Building a Design-Based Matching Pipeline: From Principles to Practical Implementation in R

Thomas Leavitt and Luke W. Miratrix

Abstract

Matching, a canonical design for observational studies, takes many forms that often rest on distinct — yet implicit — statistical principles. We construct a matching pipeline for practitioners that makes these principles explicit by grounding each step in a coherent design-based framework. The pipeline begins with the conceptual ideal of a randomized experiment, traces how observational studies deviate from that ideal, and uses matching to approximate it. The next stage is inference under the as-if randomization assumption of matched sets' being equivalent to a collection of miniature randomized experiments within blocks. Under this assumption, we consider inference on all individual effects in the "sharp" framework and the average effect in the "weak" framework. The final stage is a sensitivity analysis to assess, under either framework, how inferences change under departures from as-if randomization. Each step includes extensively commented `R` code that equips practitioners to implement both established and newly developed procedures, including several not yet available in existing `R` packages. By integrating methods that are often considered separately into an overall pipeline, we aim to help practitioners understand why each step matters and how the pipeline can be tailored to their own data. We illustrate the full workflow through an application examining the effect of United Nations peacekeeping interventions on the duration of post-conflict peace.

Design-Based Foundations of the Matching Pipeline

The Randomized Experimental Ideal

Imagine a randomized experiment in which a researcher flips a fair coin independently for each individual in the study. Heads means assignment of the individual to control, while tails means assignment to treatment. After assignments to treatment and control, the researcher administers the conditions and then compares outcomes between treated and control groups.

Why is this procedure effective? Randomization is a *fair lottery*: Every individual has the same chance of being assigned to treatment. This means that individuals who would respond more strongly to treatment are no more likely to receive it than those who would respond less strongly, and the same is true for control. Because each individual's assignment is determined by the same coin toss (with the same probability of landing heads or tails), randomization leaves only two possibilities: (1) the difference in outcomes between treatment and control groups reflects the true causal effect, or (2) chance variation produced a misleading difference. Although misleading differences can occur by chance, randomization is valuable because it enables us to use statistical tools to quantify and limit the chance of such errors. As a result, randomized experiments yield especially credible causal conclusions.

Randomization is useful not only as a procedure, but also as an idea. It helps us understand when statistical tools will (and will not) yield credible conclusions, even when we have not directly randomized. In an observational study, the researcher does not control who receives the treatment and instead observes units after they have been assigned to treatment and control groups. In such settings, the idea of randomization can guide how we design studies so that they yield more credible causal conclusions.

Bridging Randomized Experiments and Observational Studies

A useful framework for connecting randomized experiments to observational studies is what Rubin (1977) calls "assignment to treatment group on the basis of a covariate," where a covariate is a pre-treatment characteristic of a study's units. Rubin (1977) supposes independent coin tosses for each individual in which the probability of heads or tails now depends on that individual's value of a single covariate. Consequently, all individuals with the same covariate value share the same chance of ending up in treatment, while those with a different covariate value share a different chance.

In this setting, we can envision forming groups (i.e., matched sets) so that all individuals within a group share the same covariate value. Each group then functions as a miniature randomized experiment in that random chance alone explains why some individuals in the group ended up in treatment while others did not. Importantly, to justify this interpretation, we do not need to know each individual's actual probability of treatment. It is enough to know that the probability of treatment depends only on the covariate, which ensures that all individuals with the same covariate value have the same chance of treatment.

Matching to Approximate the Randomized Experimental Ideal

The same intuition applies to matching when the probability of treatment depends on many baseline covariates. The underlying idea is that individuals with similar covariates have similar treatment assignment probabilities. Thus, in an effort to recreate a randomized experiment, we use the covariates we observe and believe determine treatment chances in order to divide individuals into groups, with each group containing both treated and control subjects who are homogeneous in those covariates. We are, in effect, constructing a new single variable: group (i.e., matched set) membership. We can think of this membership variable as the single covariate in the framework of "assignment to treatment group on the basis of a covariate" (Rubin, 1977) discussed above. Although we still do not know each individual's treatment probability, the hope is that all individuals within a group share the same probability, whatever that probability may be.

We refer to the condition that all individuals within matched sets share the same treatment probability as *as-if randomization*, although other terms, such as *selection on observables*, are also common. We use the term as-if randomization because, when all units in a matched set share the same treatment probability, conditioning on the number of treated units makes every possible assignment within the set equally likely, creating a situation that is as if we had actually randomized. When as-if randomization holds, we can use the same statistical tools that we would use in a randomized experiment to draw credible causal inferences from our observational study.

Why Sensitivity Analysis Matters

Unlike a randomized experiment, even the best matched designs rely on the strong assumption of as-if randomization. When this assumption holds, we can draw causal conclusions by analyzing the data as if they came from a randomized experiment. However, if the assumption is wrong, our causal claims are no longer guaranteed to be credible.

How can this assumption fail? First, individuals within a group may be similar on observed covariates, but not similar enough to have the same treatment probabilities. Second, treatment assignment may depend on covariates we did not measure. If so, even if individuals within groups appear comparable on observed covariates, those individuals may still differ on hidden covariates that determine the probability of treatment.

For these reasons, it is important to assess the sensitivity of our causal conclusions to departures from as-if randomization. Conclusions are especially convincing when they hold not only under this assumption, but also under moderate violations of it. Conclusions that collapse under only mild departures are much less convincing.

Roadmap of the Design-Based Matching Pipeline

Building on these design-based foundations, we now outline a pipeline that starts with matching and then proceeds to inference and sensitivity analysis:

- **Construct and evaluate matched sets**. We begin with the mechanics of optimal matching (Hansen, 2004; Hansen and Klopfer, 2006): choosing a distance measure that defines similarity on the covariates, setting calipers — maximum allowable distances between treated and control units for inclusion in the same matched set — and imposing structural constraints (e.g., requiring matches to be pairs). We then show how to evaluate the resulting design in terms of both effective sample size and covariate balance. In particular, we focus on tests proposed by Hansen and Bowers (2008) that compare the matched design's covariate balance to what one would expect under an equivalent completely randomized experiment within blocks (that is, random assignment with a fixed number of treated units per block).
- **Draw causal inferences**. Once a matched design is chosen, practitioners can conduct inference

under either a “sharp” causal framework, which pertains to individual-level effects for all units, or a “weak” framework, which pertains to a summary quantity of the unit-level causal effects, typically the average treatment effect (ATE). For the sharp framework, we focus on how researchers can use both simulation- and Normal-based approximations to the exact randomization distribution, either to perform hypothesis tests or to obtain point estimates by inverting those tests. For the weak framework, we cover estimation of the ATE and hypothesis tests about it. We emphasize exposition and code for recent variance estimators tailored to designs with only 1 treated or 1 control unit per matched set (Fogarty, 2018; Pashley and Miratrix, 2021) — an important case, since such designs are optimal in terms of balance and effective sample size (Gu and Rosenbaum, 1993; Rosenbaum, 1991; Hansen, 2004).

- **Assess sensitivity**. Finally, we turn to sensitivity analyses under both inferential frameworks. We review established methods for conducting sensitivity analysis for tests of sharp nulls under possible violations of as-if randomization (Rosenbaum and Krieger, 1990; Gastwirth et al., 2000; Rosenbaum, 2018). We then describe and implement new methods that extend sensitivity analysis to tests of weak null hypotheses (Fogarty, 2023).

Below we present a flow diagram that summarizes the overall pipeline. The diagram shows each step and decision point, the relevant `R` tools (whether existing packages or custom functions included herein) and core references.

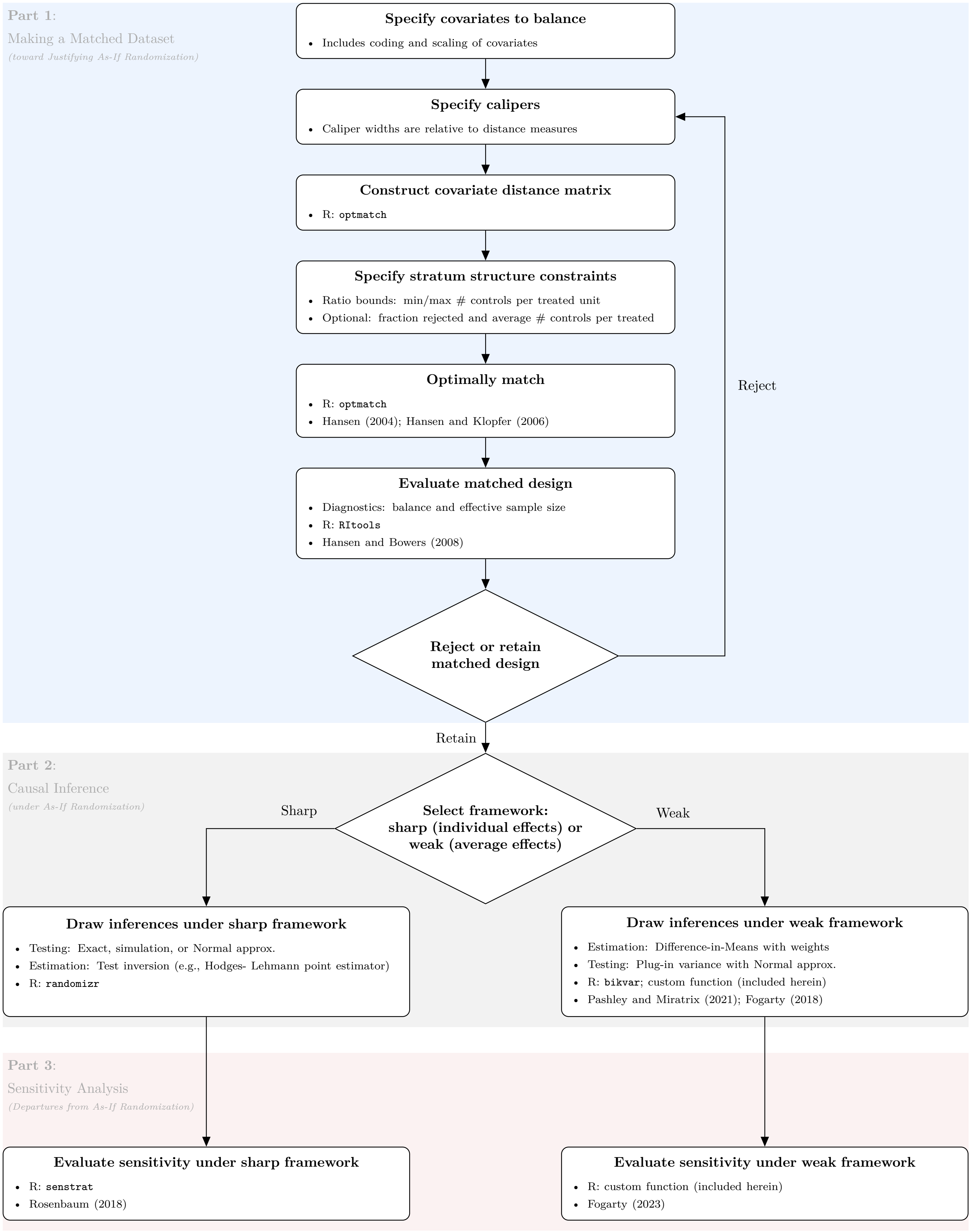

Part 1:
Making a Matched Dataset
(toward Justifying As-If Randomization)
Specify covariates to balance
• Includes coding and scaling of covariates
Specify calipers
• Caliper widths are relative to distance measures
Construct covariate distance matrix
• R: optmatch
Specify stratum structure constraints
• Ratio bounds: min/max # controls per treated unit
• Optional: fraction rejected and average # controls per treated
Optimally match
• R: optmatch
• Hansen (2004); Hansen and Klopfer (2006)
Evaluate matched design
• Diagnostics: balance and effective sample size
• R: RItools
• Hansen and Bowers (2008)
Reject or retain matched design
Reject
Retain
Part 2:
Causal Inference
(under As-If Randomization)
Select framework: sharp (individual effects) or weak (average effects)
Sharp
Weak
Draw inferences under sharp framework
• Testing: Exact, simulation, or Normal approx.
• Estimation: Test inversion (e.g., Hodges- Lehmann point estimator)
• R: randomizr
Draw inferences under weak framework
• Estimation: Difference-in-Means with weights
• Testing: Plug-in variance with Normal approx.
• R: bikvar; custom function (included herein)
• Pashley and Miratrix (2021); Fogarty (2018)
Part 3:
Sensitivity Analysis
(Departures from As-If Randomization)
Evaluate sensitivity under sharp framework
• R: senstrat
• Rosenbaum (2018)
Evaluate sensitivity under weak framework
• R: custom function (included herein)
• Fogarty (2023)

Figure 1: Flow diagram of the design-based matching pipeline

Implementation of the Matching Pipeline

We provide a high-level overview of the ideas behind matching and include code that demonstrates how to implement those ideas with various R packages. We also work through some of these steps "by hand" to underscore the underlying conceptual issues. Doing so also gives practitioners more flexibility to adapt the matching pipeline to their own needs.

We break the implementation into specific decision points that practitioners commonly face, and present the pipeline in three main parts (see Figure 1):

Part 1: Making a Matched Dataset (toward Justifying As-If Randomization)

(a) How Do I Measure Similarity on My Chosen Covariates?

- Similarity on the Estimated Propensity Score

(b) How Can I Apply Rules for Matches to Ensure Comparability?

(c) How Can I Apply Rules for Matches to Improve Effective Sample Size?

(d) How Do I Actually Form the Matches?

(e) How Do I Decide Whether to Move Ahead with My Matched Design?

Part 2: Causal Inference (under As-If Randomization)

(a) How Do I Draw Inferences under Sharp Framework?

(b) How Do I Draw Inferences under Weak Framework?

Part 3: Sensitivity Analysis (Departures from As-If Randomization)

(a) How Do My Inferences under the Sharp Framework Change under these Departures?

- Finding the Worst-Case Scenario of Hidden Confounding to Ensure Valid Inference
  - Separable Approximation
  - Taylor Series Approximation
- Conducting Sensitivity Analysis under the Worst-Case Scenario

(b) How Do My Inferences under the Weak Framework Change under these Departures?

Before turning to Part 1, we introduce a running example taken from Gilligan and Sergenti (2008), which we use throughout this document.

Running Example: United Nations Peacekeeping and Post-Conflict Peace

We introduce matching through an example that examines the causal effect of United Nations (UN) peacekeeping missions on the durability of post-conflict peace, a question of central importance for both academic research and policy. Our example draws on data from Gilligan and Sergenti (2008), whose title includes the phrase "Matching to Improve Causal Inference," underscoring the value of applying matching to study the UN's causal impact on post-conflict peace. These data are publicly available in the supplementary information of the article's webpage in the *Quarterly Journal of Political Science* (DOI: 10.1561/100.00007051).

The dataset from Gilligan and Sergenti (2008) includes 87 observations, each corresponding to a country's peace-period episode following a civil war, with episodes beginning as early as January 1989 and data extending through December 2003. In some episodes, UN peacekeepers intervened (e.g., Sierra Leone, Jan 2001 - Dec 2003), while in others they did not (e.g., Macedonia, Sep 2001 - Dec 2003). The treatment variable is UN intervention (`UN`), coded as 1 if a UN mission was present during the peace period and 0 otherwise. The outcome variable is the duration of the peace spell, which Gilligan and Sergenti (2008) measure as the log of the number of days from the start of peace until either the outbreak of a new conflict or right-censoring at December 2003. This log-transformed outcome (`ldur`) captures the total length of the peace period rather than the time elapsed after a potential UN intervention. In practice, however, UN interventions almost always began immediately after the onset of peace: "Of the 19 post-conflict UN interventions, the United Nations was present within the first month for 16 of them" (Gilligan and Sergenti, 2008, p. 118). Going forward, we set aside these two measurement details.

As Gilligan and Sergenti (2008) state, "UN missions are not randomly assigned" (p. 89). Whether peacekeepers are present in a country during a given peace period depends on baseline covariates such as the logged number of deaths (`lwdeaths`), the logged duration (`lwdurat`) of the last war, ethnic fractionalization (`ethfrac`, a 0 - 100 index, where dividing by 100 is intended to represent the probability that two randomly chosen individuals belong to different ethnic groups), logged population size (`pop`) and others. We implement matching using these same covariates, but emphasize that our exercise is expository and not intended as a replication of the original findings.

Loading the Data for the Running Example

To load the data, you could first download the replication files from the supplementary information on the article's webpage (DOI: 10.1561/100.00007051), save the files to your working directory, and then use a package such as `haven` to load the Stata (.dta) file, `peace_pre_match.dta`, into `R`. However, for our purposes, we recommend loading our pre-created .rds file (`peace_pre_match.rds`), in which Stata's monthly numeric dates have been converted to `R`'s year-month format and the geographic region indicators recoded into a single factor variable (`region`). The command below loads this pre-created dataset (`peace_pre_match.rds`) into the `R` environment as an object named `data`.

```r
# Define base URL for the Matching Guide GitHub repository
base_url <- "https://raw.githubusercontent.com/tl2624/matching-guide/main"

# Load the cleaned dataset
data <- readRDS(url(paste0(base_url, "/data/peace_pre_match.rds")))
```

Part 1: Making a Matched Dataset (toward Justifying As-If Randomization)

In an observational setting, treatment is not assigned by the flip of a coin but depends on individuals' covariates. The goal of matching is to compare treated and control units that have the same chances of treatment based on those covariates — i.e., the same *propensity scores*. If we could observe propensity scores, it would be straightforward to compare treated and control units by matching directly on them. Because propensity scores are not directly observed, we instead aim to create a collection of matched sets that is *balanced* — meaning that treated and control observations are similar in their covariates.

Before turning to questions of covariate similarity and matching, it is important to note that substantive transformations of covariates play a central role, as they determine the inputs on which subsequent notions of similarity between units are based. These transformations reflect substantive judgments and are often among the most important decisions in practice. In this document, however, we do not address these substantive choices. Instead, we take the transformations used by Gilligan and Sergenti (2008) as given. For example, they measure ethnic fractionalization (`ethfrac`) on a 0-100 scale (rather than a 0–1 scale) and apply logarithmic transformations to covariates such as the number of deaths (`lwdeaths`) and the duration of the previous war (`lwdurat`), among others. Our focus is therefore on design and analysis decisions conditional on these substantively chosen scales.

How Do I Measure Similarity on My Chosen Covariates?

In the simplest terms, matching is about ensuring apples-to-apples, rather than apples-to-oranges, comparisons between treated and control observations (Rubin and Waterman, 2006). To create matched sets in which treated and control groups are similar in their covariates, we first need a distance measure that quantifies how close any two observations are. With such a measure in hand, we can then construct sets of treated and control observations that are close on this measure — i.e., apples-to-apples in their pre-treatment characteristics.

We record the distances between treated and control units in a distance matrix: The rows correspond to treated units and the columns to control units. Each cell of the matrix records the distance between a specific treated unit and a specific control unit, as defined by a distance measure. This distance measure takes the baseline covariates of the two units and maps them to a single nonnegative number, with smaller values indicating greater similarity.

In our setting, we are interested in similarity across 9 covariates, named in the object `covs`.

```r
# Define character vector of the 9 covariate names in the dataset
covs <- c("lwdeaths", "lwdurat", "ethfrac", "pop", "lmtnest", "milper", "bwgdp",
          "bwplty2", "region")
```

The first four — `lwdeaths`, `lwdurat`, `ethfrac`, and `pop` — were introduced earlier. The others include a logged measure of the proportion of a country's land area that is mountainous (`lmtnest`), the logged total number of military personnel in a country (`milper`), logged GDP per capita before the last civil war (`bwgdp`), the Polity score (a standard –10 to 10 scale of democratic versus autocratic institutions) before the last civil war (`bwplty2`), and a region factor (`region`) with categories for Eastern Europe, Latin America, Asia, Sub-Saharan Africa, and North Africa/Middle East.

All of these covariates are measured before treatment and presumably determine the chance of a UN intervention during a country's peace period. Our interest in them stems primarily from their role in determining those intervention probabilities. Yet many of these covariates may also be prognostic; that is, they help predict the outcome of interest in Gilligan and Sergenti (2008), the log duration of the peace period (`ldur`) that countries would potentially experience with or without a UN intervention. This prognostic value of covariates can provide an additional reason to match on them (Hansen, 2008a; Sales et al., 2018).

There are many ways to measure the distance between a treated and a control unit. For example, we might compare units using the Euclidean distance — i.e., the square root of the sum of squared differences — across all baseline covariates. To do so, we first convert the factor variable `region`, which stores categorical labels, into a set of dummy (0 or 1) variables, one for each region. This conversion ensures that distances between treated and control units can be computed, since distance measures require numeric variables rather than categorical labels.

```r
# Install "dplyr" package (only run if you don't already have it installed)
# Install.packages("dplyr")

# Load dplyr package for data manipulation (mutate, group_by, summarize, etc.)
library(dplyr)

# Convert categorical variable 'region' into 0/1 dummy indicators
data <- data |>  # Pipe (|>) to pass left-hand result into next function call
  mutate(
    eeurop   = ifelse(test = region == "eeurop",   yes = 1, no = 0),
    lamerica = ifelse(test = region == "lamerica", yes = 1, no = 0),
    asia     = ifelse(test = region == "asia",     yes = 1, no = 0),
    ssafrica = ifelse(test = region == "ssafrica", yes = 1, no = 0),
    nafrme   = ifelse(test = region == "nafrme",   yes = 1, no = 0)
  )

# Remove 'region' and replace with dummy indicators
expanded_covs <- c(
# Setdiff() returns elements in 'x' that are not in 'y'
  setdiff(x = covs, y = "region"),  # Drops "region" from the covariate list
  "eeurop", "lamerica", "asia", "ssafrica", "nafrme"
)
```

Here we calculate the Euclidean distance between post-conflict Liberia, where the UN did intervene, and post-conflict Guinea-Bissau, where the UN did not.

```r
# Extract covariate values for Liberia and Guinea-Bissau (cname = country name)
liberia <- data[data$cname == "Liberia", expanded_covs]
guinea_bissau <- data[data$cname == "Guinea-Bissau", expanded_covs]

# Compute Euclidean distance between the two countries on these covariates
sqrt(sum((liberia - guinea_bissau)^2))  # Display the Euclidean distance
```

```
[1] 57.44263
```

We can obtain the full matrix of pairwise distance values using `match_on()` from the `optmatch` package. We do not need to manually recode factor variables into dummy indicators because `match_on()` handles this conversion automatically.

```r
# Create a formula: UN (treatment indicator) ~ covariates
# Note: we keep "region" in covs as a factor
cov_fmla <- reformulate(termlabels = covs,
                        response = "UN")

# Install optmatch if not already installed
# Install.packages("optmatch")

# Load optmatch, which provides the match_on() function
library(optmatch)

# Compute Euclidean distance matrix between treated (UN = 1) and control (UN = 0)
dist_mat_euc <- match_on(x = cov_fmla,                     # Formula for covariates
                         data = data,                      # Dataset used
                         standardization.scale = NULL, # No rescaling of covariates
                         method = "euclidean")         # Use Euclidean distance

# Add country names (cname) as row/column labels for clarity
dimnames(dist_mat_euc) <- list(data$cname[data$UN == 1], data$cname[data$UN == 0])

# Display a submatrix of distances: treated units 11-15 vs control units 27-30
round(x = dist_mat_euc[11:15, 27:30],
      digits = 2)  # Number of decimal places to round
```

```
              Niger Guinea   Togo Central African Republic
Sierra Leone  94.21 101.18 116.32                   116.94
Zaire         26.68  29.87  45.63                    46.57
Rwanda        66.75  71.57  77.04                    76.03
Mozambique   133.47 140.60 155.40                   155.80
Namibia      246.44 253.23 268.35                   268.08
```

In this little subset of the full distance matrix, the first entry is the Euclidean distance of Sierra Leone (treated) from Niger (control). The last listed entry is the distance between Namibia (treated) and the Central African Republic (control).

One concern with Euclidean distance is that it depends on the scale of the covariates. For example, the difference between a country in Sub-Saharan Africa (`ssafrica` = 1) and a country in Latin America (`ssafrica` = 0) would contribute the same to the Euclidean distance as the difference between two countries with GDP per capita values of $3000 and $3001. Intuitively, we would not want such a tiny difference in economic size to be treated as equally important as belonging to different regions of the world. More generally, we want differences across variables to be placed on a comparable scale, so that a meaningful difference in one variable counts about the same as a difference of similar importance in another.

Another concern with Euclidean distance is that it ignores correlations among covariates. For example, countries with larger populations (`pop`) usually have more military personnel (`milper`), if only because a larger population provides a greater pool of potential recruits. Therefore, differences in both covariates may largely reflect the same underlying factor — population size. Yet Euclidean distance adds these differences separately, as if they were unrelated, which can exaggerate the overall distance between two observations.

The Mahalanobis distance (Mahalanobis, 1936) addresses both of these concerns. First, it standardizes covariates so that differences are placed on a comparable scale. This standardization is used solely to compute distances across covariates that typically differ in scale, regardless of the substantively chosen transformations, with the aim of constructing matches that ultimately support apples-to-apples comparisons on these original, substantively chosen scales. Second, the Mahalanobis distance adjusts for correlations among covariates, ensuring that highly related variables are not effectively counted twice. We can therefore construct a distance matrix based on Mahalanobis rather than Euclidean distances as follows.

```r
# Compute Mahalanobis distance matrix between treated (UN = 1) and control (UN = 0)
dist_mat_mah <- match_on(
  x = cov_fmla,
  data = data,
  standardization.scale = NULL,
  method = "mahalanobis" # Use Mahalanobis distance
)
```

Similarity on the Estimated Propensity Score So far, we have focused on ways to measure distance between treated and control units across many covariates in order to identify which units are most similar and group them together to achieve covariate balance. However, when there are many covariates, it becomes difficult to find treated and control units that are similar on all of them. This challenge is often referred to as the "curse of dimensionality."

A common way to address this problem is to reduce the information from many covariates into a lower-dimensional form. The *estimated propensity score* does this by collapsing information from all covariates into a single number. This number represents a transformation of a linear index of covariates that accounts for how strongly each covariate predicts treatment.

Consider, for example, a logistic model for the estimated propensity score of an individual unit i. We write this model as

$$\hat{\lambda}(\boldsymbol{x}_i) := \frac{1}{1 + \exp(-\hat{\boldsymbol{\beta}}^\top \boldsymbol{x}_i)}, \tag{1}$$

where $\boldsymbol{x}_i$ is the covariate vector, $\hat{\boldsymbol{\beta}}$ is the vector of estimated coefficients, and the superscript $\top$ denotes the transpose, turning the row vector $\hat{\boldsymbol{\beta}}$ into a column vector. The quantity $\hat{\boldsymbol{\beta}}^\top \boldsymbol{x}_i$ is the linear index, and the inverse logistic function, $1/(1 + \exp(-x))$, maps any real-valued input, x, onto the interval (0, 1). The linear covariate index for unit i, $\hat{\boldsymbol{\beta}}^\top \boldsymbol{x}_i$, is simply the logit, i.e., log odds, transformation of $\hat{\lambda}(\boldsymbol{x}_i)$ in (1).

This quantity, $\hat{\lambda}(\boldsymbol{x}_i)$, is a simple transformation of a linear index of covariates that best “separates” treated from control units. The estimated coefficients ($\hat{\boldsymbol{\beta}}$) “separate” treated from control units on the linear index because the coefficients are chosen to maximize a likelihood that rewards large differences between the groups. When treated and control units have little or no covariate overlap, the linear indices can diverge substantially, very positive for treated units and very negative for controls. With greater overlap, the linear indices for treated and control units are similar, clustering near zero.

This estimation process reflects how predictive each covariate is of treatment. When treated and control observations lack overlap on a covariate, that covariate is highly predictive of treatment and therefore receives an estimated coefficient with a large magnitude. When there is substantial overlap on a covariate, it is less predictive of treatment, and the magnitude of its coefficient is small. Consequently, when we assess similarity on the linear index of covariates in (1), the estimated coefficients assign greater importance to covariates that strongly predict treatment and less importance to those that do not.

To see this logic in action, first estimate a logistic propensity score model using all covariates except for `region`, which we exclude because some regions almost perfectly predict treatment, leading to near-complete separation (Albert and Anderson, 1984).

```r
# Formula for UN ~ covariates (excluding "region")
psm_cov_fmla <- reformulate(termlabels = setdiff(x = covs, y = "region"),
                            response = "UN")

# Fit logistic regression for propensity score model
psm <- glm(
  formula = psm_cov_fmla,             # Treatment ~ covariates
  family = binomial(link = "logit"), # Logistic regression (logit link)
  data = data                         # Dataset used for model fitting
)
```

We can extract units' linear covariate indices from this model like so.

```r
# Extract logit propensity scores (linear predictors from fitted model)
lin_cov_inds <- psm$linear.predictors  # Same as model.matrix(psm) %*% coef(psm)
```

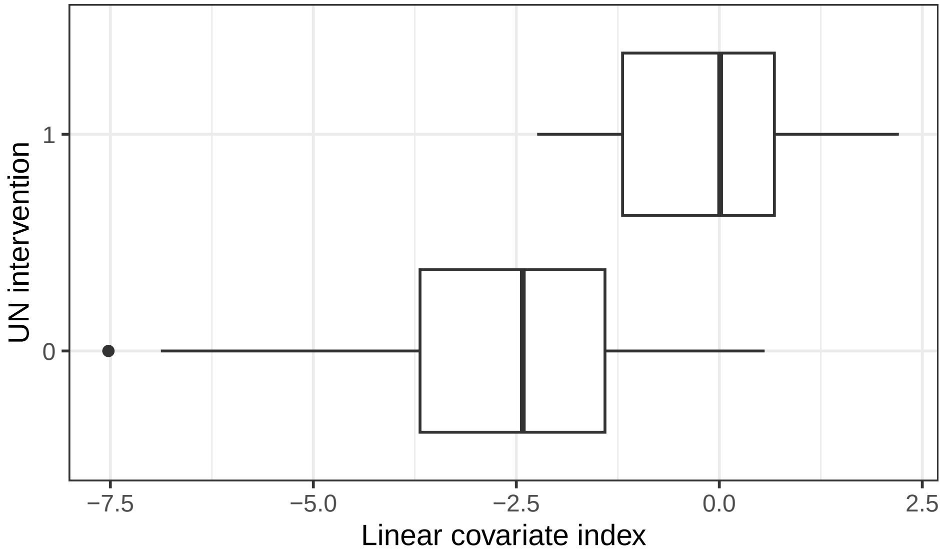
Below we can see that the linear covariate indices from the model `psm` correspond exactly to the logits (i.e., the log-odds transformations) of the model's predicted probabilities of treatment, which range between 0 and 1.

```r
# Extract estimated propensity scores (predicted probabilities of UN = 1)
p_scores <- psm$fitted.values

# Convert propensity scores to log-odds (logit scale)
log(p_scores/(1 - p_scores))

# Check that p_scores equals logistic(lin_cov_inds)
all.equal(p_scores, 1/(1 + exp(-lin_cov_inds)))
```

To see how two observations that differ on many covariates can still have similar estimated propensity scores, consider post-conflict Namibia (treated) and Burundi (control). The two are similar on some covariates, such as logged military personnel (`milper`), but — consistent with the "curse of dimensionality" — very different on others, such as duration of the last war (`lwdurat`) and ethnic fractionalization (`ethfrac`). Yet Namibia's and Burundi's linear indices differ by only 0.5. This small difference occurs because the covariates on which Namibia and Burundi differ greatly have small coefficients (e.g., `lwdurat` ≈ -0.01, `ethfrac` ≈ 0), while those on which the two observations are similar, such as `milper`, have large coefficients (approximately -0.92).

In some cases, differences on individual covariates may offset each other; for example, when one covariate has a large negative coefficient and another a large positive one. Conversely, even if a treated-control pair is close in Euclidean distance — say, closer than the 183.12 distance between Namibia and Burundi — the two observations may still differ more in their linear covariate indices if that pair differs on covariates that are especially predictive of treatment.

To illustrate these broader patterns, the boxplot below compares the empirical distributions of the linear covariate index for treated and control groups.



As the figure above shows, there is some, but not a lot of, overlap between treated and control groups. In accordance with our earlier discussion of how the linear covariate index "separates" treated from control units, many control observations have very negative values (as low as -7.52), far from the treated units' range of -2.24 to 2.21. Nevertheless, a sufficient number of treated and control observations have linear covariate indices that cluster around 0, indicating covariate overlap for at least a subset of treated and control observations.

How Can I Apply Rules for Matches to Ensure Comparability?

The discussion above on measuring covariate distances between treated and control observations helps identify which treated-control pairs are similar. The goal of identifying these similar treated-control pairs is to form matched sets that are closely aligned on their covariates, thereby improving balance between treatment and control groups. In practice, we do this by excluding potential matches that are "too dissimilar." There are two common ways to do this:

- **Exact matching**: Require that units be identical on some subset of important covariates, typically those that are categorical or coarse enough for units to take the same values.
- **Calipers**: Impose a threshold for the maximum allowable distance so that no matched set may include a treated and a control unit that are farther apart than this caliper.

As a simple example to build intuition, we will impose the following constraints:

- Observations can only be in the same matched stratum if they are in the same geographic region.
- Treated and control observations more than two points apart on the Polity score cannot be in the same matched set.

Below we impose the first constraint, requiring an exact match on region. Doing so produces separate distance matrices containing the Euclidean distances on the covariate used to define the exact match (`region`), where all entries are 0, indicating that treated and control units belong to the same region.

```r
# Create distance structure: 0 if units are in the same region, Inf otherwise
em_region <- exactMatch(x = UN ~ region,
                        data = data)
```

Now we impose the second constraint: We construct a distance matrix based on the Polity score, `bwplty2`, with a caliper of 2. This matrix records the Euclidean difference in Polity scores when the difference is 2 or less, and assigns a value of ∞ (denoted in `R` as `Inf`) when the difference exceeds 2. The ∞ entries are crucial because `optmatch` minimizes the sum, across all matched sets, of the within-set sums of covariate distances between each treated-control pair. Consequently, any treated–control pair differing by more than 2 points on the Polity score is assigned an overall distance of ∞, which prevents them from being matched.

```r
# Euclidean distance on Polity score (bwplty2) with caliper = 2
# Pairs differing by >2 are set to Inf
euc_dist_polity_cal_2 <- match_on(x = UN ~ bwplty2,
                                  caliper = 2,  # Set caliper
                                  data = data,
                                  standardization.scale = NULL,
                                  method = "euclidean")
```

Note that we used Euclidean distance here because, after exactly matching on geographic region, matching proceeds on only 1 covariate (Polity score), so we do not have to worry about covariates' relative scales.

Finally, we combine the two constraints into distance matrices defined within each region. We construct these region-specific matrices by "adding" the em_region and euc_dist_polity_cal_2 objects, as shown below.

```r
# Create overall distance matrix by element-wise addition of two distance matrices
overall_dist_mat <- em_region + euc_dist_polity_cal_2
```

How Can I Apply Rules for Matches to Improve Effective Sample Size?

Beyond comparability in covariates, we also care about the matched study's effective size. The effective sample size — i.e., how much information the matched design provides — depends not simply on the total number of units included in our matches. Effective sample size also depends on how those units are arranged across the matched sets.

The optmatch package will always produce matches with a particular arrangement of units across sets. In particular, all sets contain either 1 treated unit or 1 control unit — an overall structure that minimizes imbalance while excluding as few units as possible (Rosenbaum, 1991; Gu and Rosenbaum, 1993; Hansen, 2004). In practice, optimal full matching can yield lopsided sets, with either 1 treated matched to many controls or 1 control matched to many treated units, which has implications for the effective sample size.

The formal definition of effective sample size used by the optmatch package is the sum, across matched sets, of the harmonic mean of the numbers of treated and control units in each set:

$$\sum_{s=1}^{S} \left[\left(m_s^{-1} + (n_s - m_s)^{-1} \right) / 2 \right]^{-1} . \tag{2}$$

In this definition, the index s runs over the $\{1, \ldots, S\}$ matched sets, with m_s denoting the number of treated units in set s and $n_s - m_s$ the number of control units. With n_s denoting the number of units in set s, the total number of individuals included in the matched study is $n = \sum_{s=1}^{S} n_s$.

From the formula in (2), we can see precisely how the effective sample size depends on the arrangement of units across sets. For example, in a study with 4 total units, the effective sample size would be 2 if the units were arranged into 2 matched pairs. By contrast, in a study of the same total size but arranged as a single set with 1 treated unit and 3 controls, the effective sample size would be 1.5. The effective sample size is larger in the former arrangement because it provides 2 distinct treated-versus-control comparisons, whereas the latter provides only 1.

This definition of effective sample size connects directly to the precision of estimators and the power of hypothesis tests. Assuming constant, additive treatment effects within a set, the variance of the Difference-in-Means — the average outcome among treated units minus the average outcome among control units — in that set is minimized when the harmonic mean is largest (Hansen and Bowers, 2008; Hansen, 2011). The harmonic mean reaches its maximum when the numbers of treated and control units are equal. Thus, all else equal, matched pairs and other balanced sets provide more information about causal effects than lopsided sets with unequal treated-to-control ratios.

One straightforward way to increase effective sample size is to relax restrictions on which units can be matched. For instance, we might widen the caliper on Polity score from 2 to 3 and then rebuild the distance matrix.

```r
# Apply a caliper of width 3 to the polity Euclidean distance matrix
euc_dist_polity_cal_3 <- match_on(x = UN ~ bwplty2,
                                  caliper = 3,
                                  data = data,
                                  standardization.scale = NULL,
                                  method = "euclidean")

# Combine regional exact match distance with polity distance
em_region + euc_dist_polity_cal_3
```

Relaxing calipers allows additional units to be included in the matched design, since units without eligible matches would otherwise be discarded. Including these additional units can increase the effective sample size, but such gains need not be large. In particular, as the formula for a set's harmonic mean in (2) makes clear, the number of treated-versus-control comparisons does not increase because optimally matched sets, as we noted earlier, contain a single treated unit with possibly many controls or a single control unit with possibly many treated units. As a result, the contribution of newly admitted units to the effective sample size is limited. The magnitude of any increase in effective sample size — given by the sum of harmonic means across sets — depends on how these newly admitted units are distributed across matched sets: If these units are concentrated within only a few sets, the resulting gain will be modest.

Instead of relying solely on caliper width, researchers can also shape the effective sample size by controlling the structure of matched sets via the `min.controls` and `max.controls` arguments in `optmatch`'s fullmatch() function. These arguments specify lower and upper bounds, respectively, on the ratio of control to treated units within each matched set. By default, min.controls = 0 and max.controls = Inf, which impose no restrictions on matched-set composition. Departing from these defaults allows researchers to control how balanced or lopsided matched sets may be, thereby influencing the effective sample size.

To illustrate, suppose we want to restrict matches using the overall_dist_mat introduced earlier. When we ultimately construct our matches, full matching will divide the data into matched sets containing one treated unit and any positive number of controls, or one control unit and any positive number of treated units. However, we can impose additional constraints on this full matching, such as requiring a minimum control-to-treated ratio of 1:2 — that is, at least one control for every two treated units (min.controls = 0.5) — and no more than two controls per treated unit (max.controls = 2). Under these restrictions, allowable matched sets could include 2 treated units with 1 control, 1 treated unit with 1 control (a matched pair), or 1 treated unit with 2 controls.

```r
# Full matching using overall distance matrix
fullmatch(
  x = overall_dist_mat,
  min.controls = 0.5,  # At least 0.5 controls per treated unit
                       # (i.e., no more than 2 treated per control)
  max.controls = 2,    # At most 2 controls per treated unit
  data = data
)
```

Imposing ratio constraints can introduce trade-offs. In some cases, balance may worsen if a control is forced to match with a less similar treated unit — though still within the specified calipers — in order to satisfy the minimum and maximum ratio rules. In other cases, such constraints may improve effective sample size by redistributing how units are grouped. However, if the restrictions are too stringent, they can reduce effective sample size by forcing too many units to be discarded.

In applied settings, final choices of calipers and ratio constraints typically follow iterative checks of both covariate balance and effective sample size. Practitioners often compare several specifications to identify the most useful trade-off between these two goals. Hansen and Sales (2015) outline how this process can be carried out in a structured way, drawing on the stepwise-intersection-union principle (SIUP) of hypothesis testing.

How Do I Actually Form the Matches?

The simple matching example above — based on an exact match on geographic region and a caliper on Euclidean distance for a single covariate (Polity score) — serves to illustrate the basic ideas. When matching on many covariates, however, we will often prefer some combination of Mahalanobis distance and propensity score matching, sometimes adding calipers on specific covariates. In what follows, we use *rank-based* Mahalanobis distance, which has the advantage of being less sensitive to outliers and differences in scales across covariates (Rosenbaum, 2010). We further constrain the matching by imposing a caliper on the estimated propensity score, requiring treated and control observations to come from the same geographic region, and applying additional calipers directly to two covariates: ethnic fractionalization (`ethfrac`) and logged GDP per capita (`bwgdp`).

We impose a caliper equal to 0.5 standard deviations of the logit of the estimated propensity score (the linear covariate index defined above). This choice is larger than one rule of thumb emanating from Cochran and Rubin (1973), which recommends a caliper less than or equal to 0.20 standard deviations. Given that the standard deviation of the logit index is approximately 1.96, our choice of 0.5 permits treated and control units to differ by up to roughly 0.98 units on the logit scale, compared to about 0.39 under the 0.20 guideline.

```
# Add linear predictors from logistic regression (psm$linear.predictors) to dataset
data$logit_p_score <- lin_cov_inds

# Population standard deviation of logit_p_score (divides by n, not n - 1)
pop_sd_logit <- sqrt(mean((data$logit_p_score - mean(data$logit_p_score))^2))

# Distance matrix from propensity score (logit of estimated treatment probability)
ps_mat <- match_on(x = UN ~ logit_p_score,
                   caliper = 0.5 * pop_sd_logit,
                   data = data,
                   standardization.scale = NULL,
                   method = "euclidean")
```

Below we construct the distance matrix for rank-based Mahalanobis distance.

```
# Rank-based Mahalanobis distance on covariates
# Covs was defined earlier as the set of covariate names; here we drop "region"
rank_mah_mat <- match_on(
  x      = reformulate(termlabels = setdiff(x = covs, y = "region"),
                       response   = "UN"),
  data   = data,
  standardization.scale = NULL,
  method = "rank_mahalanobis" # Use rank-based Mahalanobis distance
)
```

Finally, we construct the Euclidean distance matrix for ethnic fractionalization (`ethfrac`) and logged GDP per capita (`bwgdp`) using calipers of 35 and 2, respectively. The exact-match constraint on region has already been defined through the object `em_region` above.

```
# Compute Euclidean distance matrix for ethnic fractionalization
eth_mat <- match_on(
  x = UN ~ ethfrac,
  caliper = 35,
  data = data,
  standardization.scale = NULL,
  method = "euclidean"
)

# Compute Euclidean distance matrix for logged GDP per capita
bwgdp_mat <- match_on(
  x = UN ~ bwgdp,
  caliper = 2,
  data = data,
  standardization.scale = NULL,
  method = "euclidean"
)
```

We then combine the `ps_mat`, `rank_mah_mat`, `eth_mat`, `bwgdp_mat`, and `em_region` objects to form the overall distance structure. We then pass the combined object to `optmatch`'s `fullmatch()` function, imposing a constraint that no more than 4 control units may be matched to any treated unit. If instead we wanted to perform pair matching, the `optmatch` package allows users to directly specify a matched-pair structure via the `pairmatch()` function. Equivalently, we could implement pair matching by setting `min.controls = 1` and `max.controls = 1` in the `fullmatch()` call. Below, we proceed with full matching under the constraint that no more than 4 controls may be matched to any treated unit.

```
# Full matching on PS + rank-based Mahalanobis + separate Euclidean distances
# (ethfrac, bwgdp) + region exact match
fm <- fullmatch(
  # x specifies the distance for matching: it can be a formula, a precomputed
  # distance matrix, or (as here) a sum of distance specifications from match_on()
  x            = ps_mat + rank_mah_mat + eth_mat + bwgdp_mat + em_region,
  data         = data,
  max.controls = 4  # Up to 4 controls per treated; min.controls = 0 by default
)
```

To calculate the effective sample size of the matched observations, we use the following function.

```
# Effective sample size of matched sets
effectiveSampleSize(fm)
```

```
[1] 15.26667
```

This reported effective sample size of approximately 15.27 is the sum across sets of the within-set harmonic mean of the number of treated and control subjects. This effective sample size reflects the matched structure in which no set contains more than 4 controls for any 1 treated observation.

```
# Summarize matched sets (set sizes, structure) and report effective sample size
summary(fm)
```

```
Structure of matched sets:
1:0 2:1 1:1 1:2 1:3 1:4 0:1 
  6   1   4   4   2   1  45 
Effective Sample Size:  15.3 
(equivalent number of matched pairs).
```

The notation in the `summary()` output of `optmatch` indicates the ratio of treated to control observations within each matched set. For example, `1:0` indicates 1 treated unit and no controls (effectively an unmatched treated unit). Similarly, `1:2` indicates 1 treated unit and 2 controls, and `0:1` indicates 1 control and no treated units (effectively an unmatched control). Below each label, the output shows how many matched sets have that particular structure.

If we examine the object (`fm`) returned by the matching call, we see that `optmatch` labels each observation according to its matched set, assigning `NA` to those not included. Because we performed exact matching by region, `optmatch` labels each matched set using the name of the exact-match stratum followed by a set index within that stratum. For example, the label `lamerica.1` denotes the first matched set within the Latin America stratum.

To see which units were matched together, we can add the `fm` object to the dataframe and then tabulate. For example, to view the `ssafrica.3` set, we run the following.

```r
# Add matched set ID to data for each unit
data$fm <- fm

# Look at one matched set ("ssafrica.3") for illustration
data |>
  filter(fm == "ssafrica.3") |>  # Keep only units in set "ssafrica.3"
# Display selected variables
  select(cname, UN, region, logit_p_score, ethfrac, bwgdp, bwplty2)
```

```
# A tibble: 4 x 7
  cname      UN region   logit_p_score ethfrac bwgdp bwplty2
  <chr>   <dbl> <fct>            <dbl>   <dbl> <dbl>   <dbl>
1 Burundi     0 ssafrica         0.508    3.55  5.35      -7
2 Rwanda      1 ssafrica         0.148   12.9   5.68      -7
3 Somalia     0 ssafrica         0.209    7.67  6.61      -7
4 Lesotho     0 ssafrica         0.558   22.2   6.28       0
```

We can see that the exact match on geographic region holds: All 4 countries are located in Sub-Saharan Africa. The logit of the estimated propensity score is similar across units, though not identical. The treated country, Rwanda, is matched to three controls — Burundi, Somalia, and Lesotho — and in each case the distance falls within 0.5 standard deviations of the estimated logit propensity scores (approximately 0.98), the caliper we specified. Likewise, Rwanda's distances to each of the 3 control units fall within the calipers of 35 for ethnic fractionalization (`ethfrac`) and 2 for logged GDP per capita (`bwgdp`). By contrast, distances between control countries may exceed these thresholds, since `optmatch` enforces calipers only between treated and control units, not among controls.

Some covariates used in the propensity score model and the rank-based Mahalanobis distance — such as Polity score (`bwplty2`) — still show modest imbalance within this matched set. This imbalance is unsurprising. We applied a caliper on the logit of the estimated propensity score and matched on the rank-based Mahalanobis distance including `bwplty2`, but we did not apply a caliper directly to `bwplty2`, though such a caliper could easily be added if desired.

How Do I Decide Whether to Move Ahead with My Matched Design?

Once we have constructed our matched sets, we want to evaluate the overall quality of the matched design. Covariate balance is an important aspect of this evaluation. To assess covariate balance, we compare the balance in our matched observational study with the balance we would expect to see in a completely randomized experiment within blocks (Hansen and Bowers, 2008).

Below we calculate adjusted covariate means for the treatment and control groups by averaging within matched sets, weighting each set by its contribution to the effective sample size. We also report standardized differences, defined as the adjusted mean difference scaled by the pooled standard deviation of the covariate across treated and control units, where the pooling and weighting are carried out under the matched-set design.

```r
# Install "RItools" package (only run if you don't already have it installed)
# Install.packages("RItools")

# Load RItools package for balance diagnostics (balanceTest)
library(RItools)
# Covariate balance test
cov_bal <- balanceTest(
  # Formula: treatment ~ covariates
  # update(): keep the original formula (. ~ .)
  # and add stratification by matched set, strata(fm)
  fmla = update(cov_fmla, . ~ . + strata(fm)),
  data = data,
  p.adjust.method = "none"
)
```

	Before matching			**After matching**		
Covariate	Control mean	Treated mean	Std. diff	Control mean	Treated mean	Std. diff
Log Cumulative Battle Deaths from Last War	6.65	8.98	0.84*	8.34	8.57	0.08
Duration of Last War	50.28	80.53	0.39	60.51	73.23	0.16
Ethnic Fractionalization	56.50	49.21	-0.28	51.75	57.50	0.22
Log Population Size	9.51	8.75	-0.64*	8.89	8.91	0.02
Log Mountainous	2.22	2.80	0.43	2.69	2.82	0.09
Log Military Personnel	3.87	3.25	-0.42	3.63	3.54	-0.06
Log GDP per Capita Before Last War	6.56	6.59	0.03	6.73	6.55	-0.17
Democracy (Polity Score) Before Last War	-0.84	-2.58	-0.32	-2.19	-2.54	-0.07
Asia	0.19	0.00	-0.68*	0.00	0.00	0.00
Eastern Europe	0.15	0.37	0.51*	0.46	0.46	0.00
Latin America	0.12	0.21	0.25	0.08	0.08	0.00
North Africa & Middle East	0.12	0.11	-0.04	0.08	0.08	0.00
Sub-Saharan Africa	0.43	0.32	-0.23	0.38	0.38	0.00

In the table above, the adjusted means offer a direct description of balance. The standardized differences place all covariates on a common scale, making imbalances comparable across variables. The stars indicate cases where the adjusted mean difference would be unusually extreme under complete random assignment within matched sets (based on a Normal approximation to the distribution of the adjusted mean differences). Because no adjustments are made for multiple comparisons, these stars are conservative, meaning they are, if anything, more likely than the nominal rate to detect a significant difference on a covariate.

One concern with balance tests is that high p-values may arise not from improved covariate balance but from the reduction in effective sample size that typically accompanies the matching process (Austin, 2008; Imai et al., 2008). As Hansen (2008b) notes, however, this possibility is less troubling than it first appears. The same increase in standard errors that produces high p-values for covariate balance tests will also carry over to subsequent causal inferences, meaning that those high p-values remain informative: They suggest that we are, if anything, less likely to overstate our causal conclusions than if the p-values had been significant.

Regardless of whether the balance tests are statistically significant, there are also established guidelines for what constitutes sufficient balance. While precise thresholds depend on context and substantive judgment about each covariate's importance, two commonly cited rules of thumb appear in Austin (2009) and Stuart (2010). Austin (2009) suggests that standardized differences of 0.1 or greater indicate inadequate balance on a covariate, whereas Stuart (2010), following Rubin (2001), proposes a more lenient threshold of 0.25. In our case, all covariates meet this latter standard, and none show statistically significant differences, indicating that observed imbalances would not be unusual under a completely randomized experiment within blocks (i.e., under as-if randomization).

In addition to assessing balance on each covariate individually, Hansen and Bowers (2008) also propose an omnibus test that evaluates balance across all covariates and their linear combinations simultaneously. We conduct this test below.

```r
# Extract overall chi-square balance test results, stratified by matched set (fm)
cov_bal$overall["fm", ]
```

```
   chisquare df   p.value
fm  2.631007  8 0.9553417
```

This χ^2 balance test yields an observed test statistic of 2.63 and a corresponding p-value of 0.96. Our high p-value indicates that the observed level of covariate balance is consistent with what we would expect in a completely randomized experiment within blocks. Despite the high p-value, there is no guarantee that balance is sufficient. Residual imbalance on observed covariates and hidden imbalance on unobserved ones may undermine the as-if randomization assumption. For now, we proceed under the as-if randomization assumption, but we will later assess how sensitive our inferences are to violations of it due to such imbalances.

Part 2: Causal Inference (under As-If Randomization)

After forming matched sets that ideally satisfy the as-if randomization assumption, researchers must decide which framework to use for inference. In the sharp framework, inference pertains to unit-level causal effects for every subject, thereby fully specifying the pattern of counterfactual outcomes. The weak framework, by contrast, pertains to a summary quantity of the unit-level causal effects, such as the ATE, and thus admits multiple configurations of individual effects consistent with it.

In both frameworks, the causal targets of inference are defined through potential outcomes. Under the stable unit treatment value assumption (SUTVA), each unit has two potential outcomes: a value the outcome would take if that unit were assigned to treatment and a value the outcome would take if that unit were assigned to control. Let $y_{si}(1)$ and $y_{si}(0)$ denote these potential outcomes for unit i in set s, where the index i runs over the $\{1, \ldots, n_s\}$ units in set s. The individual treatment effect is $\tau_{si} = y_{si}(1) - y_{si}(0)$. With $n = \sum_{s=1}^{S} n_s$ total units, let $\boldsymbol{\tau} = (\tau_{1,1}, \tau_{1,2}, \ldots, \tau_{S,n_s})^{\top}$ be the collection all n unit-level effects, and write the ATE as $\tau = (1/n) \sum_{s=1}^{S} \sum_{i=1}^{n_s} \tau_{si}$.

Note, however, that the matched design may alter the study population over which these causal targets are defined relative to the original set of units (Rosenbaum and Rubin, 1985; Rosenbaum, 2012). Units excluded from the matched design — because they lack suitable matches under the chosen calipers or exact-match constraints — are no longer part of the study population to which inference applies. Consequently, both $\boldsymbol{\tau}$ and τ refer only to the subpopulation of units retained after matching. This resulting study population is often difficult to interpret. For this reason, Rosenbaum (2010) cautions that "it is usually better to go back to the covariates themselves … perhaps redefining the population under study to be a subpopulation of the original population" (p. 86), a point echoed in Stuart (2010). Several approaches explicitly address how to define and interpret these study populations, including methods that trim units with extreme estimated propensity scores — where overlap is limited (Crump et al., 2009; King and Zeng, 2006) — as well as approaches developed for matching more generally (Rosenbaum, 2012; Fogarty et al., 2016; Traskin and Small, 2011).

The breadth of an interpretable study population with meaningful treatment–control comparisons depends in part on the relative abundance of treated and control units across covariate space. When either treatment arm is thinly represented in regions of covariate space dominated by the other, overlap may be confined to a restricted region of covariate space and include relatively few units. All else equal, a larger pool of units available for matching supports overlap over a wider region of covariate space and yields a larger, more interpretable study population over which the target of inference is defined.

Neither $\boldsymbol{\tau}$ nor τ can be directly calculated. Even in a randomized experiment, we cannot assign an individual to treatment, measure the outcome, then rewind time to assign the same individual to control and measure again. We therefore can observe only one potential outcome per unit. Denote this observed outcome by y_{si}, which is the treated potential outcome when individual i in set s is in the treatment condition or the control potential outcome when in the control condition. Because we observe only one of each unit's two potential outcomes, rather than the causal effect itself, we must rely on statistical inference.

We consider inference under the sharp and weak frameworks, targeting $\boldsymbol{\tau}$ and τ, respectively. Ongoing work shows how both types of effects can be inferred simultaneously under as-if randomization (Chung and Romano, 2013; Ding, 2017; Wu and Ding, 2021; Cohen and Fogarty, 2022), though they cannot generally be unified in sensitivity analyses (Fogarty, 2023). When researchers must choose between the two, the decision depends on both statistical properties and substantive goals.

- Statistically, the sharp framework specifies all missing potential outcomes, allowing exact randomization inference under minimal assumptions. The weak framework leaves some outcomes unspecified and instead relies on variance estimation and a Normal approximation, which can perform poorly in small samples or when outcomes are skewed with extreme outliers. In such cases — or whenever one wants exact p-values under minimal assumptions — permutation inference under the sharp framework may be preferable.
- Substantively, researchers usually test a constant effect in the sharp framework. Such a hypothesis may be unrealistic or of limited scientific interest (Gelman, 2003, 2011). The weak framework, by contrast, accommodates heterogeneous effects across units, making the ATE a more relevant target in many settings. Nevertheless, testing a constant effect can provide a useful approximation to a more complex hypothesis with heterogeneous effects (Rosenbaum, 2010, pp. 44–46), and such tests

remain valid for a range of bounded but heterogeneous effects (Caughey et al., 2023).

The Assignment Process as the Basis for Inference Regardless of the framework, inference is based on the treatment assignment process. Let z_{si} denote an indicator for whether unit i in matched set s is treated ($z_{si} = 1$) or not ($z_{si} = 0$). We collect these indicators for all units in set s into the vector $\boldsymbol{z}_s \coloneqq (z_{s1}, \dots, z_{sn_s})^\top$. Stacking these vectors across all sets gives the full assignment vector $\boldsymbol{z} = (z_{11}, \dots, z_{Sn_S})^\top$. For inference, we condition on the number of treated units within each set, even if the actual assignment mechanism consisted of n_s independent assignments. This conditioning represents a legitimate form of a "conditional as-if analysis" (Pashley et al., 2021).

We denote by Ω_s the set of all possible treatment assignments in set s that have a fixed number of treated units. Formally, Ω_s includes every possible way the n_s units in set s could be assigned to treatment and control such that exactly m_s units are treated. The number of possible assignments in Ω_s is denoted by $|\Omega_s|$, where the notation $|\cdot|$ indicates the number of elements in a set. This quantity equals $|\Omega_s| = \binom{n_s}{m_s} = \frac{n_s!}{m_s!(n_s - m_s)!}$, where "!" denotes the factorial operator (e.g., $4! = 4 \times 3 \times 2 \times 1$).

In the `ssafrica.3` set, for example, there are 4 observations — 1 treated and 3 control — so $|\Omega_s|$ for the `ssafrica.3` set is $\binom{4}{1} = 4$. The corresponding set of possible assignments with this treated count is shown in the table below.

	Assignment 1	Assignment 2	Assignment 3	Assignment 4
Burundi	0	0	0	1
Rwanda	0	0	1	0
Somalia	0	1	0	0
Lesotho	1	0	0	0

The column labeled Assignment 3 shows the assignment that actually occurred. The other possible assignments — Assignment 1, Assignment 2, and Assignment 4 — represent cases in which Lesotho, Somalia, or Burundi is treated instead of Rwanda. The set excludes any assignments with more than 1 treated unit.

The set of possible treatment assignments across all matched sets, given the number treated in each, is $\Omega \coloneqq \Omega_1 \times \ldots \times \Omega_S$, which is all the ways one assignment can be chosen from each Ω_s at the same time. Although the assignment itself can vary, the underlying causal quantities of interest — $\boldsymbol{\tau}$ and τ — remain fixed across all possible assignments. What changes from one assignment to another is which potential outcomes we actually observe. In the observed Assignment 3, we see Rwanda's treated potential outcome and the control potential outcomes of Burundi, Somalia, and Lesotho, but not Rwanda's control potential outcome or the treated potential outcomes of the others. Under a different assignment, a different set of treated and control potential outcomes would have been observed. No matter which assignment occurs, we observe only partial information about our causal targets.

Inference from the partial information contained in the data to our causal targets is predicated on a probability distribution defined over the set of assignments, Ω. This distribution constitutes the uncertainty underlying our inferences — what Fisher (1935, p. 14) famously called the "reasoned basis" for inference — but, unlike in a randomized experiment, this distribution is unknown in an observational study. Because this distribution is unknown, we must make assumptions about it when drawing inferences from observational data. Under the assumption of as-if randomization, every possible assignment within each Ω_s is equally likely, making each overall assignment in Ω equally likely as well. In this case, all individuals within the same matched set have the same probability of treatment. When as-if randomization does not hold, however, assignments are no longer equally likely, and some individuals have a higher probability of treatment than others.

How Do I Draw Inferences under the Sharp Framework?

To set the stage for inference under the sharp framework, consider a thought experiment. Suppose we were to subtract the true individual effects from the outcomes of the treated units. Doing so would yield, for each unit, the outcome it would have had under control. In other words, we may imagine reconstructing the dataset so that, under any possible treatment assignment, the outcomes would appear exactly as they would have if no one had been treated. In this reconstructed world, there would be no effect since every unit's outcome would reflect what it would have been without treatment.

Of course, we do not know the true collection of individual effects, $\boldsymbol{\tau}$, but we can test hypotheses about it, such as the hypothesis of a homogeneous effect for all units, denoted by τ_h. We do so by evaluating whether the data would look consistent with no treatment effect when that hypothesized value is subtracted from the treated outcomes. If, after this reconstruction, the data still show a positive effect, then the hypothesized value is presumably too small; if they show a negative effect, then the hypothesized value is presumably too large.

More formally, when outcomes are reconstructed under a null hypothesis, the observed test statistic will tend to fall in the upper tail of the null distribution if the hypothesized effect is too small, and in the lower tail if it is too large. To generate this null distribution, we reconstruct the outcomes under the hypothesized effect. Under the null, these reconstructed outcomes would remain fixed across assignments, so we hold them constant and recalculate the test statistic for every possible assignment.

A canonical choice of test statistic in this setting is a *sum statistic*, which first adds up the treated outcomes within each set, and then adds those set-level sums across all sets. Many familiar test statistics can be expressed in this form by applying to the outcomes scale and shift transformations that do not depend on the treatment assignments. One such useful test-statistic that can be expressed as a sum statistic is the Difference-in-Means, which is often the default choice for practitioners, computed within sets and then averaged across sets, with each set's contribution weighted by its effective sample size.

This harmonic-mean–weighted Difference-in-Means is useful because, as discussed earlier, assuming constant, additive treatment effects within a set, the variance of the within-set Difference-in-Means is minimized when the harmonic mean of the numbers of treated and control units is largest. Sets with larger harmonic means therefore yield more precise within-set comparisons, while highly lopsided sets contribute relatively little information. By consequence, a test will have greater power when it places more weight on sets in which the Difference-in-Means is most informative, rather than weighting sets solely by their share of units.

The harmonic-mean–weighted Difference-in-Means also has the practical advantage of coinciding with two other common approaches to analyzing matched data:

- First, the harmonic-mean–weighted Difference-in-Means equals the coefficient on the treatment indicator from a fixed-effects regression that includes matched-set indicators (Hansen and Bowers, 2008, pp. 228-229), an approach commonly used in practice after matching.
- Second, the harmonic-mean–weighted Difference-in-Means coincides with the overlap-weighted Difference-in-Means (Li et al., 2018) when overlap weights are constructed using the treatment assignment probabilities implied by as-if randomization. Overlap weights place the greatest weight on units with intermediate values of these assignment probabilities and downweight units with

extreme values. In particular, using the assignment probability under as-if randomization, control units are weighted by that probability, while treated units are weighted by one minus that probability. Because these assignment probabilities are constant within each matched set, the resulting overlap weights are constant within treated units and within control units in a set, and they aggregate exactly to the harmonic-mean weights implied by the matched strata.

To implement a sum statistic equivalent to the harmonic-mean–weighted Difference-in-Means for hypothesis testing, we first reconstruct the outcomes that treated units would have exhibited under the null hypothesis $\tau_h = 0$.

```r
# Keep only rows assigned to a matched set (drop NA in fm)
data_matched <- filter(.data = data, !is.na(fm))

# Null hypothesis value
tau_h <- 0

# Reconstruct outcomes under sharp null (tau_h = 0)
data_matched <- data_matched |>
  mutate(ldur_tilde = ldur - tau_h * UN)
```

We now source a helper function that rescales the outcome variable so that the sum of the rescaled values among treated units equals the harmonic-mean–weighted Difference-in-Means statistic. After sourcing the function, we apply it to the matched dataset to generate a new column containing the rescaled outcomes. We then use this rescaled outcome to compute the observed test statistic.

```r
# Load the hm_stat_rescale() function from the GitHub repo
# Base_url (defined earlier as
# "https://raw.githubusercontent.com/tl2624/matching-guide/main")
# Points to the main GitHub repo URL
source(paste0(base_url, "/R/hm_stat_rescale.R"))

# Apply the rescaling function: adds a new column (.hm_scaled)
# And returns the full matched dataset with this rescaled outcome
data_matched <- hm_stat_rescale(
  data = data_matched,  # Set dataset containing matched observations
  outcome = ldur_tilde, # Set outcome variable to be rescaled within matched sets
  treat = UN,           # Set name of treatment indicator variable
  strata = fm           # Set name of matched strata (block) variable
)

# Observed HM-weighted diff-in-means statistic
```

```r
obs_stat <- sum(data_matched$ldur_tilde_hm_scaled[data_matched$UN == 1])
```

We can verify in R that this observed sum statistic coincides with the harmonic-mean-weighted Difference-in-Means, the coefficient on the treatment indicator from a fixed-effects regression that includes matched-set indicators, and the overlap-weighted Difference-in-Means constructed from the assignment probabilities implied by as-if randomization.

```r
# Harmonic-mean-weighted difference in means (weights computed within sets)
dim_hm <- data_matched |>
  group_by(fm) |>
  summarize(
    n_treated = sum(UN == 1),
    n_control = sum(UN == 0),
    # Within-set Difference-in-Means
    dim_set   = mean(ldur_tilde[UN == 1]) - mean(ldur_tilde[UN == 0]),
    # Harmonic-mean weight (set's contribution to effective sample size)
    w_hm      = 2 * n_treated * n_control / (n_treated + n_control),
    .groups   = "drop"
  ) |>
  summarize(
    # Harmonic-mean-weighted average of within-set Differences-in-Means
    dim_hm = sum(w_hm * dim_set) / sum(w_hm)
  ) |>
  pull(dim_hm)

# Fixed-effects (FE) regression coefficient on UN (matched-set indicators as FE)
fe_fit <- lm(formula = ldur_tilde ~ UN + fm,
             data = data_matched)

# Install "PSweight" package (only run if you don't already have it installed)
# Install.packages("PSweight")
library(PSweight) # For overlap weights from Li et al (2018)

# Encode the treatment assignment probabilities implied by as-if randomization
# within matched sets: n_treated / n in each set
ps_fit <- PSmethod(
  ps.formula = UN ~ factor(fm),    # Reproduces n_treated / n in each set
  method     = "glm",              # Logistic link for binary treatment
  data       = as.data.frame(data_matched),
  ncate      = 2L                  # Binary treatment (treated vs. control)
)

assign_prob <- ps_fit$e.h[, "1"]  # Column corresponding to treatment level "1"

# Unit-level overlap weights implied by these assignment probabilities
# Control units get weight equal to their assignment probability
# Treated units get weight equal to one minus their assignment probability
w_ow <- ifelse(
  test = data_matched$UN == 1,
  yes  = 1 - assign_prob,
  no   = assign_prob
)

# Overlap-weighted difference in means
dim_ow <- data_matched |>
  dplyr::summarize(
    # Overlap-weighted mean outcome for treated units
    treated_weighted_mean =
      sum(w_ow[UN == 1] * ldur_tilde[UN == 1]) / sum(w_ow[UN == 1]),

    # Overlap-weighted mean outcome for control units
    control_weighted_mean =
      sum(w_ow[UN == 0] * ldur_tilde[UN == 0]) / sum(w_ow[UN == 0]),

    # Difference between overlap-weighted treated and control means
    dim_ow = treated_weighted_mean - control_weighted_mean
  ) |>
  dplyr::pull(dim_ow)

# All three approaches coincide with the sum statistic
all.equal(
  dim_hm,
  dim_ow,
  unname(coef(fe_fit)["UN"]),  # Drop name so comparison is purely numeric
  obs_stat
)
```

```
[1] TRUE
```

Then, to generate the distribution to which we refer our observed sum statistic, we can hold the reconstructed and rescaled outcomes fixed and then calculate the sum statistic over all possible assignments holding the numbers of treated observations in each set fixed at their observed values. In our application, with 12 sets ranging in size from 2 to 5, the total number of assignments is 311040.

```
# For each matched set (fm), record:
# N   = total units in the set
# M = number treated (UN == 1)
block_ns <- data_matched |>
  group_by(fm) |>    # Group results by key variables
  summarise(         # Aggregate to one row per group
    n = n(),         # Row count per group
    m = sum(UN),
    .groups = "drop" # Drop grouping after summarise
  )

# Total possible treatment assignments = product of binomial coefficients
# (choose n_s units for treatment in each set and multiply across sets)
prod(choose(n = block_ns$n, k = block_ns$m))
```

Because our matched study is relatively small, we can enumerate all possible treatment assignments exactly.

```
# Install "randomizr" (only run if not already installed)
# Install.packages("randomizr")

# Load randomizr for generating random assignments
library(randomizr)

exact_assigns <- obtain_permutation_matrix(
  declaration = declare_ra(   # Declare assignment procedure
    N = nrow(data_matched),   # Total number of units
    blocks = data_matched$fm, # Matched set membership
    block_m = block_ns$m      # Number treated in each set
  ),
# Total number of feasible assignments across all matched sets
  maximum_permutations = prod(choose(n = block_ns$n, k = block_ns$m))
)
```

However, in most applications, exactly enumerating all possible assignments is computationally infeasible. Instead, we typically draw a random subset of assignments (e.g., 10,000) to approximate the exact randomization distribution, as illustrated below.

```
# Set RNG seed for reproducibility
set.seed(11242017)

# Generate permutation matrix of treatment assignments
# Each column = one possible assignment consistent with block structure
sim_assigns <- obtain_permutation_matrix(
  declaration = declare_ra(
    N = nrow(data_matched),
    blocks = data_matched$fm,
    block_m = block_ns$m
  ),
  maximum_permutations = 10^4  # Cap at 10,000 random draws
)
```

Now, to generate the null distribution of the sum statistic, we apply the statistic to each of these 10,000 assignments while holding the reconstructed and rescaled outcomes fixed under the null.

```r
# Randomization distribution under sharp null of no effect:
# Apply sum statistic to each assignment column in 'assigns'
# Outcome has been transformed so that
# Sum statistic = harmonic-mean weighted diff in means
sim_sharp_null_dist <- apply(
  X = sim_assigns,     # Matrix of treatment assignments
  MARGIN = 2,          # Iterate over columns (assignments)
  FUN = function(x) {
    # Sum transformed outcomes among treated
    sum(data_matched$ldur_tilde_hm_scaled[x == 1])
  }
)
```

From this null distribution, we can now compute a one-sided, upper p-value as follows.

```r
# One-sided, upper p-value: proportion of simulated randomization stats >= observed
round(x = mean(sim_sharp_null_dist >= obs_stat), digits = 4)
```

```
[1] 0.0346
```

The upper p-value of a test of the sharp null of no effect against the alternative of a larger effect is 0.0346.

In this particular case — unlike in most applications — the matched study is small enough to enumerate all possible assignments exactly, allowing us to compute the exact p-value and assess the accuracy of the simulation-based approximation.

```r
# Randomization distribution under sharp null of no effect:
# Apply sum statistic to each assignment column in 'assigns'
# Outcome has been transformed so that
# Sum statistic = harmonic-mean weighted diff in means
exact_sharp_null_dist <- apply(
  X = exact_assigns,
  MARGIN = 2,
  FUN = function(x) {
    sum(data_matched$ldur_tilde_hm_scaled[x == 1])
  }
)

# Exact one-sided, upper p-value: proportion of randomization stats >= observed
round(x = mean(exact_sharp_null_dist >= obs_stat), digits = 4)
```

```
[1] 0.0363
```

The exact p-value of 0.0363 is nearly identical to the simulation-based p-value of 0.0346. At the conventional significance level of $\alpha = 0.05$, we would reject the null hypothesis in favor of the alternative, regardless of whether we use the exact or simulation-based p-value.

As an alternative to randomly sampling assignments and computing the test statistic for each one, we can use a much faster Normal approximation to the null distribution when the matched design is sufficiently large. The approximation relies on closed-form expressions for the expected value and variance of a sum statistic derived in Rosenbaum and Krieger (1990). Using these expressions, we standardize the observed test statistic and then compare it to the standard Normal distribution, which gives us a corresponding p-value.

```r
# Install "senstrat" package (only run if you don't already have it installed)
# Install.packages("senstrat")

# Load senstrat for computing stratum-level null expectations/variances
# (Rosenbaum & Krieger 1990) and later sensitivity analysis
library(senstrat)

# Compute per-block null expectations and variances
per_block_moms <- data_matched |>
  group_by(fm) |>
  summarize(
    expect   = ev(
      sc = ldur_tilde_hm_scaled, # Transformed outcomes for the stratum
      z = UN,                    # Treatment indicator
      m = 1,                     # Number of "1"s in vector of hidden confounder
                                 # Irrelevant here since Gamma = 1
      g = 1,                     # Sensitivity parameter Gamma
      method = "RK"              # Use formula from Rosenbaum and Krieger (1990)
    )$expect,
    variance = ev(
      sc     = ldur_tilde_hm_scaled,
      z      = UN,
      m = 1,
      g      = 1,
      method = "RK"
    )$vari,
    .groups  = "drop"
  )

# Sum across blocks to get overall null expectation and variance
null_ev  <- sum(per_block_moms$expect)
null_var <- sum(per_block_moms$variance)
```

```r
# Standardized test statistic and one-sided Normal p-value (upper tail)
norm_upper_p_value <- pnorm(
  q = (obs_stat - null_ev) / sqrt(null_var), # Standardized statistic
  lower.tail = FALSE                          # Compute upper-tail probability
)
```

This Normal-approximation p-value of 0.0343 is close to the exact and simulation-based p-values of 0.0363 and 0.0346, respectively. All p-values lead to the same conclusion in which we reject the sharp null of no effect in favor of a larger effect.

To form a confidence set for a homogeneous treatment effect using a Normal approximation, one can invert the hypothesis test over a grid of constant-effect null hypotheses. We first construct a one-sided confidence set by retaining all null values that are not rejected by the upper-tail test at the chosen α-level. We then form a two-sided confidence set by allocating $\alpha/2 = 0.025$ to each tail, corresponding to the upper- and lower-tail rejection regions.

```r
# Significance level
alpha <- 0.05

# ----------------------------
# Upper-tail test (H1: tau > tau_0):
# The lower endpoint is the smallest null value tau_0 for which the
# upper-tail p-value is still >= alpha; any smaller tau_0 would be rejected.
cs_sharp_lower_one_sided <- c(
  lower = obs_stat - qnorm(1 - alpha) * sqrt(null_var),
  upper = Inf
)
# qnorm(1 - alpha): standard Normal critical value for upper one-sided test

cs_sharp_two_sided <- c(
  lower = obs_stat - qnorm(1 - alpha / 2) * sqrt(null_var),
  upper = obs_stat + qnorm(1 - alpha / 2) * sqrt(null_var)
)
# At the lower endpoint, the upper-tail one-sided p-value equals alpha/2;
# any smaller null value would be rejected by the two-sided test
# At the upper endpoint, the lower-tail one-sided p-value equals alpha/2;
# any larger null value would be rejected by the two-sided test
```

An analogous procedure applies when constructing confidence sets without relying on a Normal approximation, instead using a simulation-based approximation to the null randomization distribution.

```r
# Grid of constant-effect null values (sharp framework)
tau_grid <- seq(from = -0.02, to = 1.5, by = 0.0001)

# For each tau_0, compute upper- and lower-tail randomization p-values
p_mat <- sapply(X = tau_grid, FUN = function(tau_0) {

  # Shift outcomes under null: ldur_i - tau_0 * UN_i
  dat_tau <- hm_stat_rescale(
    # transform(): create copy of data_matched with shifted-outcome column
    data    = transform(data_matched,
                        ldur_tilde_shift = ldur - tau_0 * UN),
    outcome = ldur_tilde_shift,
    treat   = UN,
    strata  = fm
  )

  q <- dat_tau$ldur_tilde_shift_hm_scaled

  # Observed statistic under null tau_0
  obs_stat_tau <- sum(dat_tau$UN * q)

  # Randomization distribution via simulated assignments (defined above)
  sim_null_dist <- as.numeric(t(q) %*% sim_assigns)

  # Upper-tail and lower-tail randomization p-values
  p_upper <- mean(sim_null_dist >= obs_stat_tau)
  p_lower <- mean(sim_null_dist <= obs_stat_tau)

  c(upper_tail = p_upper,
    lower_tail = p_lower)
})

# Extract vectors of p-values
p_upper <- p_mat["upper_tail", ]
p_lower <- p_mat["lower_tail", ]

# Simulation-based confidence sets

# Upper-tail confidence set (inversion of H1: tau > tau_0):
# retain all tau_0 with upper-tail p >= alpha
cs_sharp_upper_tail_sim <- tau_grid[p_upper >= alpha]

# Lower bound of the upper-tail confidence set
cs_sharp_upper_tail_sim_bound <- min(cs_sharp_upper_tail_sim)

# Two-sided confidence set (inversion using alpha/2 in each tail):
# retain tau_0 only if neither one-sided test rejects at level alpha/2
cs_sharp_two_sided_sim <- tau_grid[
  p_upper >= alpha / 2 &
  p_lower >= alpha / 2
]

# Two-sided confidence set summarized by its bounds
cs_sharp_two_sided_sim_bounds <- c(
  lower = min(cs_sharp_two_sided_sim),
  upper = max(cs_sharp_two_sided_sim)
)
```

Likewise, for a point estimate, one could follow Hodges Jr. and Lehmann (1963) and Rosenbaum (1993) by identifying the null value that makes the observed sum statistic equal to its null expectation. With our test statistic, this would occur when the null equals the harmonic-mean weighted Difference-in-Means computed from the observed outcomes (roughly 0.673), which yields a null expectation of 0.

How Do I Draw Inferences under the Weak Framework?

When the target is the ATE, it can be written as $\tau = \sum_{s=1}^{S} (n_s/n)\tau_s$, a weighted average of the set-level ATEs with weights equal to each set's share of the total study size. Thus, a straightforward way to estimate the ATE is to compute the Difference-in-Means within each matched set, and then combine these set-level estimates using the same weights. Formally, the overall Difference-in-Means is $\hat{\tau} = \sum_{s=1}^{S} (n_s/n)\hat{\tau}_s$, where $\hat{\tau}_s$ is the Difference-in-Means in set s.

Under as-if randomization, the Difference-in-Means is an unbiased estimator of the ATE. Formally, the expected value of the estimator — i.e.„ the average of $\hat{\tau}$ taken over all treatment assignments consistent with the matched design and their associated probabilities — equals the true ATE. Nevertheless, although correct in expectation, the Difference-in-Means can vary substantially across assignments. In any given assignment, the estimate may lie far from the target ATE.

Because the estimator can fluctuate across different treatment assignments, it is important to quantify the typical squared distance between an estimate and the true ATE. To this end, Neyman (1923) introduced a particular variance estimator. Under as-if randomization, this estimator is exactly unbiased when individual treatment effects are homogeneous; otherwise, it is conservative, meaning its expected value is greater than or equal to the Difference-in-Means' true variance. In principle, we could apply this approach by estimating the variance of the Difference-in-Means within each matched set and then taking a weighted sum of these estimates (see, e.g., Miratrix et al., 2013).

In our setting, however, this approach is infeasible because each matched set contains only 1 treated or 1 control unit. The usual Neyman formula relies on computing sample variances separately within the treated and control groups of each set. Sample variance requires at least two observations because `var()` divides by the number of observations minus 1. As a result, we cannot compute the sample variance when there is only a single treated or control unit.

How, then, can we estimate the variance of the Difference-in-Means in a matched observational study? Pashley and Miratrix (2021) and Fogarty (2018) offer distinct solutions. Pashley and Miratrix (2021) propose two approaches, each valid under different conditions on the matched structure, while Fogarty (2018) develops a single method that applies more broadly to finely stratified designs.

The first approach in Pashley and Miratrix (2021), `hybrid_m`, requires at least two matched sets of each unique set size in the data. The second approach, `hybrid_p`, permits variation in set sizes as long as no single set contains half or more of the total study size. Our matched design meets the condition for the second approach, not the first. We therefore estimate the variance using `hybrid_p` through the `blkvar` package, the companion software for Pashley and Miratrix (2021).

```r
# Install blkvar (only run if not already installed)
# Install.packages("remotes")
# Remotes::install_github("lmiratrix/blkvar")

# Load blkvar for block randomization variance estimators
library(blkvar)

# Compute results with hybrid_p method
res <- block_estimator(
  Yobs = ldur,          # Observed outcomes
  Z = UN,               # Treatment indicator
  B = fm,               # Block (matched set) membership
  data = data_matched, # Dataset
  method = "hybrid_p"  # P-value method
)

# Extract variance estimate
res$var_est
```

```
[1] 0.1187282
```

We can also estimate the variance of the Difference-in-Means in a finely stratified design using the approach of Fogarty (2018). Rather than defining the function within the script, we load a custom R function (fine_strat_var_est()) from the accompanying replication files.

```r
# Load the fine_strat_var_est() function from the GitHub repo
source(paste0(base_url, "/R/fine_strat_var_est.R"))
```

The function takes as inputs the set-specific treatment effect estimates and the number of units in each set, and it returns a single scalar representing the estimated variance. After loading the function, we compute the set-specific estimates and corresponding set sizes, and then pass these two vectors as inputs to fine_strat_var_est().

```r
# Compute stratum sizes and stratum-specific differences in means
strat_stats <- data_matched |>
  group_by(fm) |>
  summarize(
    n = n(),                                # Stratum size
    diff_in_means = mean(ldur[UN == 1L]) - # Treated mean minus
      mean(ldur[UN == 0L]),                 # control mean
    .groups = "drop"
  )

# Apply Fogarty (2018/2023) variance estimator
fine_strat_var_est(
  strat_ns = strat_stats$n,                # Vector of stratum sizes
  strat_ests = strat_stats$diff_in_means  # Vector of stratum estimates
)
```

```
[1] 0.1163934
```

This variance estimate is nearly identical to the one we obtained using the hybrid_p approach of Pashley and Miratrix (2021).

Now that we have estimates of both the ATE and the variance of the ATE estimator, we can form a standardized test statistic by subtracting the expected Difference-in-Means implied by the null and dividing the result by the estimated standard error (the square root of the estimated variance). We then compare this statistic to a standard Normal distribution to calculate p-values. Below, we calculate the upper one-sided p-value for a test the null hypothesis that the ATE is 0 against the alternative that it is positive.

```r
pnorm(
  q = (res$ATE_hat - 0) / sqrt(res$var_est),
  lower.tail = FALSE
# By default: mean = 0 and sd = 1 (standard normal distribution)
)
```

```
[1] 0.03722157
```

We reject the weak null hypothesis that the ATE equals 0 in favor of the alternative that the ATE is greater than 0, using the conventional significance level of $\alpha = 0.05$.

As in the construction of confidence sets for a homogeneous treatment effect under the sharp framework, we form 95% confidence sets by inverting the corresponding hypothesis tests. We first construct a one-sided confidence set obtained by inverting the upper-tail test and retaining all null values that are not rejected at the $\alpha = 0.05$ level. We then construct a two-sided confidence set by allocating $\alpha/2 = 0.025$ to each tail, reflecting that each null value is assessed against both directions of departure.

```r
cs_weak_upper_tail <- c(
  lower = res$ATE_hat - qnorm(1 - alpha) * sqrt(res$var_est),
  upper = Inf
)

cs_weak_two_sided <- c(
  lower = res$ATE_hat - qnorm(1 - alpha / 2) * sqrt(res$var_est),
  upper = res$ATE_hat + qnorm(1 - alpha / 2) * sqrt(res$var_est)
)
```

Part 3: Sensitivity Analysis (Departures from As-If Randomization)

Up to this point, we have supposed a framework in which each observation's chance of a UN intervention is like flipping a weighted coin. Each coin flip is independent across observations, but the probability of landing tails (i.e., receiving treatment) can differ from one unit to another depending on its baseline characteristics. With matching, we make our crucial assumption that all units within a set are similar enough on these characteristics that their coins have the same probability of landing tails.

The as-if randomization assumption may fail because of imbalances on hidden covariates (or residual imbalances on observed covariates, though the sensitivity analysis to follow subsumes both within the same framework). To represent such imbalances, consider a single hidden covariate, $\boldsymbol{u} = (u_{11}, \dots, u_{Sn_S})^\top$

with each u_{si} constrained to lie in the interval from 0 to 1. Although the restriction that each element of $\boldsymbol{u}$ is between 0 and 1 may seem strong, Rosenbaum (2017, p. 300, fn. 33) shows that any departure from complete random assignment within blocks can be represented by such a $\boldsymbol{u}$ in the sense that as-if randomization would hold if we had access to and exactly matched on this $\boldsymbol{u}$.

Rosenbaum (1987) and Rosenbaum and Krieger (1990) then propose a model in which each unit's independent probability of assignment to treatment is given by

$$\pi_{si} := \frac{\exp\left[\kappa_s + \log(\Gamma) u_{si}\right]}{1 + \exp\left[\kappa_s + \log(\Gamma) u_{si}\right]}. \tag{3}$$

The parameter κ_s is a set-specific intercept that captures the baseline propensity for treatment shared by all units in matched set s before accounting for any differences in the hidden covariate. In other words, κ_s reflects the central idea in matching that, after forming matched sets homogeneous in observed covariates, all individuals within a set share the same treatment propensity based on those covariates. The parameter $\Gamma \geq 1$, by contrast, quantifies how strongly the hidden covariate u_{si} can alter treatment odds. When $\Gamma = 1$, all individuals in matched set s have the same probability of treatment, $\exp[\kappa_s]/(1 + \exp[\kappa_s])$. However, when $\Gamma > 1$, two individuals in the same set who differ in the hidden covariate may differ in their odds of treatment by as much as a factor of Γ.

An important point to reiterate is that we condition on assignments that belong to the set Ω, meaning that the number of treated units is fixed within each matched set. It turns out that this conditioning removes dependence on the set-specific baseline κ_s in the probability distribution over assignments in Ω. Eliminating this dependence is crucial because it allows us to characterize both as-if randomization and departures from it using just a single sensitivity parameter, Γ.

To build intuition for the sensitivity parameter Γ, note that the model in (3) implies the following restriction:

$$\frac{1}{\Gamma} \leq \frac{\pi_{si}\left(1 - \pi_{sj}\right)}{\pi_{sj}\left(1 - \pi_{si}\right)} \leq \Gamma \text{ for all } i,\ j \text{ and } s. \tag{4}$$

This restriction states that, within any set, no two units can differ in their odds of treatment by more than a factor of Γ. When $\Gamma = 1$, all units share the same odds of treatment, corresponding to as-if randomization. By contrast, larger values of Γ represent increasingly severe departures from this assumption.

Rosenbaum (1995, pp. 1424–1425) shows that the converse also holds: The restriction in (4) implies a model of the form in (3). For any collection of treatment probabilities satisfying the bound in (4), it is always possible to find values of $u_{si} \in [0, 1]$ and a scalar $\Gamma \geq 1$ such that the two formulations yield the same probability distribution over assignments in Ω. In other words, (4) describes a general restriction on treatment probabilities that does not assume any particular functional form. The logistic model in (3), by contrast, provides one convenient parametric representation of that general restriction.

Thus far, we have conducted inference assuming $\Gamma = 1$, meaning that all units within a matched set share the same treatment probability. Under this as-if randomization assumption, tests of both sharp and weak null hypotheses are valid; that is, the probability of rejecting the null does not exceed α. When we allow departures from as-if randomization, governed by the sensitivity parameter $\Gamma \geq 1$, we seek new p-values that remain valid in the same sense: The probability of a false rejection should not exceed α, regardless of the hidden covariate $\mathbf{u}$ or the potential outcomes consistent with the null. The way we ensure this validity, however, differs between sharp and weak nulls, which we consider in turn.

How Do My Inferences under the Sharp Framework Change under these Departures?

For a sharp null, if the hidden $\boldsymbol{u}$ were known, we could calculate the exact distribution of assignments consistent with the matched design for any given value of $\Gamma \geq 1$. This distribution would provide the correct reference for computing p-values under the null. In particular, the upper one-sided p-value could be obtained by summing the probabilities of all assignments whose test statistic under the null is at least as large as the observed statistic.

In practice, $\boldsymbol{u}$ is unknown, so, to ensure validity of our tests, we compute p-values under the worst-case choice of $\boldsymbol{u}$ — the one that makes the p-value as large as possible. Rejection under the worst-case choice of $\boldsymbol{u}$ guarantees rejection under the actual (but hidden) $\boldsymbol{u}$. Because a test based on the true $\boldsymbol{u}$ already controls the false rejection probability at or below α, and the worst-case test can only reject as often or less often, the false rejection probability under the worst-case choice of $\boldsymbol{u}$ must also be at most α.

Finding the Worst-Case Scenario of Hidden Confounding to Ensure Valid Inference Actually finding this worst-case $\boldsymbol{u}$ is difficult. As a first step, Rosenbaum and Krieger (1990) show that in the unmatched (i.e., two-sample) case, the vector $\boldsymbol{u}$ that maximizes the p-value under any fixed $\Gamma \geq 1$ must belong to a restricted set of possibilities, denoted by $\mathcal{U}_+$. Specifically, once the subjects are ordered from the largest to the smallest outcome, the possible worst-case $\boldsymbol{u}$ vectors all look the same: a sequence of 1s at the top followed by 0s below. We do not know *how many* 1s should precede the 0s, and this number determines which configuration of $\boldsymbol{u}$ yields the worst-case p-value. Fortunately, in an unmatched study, it is straightforward to enumerate all $n-1$ such candidate vectors, and then identify which one yields the largest p-value for a fixed $\Gamma \geq 1$.

Unfortunately, in matched designs, the overall worst-case vector $\boldsymbol{u}$ cannot be obtained by simply stitching together the worst-case vectors from each matched set. That is, the global worst-case is not just the collection of local worst-cases. Instead, $\mathcal{U}_+$ consists of $\prod_{s=1}^{S}(n_s - 1)$ total candidate vectors for the worst-case $\boldsymbol{u}$ — a quantity that quickly becomes infeasible to enumerate directly. For example, with only 20 matched sets of 4 units each, the number of candidate vectors already exceeds 3.5 billion.

Separable Approximation To address this challenge, Gastwirth et al. (2000) propose a practical shortcut called the *separable approximation.* The idea is to select, within each matched set, the worst-case $\boldsymbol{u}_s$ from the candidate sequences of 1s and 0s introduced above, choosing the one that maximizes the expected value of the test statistic for that set under the null. If more than one candidate yields the same expectation, the choice goes to the $\boldsymbol{u}_s$ that produces the larger variance of the test statistic under the null. The method is called "separable" because it then stitches together the choices of $\boldsymbol{u}_s$ made separately within each matched set, rather than searching over all possible combinations across sets. The resulting $\boldsymbol{u}$ may not give the exact worst-case p-value; yet in designs with many small matched sets, the error is negligible.

Taylor Series Approximation The separable approximation is a useful shortcut, but it can fail in designs with relatively few strata or with highly unbalanced strata sizes. In such cases, the $\boldsymbol{u}$ selected by the separable approximation may yield p-values that are smaller than the true worst-case p-value. Rosenbaum (2018) therefore introduces a refinement that guarantees valid inference.

The key idea is to reframe hypothesis testing in terms of a function that, for each configuration of hidden confounding $\boldsymbol{u} \in \mathcal{U}_+$, combines the gap between the null expectation and the observed test statistic with a corresponding Normal critical buffer. Concretely, for $\alpha = 0.05$ and corresponding critical value 1.64, this function is the expected value of the test statistic under $\boldsymbol{u}$ minus the observed test statistic, plus 1.64 times the standard deviation of the test statistic under $\boldsymbol{u}$. The function is nonpositive exactly when the observed test statistic lies at least 1.64 standard deviations above its null expectation, and positive otherwise. Therefore, the null hypothesis is rejected if and only if this function is nonpositive.

Crucially, this function is concave over $\mathcal{U}_+$, which means that a tangent line drawn at *any* $\boldsymbol{u} \in \mathcal{U}_+$ lies above the function *everywhere* on $\mathcal{U}_+$. Rosenbaum (2018) draws the tangent line at the configuration of $\boldsymbol{u}$ chosen by the separable approximation. Because the original function is concave, this tangent line provides a global upper bound on the function for all candidate configurations of hidden confounding. Consequently, instead of searching for the largest value of a curved function, we can search for the largest value of a straight line.

Straight lines are easy to work with because they decompose additively across strata. We can therefore determine, within each stratum separately, which hidden-bias pattern makes the tangent line as large as possible, and then add up these stratum-specific contributions. The resulting sum is the maximum value of the tangent line over all $\boldsymbol{u} \in \mathcal{U}_+$.

Because the original function is concave, if the maximum value of its tangent-line approximation over $\boldsymbol{u} \in \mathcal{U}_+$ is at or below zero, then the original function itself must be nonpositive for every $\boldsymbol{u} \in \mathcal{U}_+$. Therefore, we can safely base the decision to reject the null hypothesis or not on the Taylor approximation. Although valid, this procedure can be conservative: The tangent line may remain above zero even when the original function would be below zero at its true worst case. By contrast, the separable approximation can be liberal, as it may reject even when the worst-case value of the function is positive.

Conducting Sensitivity Analysis under the Worst-Case Scenario Below we report the upper one-sided p-value obtained under the separable approximation, along with an indication of whether the resulting decision to reject or not reject the null hypothesis agrees with the decision implied by the Taylor series approximation. Agreement or disagreement between the two methods is determined by the sign of the Taylor-series–based bound on the function described above that governs rejection of the null hypothesis. Disagreement can occur only when the separable approximation yields a p-value slightly below α, indicating rejection, while the Taylor series approximation yields a positive value, leading to non-rejection.

```r
# Grid of Gamma values
Gamma_vals <- seq(from = 1, to = 1.5, by = 0.0001)

# Collect results for each Gamma
sens_results <- lapply(
  X = Gamma_vals,
  FUN = function(g) {
    out <- senstrat(
      sc          = data_matched$ldur_tilde_hm_scaled, # outcome
      z           = data_matched$UN,                   # treatment indicator
      st          = data_matched$fm,                   # strata
      gamma       = g,                                 # Gamma
      alternative = "greater",
      detail      = TRUE
    )

    # Separable p-value
    p_sep <- as.numeric(out$Separable["P-value"])

    # Taylor-series margin (cvA)
    cvA <- as.numeric(out$lambda["Linear Taylor bound"])

    # Decisions
    sep_reject <- p_sep <= alpha          # separable rejects?
    tay_reject <- cvA <= 0                # Taylor rejects?

    data.frame(
      Gamma     = g,
      p_sep     = p_sep,
      cvA       = cvA,
      agree_tay = (sep_reject == tay_reject)
    )
  }
)

# Bind into a single data frame
sens_df <- do.call(rbind, sens_results)

# Smallest Gamma where the separable p-value is >= alpha
sens_value_sep <- min(sens_df$Gamma[sens_df$p_sep >= alpha])

# Taylor-series sensitivity value:
# smallest Gamma at which Taylor margin becomes positive (non-rejection)
sens_value_tay <- min(sens_df$Gamma[sens_df$cvA > 0])
```

For this matched dataset, the conclusions from the separable and Taylor series approximations agree for nearly all values of Γ, differing only over a narrow interval from 1.1361 to 1.1399. We can reject the sharp null of no effect under as-if randomization (i.e., $\Gamma = 1$), but this conclusion is sensitive to departures from that assumption. At a Γ of roughly 1.14, we are no longer able to reject the sharp null of no effect against the alternative of a larger effect at an α-level of 0.05.

The plot below illustrates this comparison and clarifies how the two approaches relate across values of Γ

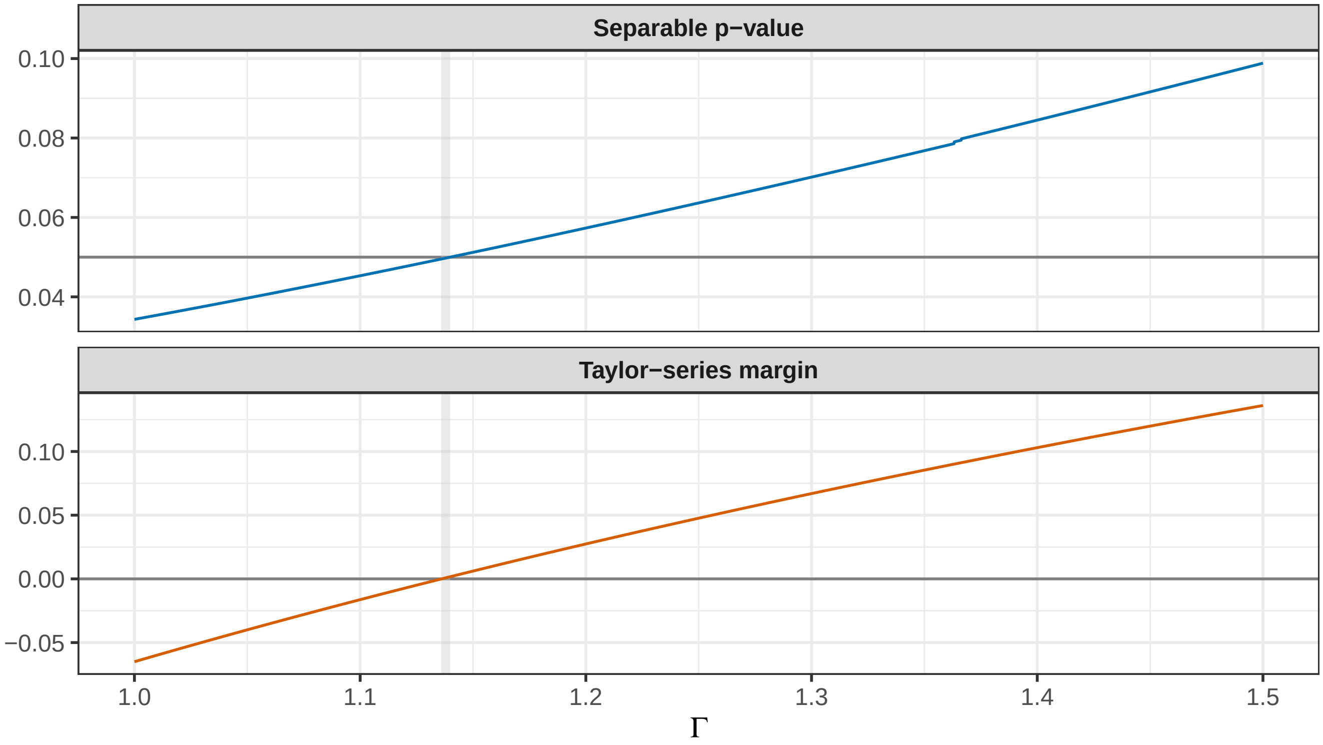


Figure 2: The top panel reports the upper one-sided p-value under the separable approximation, while the bottom panel reports the Taylor-series margin (with zero marking the rejection boundary). The shaded vertical band highlights values of Γ for which the separable approximation and the Taylor-series approximation lead to different rejection decisions.

These sensitivity analyses rely on multiple approximations. For example, the separable approximation is used to approximate the worst-case configuration of hidden bias $\boldsymbol{u} \in \mathcal{U}_+$, and inference additionally relies on a Normal approximation to the null randomization distribution of the test statistic. If the exact randomization distribution were used instead, the resulting sensitivity conclusions could differ for certain values of Γ. In principle, one could assess sensitivity at a given Γ using the exact randomization distribution by fixing a configuration of hidden bias $\boldsymbol{u}$ — for example, the separable choice — and deriving, from (3), the implied probability distribution on Ω, which can then be combined with the previously defined exact_sharp_null_dist to compute the corresponding p-value.

```
# Load helper that constructs all block-respecting candidate u vectors
source(paste0(base_url, "/R/construct_cand_u.R"))

# U_plus: n x K matrix, each column a candidate u vector in original row order
u_plus <- construct_cand_u(
  set      = data_matched$fm,                  # matched-set IDs
  y_scaled = data_matched$ldur_tilde_hm_scaled # scaled outcomes used for ordering
)$U
```

We now compute the exact worst-case upper-tail p-value for the specific value $\Gamma = rGamma$ by evaluating the upper p-value for every $\boldsymbol{u} \in \mathcal{U}_+$. For each candidate $\boldsymbol{u}$, the probability of each assignment in Ω is derived from the sensitivity model in (3).

```
Gamma <- 1.1161

# Log-odds sensitivity parameter
gamma <- log(Gamma)

# Number of candidate u vectors
K <- ncol(u_plus)

# Initialize worst-case p-value and the column index of the worst u
worst_p <- -Inf
worst_pos_u <- NA_integer_

# Loop over all block-respecting candidate u's
for (k in seq_len(K)) {

    # Candidate u_k (length n, original row order)
    u_vec <- u_plus[, k]

    # For each assignment z_j in Omega (columns of exact_assigns), compute u^T
    # z_j; result is a length-B vector
    zu <- as.numeric(crossprod(u_vec, exact_assigns))

    # Assignment probabilities under the sensitivity model: P(Z = z_j | u)
    # proportional to exp{ gamma * (u^T z_j) }
    w <- exp(gamma * zu)
    w <- w/sum(w)

    # Exact upper-tail p-value for this u_k
    p_val_k <- sum(w[exact_sharp_null_dist >= obs_stat])

    # Keep track of the largest (worst-case) p-value and its u_k
    if (p_val_k > worst_p) {
        worst_p <- p_val_k
        worst_pos_u <- k
    }
}

# Worst-case exact upper-tail p-value at this Gamma
worst_p
```

```
[1] 0.05001232
```

```
# Corresponding worst-case u vector (original row order)
exact_u_worst <- u_plus[, worst_pos_u]
```

At $\Gamma = 1.1161$, the worst-case exact upper-tail p-value already exceeds $\alpha = 0.05$, so the null hypothesis can no longer be rejected for this degree of departure from as-if randomization. Consequently, the previously reported changepoint — the smallest value of Γ at which the worst-case p-value reaches or exceeds α — of approximately 1.14 is too large.

To diagnose the source of this discrepancy, we next evaluate the Normal-approximation p-value at the true worst-case configuration $\boldsymbol{u} \in \mathcal{U}_+$. By holding fixed the reference distribution and varying only the choice of $\boldsymbol{u}$, this calculation isolates the role played by the search over hidden-bias configurations from the role played by the Normal approximation itself. Comparing the resulting p-value to the separable result therefore clarifies whether the discrepancy arises from approximation error in selecting $\boldsymbol{u}$ or from the Normal approximation to the randomization distribution.

```
# Initialize worst-case Normal-approximation p-value over u in U+
worst_p_norm <- -Inf

# Loop over all candidate u's
for (k in seq_len(K)) {

  # Candidate u_k (original row order)
  u_vec <- u_plus[, k]

  # Initialize total null mean and variance of the test statistic for this u
  null_mean <- 0
  null_var  <- 0

  # Loop over matched sets to accumulate block-level moments
  for (s in levels(data_matched$fm)) {
```

```r
    # Indices for units in stratum s
    rows_s <- which(data_matched$fm == s)

    # Number of ones in u within this stratum
    m_s <- sum(u_vec[rows_s])

    # RK block-level expectation and variance for given m_s and Gamma
    moms <- ev(
      sc     = data_matched$ldur_tilde_hm_scaled[rows_s],
      z      = data_matched$UN[rows_s],
      m      = m_s,
      g      = Gamma,
      method = "RK"
    )

    # Add block-level contributions to overall mean and variance
    null_mean <- null_mean + moms$expect
    null_var  <- null_var  + moms$vari
  }

  # Standardized test statistic under this u (Normal approximation)
  z_stat <- (obs_stat - null_mean) / sqrt(null_var)

  # Upper-tail Normal p-value for this u
  p_val_norm <- as.numeric(pnorm(z_stat, lower.tail = FALSE))

  # Track the largest (worst-case) Normal p-value across u
  if (p_val_norm > worst_p_norm) {
    worst_p_norm <- p_val_norm
  }
}

# Match the row in sens_df closest to this Gamma (guards against
# floating-point issues when comparing Gamma values)
gamma_row <- which.min(abs(sens_df$Gamma - Gamma))

# Compare Normal approximation at the true worst-case u to
# the separable and Taylor-series Normal approximations
data.frame(
  Gamma        = Gamma,
  p_worst_norm = worst_p_norm,                # Normal p-value at true worst-case u
  p_sep        = sens_df$p_sep[gamma_row]     # separable approximation p-value
)
```

```
   Gamma p_worst_norm      p_sep
1 1.1161   0.04755974 0.04718824
```

This comparison shows that, in this example, the Normal-approximation p-value evaluated at the true worst-case configuration $\boldsymbol{u}$ is essentially identical to the separable-approximation p-value. Thus, the difference between the exact and approximate sensitivity analyses is attributable to the Normal approximation to the randomization distribution, rather than to the separable approximation used to select $\boldsymbol{u}$.

How Do My Inferences under the Weak Framework Change under these Departures?

Compared to tests of the sharp null hypothesis of no effect, constructing valid tests of an analogous weak null hypothesis is more challenging. A sharp null hypothesis specifies all missing potential outcomes, so we do not have to consider p-values over many different configurations of potential outcomes. By contrast, a weak null permits many possible configurations of potential outcomes, so identifying the worst-case p-value over both $\boldsymbol{u}$ and the potential outcomes consistent with that null is often computationally intractable.

Fogarty (2023) proposes an alternative approach that yields valid tests for the ATE (τ) — at least in sufficiently large matched designs — without requiring an explicit search for the worst-case p-value across all configurations of $\boldsymbol{u}$ and potential outcomes consistent with the null hypothesis about τ. To build intuition for this approach, note that the following inverse probability weighted (IPW) version of the Difference-in-Means is unbiased for the ATE, τ:

$$\sum_{s=1}^{S}(n_s/n)\left(\frac{1}{|\Omega_s|}\right)\left(\frac{\hat{\tau}_s}{p(\boldsymbol{z}_s)}\right), \tag{5}$$

where $p(\boldsymbol{z}_s)$ denotes the probability of assignment $\boldsymbol{z}_s$ conditional on $\boldsymbol{z}_s \in \Omega_s$. In practice, both $\hat{\tau}_s$ and $p(\boldsymbol{z}_s)$ are evaluated at the same realized treatment assignment $\boldsymbol{z}_s$ within matched set s. In other words, $p(\boldsymbol{z}_s)$ denotes the probability of the assignment vector under which the observed treated and control outcomes in set s — and hence the Difference-in-Means $\hat{\tau}_s$ — were realized.

When $\Gamma = 1$, this probability $p(\boldsymbol{z}_s)$ is $1/|\Omega_s|$, and the IPW Difference-in-Means reduces to the usual unweighted Difference-in-Means, $\hat{\tau}$. When $\Gamma > 1$, however, $p(\boldsymbol{z}_s)$ is no longer known exactly. Instead, Γ constrains the set of possible assignment probabilities, conditional on $\boldsymbol{z}_s \in \Omega_s$, to lie within bounds consistent with the specified level of departure from as-if randomization:

$$\frac{1}{\Gamma(n_s - 1) + 1} \leq p(\boldsymbol{z}_s) \leq \frac{\Gamma}{(n_s - 1) + \Gamma}, \tag{6}$$

for all $\boldsymbol{z}_s \in \Omega_s$ and for all matched sets s.

Since the IPW Difference-in-Means in (5) cannot be directly computed when $\Gamma > 1$, Fogarty (2023) instead uses the bounds in (6) to construct a worst-case version of this IPW Difference-in-Means. For tests of a null hypothesis about the overall ATE across all matched sets, the procedure replaces $p(\boldsymbol{z}_s)$ with the upper bound from (6) whenever $\hat{\tau}_s$ is greater than or equal to the null value of the overall ATE, and with the lower bound whenever $\hat{\tau}_s$ is less than that null value. When testing the null against a smaller alternative, the procedure reverses these substitutions.

To see the value of this procedure, consider testing a null hypothesis about the ATE against the alternative of a larger ATE. Fogarty (2023) shows that, for any $\Gamma \geq 1$, the expected value of this worst-case IPW Difference-in-Means — defined under whatever the true distribution on Ω consistent with that Γ happens to be — is always less than or equal to the null when it is true. Conversely, when testing against the alternative of a smaller ATE, this expectation is greater than or equal to the null when it is true. Importantly, these properties hold for all possible values of $\boldsymbol{u}$ and all configurations of potential outcomes that are consistent with the null hypothesis.

How does this property of the expected value ensure a valid test in sufficiently large studies? Consider testing the null hypothesis against the alternative of a larger ATE. If the expected value of the worst-case IPW Difference-in-Means is less than or equal to the null when it is true, then the worst-case IPW Difference-in-Means tends to fall at or below the null rather than above it. In other words, the procedure intentionally "tilts" the worst-case IPW Difference-in-Means in the direction opposite the alternative, making the procedure conservative.

Now note that an analogue of the variance estimator from Fogarty (2018) discussed above satisfies an important property. For any fixed $\Gamma \geq 1$, this estimator consistently overestimates (or bounds above) the true variance of the worst-case IPW Difference-in-Means. When we standardize the worst-case IPW Difference-in-Means — by dividing its deviation from the null by the square root of this variance estimate — the resulting value tends to be smaller than it would be under the true (but unknown) variance.

Together, these two properties work in the same direction:

- The expected value is at or below the null, keeping the center of the distribution at or shifted away from the upper tail, and
- the variance estimate tends to be too large, which pulls the standardized value closer to zero.

As a result, the probability that the standardized statistic falls in the upper tail of the standard Normal distribution — that is, the probability of rejecting the null when it is true — remains at or below the nominal significance level. Hence, the test is conservative (valid) in large studies. Analogous reasoning applies when testing against the alternative of a smaller effect: In that case, the probability that the standardized value falls in the lower tail of the standard Normal distribution also stays at or below the nominal significance level.

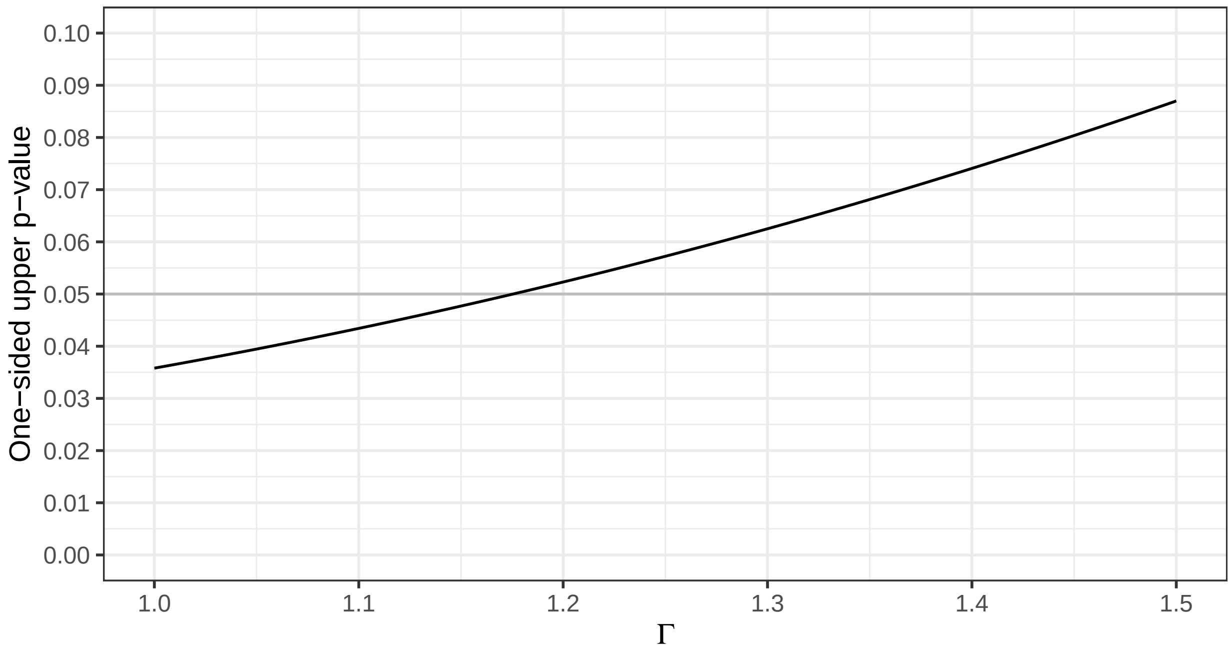
To implement this approach from Fogarty (2023), we first source an `R` function that, for a fixed $\Gamma \geq 1$, computes the worst-case IPW Difference-in-Means for a single matched set.

```r
# Load the worst_case_IPW() function from the GitHub repo
source(paste0(base_url, "/R/worst_case_IPW.R"))
```

We now use this function to calculate the worst-case IPW version of the Difference-in-Means within each matched set. To form the overall statistic, we take a weighted average of the set-specific values, using weights proportional to the number of units in each set. Finally, we standardize this overall statistic using the conservative variance estimator from Fogarty (2018), and then compare the resulting standardized test statistic to the standard Normal distribution to obtain p-values.

```r
# For each Gamma, compute one-sided weak-null p-value
weak_results_list <- lapply(
  X = Gamma_vals,
  FUN = function(G) {

# Compute per-set worst-case IPW; keep as list to preserve attributes
    strat_stats <- data_matched |>
      group_by(fm) |>
      summarise(
        n = n(),
        prop_n = n / nrow(data_matched),      # Weight n_s / n
        est = list(worst_case_IPW(             # Keep as list()
          z = UN,                             # Set treatment variable
          y = ldur,                           # Set outcome variable
          Gamma = G,                          # Set Gamma value
          tau_h = 0,                          # Set null hypothesis
          alternative = "greater"             # One-sided (upper-tail) test
        )),
        .groups = "drop"
      ) |>
      mutate(
        alt = attr(est[[1]], "alternative"),  # Read attribute once
        est = as.numeric(est)
      )

# Weighted statistic and variance
    num     <- sum(strat_stats$prop_n * strat_stats$est)
    var_hat <- fine_strat_var_est(
      strat_ns   = strat_stats$n,
      strat_ests = strat_stats$est
    )

# Z: only if variance finite and positive; else NA
    denom <- sqrt(var_hat)
    z <- if (is.finite(denom) && denom > 0) num / denom else NA_real_

# Tail from preserved attribute (same for all rows)
    alt <- strat_stats$alt[1]
    lower_tail <- if (identical(alt, "greater")) FALSE else TRUE

# One-sided p-value
    pval <- if (is.na(z)) NA_real_ else pnorm(z, lower.tail = lower_tail)

    data.frame(Gamma = G, p_value = pval, stringsAsFactors = FALSE)
  }
)

# Bind rows: final results data.frame for plotting/reporting
weak_sens_df <- do.call(what = rbind, args = weak_results_list)

# Sensitivity value: smallest Gamma with p-value >= alpha
weak_sens_value <- min(weak_sens_df$Gamma[weak_sens_df$p_value >= alpha])
```

Below, we plot the upper one-sided p-values corresponding to Γ values from 1 to 1.5.



Under as-if randomization ($\Gamma = 1$), we find evidence of a positive average effect. This conclusion about the weak null is slightly more robust to departures from as-if randomization than our earlier rejection of the sharp null of no effect at $\Gamma = 1$. However, in absolute terms, the evidence for a positive ATE remains sensitive. Our qualitative conclusion about the null hypothesis that the ATE equals zero changes once Γ reaches 1.1756.

It is important to emphasize that this sensitivity analysis guarantees validity across all configurations of potential outcomes consistent with the weak null. By contrast, Fogarty (2023) also describes an alternative sensitivity analysis that restricts attention to subsets of configurations that researchers may regard as more plausible. Such alternatives can produce smaller p-values, but at the cost of forfeiting validity for certain (possibly unrealistic) outcome configurations. Other sensitivity analyses for tests of weak nulls directly optimize — over the space of the hidden confounder and potential outcomes consistent with the null — the worst-case p-value for a given $\Gamma \geq 1$, an optimization that is tractable in restricted settings such as binary outcomes (Fogarty et al., 2017) or matched-pair designs (Fogarty, 2020).

Conclusion

We have now completed the full matching pipeline. We began by constructing matched sets and then conducted inference under the as-if randomization assumption, treating the matched design as a collection of completely randomized experiments within blocks. We examined inference under both sharp and weak null frameworks — corresponding, respectively, to unit-level causal effects and average effects — and then assessed the sensitivity of these inferences to potential violations of the as-if randomization assumption. The embedded code, accompanied by extensive comments and illustrated through the running example from Gilligan and Sergenti (2008), is intended to be readily adapted by practitioners to their own datasets.

Limitations and Related Topics Beyond Our Scope

For clarity and focus, we have excluded several important topics, though we have pointed to relevant references where appropriate. Some of these omissions are not specific to matching or observational studies. In particular, we have not addressed imperfect compliance with treatment assignment, missing outcomes, clustered (rather than individual-level) assignment, or interference settings in which a subject's outcome depends on others' treatment assignments.

Our analysis has focused on inference for two causal targets only: a constant, additive treatment effect and the average treatment effect. We have not considered alternative effect models, such as dilated effects (Rosenbaum, 1999), multiplicative effects, or Tobit effects (Rosenbaum, 2010, pp. 46–49). Throughout, we have treated causal targets as fixed quantities over the set of treatment assignments consistent with the matched design, rather than as random variables. As a result, we have excluded attributable effects (Rosenbaum, 2001, 2003; Hansen and Bowers, 2009) and the average treatment effect on the treated (Sekhon and Shem-Tov, 2021). We have also set aside methods for joint inference on sharp and weak causal hypotheses (Ding, 2017; Wu and Ding, 2021; Cohen and Fogarty, 2022), as well as strategies that improve the power of hypothesis tests by rescaling outcomes or choosing alternative test statistics, including regression-assisted approaches (Lin, 2013; Cohen and Fogarty, 2023; Guo and Basse, 2023).

Several omissions are specific to matching methodology itself:

- First, we have not covered a range of **alternative matching and balancing approaches**, including nonbipartite matching for multivalued treatments (Lu et al., 2001, 2011; Rosenbaum, 2010, pp. 207–221), template matching (Silber et al., 2014), multilevel matching (Zubizarreta and Keele, 2017; Pimentel et al., 2018), risk-set matching (Li et al., 2001), cardinality matching

(Zubizarreta et al., 2014), coarsened exact matching (Iacus et al., 2012), tapered matching (Daniel et al., 2008), or generalized full matching (Sävje et al., 2021) among others. We have also omitted balance constraints such as fine and near-fine balance (Rosenbaum et al., 2007; Yang et al., 2012; Yu, 2023; Pimentel et al., 2015a,b), which enforce exact or nearly exact equality of marginal covariate distributions. Although these methods are beyond our scope, the design-based principles and pipeline we have presented apply broadly and can be adapted to these settings.

- Second, we have not discussed the use of **prognostic covariates** identified using pilot or external data. Such approaches can substantially improve efficiency by prioritizing covariates that strongly predict outcomes. For theoretical foundations, see Hansen (2008a), with complementary theoretical results and practical insights in Sales et al. (2018).
- Third, we have considered only matching strategies aimed at justifying as-if randomization, not those designed explicitly to improve **design sensitivity** — that is, robustness to moderate violations of as-if randomization. Beyond match construction, researchers can also enhance design sensitivity by selecting particular test statistics, among other strategies (Rosenbaum, 2004; Heller et al., 2009; Hsu et al., 2013; Small et al., 2013).
- Fourth, we have excluded **extensions to covariate balance testing** beyond the framework developed in Hansen and Bowers (2008), including subsequent contributions by Gagnon-Bartsch and Shem-Tov (2019) and Branson (2021), as well as related approaches based on the stepwise intersection–union principle (Hansen and Sales, 2015).
- Fifth, we have ignored **residual imbalance on observed covariates**. Specifically, we have proceeded under the assumption that matched set membership can justify as-if randomization, even when small imbalances remain within matched sets. Under this approach, concerns about residual imbalance are absorbed into the sensitivity analysis for hidden bias. This approach is not entirely satisfying, however, because imbalances on observed covariates are not, in fact, hidden. Several methods therefore address such imbalances directly, either in approaches targeting individual effects (Rosenbaum, 1988; Pimentel and Huang, 2024; Chen et al., 2023; Heng et al., 2025) or average effects (Zhu et al., 2025).
- Sixth, we have not incorporated into the overall pipeline an explicit approach for defining an **interpretable study population** in matched designs. This omission matters because discarding units during matching can alter the population over which the causal estimand is defined (Rosenbaum

and Rubin, 1985; Rosenbaum, 2012). Common practice excludes units based on limited overlap in the estimated propensity score (Crump et al., 2009; King and Zeng, 2006), but study populations defined in this way can be difficult to characterize substantively. This concern has motivated alternative approaches that define the study population directly in terms of a small number of covariates. For example, Traskin and Small (2011) use a classification tree to define a study population that approximates one defined by trimming on the estimated propensity score, while Fogarty et al. (2016) use discrete optimization to define a maximal, rectangular region of covariate space with adequate overlap on key covariates.

- Finally, have not addressed the possibility that matched set membership itself may depend on the treatment assignments — that is, settings with ***Z*-dependence** (Pashley et al., 2021; Pimentel and Huang, 2024; Pimentel and Yu, 2024). Our exposition has relied on an analogy to Rubin's framework of "assignment to treatment on the basis of a covariate" (Rubin, 1977), where matched set membership plays the role of the covariate. This analogy assumes that membership is fixed, but in practice it could be a random variable if the matched structure depends on the observable assignments. We have set aside this issue, which may be minor when practitioners use sufficiently tight calipers for matching.

Despite these limitations, the pipeline and design-based perspective developed here provide a coherent framework for understanding how matched designs support causal inference. The emphasis on design-based foundations clarifies the logic underlying the construction of matched sets and the subsequent steps of inference and sensitivity analysis under both weak and sharp frameworks. We present this work not as exhaustive, but as a foundation on which researchers can build when incorporating more specialized methods tailored to their substantive and inferential goals.

References

Donald B Rubin. Assignment to treatment group on the basis of a covariate. *Journal of Educational Statistics*, 2(1):1–26, 1977.

Ben B Hansen. Full matching in an observational study of coaching for the SAT. *Journal of the American Statistical Association*, 99(467):609–618, 2004.

Ben B Hansen and Stephanie Olsen Klopfer. Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15(3):609–627, 2006.

Ben B Hansen and Jake Bowers. Covariate balance in simple, stratified and clustered comparative studies. *Statistical Science*, 23(2):219–236, 2008.

Colin B Fogarty. On mitigating the analytical limitations of finely stratified experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):1035–1056, 2018.

Nicole E Pashley and Luke W Miratrix. Insights on variance estimation for blocked and matched pairs designs. *Journal of Educational and Behavioral Statistics*, 46(3):271–296, 2021.

Xing Sam Gu and Paul R Rosenbaum. Comparison of multivariate matching methods: Structures, distances, and algorithms. *Journal of Computational and Graphical Statistics*, 2(4):405–420, 1993.

Paul R Rosenbaum. A characterization of optimal designs for observational studies. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):597–610, 1991.

Paul R Rosenbaum and Abba M Krieger. Sensitivity of two-sample permutation inferences in observational studies. *Journal of the American Statistical Association*, 85(410):493–498, 1990.

Joseph L Gastwirth, Abba M Krieger, and Paul R Rosenbaum. Asymptotic separability in sensitivity analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):545–555, 2000.

Paul R Rosenbaum. Sensitivity analysis for stratified comparisons in an observational study of the effect of smoking on homocysteine levels. *Annals of Applied Statistics*, 12(4):2312–2334, 2018.

Colin B Fogarty. Testing weak nulls in matched observational studies. *Biometrics*, 79(3):2196–2207, 2023.

Michael J Gilligan and Ernest J Sergenti. Do UN interventions cause peace? Using matching to improve causal inference. *Quarterly Journal of Political Science*, 3(2):89–122, 2008.

Donald B Rubin and Richard P Waterman. Estimating the causal effects of marketing interventions using propensity score methodology. *Statistical Science*, 21(2):206–222, 2006.

Ben B Hansen. The prognostic analogue of the propensity score. *Biometrika*, 95(2):481–488, 2008a.

Adam C Sales, Ben B Hansen, and Brian Rowan. Rebar: Reinforcing a matching estimator with predictions from high-dimensional covariates. *Journal of Educational and Behavioral Statistics*, 43(1): 3–31, 2018.

Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of Indian National Science Academy*, 2(1):49–55, 1936.

A Albert and J A Anderson. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71(1):1–10, 1984.

Ben B Hansen. Propensity score matching to extract latent experiments from nonexperimental data: A case study. In Neil J. Dorans and Sandip Sinharay, editors, *Looking Back: Proceedings of a Conference in Honor of Paul W. Holland*, volume 202 of *Lecture Notes in Statistics*, chapter 9, pages 149–181. Springer, New York, NY, 2011.

Ben B Hansen and Adam Sales. Comment on cochran's "observational studies". *Observational Studies*, pages 184–193, 2015.

Paul R Rosenbaum. *Design of Observational Studies*. Springer, New York, NY, 1st edition, 2010.

William G Cochran and Donald B Rubin. Controlling bias in observational studies: A review. *Sankhyā: The Indian Journal of Statistics, Series A (1961–2002)*, 35(4):417–446, 1973.

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

Kosuke Imai, Gary King, and Elizabeth A Stuart. Misunderstandings between experimentalists and observationalists about causal inference. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 171(2):481–502, 2008.

Ben B Hansen. 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*, 27(12):2050–2054, 2008b.

Peter C Austin. Balance diagnostics for comapring the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine*, 28(25):3083–3107, 2009.

Elizabeth A Stuart. Matching methods for causal inference: A review and a look forward. *Statistical Science*, 25(1):1–21, 2010.

Donald B Rubin. Using propensity scores to help design observational studies: Application to the tobacco litigation. *Health Services and Outcomes Research Methodology*, 2(3):169–188, 2001.

Paul R Rosenbaum and Donald B Rubin. The bias due to incomplete matching. *Biometrics*, 41(1): 103–116, 1985.

Paul R Rosenbaum. Optimal matching of an optimally chosen subset in observational studies. *Journal of Computational and Graphical Statistics*, 21(1):57–71, 2012.

Richard K Crump, V Joseph Hotz, Guido W Imbens, and Oscar A Mitnik. Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96(1):187–199, 2009.

Gary King and Langche Zeng. The dangers of extreme counterfactuals. *Political Analysis*, 14(2):131–159, 2006.

Colin B Fogarty, Mark E Mikkelsen, David F Gaieski, and Dylan S Small. Discrete optimization for interpretable study populations and randomization inference in an observational study of severe sepsis mortality. *Journal of the American Statistical Association*, 111(514):447–458, 2016.

Mikhail Traskin and Dylan S Small. Defining the study population for an observational study to ensure sufficient overlap: A tree approach. *Statistics in Biosciences*, 3:94–118, 2011.

Eun Yi Chung and Joseph P Romano. Exact and asymptotically robust permutation tests. *The Annals of Statistics*, 41(2):484–507, 2013.

Peng Ding. A paradox from randomization-based causal inference. *Statistical Science*, 32(3):331–345, 2017.

Jason Wu and Peng Ding. Randomization tests for weak null hypotheses. *Journal of the American Statistical Association*, 116(536):1898–1913, 2021.

Peter L Cohen and Colin B Fogarty. Gaussian prepivoting for finite population causal inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(2):295–320, 2022.

Andrew Gelman. A Bayesian formulation of exploratory data analysis and goodness-of-fit testing. *International Statistical Review*, 71(2):369–382, 2003.

Andrew Gelman. Review: Causality and statistical learning. *American Journal of Sociology*, 117(3): 955–966, 2011.

Devin Caughey, Allan Dafoe, Xinran Li, and Luke Miratrix. Randomisation inference beyond the sharp null: Bounded null hypotheses and quantiles of individual treatment effects. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 85(5):1471–1491, 2023.

Nicole E Pashley, Guillaume W Basse, and Luke W Miratrix. Conditional as-if analyses in randomized experiments. *Journal of Causal Inference*, 9(1), 2021.

Ronald Aylmer Fisher. *The Design of Experiments*. Oliver and Boyd, Edinburgh, SCT, 1935.

Fan Li, Kari Lock Morgan, and Alan M Zaslavsky. Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113(521):390–400, 2018.

Joseph L Hodges Jr. and Erich L Lehmann. Estimates of location based on rank tests. *The Annals of Mathematical Statistics*, 34(2):598–611, 1963.

Paul R Rosenbaum. Hodges–Lehmann point estimates of treatment effect in observational studies. *Journal of the American Statistical Association*, 88(424):1250–1253, 1993.

Jersey Neyman. Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes. *Roczniki Nauk Rolniczych*, 10:1–51, 1923.

Luke W Miratrix, Jasjeet S Sekhon, and Bin Yu. Adjusting treatment effect estimates by post-stratification in randomized experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(2):369–396, 2013.

Paul R Rosenbaum. *Observation and Experiment: An Introduction to Causal Inference*. Harvard University Press, Cambridge, MA, 2017.

Paul R Rosenbaum. Sensitivity analysis for certain permutation inferences in matched observational studies. *Biometrika*, 74(1):13–26, 1987.

Paul R Rosenbaum. Quantiles in nonrandom samples and observational studies. *Journal of the American Statistical Association*, 90(432):1424–1431, 1995.

Colin B Fogarty, Pixu Shi, Mark E Mikkelsen, and Dylan S Small. Randomization inference and sensitivity analysis for composite null hypotheses with binary outcomes in matched observational studies. *Journal of the American Statistical Association*, 112(517):321–331, 2017.

Colin B Fogarty. Studentized sensitivity analysis for the sample average treatment effect in paired observational studies. *Journal of the American Statistical Association*, 115(531):1518–1530, 2020.

Paul R Rosenbaum. Reduced sensitivity to hidden bias at upper quantiles in observational studies with dilated treatment effects. *Biometrics*, 55(2):560–564, 1999.

Paul R Rosenbaum. Effects attributable to treatment: Inference in experiments and observational studies with a discrete pivot. *Biometrika*, 88(1):219–231, 2001.

Paul R Rosenbaum. Exact confidence intervals for nonconstant effects by inverting the signed rank test. *The American Statistician*, 57(2):132–138, 2003.

Ben B Hansen and Jake Bowers. Attributing effects to a cluster-randomized get-out-the-vote campaign. *Journal of the American Statistical Association*, 104(487):873–885, 2009.

Jasjeet S Sekhon and Yotam Shem-Tov. Inference on a new class of sample average treatment effects. *Journal of the American Statistical Association*, 116(534):798–804, 2021.

Winston Lin. Agnostic notes on regression adjustments to experimental data: Reexamining Freedman's critique. *The Annals of Applied Statistics*, 7(1):295–318, 2013.

Peter L Cohen and Colin B Fogarty. No-harm calibration for generalized Oaxaca-Blinder estimators. *Biometrika*, 111(1):331–338, 2023.

Kevin Guo and Guillaume W Basse. The generalized Oaxaca-Blinder estimator. *Journal of the American Statistical Association*, 118(541):524–536, 2023.

Bo Lu, Elaine Zanutto, Robert Hornik, and Paul R Rosenbaum. Matching with doses in an observational study of a media campaign against drug abuse. *Journal of the American Statistical Association*, 96 (456):1245–1253, 2001.

Bo Lu, Robert Greevy, Xinyi Xu, and Cole Beck. Optimal nonbipartite matching and its statistical applications. *The American Statistician*, 65(1):21–30, 2011.

Jeffrey H Silber, Paul R Rosenbaum, Richard N Ross, Justin M Ludwig, Wei Wang, Bijan A Niknam, Nabanita Mukherjee, Philip A Saynisch, Orit Even-Shoshan, Rachel R Kelz, and Lee A Fleisher. Template matching for auditing hospital cost and quality. *Health Services Research*, 48(5):1446–1474, 2014.

José R Zubizarreta and Luke Keele. Optimal multilevel matching in clustered observational studies: A case study of the effectiveness of private schools under a large-scale voucher system. *Journal of the American Statistical Association*, 112(518):547–560, 2017.

Samuel D Pimentel, Lindsay C Page, Matthew Lenard, and Luke Keele. Optimal multilevel matching using network flows: An application to a summer reading intervention. *Annals of Applied Statistics*, 12 (3):1479–1505, 2018.

Yunfei Paul Li, Kathleen J Propert, and Paul R Rosenbaum. Balanced risk set matching. *Journal of the American Statistical Association*, 96(455):870–882, 2001.

José R Zubizarreta, Ricardo D Paredes, and Paul R Rosenbaum. Matching for balance, pairing for heterogeneity in an observational study of the effectiveness of for-profit and not-for-profit high schools in Chile. *The Annals of Applied Statistics*, 8(1):204–231, 2014.

Stefano M Iacus, Gary King, and Giuseppe Porro. Causal inference without balance checking: Coarsened exact matching. *Political Analysis*, 20(1):1–24, 2012.

Shoshana R Daniel, Katrina Armstrong, Jeffrey H Silber, and Paul R Rosenbaum. An algorithm for optimal tapered matching, with application to disparities in survival. *Journal of Computational and Graphical Statistics*, 17(4):914–924, 2008.

Fredrik Sävje, Michael J Higgins, and Jasjeet S Sekhon. Generalized full matching. *Political Analysis*, 29 (4):423–447, 2021.

Paul R Rosenbaum, Richard N Ross, and Jeffrey H Silber. Minimum distance matched sampling with fine balance in an observational study of treatment for ovarian cancer. *Journal of the American Statistical Association*, 102(477):75–83, 2007.

Dan Yang, Dylan S Small, Jeffrey H Silber, and Paul R Rosenbaum. Optimal matching with minimal deviation from fine balance in a study of obesity and surgical outcomes. *Biometrics*, 68(2):628–636, 2012.

Ruoqi Yu. How well can fine balance work for covariate balancing. *Biometrics*, 79(3):2346–2356, 2023.

Samuel D Pimentel, Rachel R Kelz, Jeffrey H Silber, and Paul R Rosenbaum. Large, sparse optimal matching with refined covariate balance in an observational study of the health outcomes produced by new surgeons. *Journal of the American Statistical Association*, 110(510):515–527, 2015a.

Samuel D Pimentel, Frank Yoon, and Luke Keele. Variable-ratio matching with fine balance in a study of the Peer Health Exchange. *Statistics in Medicine*, 34(30):4070–4082, 2015b.

Paul R Rosenbaum. Design sensitivity in observational studies. *Biometrika*, 91(1):153–164, 2004.

Ruth Heller, Paul R Rosenbaum, and Dylan S Small. Split samples and design sensitivity in observational studies. *Journal of the American Statistical Association*, 104(487):1090–1101, 2009.

Jesse Y Hsu, Dylan S Small, and Paul R Rosenbaum. Effect modification and design sensitivity in observational studies. *Journal of the American Statistical Association*, 108(501):135–148, 2013.

Dylan S Small, Jing Cheng, M Elizabeth Halloran, and Paul R Rosenbaum. Case definition and design sensitivity. *Journal of the American Statistical Association*, 108(504):1457–1468, 2013.

Johann Gagnon-Bartsch and Yotam Shem-Tov. The classification permutation test: A flexible approach to testing for covariate imbalance in observational studies. *The Annals of Applied Statistics*, 13(3): 1464–1483, 2019.

Zach Branson. Randomization tests to assess covariate balance when designing and analyzing matched datasets. *Observational Studies*, 7(2):1–36, 2021.

Paul R Rosenbaum. Permutation tests for matched pairs with adjustments for covariates. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 37(3):401–411, 1988.

Samuel D Pimentel and Yaxuan Huang. Covariate-adaptive randomization inference in matched designs. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(5):1312–1338, 2024.

Kan Chen, Siyu Heng, Qi Long, and Bo Zhang. Testing biased randomization assumptions and quantifying imperfect matching and residual confounding in matched observational studies. *Journal of Computational and Graphical Statistics*, 32(2):528–538, 2023.

Siyu Heng, Yanxin Shen, and Pengyun Wang. Reconciling overt bias and hidden bias in sensitivity analysis for matched observational studies. arXiv Preprint, https://arxiv.org/pdf/2311.11216, May 2025.

Jianan Zhu, Jeffrey Zhang, Zijian Guo, and Siyu Heng. Randomization-based inference for average treatment effects in inexactly matched observational studies. arXiv Preprint, https://arxiv.org/pdf/2311.11216, March 2025.

Samuel D Pimentel and Ruoqi Yu. Re-evaluating the impact of hormone replacement therapy on heart disease using match-adaptive randomization inference. Working Paper, https://arxiv.org/pdf/2403.01330.pdf, March 2024.