design sensitivity in observational studies. *The American Statistician*, 59(2):147–152, 2005a. [p614]

P. R. Rosenbaum. Sensitivity analysis in observational studies. *Encyclopedia of statistics in behavioral science*, 4:1809–1814, 2005b. [p614, 617, 620]

P. R. Rosenbaum. Sensitivity analysis for m-estimates, tests, and confidence intervals in matched observational studies. *Biometrics*, 63(2):456–464, 2007. [p620]

P. R. Rosenbaum. *sensitivitymw: Sensitivity analysis using weighted M-statistics*, 2014. URL https://CRAN.R-project.org/package=sensitivitymw. R package version 1.1. [p620, 624]

P. R. Rosenbaum. Two r packages for sensitivity analysis in observational studies. *Observational Studies*, 1(1):1–17, 2015. [p620]

P. R. Rosenbaum. *sensitivityfull: Sensitivity Analysis for Full Matching in Observational Studies*, 2017. URL https://CRAN.R-project.org/package=sensitivityfull. R package version 1.5.6. [p620]

P. R. Rosenbaum. *DOS2: Design of Observational Studies, Companion to the Second Edition*, 2019. URL https://CRAN.R-project.org/package=DOS2. R package version 0.5.2. [p614]

P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983. [p614, 615, 617]

P. R. Rosenbaum et al. *Design of Observational Studies*, volume 10. Springer-Verlag, 2010. [p614, 620]

D. B. Rubin. For objective causal inference, design trumps analysis. *The Annals of Applied Statistics*, 2 (3):808–840, 2008. [p615]

E. A. Stuart. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010. [p620]

E. Tipton. Improving generalizations from experiments using propensity score subclassification: Assumptions, properties, and contexts. *Journal of Educational and Behavioral Statistics*, 38(3):239–266, 2013. [p620]

Rachael C. Aikens
Interdepartmental Program in Biomedical Informatics
Stanford University
Stanford, CA
USA

Joseph Rigdon
Department of Biostatistics and Data Science
Wake Forest School of Medicine
Winston-Salem, North Carolina

Justin Lee
Quantitative Sciences Unit
Stanford University
Stanford, CA

Michael Baiocchi
Epidemiology and Population Health
Stanford University
Stanford, CA

Andrew B. Goldstone
Division of Cardiovascular Surgery
University of Pennsylvania
Philadelphia, PA

Peter Chiu
Department of Cardiothoracic Surgery
Stanford University School of Medicine
Stanford, CA

Y. Joseph Woo
Department of Cardiothoracic Surgery
Stanford University School of Medicine
Stanford, CA

Jonathan H. Chen
Biomedical Informatics Research Institute

Stanford University Medical Center
Stanford, CA
jonc101@stanford.edu