A general framework for learning about research designs[†]

Graeme Blair[‡] Jasper Cooper[§] Alexander Coppock[¶] Macartan Humphreys[††]

12/19/2017

Abstract

Researchers need to select high quality research designs and communicate those designs to readers. Both tasks are difficult. We provide a framework for formally characterizing the analytically relevant features of a research design. In standard applications, the approach to design declaration we describe requires defining a model of the world (M), an inquiry (I), a data strategy (D), and answer strategy (A). We advocate that scholars declare each feature formally in computer code, and then use Monte Carlo techniques to diagnose properties such as power, bias, expected mean squared error, external validity with respect to some population, and other "diagnosands." Declaring a design in this way lays researchers' assumptions bare. Ex ante design declarations can be used to improve designs and facilitate preregistration, analysis, and reconciliation of intended and actual analyses. Ex post design declarations are also useful for describing, sharing, reanalyzing, and critiquing existing designs. We provide an open-source software package, `DeclareDesign`, to implement the proposed approach.

[†] Authors are listed in alphabetical order. This work was supported in part by a grant from the Laura and John Arnold Foundation and seed funding from EGAP – Evidence in Governance and Politics. Errors remain the responsibility of the authors. We thank Peter Aronow, Erin HartmanJustin Grimmer, Kolby Hansen, Chad Hazlett, Tom Leavitt, Winston Lin, Matto Mildenberger, Matthias Orlowski, Molly Roberts, Tara Slough, Gosha Syunyaev, Anna Wilke, Erin York, Lauren Young, Yang-Yang Zhou, Teppei Yamamoto, and participants at the Southern California Methods Workshop and the EPSA 2016, APSA 2016, and EGAP 18 meetings for helpful comments. The methods proposed in this paper are implemented in an accompanying open-source software package, `DeclareDesign` (Blair et al., 2016).

[‡] Assistant Professor of Political Science, UCLA. graeme.blair@ucla.edu. `https://graemeblair.com`

[§] Ph.D. candidate in Political Science, Columbia University. jjc2247@columbia.edu. `http://jasper-cooper.com`

[¶] Assistant Professor of Political Science, Yale University. alex.coppock@yale.edu. `https://alexandercoppock.com`

[††] Professor of Political Science, Columbia University. mh2245@columbia.edu. `http://www.macartan.nyc`

Empirical social scientists routinely face two research design problems. First, we need to select a high-quality design, given resource constraints. Second, we need to convince readers and reviewers of the design's high quality.

To select strong designs, we often rely on rules of thumb, rudimentary power calculators, or generic best practices from the methodological literature that may assume ideal conditions that do not hold in real-world applied settings. These relatively informal practices sometimes result in the selection of suboptimal designs, or worse, designs that are simply too weak to deliver useful answers.

To convince others of the quality of our designs, we defend them with citations to previous studies that used similar approaches, power analyses that may rely on assumptions unknown even to ourselves, or in rare cases, difficult-to-parse simulation code. Communication of all design-relevant information is also frustrated by the basic problem that the social science community does not have a shared understanding of what constitutes a design.

The challenges of assessment and communication raise two questions: what information do we need in order to evaluate our designs? Given that information, how do we assess the quality of the design? In this paper we describe an approach that addresses these two problems. We first provide a framework to describe the distinct elements of a research design. The *MIDA* framework asks researchers to provide information about their background model (M), their inquiry (I), their data strategy (D), and their answer strategy (A).

We then introduce the notion of "diagnosands," or statistical summaries of the design such as the power of the design, the bias of the estimator, or the expected mean squared error (MSE) of the estimates with respect to an estimand. A design is "diagnosand-complete" in our framework when a diagnosand can be estimated from the declared features of the design. We do not have a general notion of a *complete* design, but rather adopt an approach in which the purposes of the design determine which features must be declared. Many different designs can be defined in terms of their diagnosand-completeness: including causal inference strategies employing observational, experimental or qualitative methods, as well as descriptive and exploratory strategies.

Using this framework researchers can *declare*[1] research designs mathematically and as com-

[1] We emphasize that the term "declare" does not imply a public declaration or a declaration before research takes place. A researcher may declare the features of designs in our framework for their own understanding and declaring designs may be useful before or after the research is implemented.

puter code objects and, second, *diagnose* the statistical properties of the design relying on this declaration. We recommend researchers diagnose their designs using Monte Carlo simulation, and we provide an algorithm that produces diagnosand estimates as well as bootstrapped estimates of simulation uncertainty. We implement this framework in the companion `R` package `DeclareDesign`.

The formal characterization and diagnosis of designs before implementation can serve many purposes. A researcher may wish to include design declaration and diagnosis as part of a pre-analysis plan or a funding request. Whether or not the declaration and diagnosis serves this purpose, the process of generating diagnostics provides researchers an opportunity to learn about and improve their inferential strategies. Even if only declared ex-post, formal declaration still has advantages. The complete characterization can help readers understand the properties of a research project, facilitate transparent replication, and contribute to re-analysis decisions.

In general, we do not provide specific guidance on the set of diagnosands that must be calculable in order for a design to be complete "enough." Domain-specific standards might be agreed upon among members of particular research communities. A standard set might include power, bias, root mean-squared error, and coverage. Others concerned about the policy impact of a given treatment might require a design that is diagnosand-complete for an out-of-sample diagnosand, such as bias relative to the population average treatment effect.

The approach we describe is clearly more easily applied to some types of research than others. In prospective confirmatory work, for example, researchers may have access to all design-relevant information prior to launching their study. For exploratory work, by contrast, researchers may simply not have enough information about possible quantities of interest to declare a design in advance. Although in some cases the design may still be declared ex post, in others it may not be possible to fully reconstruct the inferential procedure after the fact. For instance although researchers might be able to provide compelling grounds for their inferences, they may not be able to describe what inferences they would have drawn had different data had been realized. Thus variation in research strategy limits the utility of our procedure for different types of research.

Our framework makes five main contributions: it enables the diagnosis of designs in terms of their probative value; it assists in the improvement of research designs through comparison with alternatives; it assists learning about the properties of research designs; it enhances research transparency by making design choices explicit; and it provides tools to assist principled replication and reanalysis of published research.

1. Research Designs and Diagnosands

We consider a general description of a research design as the specification of a problem and a strategy to answer it. Our framework is an elaboration of ideas proposed by King, Keohane and Verba (1994, p. 13), who enumerate four components of a research design: a theory, a research question, data, and an approach to using the data. Formalized, this framework can be used to structure a general procedure for assessing research designs' inferential properties, by, for example, assessing the bias or precision of results from a design given the theory, the data collection process, the research question and inferential strategy. We extend this four component framework using recent advances in the theory of causal inference. Specifically, we follow Pearl's (2009) approach to structural modeling, which gives a syntax for mapping design components to design outputs. We combine this causal modeling approach with the potential outcomes framework as presented in Imbens and Rubin (2015), which clarifies the inferential targets of many research designs.

1.1 Elements of a Research Design

The specification of a problem requires a description of the world and the question to be asked about that world. The answering requires a description of what information is used and how conclusions are reached given the information.

At its most basic we think of a research design, Δ, as including four elements $< M, I, D, A >$:

1. A **model**, M, of how the world works. In general following Pearl's definition of a probabilistic causal model we will assume that a model contains three core elements. First a specification of the variables X, about which research is being conducted. This includes endogenous and exogenous variables (V and U respectively). In the formal literature this is sometimes called the *signature* of a model (e.g., Halpern, 2000). Second, a specification of how each endogenous variable depends on other variables (the "functional relations" or "potential outcomes"), F. Third, a probability distribution over exogenous variables, $P(U)$.

2. An **inquiry**, I, about the distribution of variables, X, perhaps given interventions on some

variables. In many applications I might be thought of as the estimand. Using Pearl's notation we can distinguish between questions that ask about the conditional values of variables, such as $\Pr(X_1|X_2 = 1)$ and questions that ask about values that would arise under interventions: $\Pr(X_1|do(X_2 = 1))$.[2] We let a^M denote the answer to I provided by the model. Under model M, answer a^M is generated with probability $P_M(a^M)$.

3. A **data** strategy, D, generates data d on X. Data d arises, under model M with probability $P_M(d|D)$. Note that implicitly the data strategy includes sampling P_S, random assignment of treatments P_Z, and measurement strategies.

4. An **answer** strategy, A, that generates answer a^A using data d. Under model M, answer a^A is generated with probability $P_M(a^A|D, A)$.

A key feature of this bare specification is that if M, D, and A are sufficiently well described, the answer to question I has a distribution $P_M(a^A|D)$; moreover one can construct a distribution of comparisons of this answer to the correct answer, under M, for example by assessing $P_M(a^M - a^A|D)$. One can also compare this to results under different data or analysis strategies, D', A': $P_M(a^M - a^A|D')$, $P_M(a^M - a^{A'}|D)$ and to answers generated under alternative models, as long as these possess signatures that are consistent with inquiries and answer strategies, $P_M(a^{M'} - a^A|D)$.

Many social scientists will be familiar with a statistical framework that distinguishes between an estimand, an estimator, and an estimate. In our terms, an estimate is an answer a^A. An estimator is the procedure that is jointly described by the Data Strategy D and the Answer Strategy A. An estimand is the answer a^M that the Inquiry I produces from Model M. The overlapping and imperfect mapping of the estimand-estimator-estimate framework to the *MIDA* framework highlights the special utility of *MIDA*: the distribution of an estimator is a product of both how the data are collected and how they are analyzed; an estimand is a summary of a *theoretical* model that may or may not be correct.

MIDA captures the the analysis-relevant features of a design, but it does not describe substantive elements, such as how interventions are implemented or how outcomes are measured.

[2] The distinction lies in whether the conditional probability is recorded through passive observation or active intervention to manipulate the probabilities of the conditioning distribution. For example, $\Pr(X_1|X_2 = 1)$ might indicate the conditional probability that it is raining, given that Jack has his umbrella, whereas $\Pr(X_1|do(X_2 = 1))$ would indicate the probability with which it would rain, given Jack is made to carry an umbrella.

Yet many other aspects of a design that are not explicitly labeled in these features enter into this framework if they are analytically relevant. For example, logistical details of data collection such as the duration of time between a treatment being administered and endline data collection enter into the model if the longer time until data collection affects subject recall of the treatment.

1.2 Diagnosands

The ability to calculate distributions of answers, given a model, opens multiple avenues for assessment and critique. How good is the answer you expect to get from this strategy? Would you do better with a different data strategy? With a different analysis strategy? How good is the strategy if the model is wrong in some way or another?

To allow for this kind of *diagnosis* of a design, we introduce two further concepts, both functions of research designs. These are quantities that a researcher or a third party could calculate with respect to a design.

1. A **Diagnostic Statistic** is a summary statistic generated from a run of a design—that is, the results given a possible realization of variables, given the model and data strategy. A diagnostic statistic may or may not depend on the model as well as realized data. For example the statistic: $e =$ "difference between the estimated ATE and the actual average treatment effect (ATE)" depends on the model (where "actual" is defined under the model for a given run). The statistic $s = \mathbb{I}(p \leqslant 0.05)$, interpreted as "the result is considered statistically significant at the 5% level," does not depend on the model but it does presuppose an answer strategy that reports a p value.

   Under a given model, diagnostic statistics have a distribution that result from the fact that both the model and the data generation, given the model, may be stochastic.

2. A **Diagnosand** is a summary of the distribution of a diagnostic statistic. For example, given the diagnostic statistics described above, (expected) *bias* in the estimated treatment effect is $\mathbb{E}(e)$ and statistical *power* is $\mathbb{E}(s)$.

To illustrate, consider the following design. A model M specifies three variables X, Y, Z in some population (the signature of the model) and some functional relationships between them that allow for the possibility of confounding (for example, $Y = bX + Z + \epsilon_Y; X = Z + \epsilon_X$, with

$Z, \epsilon_X, \epsilon_Z$ distributed standard normal). The question of interest is "what is the average effect of a unit change in X on Y in the population?" Note that this question depends on the signature of the model, but not the functional equations of the model (the answer provided by the model does of course depend on the functional equations). Consider now a data strategy D, in which data is gathered on X and Y for n randomly selected units. An answer a^A, is then generated using ordinary least squares as the answer strategy, A.

Is this a good research design? One way to answer this question is with respect to the diagnosand "expected error." Here the model's functional equations provide an answer, a^M to the inquiry (for any draw of β), and so the distribution of the expected "error," *given the model*, $a^A - a^M$, can be calculated.

In this example the expected performance of the design may be poor because the data and analysis strategy do not handle the confounding described by the model. In comparison, better performance may be achieved through an alternative data strategy (e.g., where D' randomly assigned X to n units before recording X and Y) or an alternative analysis strategy (e.g., A' conditions on Z). All these evaluations of designs depend on the model, and so one might reasonably ask how performance would look were the model different (for example if it allowed for spillovers or effect heterogeneity).

In all cases the evaluation depends on the assessment of a diagnosand, and comparing the diagnoses to what could be achieved under alternative designs.

1.3 Choice of Diagnosands

What diagnosands should researchers choose? Although researchers commonly focus on statistical power, a larger range of diagnosands can be examined and may provide more informative diagnoses of design quality. We list and describe some of these in Table 1, indicating for each the design information that is required in order to calculate them.

This set of statistics allows researchers to understand the properties of the estimates across possible realizations of the data and how successful their data and analysis strategies are at estimating estimands. Though these are frequentist properties, many of the diagnosands can be used to assess Bayesian estimation strategies (see Rubin, 1984), and as we illustrate below there are diagnosands unique to Bayesian strategies.

		Required:			
Diagnosand	**Description**	M	I	D	A
Power	Probability of rejecting null hypothesis of no effect	✓		✓	✓
Estimation Bias	Expected difference between estimate and estimand	✓	✓	✓	✓
Sampling Bias	Expected difference between population average treatment effect and sample average treatment effect (Imai, King and Stuart, 2008)	✓	✓	✓	
RMSE	Standard deviation of differences between estimates and estimand	✓	✓	✓	✓
Coverage	Probability that estimand falls within confidence interval	✓	✓	✓	✓
SD of Estimates	Standard deviation of estimates	✓		✓	✓
SD of Estimands	Standard deviation of estimands	✓	✓	✓	
Imbalance	Expected distance of covariates across treatment conditions (Mahalanobis, 1936; Gu and Rosenbaum, 1993)	✓		✓	
Type S Rate	Probability estimate has incorrect sign, if statistically significant (Gelman and Carlin, 2014)	✓	✓	✓	✓
Exaggeration Ratio	Expected ratio of absolute value of estimate to estimand, if statistically significant (Gelman and Carlin, 2014)	✓	✓	✓	✓
Value for money	Probability that the estimated effect is at least as large as x	✓		✓	✓
Robustness	Joint probability of rejecting the null hypothesis across multiple tests	✓		✓	✓

Table 1: Examples of diagnosands and the declarations required for completeness in each case.

Diagnosands can also be defined for design properties that are often discussed informally but rarely subjected to formal investigation. For example one might define an inference as "robust" if the same inference is made under different analysis strategies, or an intervention as having "value for money" if some set of estimates have some minimal magnitude. A diagnosis summarizing these diagnostic statistics across many simulations of the design would then report the chances that an inference considered robust or an intervention deemed to have value for money.

1.4 What is a Complete Research Design Declaration?

A declaration of a research design that is in some sense complete is required in order to implement it, communicate its essential features, and to assess its properties. Yet existing definitions make clear that there is no single conception of a complete research design that is satisfactory for all purposes: the Consolidated Standards of Reporting Trials (CONSORT) Statement widely used in medicine includes 22 features and other proposals range from nine to 60 components.[3]

We propose a conditional notion of completeness: we say a design is "diagnosand-complete" for a given diagnosand if that diagnosand can be calculated from the declared design. Thus a design that is diagnosand complete for one diagnosand may not be for another. Consider for example the diagnosand "statistical power." Power is the probability that a p-value is lower than

[3] See "Pre Analysis Plan Template" (60 features); World Bank Development Impact Blog (nine features).

some critical value. Thus, power-completeness requires that the answer strategy return a p value. It does not require a well defined estimand however (hence the lack of a checkmark under I on Table 1). Bias or RMSE completeness in contrast do not require a hypothesis test, but do require the specification of an estimand.

Our notion of diagnosand-completeness does not encompass all of the information relevant to research design. It instead clarifies the assumptions under which a design has good inferential properties.[4]

2. Existing Methods for Diagnosing Research Designs

Currently there are three main methods for researchers to assess the properties of their designs: analytical formulae for simple research designs such as a sample survey of n units from a population of N to estimate a population parameter (e.g., Cohen, 1977; Haseman, 1978; Muller and Peterson, 1984; Muller et al., 1992; Lenth, 2001); bespoke Monte Carlo simulation code, written by researchers to diagnose specific studies; and computational tools available through Web apps (e.g., the EGAP power tool), general statistical software (e.g., `easypower` for `R` and Power and Sample Size for Stata), and single-use diagnosis software for particular types of designs (e.g., Optimal Design, G*Power, nQuery, NCSS PASS, SPSS Sample Power, and so on). We show here that, with the exception of custom-written Monte Carlo simulations, even the most sophisticated of these methods cannot calculate key diagnosands for even relatively simple designs because they do not require or accommodate sufficient information about the design.

We conducted a census of the currently available computational diagnostic tools for research designs. We used two methods to search for existing tools. First, we conducted Google searches on the terms “statistical bias calculator”, “statistical power calculator” and “sample size calculator.” We assessed the first 30 results using these terms.[5] Second, we assessed the tools listed in

[4] Diagnosand-completeness only conveys what aspects of a design must be declared in order for some diagnostic feature to be queried, but it does not convey what information is required to make such diagnosis *believable*. A bias-complete design may be declared which excludes the possibility of bias from Hawthorne effects. Whether the estimated bias of the design is credible or not depends on the credibility of the model used to generate the diagnostic statistics. Different research communities set different standards for what constitutes sufficient information to make such conjectures about the world plausible. With respect to effect sizes, for example, some organizations may want to see how diagnoses vary across the entire range of conceivable effects, while others may require researchers to conduct a relevant meta-analysis or even a baseline survey in order to bolster the assumptions feeding into their design declarations.

[5] We found no admissable tools using the term “statistical bias calculator”.

four reviews of the literature, namely Kreidler et al. (2013), Guo et al. (2013), Groemping (2016) and Green and MacLeod (2016). Of the 143 tools identified as candidates, 30 were able to diagnose specific inferential properties of designs, such as their power. Online Appendix Section S2 provides further details on the methods employed to identify, admit, and code the tools.

Using these 30 tools, we attempted to diagnose the following hypothetical research design. A team of researchers wants to assess the effectiveness of a new voter turnout strategy. Their **M**odel allows for the effectiveness of the mobilization campaign to vary depending on the size of the voting precinct, but their **I**nquiry is nevertheless the average treatment effect. Because of budget constraints, the team's **D**ata strategy involves sampling five precincts at random, then randomly assigning 10 households to treatment within each precinct, i.e., blocking by voting precinct. This procedure generates differential probabilities of assignment, so the team is considering two **A**nswer strategies: a block-level fixed effects estimator (BFE) or an estimator that incorporates both inverse probability weights and block (IPW-BFE).

The team seeks to answer three diagnostic questions:

1. What is the power of each estimator?
2. What is the bias of each estimator with respect to the average treatment effect?
3. Given the answers to 1 and 2, which approach should the team pre-register as the main analysis?

Using the tools surveyed for this article, the team would be unable to answer any of these questions correctly, because those tools cannot incorporate key features of the design. As evidenced on Table 2, none of the tools was able to diagnose the design while taking account of: the posited correlation between block size, potential outcomes, and treatment assignment probabilities; the sampling strategy; the exact randomization procedure; the formal definition of the estimand as the population average treatment effect; or the use of inverse-probability weighting estimation techniques.[6] As a result, no tool was able to calculate the power for the IPW-BFE estimator. Moreover, no tool was able to calculate the design's bias, root mean squared error or coverage.

We compared the power calculations from these 30 tools to the true power of a simulated

[6] The one tool (GLIMMPSE) that was able to account for the blocking strategy encountered an error and was unable to produce diagnostic statistics.

(a) Declare Elements of Designs

(M)	Effect and block size correlated	0/30
(I)	Estimand	0/30
(D)	Sampling procedure	0/30
(D)	Assignment procedure	0/30
(D)	Block sizes vary	1/30
(A)	Probability weighting	0/30

(b) Diagnosis Capabilities

Power (DIM estimator)	28/30
Power (BFE estimator)	13/30
Power (IPW-BFE estimator)	0/30
Bias (*any* estimator)	0/30
Coverage (*any* estimator)	0/30
SD of estimates (*any* estimator)	0/30

Table 2: Existing tools cannot declare many core elements of designs and, as a result, can only calculate some diagnosands correctly.

version of the researcher's design under assumptions about the data-generating process, which we calculated in `R` using the companion software to this paper, `DeclareDesign`. (Blair et al., 2016).Starting from a very large finite population of interest in which each unit has a treatment and control potential outcome, we first calculate the true population average treatment effect (PATE), then randomly sample five blocks, assign ten units within each block to the treatment and the rest to control, reveal the outcomes that obtain under the given random assignment, and then estimate the treatment effects using naive difference-in-means (DIM),[7] BFE and IPW-BFE. The parameters from this simulation exercise are used to calculate the power of the design using the 30 identified tools.

While almost all (28/30) of the tools were able to provide an estimate of the design's power when using the DIM estimator, fewer than half (13/30) were able to provide an estimate of the power using the BFE estimator, and as noted none were able to provide a power estimate for the design when using the IPW-BFE estimator. Although they were able to estimate power, because they neglect core features of the design, the tools substantially exaggerated power estimates – by an average of 15 and 13 percentage points for the DIM and BFE estimators, respectively. The tools misstate the true power of these estimators because they assume the estimator is unbiased. However simulations show the estimates produced by the DIM and BFE estimators are lower than the true effect on average, due to the negative correlation between a unit's treated potential outcome and its probability of assignment to treatment. Because the assessed tools base power calculations on the true underlying effect, which is larger than the estimates provided by those

[7] We include the DIM estimator not because we expect that the researcher would ever employ such a strategy, but because it is the only case that many of the tools assessed can handle and thereby enables direct comparison of their performance.

two answer strategies ($E[a^M] > E[a^A]$), they exaggerate the design's power.

Using the companion software, we show that the IPW-BFE estimator is better powered and less biased (in terms of the PATE) than the BFE estimator. However, power is a misleading indicator of the efficiency of the IPW-BFE strategy: it is better powered because it produces biased variance estimates that lead to a coverage probability that is too low. In terms of RMSE and the standard deviation of estimates, the IPW-BFE strategy does not outperform the BFE estimator. The exercise thus highlights why power and sample size calculations alone are insufficient to fully assess the tradeoffs between these relatively simple design alternatives.

Beyond this specific example, the general point is that existing tools cannot incorporate the information required to comprehensively assess the probative value of research designs. In our view, this shortcoming does not derive from any statistical weakness in the tools, which sometimes feature sophisticated mathematical underpinnings. Rather, these tools lack two core features. First, they are not guided by a framework that specifies what features of a design a researcher must declare in order to properly diagnose its inferential properties.[8] Second, because these tools' diagnostic methods are based on pre-defined formulae that abstract from core features of the design, they lack the flexibility to faithfully approximate the answers provided by real research designs. In the next section, we argue that computer-based simulation of designs provides such flexibility, and use the companion software to illustrate how computer-based diagnosis can be used in a variety of research contexts: from causal inference employing observational, experimental, and qualitative evidence, to descriptive and exploratory research.

3. Declaring and Diagnosing Research Designs in Practice

A design that can be described mathematically (as in Section 1) can also be declared in computer code and then simulated in order to diagnose its properties. The core advantage of simulation over diagnosis through analytic solutions is that diagnosands can be quantified even where closed-form solutions do not exist or are extremely difficult to derive. Whereas researchers with advanced coding skills will be able to write their own custom-built simulations, coding diagnoses of designs from scratch can be complicated, and is difficult to compare to other examples. The

[8] Even though they are power calculators for the most part, many of the tools' definitions are not "power-complete" in the sense defined above, because they do not require information on core parts of the Data or Answer strategy (such as the assignment procedure).

	Design Declaration	Code
M	Declare background variables $P(U)$	`P_U <- declare_population(x_1 = rnorm(N), N = 100)`
	Declare functional relations F	`F <- declare_potential_outcomes(Y ∼ x_1 + Z)`
I	Declare inquiry I	`I <- declare_estimand(PATE = mean(Y_Z_1 - Y_Z_0))`
D	Declare sampling p_S	`p_S <- declare_sampling(n = 50)`
	Declare assignment p_Z	`p_Z <- declare_assignment(m = 25)`
A	Declare answer strategy, A	`A <- declare_estimator(Y ∼ Z, estimand = I)`
	Declare design, <M, I, D, A>	`D <- declare_design(P_U, F, I, p_S, p_Z, A)`

	Design Simulation (1 draw)	Code
1	Draw a population u using $P(U)$	`u <- P_U()`

2	Calculate an answer a^M to I using F and u	`uv <- F(u)`
		`a_M <- I(uv)`

3	Draw data, d, given sampling and treatment assignments specified in D and data realizations as determined by F and u	`d_1 <- p_S(uv)`
		`d <- p_Z(d_1)`

4	Calculate answers, a^A using A and d:	`a_A <- A(d)`
5	Calculate a diagnostic statistic t using a^A and a^M	`t <- a_A["est"] - a_M["estimand"]`

	Design Diagnosis (m draws)	Code
	Declare a diagnosand	`bias <- declare_diagnosands(bias = mean(est - estimand))`
	Calculate a diagnosand	`diagnose_design(D, diagnosands = bias, sims = m)`

Table 3: A procedure for design diagnosis through simulation, using the companion software `DeclareDesign` (Blair et al., 2016).

top panel of Table 3 shows how to declare a design in code using the companion software to this paper, `DeclareDesign` (Blair et al., 2016). The resulting set of objects (`P_U`, `F`, `I`, `p_S`, `p_Z`, and `A`) are functions. A single simulation calls each of these functions successively as shown in steps 1-5. A design diagnosis conducts m simulations, then summarizes the resulting distribution of diagnostic statistics in order to estimate the diagnosand.

Diagnosands can be estimated with higher levels of precision by increasing m. However, simulations are often computationally expensive. In order to assess whether researchers have conducted "enough" simulations to be confident in their diagnosand estimates, we recommend estimating the sampling distributions of the diagnosands via the nonparametric bootstrap. With the estimated diagnosand and its standard error, we can characterize our uncertainty about whether the range of likely values of the diagnosand compare favorably to reference values such as statistical power of 0.8. We emphasize that the standard error reflects both estimation uncertainty (simulation error) and fundamental uncertainty (true variability in the diagnosand, for example across possible population draws).[9]

Our companion software facilitates design diagnosis for beginner to intermediate coders in `R`. Those with no coding experience can use the online design declaration and diagnosis wizard, available at `DeclareDesign.org`. The website also contains instructions for implementing this framework in Stata. In the Appendix, we provide a simple example and explains how each step corresponds to the *MIDA* framework.

Design diagnosis through simulation does place a burden on researchers to come up with a substantive model, M. Since researchers presumably want to learn about the model, declaring it in advance may seem to beg the question. Yet declaring a model is unavoidable when conducting diagnosis. In practice it is already familiar to any researcher who has calculated the power of a design, which requires the specification of effect sizes. The seeming arbitrariness of the declared model can be mitigated by assessing the sensitivity of diagnosis to alternative models and strategies, which is relatively straightforward given a diagnosand-complete design declaration, as evidenced in examples provided below. In a sense, similar to the focus on minimal detectable effects for power calculators, what design declaration offers is not only a tool to establish that a design has desirable qualities but a tool to lay bare *under what assumptions* a design has desirable properties.

In the next three sections, we outline how research designs that aim to answer causal, descriptive, and exploratory research questions can be declared and diagnosed in practice.

3.1 Causal Inference

The approach to design diagnosis we propose can be used to declare and diagnose a range of research designs typically employed to answer causal questions in the social sciences.

Observational Regression-Based Strategies. Many observational studies seek to make causal claims but do not explicitly employ the potential outcomes framework, instead describing in-

[9] This procedure depends on the researcher choosing a "good" diagnosand estimator. In nearly all cases, diagnosands will be features of the distribution of a diagnostic statistic that, given i.i.d. sampling, can be consistently estimated via plug-in estimation (for example taking sample means). Our simulation procedure, by construction, yields i.i.d. draws of the diagnostic statistic.

quiries in terms of model parameters. Consider a study that seeks to estimate parameter β from a **M**odel of the form $y_i = \alpha + \beta x_i + \epsilon_i$. What is the estimand here? If we believe that this model describes the true data generating process then β *is* an estimand: it is the true (constant) marginal effect of x on y. But what if we are wrong about the model? We run into a tautology if we want to assess the properties of strategies under different assumptions about data generation when the inquiry itself depends on the data generating model.

We can declare an **I**nquiry as some summary of differences in potential outcomes across conditions, β. For example we might define α and β as the solutions to:

$$\min_{(\alpha,\beta)} \sum_i \int (y_i(x) - \alpha - \beta x)^2 f(x)dx$$

Here $y_i(x)$ is the (unknown) potential outcome for unit i in condition x. Estimand β can be thought of as the coefficient one would get on x if one were to able to regress all possible potential outcomes on all possible conditions for all units (given density of interest $f(x)$).[10] Our **D**ata strategy will simply consist of the passive observation of units in some population, and we assess the performance of an **A**nswer strategy employing an OLS model to estimate β under different conditions.

To illustrate this design, we declare a design using the `R` package `DeclareDesign` in which the properties of a regression estimate are assessed under the assumption that in the true data-generating process y is in fact a nonlinear function of x (for the full declaration, see Online Appendix Section S1.8). Diagnosis of the design shows that under uniform random assignment of x, the linear regression returns an unbiased estimate of a (linear) estimand, even though the true data generating process is non linear. Interestingly, with the design in hand, it is easy to see that unbiasedness is lost in a design in which different values of x_i are assigned with different probabilities.

Process Tracing. While many qualitative researchers employ frameworks that may seem incompatible with the type of design declaration we have described, formal design declaration and diagnosis may still be of use to qualitative designs that aim to confirm the presence or

[10] An alternative might be to imagine some analogue of the ATT estimand, for example for some x_i defined on the real line we might define $E[Y_i(x_i) - Y_i(x_i - 1)]$ where x_i is the observed treatment received by unit i.

absence of a causal relationship (i.e., that are not focused on theory generation). Consider a stylized "process-tracing" design similar to ones described for example by Mahoney (2012) or in the Online Appendix to Bennett and Checkel (2014). A researcher selects a case in which some outcome is observed (a revolution, say) and some possible driver is present (a strong middle class, say). The researcher seeks evidence in archives that they believe to be "smoking gun evidence" (Van Evera, 1997) that the driver was indeed important for the outcome—for example they look for evidence that the revolution was financed by domestic industry—and are prepared to draw different inferences depending on what they find in this causal process observation (CPO).

Declared in terms of MIDA, the **M**odel in such a study could stipulate a population $P(U)$ of N cases. The unobserved variable $T \in \{A, B, C, D\}$ gives the causal type of each case. In combination with the potential outcomes function F, the type variable creates a mapping between the presence or absence of the causal driver $X \in$ { No strong middle class, Strong middle class } and the presence or absence of the outcome $Y \in$ { No revolution, Revolution }. A types only have revolutions when there is no strong middle class, B types only have revolutions when there is a strong middle class, C types never have revolutions, and D types have them irrespective of the middle class's strength. A clue $K \in$ { Revolution not financed by domestic industry, Revolution financed by domestic industry } is generated with probability .2 only if the case is a B type and the causal driver is present. The **D**ata strategy involves selecting one case at random for process-tracing, specifically one in which there is a middle class and a revolution. This leads to the **I**nquiry: is the case a B type, given the CPO? Or, formally, $Pr(T = B \mid K)$. A priori, since the case can only be a B or a D type given that X and Y are both present, the researcher might assign equal probabilities to the case being of either type. Suppose that the **A**nswer strategy involves inferring with certainty that the case is a type B when the clue is observed and remaining agnostic when it is not, such that $Pr(T = B \mid \neg K) = .5$. With these elements in hand, it is relatively straightforward to generate a distribution of diagnostic statistics $t = a^A - a^M$. Our implementation of this procedure using the `R` package `DeclareDesign` in Online Appendix Section S1.1 shows that the researcher's inference will be unbiased in the cases in which the CPO is observed ($E[t^K] = 0$), but not in those cases in which it is not ($E[t^{\neg K}] \neq 0]$), and so not overall ($E[t] \neq 0]$). The bias arises from the non-Bayesian property of the answer strategy: the researcher does not sufficiently discount the causal theory under investigation when disconfirmatory evidence comes to light.

Selection-on-Observables with Matching. In many observational research designs, the processes by which units are assigned to treatment are not known with certainty. In matching designs, the unknown assignment procedure is approximated by matching units on their observable traits to justify an assumption of as-if random assignment.[11] Diagnosis in such instances can be helpful as a tool to explore the conditions under which such assumptions are justified. In Online Appendix Section S1.2, we declare a design with a **M**odel in which three observable random variables are combined in a probit process that assigns the treatment variable, Z. The **I**nquiry pertains to the average treatment effect of Z on the outcome Y among those actually assigned to treatment, which we estimate using an **A**nswer strategy that reconstructs the assignment process to calculate a^A. Our diagnosis shows that matching improves mean-squared-error ($E[(a^A - a^M)^2]$) relative to a naive difference-in-means estimator of the treatment effect on the treated (ATT), but can nevertheless remain biased ($E[a^A - a^M] \neq 0$) if the matching algorithm does not successfully pair units with equal probabilities of assignment.

Regression Discontinuity. While in matching applications researchers do not typically know the assignment process, in other observational settings researchers may know how assignment works without necessarily controlling it. In regression discontinuity designs causal identification is premised on the claim that potential outcomes are continuous at a critical threshold (and not from a claim of random placement of units around a threshold). The declaration of such designs involves a **M**odel that defines the unknown potential outcomes functions mapping average outcomes to the running and treatment variables. Our **I**nquiry concerns the average difference in potential outcomes as they limit toward the threshold of the running variable at which the assignment variable changes values. The **D**ata strategy involves passive observation and collection of the data. The **A**nswer strategy is a polynomial regression in which the assignment variable is linearly interacted with a fourth order polynomial transformation of the running variable. In Online Appendix Section S1.3, we declare and diagnose such a design. A key point to arise from the simulation is that the estimand involved in many regression discontinuity designs is rarely

[11] Matching is sometimes used for a different purpose, as a non-parametric form of covariate adjustment with no special claims of as-if random assignment.

an average of potential outcomes of all units, but rather an unobservable quantity defined at the limit of the discontinuity. Assessing the external validity this design can be complicated: unless one postulates unobservable counterfactuals (such as the 'treated' outcome for a unit located below the treatment threshold), it is difficult to declare designs that are bias-complete with respect to the population or even sample average treatment effects.

Experimental Design. Experimental research may call particularly for design declaration and diagnosis because researchers are typically in direct control of many features of the design, beginning with assignment of treatments. One example of such a choice is that between a 2-by-2 factorial design or a three-arm trial where the "both" condition is excluded. Consider a researcher studying two treatments who is interested in the effect of each treatment *conditional on the other treatment being in the control condition*. Should she choose a factorial design or a three-arm design? Focusing for simplicity on the effect of a single treatment, we declare two designs under a range of alternative models to help assess the tradeoffs. For both designs, we consider **M**odels $M_1, ..., M_K$, where we set the interaction between treatments to 0 for M_1, and increment it by $.5/(K-1)$ for each $M_{k \in 2,...,K}$. Our **I**nquiry is always the average treatment effect of treatment 1 given all units are in the control condition for treatment 2. We have two alternative **D**ata strategies under consideration: d' using an assignment strategy p'_Z, in which subjects are assigned to a control condition, treatment 1, or treatment 2, each with probability $1/3$; and d'' using p''_Z to assign subjects to each cell of a 2×2 with probability $1/4$. The **A**nswer strategy in both cases involves a regression of the outcome on both treatment indicators.

We declare and diagnose this design and find that neither design exhibits bias when the true interaction term is equal to zero (Figure 1 left panel). The details of the declaration can be found in Online Appendix Section S1.4. However, as the interaction between the two treatments is stronger, the factorial design renders estimates of the effect of treatment 1 that are more and more biased relative to the "pure" main effect estimand. Moreover, there is a bias-variance tradeoff in choosing between the two designs (Figure 1 right panel). When the interaction term is small or close to zero, the factorial design is preferred, because it is more powerful: it compares one half of the subject pool to the other half, whereas the three arm design only compares a third to a third. However, as the magnitude of the interaction term increases, the precision gains are offset by the increase in bias documented in the left-panel. In cases of high heterogeneity, the three-

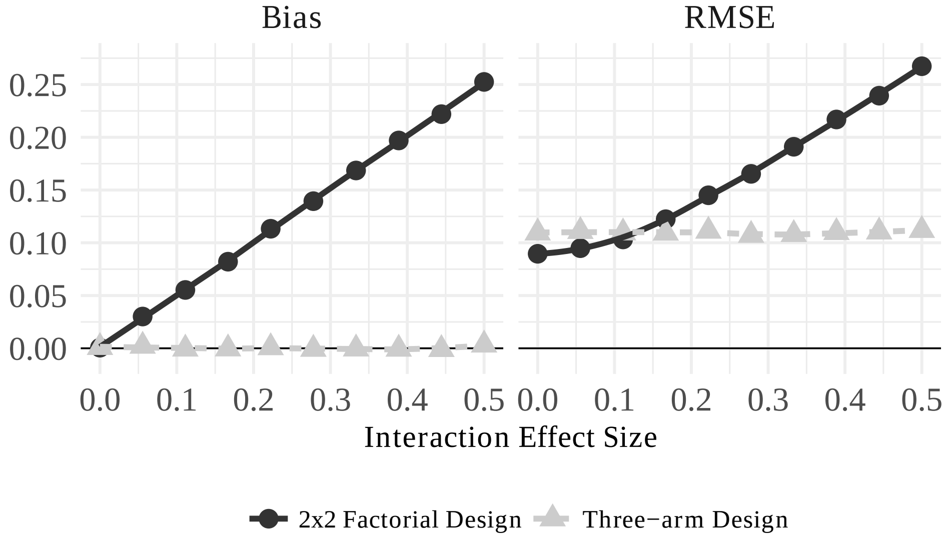

Figure 1: Diagnoses of Designs with Factorial or Three-Arm Assignment Strategies Illustrate a Bias-Variance Tradeoff. Bias (left) and root mean-squared-error (right) are displayed for two assignment strategies, a 2 × 2 treatment arm factorial design (black solid lines; circles) and a three-arm design (gray dashed lines; triangles) according to varying interaction effect sizes specified in the potential outcomes function (x axis).

arm design is then preferred. This exercise highlights key points of design guidance. Researchers often select factorial designs because they expect interaction effects: and indeed factorial designs are required to assess these. However if the scientific question of interest is the pure effect of each treatment, researchers should (perhaps counterintuitively) use a factorial design if they expect *weak* interaction effects.

3.2 Descriptive Inference

Descriptive research questions often center on measuring a parameter in a sample or in the population, such as the proportion of voters in the United States who support the democratic candidate for president. Although seemingly very different from designs that focus on causal inference—because often there are no explanatory variables—the formal differences are not great.

Survey Designs. We examine an estimator of candidate support that conditions on being a "likely voter." For this problem the data that help researchers predict who will vote is of critical importance. In Online Appendix Section S1.5, we declare a **M**odel in which latent voters are likely to vote for a candidate, but unlikely to reveal to interviewers their true propensity to vote. The **I**nquiry concerns the true underlying support for the candidate, while the **D**ata strategy involves a random population sample. The **A**nswer strategy involves looking at support for the candidate among likely voters. The design can be diagnosed to assess the risk of falsely concluding that the general election support of the democratic candidate is above 50%, given assumptions about how people report their voting proclivities.

Bayesian Descriptive Inference. In addition to modes of analysis that employ a classic null-hypothesis testing approach to statistical inference, our framework can also be of use to Bayesian strategies. In Online Appendix Section S1.6, we declare a Bayesian descriptive inference design. The **M**odel stipulates a latent probability of success for each unit, and makes one binomial draw for each based off of this probability. The **I**nquiry pertains to to the latent probability, and the **D**ata strategy involves a random sample of relatively few units. There are two alternative **A**nswer strategies under consideration: in the first, the researcher stipulates uniform priors, with a mean of .5 and a standard deviation of .29; in the second, the priors place more probability mass at .5, with a standard deviation of .11. The design can be diagnosed not only in terms of its bias, but also as a function of quantities specific to Bayesian estimation approaches, such as the expected shift in the location and scale of the posterior distribution relative to the prior distribution. The diagnosis shows that the informative prior approach yields more certain and more biased inferences than the uniform prior approach. In terms of the bias-variance tradeoff, the informative priors decrease the posterior standard deviation by about 40% relative to the uniform priors, but increase the bias by about 33%.

3.3 Designs for Discovery

In some research projects the ultimate hypotheses that are assessed are not known at the the design stage. Some inductive designs are entirely unstructured and explore a variety of data sources with a variety of methods within a general domain of interest until a new insight of some type is uncovered. Yet many can be described in a more structured way.

In studying textual data, for example, a researcher may have a procedure for discovering the "topics" that are discussed in a corpus of documents. Before beginning the research, the set of topics and even the number of topics is unknown. Instead, the researcher selects a model for estimating the content of a fixed number of topics (i.e., Blei, Ng and Jordan, 2003) and a procedure for evaluating the model fit used to select which number of topics fits the data best.

Such a design is inductive, yet the analytical *procedure of discovery* can be described and evaluated.

We examine an exploratory data analysis *procedure* in which in a first stage the researcher explores possible analysis strategies on half of the data and in the second stage apply their preferred procedure to the second half of the data. Split-sample procedures such as this enable researchers to learn about the data inductively while still protecting against Type I errors (for an early discussion of the design, see Cox, 1975). In Online Appendix Section S1.7, we declare a design in which the **M**odel stipulates that education is a confounder for the effect of income on the outcome of interest, Y, while the **I**nquiry pertains to the unconfounded effect of income on Y. The **D**ata strategy simply involves passively recording the variables of interest. We compare three **A**nswer strategies: the "right" and "wrong" models, which do and don't condition the analysis of income on the concurrent effect of education, on the one hand, and on the other, a split-sample procedure that estimates effects on one half of the sample using the one of three models that has the best goodness of fit when estimated on the other half of the sample. The design is complete for a range of diagnosands (power, bias, RMSE, Type-S, etc.). The split-sample procedure reduces bias and power relative to selection of the "wrong" estimator. Any exploratory procedure in which the domain of exploration (for example, the set of tests that will be conducted) and the decision rules (how the researcher selects among models or changes the analysis in response to test values) are known can be declared and diagnosed in this manner.

4. Putting Declarations and Design Diagnosis to Use

We have described and illustrated a strategy for declaring research designs for which "diagnosands" can be estimated given conjectures about the world. How might declaring and diagnosing research designs in this way affect the practices of authors, readers, and replication authors? We describe implications for how designs are chosen, communicated, and challenged.

4.1 Making Design Choices

The move towards increasing credibility of research in the social sciences places a premium on considering alternative data strategies and analysis strategies at early stages of research projects, not only because it reduces researcher discretion, but more importantly because it can improve the quality of the final research design. While there is nothing new about the idea of determining features such as sampling and estimation strategies ex ante in order to maximize power, for example, in practice many designs are finalized late in the research process, after data are collected. Frontloading design decisions is difficult not only because existing tools are rudimentary and often wrong, as illustrated in Section 2, but because it is not clear in current practice what features of a design must be considered ex ante.

We provide a framework for identifying *which* features of a design affect the assessment of a design's properties, a means to declare them and diagnose their properties, and to frontload design decisions. Declaring the design's features in code makes possible directly exploring alternative data strategies and especially analysis strategies using simulated data; evaluating alternative data and analysis strategies through diagnosis; and exploring the robustness of a chosen design to alternative models through diagnosis. Each step can be undertaken before the study is implemented or data is collected.

4.2 Communicating Design Choices

Bias in published results can arise for many reasons. For example, researchers may deliberately or inadvertently select analysis strategies because they produce statistically significant results. Proposed solutions to reduce this kind of bias focus on various types of preregistration of analysis strategies by researchers (Rennie, 2004; Zarin and Tse, 2008; Casey, Glennerster and Miguel, 2012; Nosek, 2014; Green and Lin, 2016). Study registries are now operating in numerous areas of social science, including those hosted by the American Economic Association, Evidence in Governance and Politics, and the Center for Open Science. Bias may also arise from reviewers implicitly selecting papers to publish based on their statistical significance. Results-blind review processes are being introduced in some journals to address this form of bias (e.g. Findley et al., 2016).

However, the effectiveness of design registries and results-blind-review in reducing the scope for either form of publication bias depends on clarity over which elements must be included to describe the design. In practice some registries rely on checklists and pre-analysis plans exhibit great variation, ranging from lists of written hypotheses to all-but-results journal articles. In our view, the solution to this problem does not lie in ever-more-specific questionnaires, but rather in a new way of characterizing designs whose analytic features can be diagnosed through simulation.

The requirement that design declarations be diagnosand-complete can clarify for researchers and third parties what aspects of a study need to be specified in order to meet standards for effective preregistration. Rather than asking: "are the boxes checked?" the question becomes: "can it be diagnosed?" A design can only be diagnosed when sufficient detail has been provided to analytically characterize diagnosands or to conduct Monte Carlo simulations of the implementation of the design from beginning to end.

Declaration of a diagnosand-complete design also enables a final and infrequently practiced step of the registration process, in which the researcher "reports and reconciles" the final with the planned analysis. Understanding how and whether the features of a design diverge between ex ante and ex post declarations highlights deviations from the pre-analysis plan. The magnitude of such deviations determines whether results should be considered exploratory or confirmatory. At present, this exercise requires a review of dozens of pages of text, such that differences (or similarities) are not immediately clear even to close readers.

4.3 Challenging Design Choices

The independent replication of the results of studies after their publication is an essential component of the shift toward more credible science. Replication — whether verification, reanalysis of the original data, or reproduction of results using fresh studies — provides incentives for researchers to be clear and transparent in their analysis strategies, and can build confidence in findings.[12]

In addition to rendering the design more transparent, diagnosand-complete declaration can allow for a different approach to the re-analysis and critique of published research. A standard practice for replicators engaging in reanalysis is to propose a range of alternative strategies and assess the robustness of the *data*-dependent estimates to different analyses. The problem with this approach is that when divergent results are found, third parties do not have clear grounds to decide which results to believe. This issue is compounded by the fact that, in changing the analysis strategy, replicators risk departing from the estimand of the original study, possibly providing different answers to different questions. In the worst case scenario, it can be difficult to determine what is learned both from the original study and from the replication.

A more coherent strategy facilitated by design simulations would be to use a diagnosand-

[12] For a discussion of the distinctions between these different modes of replication, see Clemens (2015).

	Author's assumed **M**odel	Alternative claims on **M**odel
Author's proposed **A**nswer strategy	1	2
Alternative **A**nswer strategy	3	4

Table 4: Diagnosis Results Given Alternative Assumptions on Model and Alternative Answer Strategies. Four scenarios encountered by researchers and reviewers of a study are considered depending on whether the model or the answer strategy differs from the author's original strategy and model.

complete declaration to conduct "design replication." In a design replication, a scholar restates the essential design characteristics to learn about what the study *could have* revealed, not just what the original author reports *was* revealed. This helps to answer the question: under what conditions are the results of a study to be believed? By emphasizing abstract properties of the design, design replication provides grounds to support alternative analyses on the basis of the original authors' intentions and not on the basis of the degree of divergence of results. Conversely, it provides authors with grounds to question claims made by their critics.

Table 4 illustrates situations that may arise. In a declared design an author might specify situation 1: a set of claims on the structure of the variables and their potential outcomes (the model) and an estimator (the answer strategy). A critic might then question the claims on potential outcomes (for example questioning SUTVA) or question estimation strategies (for example arguing for the need to include or exclude some control variables from an analysis), or both.

In this context here are several possible criteria for admitting alternative answer strategies:

- **Home ground dominance.** If ex ante the diagnostics for situation 3 are better than for 1 then this gives grounds to switch to 3. That is, a critic can demonstrate that an alternative estimation strategy outperforms an original estimation strategy even under the data generating process assumed by an original researcher, then they have strong grounds to propose a change in strategies. Conversely if an alternative estimation strategy produces different results, conditional on the data, but does not outperform the original strategy given the original assumptions, this gives grounds to question the reanalysis.

- **Robustness to alternative models.** If the diagnostics in situation 2 are as good as in 1 but are better in situation 4 than in situation 3 this provides a robustness argument for altering estimation strategies.

- **Model plausibility.** If the diagnostics in situation 1 are better than in situation 2, but the diagnostics in situation 4 are better than in situation 3, then this is cause for worry and the justification of a change in estimators depends on the plausibility of the different assumptions on potential outcomes.

Without a declared design—in particular the model and inquiry—none of the three of these criteria can be evaluated, making defending claims difficult both for the critic and the original author.

We illustrate an application of these principles through a design replication of Björkman and Svensson (2009). Importantly, we are able to conduct a results-independent design replication because sufficient detail to simulate the design is provided in the original article and its supporting materials. The independence of the replication to the data not only makes replication possible in a context where the data is not yet publicly available, but more significantly focuses attention on the abstract features of the design.

We draw upon the "robustness to alternative models" criterion to claim that an alternative answer strategy would be superior to the original strategy employed by the authors, in the sense that the alternative approach exhibits less bias under plausible conjectures about the world. In the original study, Björkman and Svensson (2009) investigate whether community-based monitoring can improve health outcomes in rural Uganda. They focus on improvements in two important indicators: child mortality, defined as the number of deaths per 1000 live births among under-5 year-olds, taken at the catchment-area-level; and weight-for-age z-scores, which are calculated by subtracting from an infant's weight the median for their age from some reference population, and dividing by the standard deviation of that population. In the original design, the authors estimate a positive effect of the intervention on weight among surviving infants. However, they also find that the treatment greatly decreases child mortality.

The weight of infants in control areas whose lives would have been saved if they had been in the treatment cannot be observed. We posit that unobserved variables, "family health" and "community health", may determine both whether infants survive early childhood and whether they are malnourished. We discuss these claims in detail in our data-independent design replication of Björkman and Svensson (2009) in Online Appendix Section S3. Figure 2 illustrates how the existence of an effect on mortality can pose problems for the unbiased estimation of an ef-

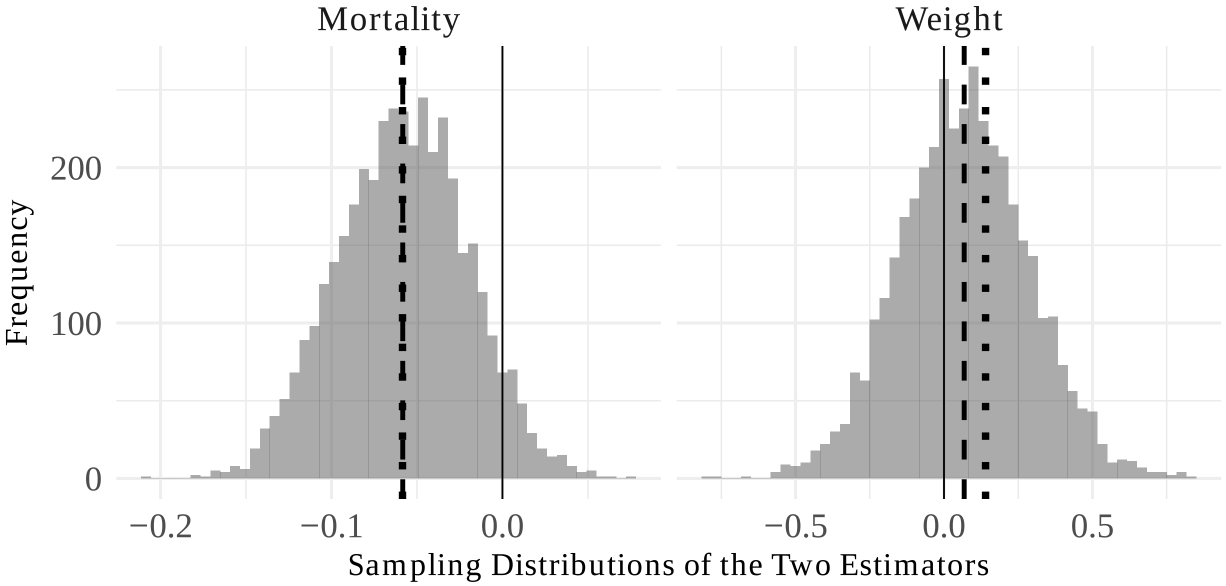

Figure 2: Data-independent replication of Björkman and Svensson (2009). Histograms display the frequency of simulated estimates of the effect of community monitoring on infant mortality (left) and on weight-for-age (right). The dashed vertical line shows the average estimate, the dotted vertical line shows the average estimand.

fect on weight-for-age when the two outcomes are correlated by community or family health. The histograms represent the frequency with which the design gives different answers to the inquiry about the effect of community monitoring on infant mortality and weight-for-age. The differences arise because the random sampling and assignment procedures select and assign different units on each run of the design. The dotted vertical line represents the true average effect, whereas the dashed line represents the average answer, i.e. the answer we expect the design to provide given our assumptions. Under our proposed model of the world the estimates of the effect on weight-for-age are biased downwards because it is precisely those infants with low health outcomes whose lives the treatment saves. This pulls down the average weight outcome for those in the treatment group.

An alternative answer strategy is to attempt to subset the analysis of the weight effects to a group of infants whose survival does not depend on the treatment. In the original study, for example, the effects on survival are much larger among infants younger than two years old. If indeed the survival of infants above this age threshold is unaffected by the treatment, then it is possible to provide unbiased estimates the weight-for-age effect by subsetting to this group (assuming effect homogeneity). In terms of bias, such an approach does at least as well if we assume that there is no correlation between weight and mortality, and better if such a correlation does exist. It thus satisfies the "robustness to alternative models" criterion.

A reasonable counter to this replication effort might be to say that the alternative answer strategy does not meet the criterion of home ground dominance with respect to RMSE: the power loss from subsetting to a smaller group may outweigh the bias reduction that it entails. In both cases, transparent arguments can be made by formally declaring and comparing the original and modified designs. While such criteria will not eliminate disputes they should at least help focus the discussion on the analytically relevant issues.

4.4 Risks

The creation of a set of tools to evaluate the completeness and quality of research designs also creates a set of risks. We outline four here. The first risk is that evaluative weight gets placed on essentially meaningless diagnoses. Given that design declaration includes declarations of conjectures about the world it is possible to choose numbers so that a design passes any diagnostic test set for it. Fortunately, however, the advantage of the formal declaration is that the basis for the diagnoses can be examined and new diagnostics can be generated quickly given alternative specifications of data generating processes while keeping other design elements intact. Even still, the risk remains that if the grounds for diagnoses are not inspected, designs may be favored because of the optimism of the designers rather than inherent qualities of the design.

A second risk is that research gets evaluated on the basis of a narrow but perhaps inappropriate set of diagnosands, such as power, bias, or RMSE. In fact, the appropriateness of the diagnosand depends on the purposes of the study. The optimal bias-variance tradeoff for example might depend on whether the interest is in assessing properties of a specific case or whether a study is contributing to a larger literature. To help guard against this risk we provide a range of diagnosands as defaults in our software and allow users to define their own. In this way, the evaluative grounds for research may be widened for example by making it easier for researchers to demonstrate the value of a design that carries a risk of bias but has other valuable properties.

A third risk is that as the evaluation of formal properties of a design become easier, evaluative weight shifts away from the substantive importance of a question being answered.[13] Similarly there could be a risk that less attention is paid to measurement issues, which largely fall outside our framework. Simplification of the evaluation of formal properties of a design could instead, however, allow for a shift in attention towards examining other properties of a design such as measurement strategy or substantive and theoretical relevance.[14]

A fourth risk is that the variation in the suitability of design declaration to different research strategies that we outlined above is taken as evidence of the relative superiority of different types of research strategies. We believe that the range of strategies that can be declared and diagnosed is wider than what one might at first think possible, and we sketch above outlines for declarations of descriptive, experimental, observational, quasi-experimental, and qualitative strategies. We argue that there is value in formally declaring designs when this is possible. There is no reason to believe, however, that all strong designs can be declared either ex ante or ex post. An advantage of our framework, we hope, is that it can help clarify when a strategy can or cannot be completely declared. When a design cannot be declared, nondeclarability is all the framework provides, and in such cases we urge caution in drawing conclusions about design quality.

5. Conclusion

How can researchers assess the properties of research designs and improve them before implementation? Today, tools upon which many scholars rely prevent them from faithfully characterizing the features of common applied research designs. We show that these tools often provide misleading assessments of the design properties and sometimes are not able to provide an assessment at all. Our method allows researchers to fully characterize, and thus to diagnose their designs in a manner consistent with their assumptions and plans. Of course, even a simulation-based claim to unbiasedness that incorporates all features of a design is still only good with respect to the conditions of the simulation; for example conditional on the potential outcomes functions posited. In this sense, claims for properties of strategies are more robustly made based on analytic results. Often however, the complexity of a given research design prohibits analytic

[13] A similar concern has been raised regarding the “identification revolution” where a focus on identification risks crowding out attention to the importance of questions being addressed (Huber, 2013).

[14] More creatively, it may also be possible to think of substantive importance as a diagnosand—for example one could declare as a diagnosand the likelihood that the research will contribute new knowledge to a given question (whether or not it has good statistical properties).

interrogation of diagnosands. Conversely, a simulation based *critique* of a strategy—a demonstration that a strategy is biased for some estimand—may be powerful even when general analytic results do not exist.

We describe a procedure for characterizing and diagnosing designs before implementation. Ex ante declaration and diagnosis of designs can help researchers improve their properties. It can make it easier for readers to evaluate a research strategy prior to implementation and without access to results. It can also make it easier for designs to be shared and to be critiqued. Our proposed framework and software aim to facilitate these steps.

References

Bennett, Andrew and Jeffrey T. Checkel, eds. 2014. *Process tracing*. Cambridge: Cambridge University Press.

Björkman, Martina and Jakob Svensson. 2009. "Power to the People: Evidence from a Randomized Field Experiment of a Community-Based Monitoring Project in Uganda." *Quarterly Journal of Economics* 124(2):735–769.

Blair, Graeme, Jasper Cooper, Alexander Coppock and Macartan Humphreys. 2016. "DeclareDesign Version 1.0." Software package for R, available at http://declaredesign.org.

Blei, David M., Andrew Y. Ng and Michael I. Jordan. 2003. "Latent dirichlet allocation." *Journal of machine Learning research* 3(Jan):993–1022.

Casey, Katherine, Rachel Glennerster and Edward Miguel. 2012. "Reshaping Institutions: Evidence on Aid Impacts Using a Pre-Analysis Plan." *The Quarterly Journal of Economics* 127(4):1755–1812.

Clemens, Michael A. 2015. "The Meaning of Failed Replications: A Review and Proposal." *Center for Global Development Working Paper* 399.

Cohen, Jacob. 1977. "Statistical power analysis for the behavioral sciences (revised ed.).".

Cox, D. R. 1975. "A note on data-splitting for the evaluation of significance levels." *Biometrika* 62(2):441–444.

Findley, Michael G., Nathan M. Jensen, Edmund J. Malesky and Thomas B. Pepinsky. 2016. "Can Results-Free Review Reduce Publication Bias? The Results and Implications of a Pilot Study." *Comparative Political Studies* 49(13):1667–1703.

Gelman, Andrew and John Carlin. 2014. "Beyond Power Calculations Assessing Type S (Sign) and Type M (Magnitude) Errors." *Perspectives on Psychological Science* 9(6):641–651.

Green, Donald P. and Winston Lin. 2016. "Standard Operating Procedures: A Safety Net for Pre-Analysis Plans." *PS: Political Science and Politics* 49(3):495–499.

Green, Peter and Catriona J MacLeod. 2016. “SIMR: an R package for power analysis of generalized linear mixed models by simulation.” *Methods in Ecology and Evolution* 7(4):493–498.

Groemping, Ulrike. 2016. “Design of Experiments (DoE) & Analysis of Experimental Data.”.
URL: *https://CRAN.R-project.org/view=ExperimentalDesign*

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

Guo, Yi, Henrietta L. Logan, Deborah H. Glueck and Keith E. Muller. 2013. “Selecting a sample size for studies with repeated measures.” *BMC Medical Research Methodology* 13(1):100.
URL: *http://dx.doi.org/10.1186/1471-2288-13-100*

Halpern, Joseph Y. 2000. “Axiomatizing causal reasoning.” *Journal of Artificial Intelligence Research* 12:317–337.

Haseman, JK. 1978. “Exact sample sizes for use with the Fisher-Irwin test for 2 x 2 tables.” *Biometrics* pp. 106–109.

Huber, John. 2013. “Is theory getting lost in the “identification revolution”?” *Monkey Cage* blog post.

Imai, Kosuke, Gary King and Elizabeth A Stuart. 2008. “Misunderstandings between experimentalists and observationalists about causal inference.” *Journal of the royal statistical society: series A (statistics in society)* 171(2):481–502.

Imbens, Guido W. and Donald B. Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge: Cambridge University Press.

King, Gary, Robert O. Keohane and Sidney Verba. 1994. *Designing Social Inquiry: Scientific Inference in Qualitative Research*. Princeton University Press, Princeton, New Jersey.

Kreidler, Sarah M, Keith E Muller, Gary K Grunwald, Brandy M Ringham, Zacchary T Coker-Dukowitz, Uttara R Sakhadeo, Anna E Barón and Deborah H Glueck. 2013. “GLIMMPSE: online power computation for linear models with and without a baseline covariate.” *Journal of statistical software* 54(10).

Lenth, Russell V. 2001. “Some practical guidelines for effective sample size determination.” *The American Statistician* 55(3):187–193.

Mahalanobis, Prasanta Chandra. 1936. “On the generalised distance in statistics.” *Proceedings of the National Institute of Sciences of India, 1936* pp. 49–55.

Mahoney, James. 2012. “The Logic of Process Tracing Tests in the Social Sciences.” *Sociological Methods and Research* 41(4):570–597.

Muller, Keith E and Bercedis L Peterson. 1984. “Practical methods for computing power in testing the multivariate general linear hypothesis.” *Computational Statistics & Data Analysis* 2(2):143–158.

Muller, Keith E, Lisa M Lavange, Sharon Landesman Ramey and Craig T Ramey. 1992. “Power calculations for general linear multivariate models including repeated measures applications.” *Journal of the American Statistical Association* 87(420):1209–1226.

Nosek, Brian A. et al. 2014. “Guidelines for Transparency and Openness Promotion (TOP) in Journal Policies and Practices.” Transparency and Openness Committee Report.

Pearl, Judea. 2009. *Causality*. Cambridge: Cambridge University Press.

Rennie, Drummond. 2004. “Trial registration.” *JAMA: the Journal of the American Medical Association* 292(11):1359–1362.

Rubin, Donald B. 1984. “Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician.” *The Annals of Statistics* 12(4):1151–1172.

Van Evera, Stephen. 1997. *Guide to methods for students of political science*. Ithaca: Cornell University Press.

Zarin, Deborah A. and Tony Tse. 2008. “Moving towards transparency of clinical trials.” *Science* 319(5868):1340–1342.

Appendix

Below, we demonstrate how each diagnosand-relevant feature of a simple design can be defined in code, with an application in which the assignment procedure is known. This could represent an experimental or quasi-experimental design.

$P(U)$ **The population**. Defines the population variables, including both observed and unobserved X. In the example below we define a function that returns a normally distributed variable of a given size. Critically, the declaration is not a declaration of a particular realization of data but of a data generating *process*. Researchers will typically have a sense of the distribution of covariates from previous work, and may even have an existing dataset of the units that will be in the study with background characteristics. Researchers should assess the sensitivity of their diagnosands to different assumptions about p_X.

```
my_population <- declare_population(N = 1000, u = rnorm(N))
```

Each `declare` step creates a function, in this case a function that returns a data set of N observations with a variable named `u` drawn from a random normal distribution. For example, the population step $P(U)$ could have equivalently been created using the following function:

```
my_population_function <- function(N) { data.frame(u = rnorm(N)) }

my_population <- declare_population(
  population_function = my_population_function, N = 1000)
```

$D(1)$ **Assignment 1: The sampling strategy**. Defines the distribution over possible samples for which outcomes are measured, p_S. In the example below each unit generated by p_X is sampled with 10% probability. Again `my_sampling` describes a sampling strategy and not an actual sample.

```
my_sampling <- declare_sampling(n = 100)
```

$D(2)$ **Assignment 2: Treatment assignment**. Defines the strategy for assigning variables under the notional control of researchers. In this example each sampled unit is assigned to treatment independently with probability 0.5. The default assumption in our code is that treatment assignment takes place after sampling though as a general matter this need not be the case. In designs in which the sampling process or the assignment process are in the control of researchers, p_z is known. In observational designs, researchers either know or assume p_z based on substantive knowledge.

```r
my_assignment <- declare_assignment(m = 50)
```

F **The structural equations, or potential outcomes function**. The potential outcomes function defines conjectured potential outcomes given interventions Z and parents. In the example below the potential outcomes function maps from a treatment condition vector (Z) and background data u, generated by p_X, to a vector of outcomes. In this example the potential outcomes function satisfies a SUTVA condition—each unit's outcome depends on its own condition only, though in general since Z is a vector, it need not.[15] It also assumes that potential outcomes depends on treatment assignment and not on sampling. Again, the declaration describes the function and not a particular set of potential outcomes.

```r
my_potential_outcomes <- declare_potential_outcomes(Y_Z_0 = u, Y_Z_1 = u + .25)
```

In many cases, the potential outcomes function (or features of it) is the very thing that the study sets out to learn, so it can seem odd to assume features of it. We suggest two approaches to developing potential outcomes functions that will yield useful information about the quality of designs. First, set a potential outcomes function in which the variables of interest are set to have no effect on the outcome whatsoever. Diagnosands such as bias can then be assessed without having to assume a particular relationship between treatments and outcomes. This approach will not work for some diagnosands such as power or Type-S errors. Second, consider setting a series of potential outcomes functions that correspond to competing theories. This enables the researcher to judge whether the design yields answers that help adjudicate between the theories and whether the design has desirable properties (i.e., sufficient power) under the potential outcomes implied by each theory.

I **The estimands**. The estimand function τ creates a summary of potential outcomes us-

[15] For an example of a function that does not satisfy SUTVA consider $Y = Z + \min(Z \times u)$, for vectors Y, Z, u.

ing 'superdata' that can be generated from the elements declared above. In principle the estimand function can also take realizations of assignments as arguments, in order to calculate post-treatment estimands. Below, the estimand takes the mean difference between the potential outcomes for units in a treated condition and units in a control condition.

```
my_estimand <- declare_estimand(ATE = mean(Y_Z_1 - Y_Z_0))
```

A **The answer strategies** are functions that use information from realized data and the design, but do not have access to the full schedule of potential outcomes. In the declaration we associate estimators with estimands and we record a set of summary statistics that are required to compute diagnostic statistics. In the example below an estimates function takes data and returns an estimate of a treatment effect using the difference-in-means estimator, as well as a set of associated statistics, including the standard error, p-value, and the confidence interval.

```
my_estimator <- declare_estimator(Y ~ Z, estimand = my_estimand)
```

We then declare the design, which in this case primarily describes the order of the features, though it could include other changes to the data such as subsetting or adding variables.

```
my_design <- declare_design(my_population,
                            my_potential_outcomes,
                            my_estimand,
                            my_sampling,
                            my_assignment,
                            my_estimator)
```

These six features represent the study. In order to assess the completeness of a declaration and to learn about the properties of the study, we also define functions for the diagnostic statistics, $t(D, Y, f)$, and diagnosands, $\theta(D, Y, f, g)$. For simplicity, the two can be coded as a single function. For example, to calculate the bias of the design as a diagnosand is:

```
diagnosand <- declare_diagnosands(bias = mean(est - estimand))
```

These eight functions could be written in many code languages. In the companion software for this paper, `DeclareDesign` (Blair et al., 2016), we implement it for the widely-used `R` platform.

A general framework for learning about research designs

Online Appendix

Graeme Blair Jasper Cooper Alexander Coppock Macartan Humphreys

Contents

S1. Diagnoses for the Examples in Sections 3.1 and 3.3

The code examples can be downloaded from the internet and run using the free, open source statistical package R. First, install the DeclareDesign software as follows:

```
install.packages("DeclareDesign", dependencies = TRUE,
  repos = c("http://R.declaredesign.org", "https://cloud.r-project.org"))
```

Code for running the examples is below. Further details on the R software package, including other examples and documentation, can be found at declaredesign.org.

S1.1 Process tracing

M *Model*: We posit a population comprising four causal types, A, B, C and D. For all types, X is present with a probability of .7, absent otherwise. The variable K is present with probability .2 only if the case is a B type and X is present, absent otherwise. Y is present for A types when X is absent, present otherwise. Y is present for B types only if X is present, and always absent for C types. For D types, Y is always present.

I *Inquiry*: The estimand is the probability with which the case is a B, 1 when the case is truly a B and 0 otherwise.

D *Data Strategy*: The data strategy involves randomly selecting a case where X and Y are both present, i.e. cases that must be B or D cases by definition.

A *Answer Strategy*: We search for clue K. If it is present we conclude the case is a B type with probability 1, and if it is absent we remain agnostic as to whether the case is a B or a D type (i.e., infer that it is B with probability .5).

```
population <- declare_population(
  N =  200,
  type = sample(c('A','B','C','D'), N, TRUE),
  X = rbinom(N, 1, .7),
  K = ifelse(X == 1 & type == 'B', rbinom(1, 1, .2), 0),
  Y = (type == 'A' & !X) | (type == 'B' & X) | (type == 'D'))
sampling <- declare_sampling(
  sampling_function = function(data) {
  eligible_cases <- with(data, which(X & Y))
  return(data[sample(eligible_cases, 1), ])})
estimand <- declare_estimand(is_B = type == 'B')
estimator <- declare_estimator(
  estimator_function = function(data) {
    with(data, data.frame(guess = ifelse(K, 1, .5), K_seen = K))},
  estimand = estimand)
process_tracing_diagnosands <- declare_diagnosands(
  truth = mean(estimand),
  mean_guess = mean(guess),
  bias = mean(guess - estimand),
  bias_given_K_seen = mean(guess[K_seen] - estimand[K_seen]),
  bias_given_no_K_seen = mean(guess[!K_seen] - estimand[!K_seen]))
process_tracing <- declare_design(
  population, sampling, estimand, estimator)


process_tracing_diagnosis <- diagnose_design(
  process_tracing, diagnosands = process_tracing_diagnosands, bootstrap = FALSE,
  sims = sims)
```

estimand_label	is_B
estimator_label	my_estimator
truth	0.3
mean_guess	0.5
bias	0.2
bias_given_K_seen	NA
bias_given_no_K_seen	0.2

S1.2 Matching

M *Model*: We posit a population that has three standard normally distributed variables, X_1, X_2 and X_3. The potential outcomes of units in the population are an additive function of these variables and the treatment.

I *Inquiry*: We wish to know the average effect of the treatment among those who were actually treated in a given implementation of the design.

D *Data Strategy*: We imagine that units are assigned to treatment through a probit process that is a function of the X variables.

A *Answer Strategy*: We match the units to one another using the three X variables and estimate the difference between treated and control among the matches.

```r
library(Matching)
population <- declare_population(
  N = 1000, X1 = rnorm(N), X2 = rnorm(N), X3 = rnorm(N))
potential_outcomes <-
  declare_potential_outcomes(formula = Y ~ X1 + X2 + X3 + Z)
assignment <- declare_assignment(
  assignment_function = function(data) {
    prob <- with(data, pnorm(X1 + X2 + X3))
    data$Z <- rbinom(nrow(data), 1, prob)
    return(data)})
estimand <- declare_estimand(att = mean(Y_Z_1[Z == 1] - Y_Z_0[Z == 1]))
estimator_d_i_m <- declare_estimator(Y ~ Z, estimand = estimand, label = "dim")
estimator_m <- declare_estimator(
  estimator_function = function(data) {
    match_out <- with(data, Match(Y = Y, Tr = Z, X = cbind(X1, X2, X3)))
    return(data.frame(
      coefficient_name = NA,
      est = match_out$est,
      se = NA,
      p = NA,
      ci_lower = NA,
      ci_upper = NA))},
  estimand = estimand,
  label = "matching")
matching <- declare_design(
  population,
  potential_outcomes,
  assignment,
  estimand,
  reveal_outcomes,
  estimator_d_i_m,
  estimator_m)

matching_diagnosis <- diagnose_design(
  matching, diagnosands = declare_diagnosands(bias = mean(est - estimand)),
  bootstrap = FALSE, sims = sims)
```

estimand_label	att	att
estimator_label	dim	matching
bias	2.4	0.5

S1.3 Regression Discontinuity

M *Model*: We posit two potential outcomes functions, one for the treatment condition and another for the control. These functions are fourth order polynomial equations that map the running variable X, to the outcome, Y. We suppose that X is drawn from a uniform distribution, and that units experience an idiosyncratic, normally distributed shock. The treatment variable is 1 when the running variable is greater than .5 (the cutoff) and 0 otherwise.

I *Inquiry*: We wish to know the true difference in the potential outcomes functions at exactly the point on the running variable where the cutoff is located.

D *Data Strategy*: We passively observe the available data.

A *Answer Strategy*: Our estimator is a fourth order polynomial regression in which the terms are fully interacted with the treatment variable.

```
cutoff <- .5
control <- function(X) {
  as.vector(poly(X, 4, raw = T) %*% c(.7, -.8, .5, 1))}
treatment <- function(X) {
  as.vector(poly(X, 4, raw = T) %*% c(0, -1.5, .5, .8)) + .15}
population <- declare_population(
  N = 1000,
  X = runif(N,0,1) - cutoff,
  noise = rnorm(N,0,.1),
  Z = 1 * (X > 0))
potential_outcomes <- declare_potential_outcomes(
  Y_Z_0 = control(X) + noise,
  Y_Z_1 = treatment(X) + noise)
estimand <- declare_estimand(LATE = treatment(0) - control(0))
estimator <- declare_estimator(
  formula = Y ~ poly(X, 4) * Z,
  model = lm,
  estimand = estimand)
rdd <- declare_design(
    population, potential_outcomes, estimand, reveal_outcomes, estimator)

rdd_diagnosis <- diagnose_design(rdd = rdd, bootstrap = FALSE, sims = sims)
```

estimand_label	LATE
estimator_label	my_estimator
bias	-0.22
rmse	0.75
power	0
coverage	1
mean_estimate	-0.07
sd_estimate	0.73
type_s_rate	0
mean_estimand	0.15

S1.4 Experimental Design

M *Model*: Our model posits that potential outcomes are a combination of background noise and three additive treatment effects: β_1 arises from treatment 1, β_2 arises from treatment 2, and β_3 arises from the interaction between the two treatments. We posit that the interaction effect varies from 0 to .5.

I *Inquiry*: We wish to know the effect of treatment 1 compared to the outcome when both treatments are absent (i.e. β_1).

D *Data Strategy*: We compare two strategies: one in which no units are assigned to have both treatments, and another in which units are assigned to one of four conditions: no treatment, treatment 1, treatment 2, or both treatments.

A *Answer Strategy*: We regress the outcome on indicators for both treatment conditions.

```
multi_arm_template <-
  function(N, beta_1 = 0, beta_2 = 0, beta_3 = 0, n_conditions = 3) {
    population <- declare_population(noise = rnorm(N), N = N)
    potential_outcomes <- declare_potential_outcomes(
      Y_Z_0 = noise,
      Y_Z_1 = beta_1 + noise,
      Y_Z_2 = beta_2 + noise,
      Y_Z_3 = beta_1 + beta_2 + beta_3 + noise)
    assignment <- declare_assignment(condition_names = 0:n_conditions)
    estimand <- declare_estimand(main_effect = mean(Y_Z_1 - Y_Z_0))
    estimator <- declare_estimator(formula = Y ~ Z1 + Z2,
                                   coefficient_name = "Z1",
                                   model = lm_robust,
                                   estimand = estimand)
    return(declare_design(population,potential_outcomes, assignment,
                          mutate(Z1 = as.numeric(Z %in% c(1, 3)),
                                 Z2 = as.numeric(Z %in% c(2, 3))),
                          reveal_outcomes, estimator, estimand))}
designs <- quick_design(multi_arm_template,
                        N = 500,
                        beta_3 = seq(0, 0.5, length.out = 10),
                        n_conditions = 2:3)


diagnoses <- diagnose_design(designs, sims = k, bootstrap = FALSE)
```

design_ID	estimand_label	estimator_label	bias	rmse	power	interaction
design_1	main_effect	my_estimator	0.027	0.064	0.0	0.000
design_2	main_effect	my_estimator	0.066	0.129	0.2	0.056
design_3	main_effect	my_estimator	0.060	0.080	0.0	0.111
design_4	main_effect	my_estimator	0.064	0.174	0.4	0.167
design_5	main_effect	my_estimator	-0.120	0.121	0.0	0.222
design_6	main_effect	my_estimator	-0.075	0.095	0.0	0.278
design_7	main_effect	my_estimator	-0.008	0.057	0.0	0.333
design_8	main_effect	my_estimator	0.007	0.093	0.0	0.389
design_9	main_effect	my_estimator	-0.035	0.098	0.0	0.444
design_10	main_effect	my_estimator	0.008	0.087	0.0	0.500
design_11	main_effect	my_estimator	0.069	0.109	0.2	0.000
design_12	main_effect	my_estimator	-0.024	0.061	0.0	0.056
design_13	main_effect	my_estimator	0.114	0.140	0.2	0.111
design_14	main_effect	my_estimator	0.108	0.126	0.2	0.167
design_15	main_effect	my_estimator	0.118	0.131	0.0	0.222
design_16	main_effect	my_estimator	0.153	0.168	0.4	0.278
design_17	main_effect	my_estimator	0.102	0.190	0.4	0.333
design_18	main_effect	my_estimator	0.211	0.224	0.4	0.389
design_19	main_effect	my_estimator	0.243	0.245	1.0	0.444
design_20	main_effect	my_estimator	0.296	0.309	1.0	0.500

S1.5 Descriptive Inference

M *Model*: We posit that voters have a latent probability of voting that is realized as actual voting through a probit process. The probability of voting is positively correlated with support for the Democratic candidate. People reveal their true support honestly, but in general are more likely to turnout to vote than they reveal.

I *Inquiry*: We wish to know the true support for the Democratic candidate given that the respondent actually votes.

D *Data Strategy*: We randomly sample 500 respondents.

A *Answer Strategy*: We estimate the true support for the Democratic candidate among those who will vote by taking the mean of stated support for the Democratic candidate among those who indicate they are likely voters.

```
population <- declare_population(
  N = 1000,
  latent_voting = rnorm(N),
  latent_HRC_support = .1 * latent_voting + rnorm(N) - .1,
  voter = rbinom(N, 1, prob = pnorm(latent_voting)),
  HRC_supporter = rbinom(N, 1, prob = pnorm(latent_HRC_support)),
  likely_voter = rbinom(N, 1, prob = pnorm(latent_voting - 2)))
sampling <- declare_sampling(N = 500)
estimand <- declare_estimand(true_support = mean(HRC_supporter[voter == 1]))
estimator <- declare_estimator(HRC_supporter ~ 1,
                               model = lm,
                               subset = (likely_voter == 1),
                               coefficient_name = "(Intercept)",
                               estimand = estimand)
descriptive_inference <- declare_design(population, sampling, estimand, estimator)

descriptive_inference_diagnosis <- diagnose_design(
  descriptive_inference = descriptive_inference,
  diagnosands = declare_diagnosands(bias = mean(est - estimand)),
  sims = sims,
  bootstrap = FALSE)
```

estimand_label	true_support
estimator_label	my_estimator
bias	0.04

S1.6 Bayesian Descriptive Inference

M *Model*: We posit a population of successes and failures generated through a probit process.

I *Inquiry*: We wish to know the true probability of success.

D *Data Strategy*: We sample 10 units.

A *Answer Strategy*: We estimate empirical priors and a posterior distribution using a beta-binomial model. We compare two estimators, one that uses flat priors, and another that uses priors whose probability mass is centered at .5.

```
population <- declare_population(N = 1000,
                                 noise = rnorm(N, -.1, .05),
                                 prob_success = pnorm(noise),
                                 success = rbinom(N, 1, prob_success))
sampling <- declare_sampling(n = 10)
estimand <- declare_estimand(success_prob = mean(prob_success))
beta_binom <- function(data,alpha_0,beta_0){n_successes <- sum(data$success)
n_trials <- length(data$success)
alpha <- n_successes + alpha_0 - 1
beta <- n_trials - n_successes + beta_0 - 1
post <- dbeta(seq(0,1,0.005),alpha,beta)
return(data.frame(
  post_mean = alpha / (alpha + beta),
  prior_mean = alpha_0 / (alpha_0 + beta_0),
  post_sd = sqrt((alpha*beta)/(((alpha+beta)^2)*(alpha+beta+1))),
  prior_sd = sqrt((alpha_0*beta_0)/(((alpha_0+beta_0)^2)*(alpha_0+beta_0+1)))))}
estimator_flat_priors <- declare_estimator(estimator_function = beta_binom,
                                           alpha_0 = 1,
                                           beta_0 = 1,
                                           estimand = estimand,
                                           label = "flat priors")
estimator_info_priors <- declare_estimator(estimator_function = beta_binom,
                                           alpha_0 = 10,
                                           beta_0 = 10,
                                           estimand = estimand,
                                           label = "informative priors")
bayesian_design <- declare_design(
  population, estimand, sampling, estimator_flat_priors, estimator_info_priors)
diagnosands <- declare_diagnosands(
  mean_est = mean(post_mean), mean_sd = mean(post_sd),
  bias = mean(post_mean - estimand), mean_shift = mean(post_mean - prior_mean),
  sd_shift = mean(post_sd - prior_sd))


bayesian_estimation_diagnosis <- diagnose_design(
  bayesian_design, diagnosands = diagnosands, bootstrap = FALSE, sims = sims)
```

estimand_label	success_prob	success_prob
estimator_label	flat priors	informative priors
mean_est	0.46	0.49
mean_sd	0.15	0.09
bias	0.00	0.03
mean_shift	-0.03	-0.01
sd_shift	-0.14	-0.02

S1.7 Discovery

M *Model*: We posit a population with income drawn from a standard uniform distribution, and education that is positively correlated with income. Units also have some background noise.

I *Inquiry*: We wish to know the true direct effect of income on the outcome, Y.

D *Data Strategy*: We passively observe the data.

A *Answer Strategy*: We compare three different estimation approaches. In the first, we regress the outcome on income and education. In the second we regress the outcome on income only. In the third we randomly split the sample into two halves, and use one half of the data to test the above two models, in addition to one in which income and education are interacted on the righthand side. We choose the model that minimizes the Akaike Information Criterion, and estimate the effect using the other half of the data.

```
population <- declare_population(income = runif(N),
                                  education = income + 0.25 * runif(N),
                                  noise = runif(N),
                                  Y = .5 * income + .5 * education + noise,
                                  N = 500)
estimand <- declare_estimand(true_income_effect = 0.5)
estimator_right <- declare_estimator(formula = Y ~ income + education,
                                      coefficient_name = "income",
                                      model = lm,
                                      estimand = estimand,
                                      label = "right model")
estimator_wrong <- declare_estimator(formula = Y ~ income,
                                      coefficient_name = "income",
                                      model = lm,
                                      estimand = estimand,
                                      label = "wrong model")
estimator_split_sample <- declare_estimator(
  model = function(data) {
    split_sample <- sample(0:1, nrow(data), replace = T)
    train <- data[split_sample == TRUE,]
    test <- data[split_sample == FALSE,]
    explorations <-
      list(lm(Y ~ income, data = train),
           lm(Y ~ income + education, data = train),
           lm(Y ~ income + education + income * education, data = train))
    exploration_best <- explorations[[which.min(sapply(explorations, AIC))[1]]]
    exploration_test <- lm(formula(exploration_best), data = test)
    return(exploration_test)},
  coefficient_name = "income",
  estimand = estimand,
  label = "split sample")
discovery <- declare_design(
  population, estimand, estimator_right,estimator_wrong, estimator_split_sample)


discovery_diagnosis <- diagnose_design(
  discovery = discovery, sims = sims, bootstrap = FALSE)
```

estimand_label	true_income_effect	true_income_effect	true_income_effect
estimator_label	right model	split sample	wrong model
bias	0.03	0.20	0.50
rmse	0.17	0.35	0.50
power	0.85	0.70	1.00
coverage	1.0	0.6	0.0
mean_estimate	0.53	0.70	1.00
sd_estimate	0.17	0.29	0.06
type_s_rate	0	0	0
mean_estimand	0.5	0.5	0.5

S1.8 Model based estimands

M *Model*: We posit a population whose potential outcomes are non-linearly but monotonically increasing in some variable, Z.

I *Inquiry*: We wish to know the average change in potential outcomes brought about by increasing Z by one unit.

D *Data Strategy*: We imagine two processes through which the values of Z are assigned: in the first, each value is assigned with equal probabilities; in the second, the highest value is assigned with a lower probability than the lower values.

A *Answer Strategy*: We estimate the effect of an average unit increase in Z through linear regression of the outcome on Z.

```
population <- declare_population(N = 10, u = rnorm(N))
potential_outcomes <- declare_potential_outcomes(Y_Z_1 = 0 + u,
                                                  Y_Z_2 = 3 + u,
                                                  Y_Z_3 = 4 + u)
estimand <- declare_estimand(estimand_function = function(data)  {
  YY <- with(data, c(Y_Z_1, Y_Z_2, Y_Z_3))
  XX <- rep(1:3, each = nrow(data))
  return(coef(lm(YY ~ XX))[2])},
  label = "beta")
assignment_equal  <- declare_assignment(condition_names = 1:3,
                                        prob_each = c(1, 1, 1) / 3)
assignment_unequal  <- declare_assignment(condition_names = 1:3,
                                          prob_each = c(.4, .4, .2))
estimator <- declare_estimator(formula = Y ~ Z,
                               model = lm_robust,
                               estimand = estimand,
                               label = "ols")
model_estimand_equal <- declare_design(
  population, potential_outcomes, estimand, assignment_equal, reveal_outcomes,
  estimator)
model_estimand_unequal <- declare_design(
  population, potential_outcomes, estimand, assignment_unequal, reveal_outcomes,
  estimator)

model_based_estimand_diagnosis <- diagnose_design(
  model_estimand_equal = model_estimand_equal,
  model_estimand_unequal = model_estimand_unequal,
  bootstrap = FALSE,
  sims = sims)
```

design_ID	model_estimand_equal	model_estimand_unequal
estimand_label	beta	beta
estimator_label	ols	ols
bias	0.05	-0.02
rmse	0.37	0.35
power	1.00	0.95
coverage	0.95	1.00
mean_estimate	2.05	1.98
sd_estimate	0.37	0.36
type_s_rate	0	0
mean_estimand	2	2

S2. Further details on survey of design tools

This section describes the construction of the working example used in the research design tool survey, as well as the method used to search for tools to include in the survey, the criteria by which tools were admitted for inclusion into the survey, and the rules for coding the outcomes of this survey. In the online appendix we provide the raw data from the survey, including an overview of the tools considered for inclusion and the reasons for their eventual exclusion, as well as an archive of screenshots of all of the tools included in the survey itself.

S2.1 Working Example

There are 1000 city blocks to choose from, each of which contains exactly 25 or 50 households, with the j'th block size distributed categorically, $n_j \sim \text{Cat}(\{25, 50\}, \{.5, .5\})$. Thus, the size of the sample varies as a function of which five city blocks the researcher randomly samples. Specifically, the expected sample size of the study is $N = 5 \times E[n] = 5 \times 37.5 = 187.5$.

Denoting the treatment variable $Z \in \{0, 1\}$, the i'th household respondent's potential outcomes are determined by the following system of equations

$$y_i = Z_i \alpha_j + \epsilon_i, \tag{1}$$

with

$$\alpha_j \sim \text{N}(\frac{n_j}{100}, .1) \qquad Z_i \sim \text{Bin}(\frac{10}{n_j}) \qquad \epsilon_i \sim \text{N}(0, 1). \tag{2}$$

Note that the size of the block determines respondents' potential outcomes and their probability of assignment to treatment. Specifically, the two are negatively correlated: the larger the respondent's block, the higher her treated potential outcome and the lower her probability of being assigned to the intervention.

The research design is declared and diagnosed using the following code:

```
set.seed(1:7)
population <- declare_population(
  block = level(N = 1000,
                block_size = sample(c(25, 50), N, TRUE),
                block_effect = rnorm(N, block_size / 100, .1)
  ),
  individual = level(N = block_size,
                     noise = rnorm(N)))
potential_outcomes <-
  declare_potential_outcomes(formula = Y ~ block_effect * Z + noise)
sampling <- declare_sampling(clust_var = block, n = 5)
assignment <- declare_assignment(block_var = block, m = 10)
estimand <- declare_estimand(ATE = mean(Y_Z_1 - Y_Z_0))
dim <- declare_estimator(Y ~ Z,
                         model = lm_robust,
                         label = "DIM",
                         estimand = estimand)
bfe <- declare_estimator(Y ~ Z + block,
                         model = lm_robust,
                         label = "BFE",
                         estimand = estimand)
ipw_bfe <- declare_estimator(Y ~ Z + block,
                             model = lm_robust,
                             label = "IPW-BFE",
                             weights = 1 / Z_cond_prob,
                             estimand = my_estimand)
design <- declare_design(
  population, potential_outcomes, my_estimand, sampling, assignment,
  reveal_outcomes,
  dim, bfe, ipw_bfe)

set.seed(1:7)
diagnosis <- diagnose_design(design, sims = 1000, bootstrap = FALSE)
```

This code produces the following diagnosis of the design:

estimand	estimator	bias	rmse	power	coverage	Mean est.	SD est.	type-s rate	mean_estimand
PATE	BFE	-0.027	0.186	0.598	0.930	0.383	0.184	0.000	0.409
PATE	DIM	-0.043	0.185	0.581	0.927	0.366	0.180	0.000	0.409
PATE	IPW-BFE	-0.007	0.186	0.717	0.884	0.403	0.186	0.000	0.409

Table S5: Bias, RMSE, power and coverage of design in working example.

Table S5 illustrates that the DIM and BFE estimators are negatively biased: they tend to underestimate the actual size of the treatment effect. This is because it is rarer for units with high treated potential outcomes to be assigned to treatment, a feature of the design that is not taken into account at all by the DIM estimator, and only through the estimation of a difference in intercepts by the BFE estimator. The IPW-BFE estimator has bias much closer to 0 because it reweights the data to take account of the lower probability with which units in larger blocks are assigned to treatment.

However, the IPW-BFE does not perform strictly better than the BFE estimator in this case. While its power is much higher (72% vs. 58%), this does not result from better efficiency: in fact, the standard deviation of the estimate is higher for the IPW-BFE as a result of the variance introduced by the re-weighting. As the coverage shows, the increased power appears to derive in part from biased variance estimates: the standard errors produced by the IPW-BFE estimator are too small, giving a coverage probability of .88, vs. the more correct coverage probability of the BFE estimator (.93).

In the following sections, we describe the methods by which we sought to assess the ability of available research tools to diagnose these features of the working example design.

S2.2 Search Method

The survey sought to identify computational tools to diagnose the power and bias of the working example design described above. In terms of the identification criteria, we considered any software that promised to design and diagnose prospective research as a candidate for the survey.

We used two principle methods to search for candidates. First, we entered the search terms "statistical power calculator" and "sample size calculator" into the Google web search engine, using an incognito browser window in Google Chrome. We assessed the first 30 results using these terms. Second, we assessed the tools listed in four reviews of the literature, namely Kreidler et al. (2013); Guo et al. (2013); Groemping (2016); Green and MacLeod (2016).

Using these two methods, we identified 143 candidate tools.

S2.3 Admissability Criteria

From the 143 candidate tools, we admitted 30 into the survey. We only admitted those tools that were specifically promised to calculate power or bias in a general purpose way, or in a way that was tailored to the working example. In other words, we excluded tools that were able to calculate power or bias but only for very specific designs that could not accommodate the working example. For instance, the `R` package `ThreeArmedTrials` was a candidate for inclusion because it was listed in the literature review by Groemping (2016) and promised to calculate power of experimental designs. However, because the tool was specifically set up to calculate the power of clinical non-inferiority or superiority trials, we excluded it from consideration in the survey. We also excluded research tools that serve to design research but are not set up to diagnose power or bias. For example, the `experiment` package is set up to design and analyze treatment effects in randomized experiments, but does not provide means for calculating power or bias of designs.

S2.4 Coding Rules

Tools that were included in the survey were coded according to what information on a design they employed to calculate diagnosands (principally bias and power). Some tools accommodated information on design aspects (i.e., block sizes) but did not use this information in the calculation of diagnosands. Tools were only coded as employing a given piece of information if it was included in the calculation of diagnosands.

- *Effect sizes:* When rounded to the third decimal place, the PATE is $\approx .406$ with a standard deviation of 1.01, producing a Cohen's d of approximately .4. Thus, when a tool asked for an effect size without specifying what kind of effect, we entered a value of .4. Sometimes tools require an expression of the effect size in terms of Cohen's f^2. Unlike Cohen's d, the calculation of the f^2 requires that effects be specified in the context of a multivariate regression, and is thus difficult to calculate *a priori*. To calculate the f^2 in this context, we use the companion software to generate 500 R^2 under the full (block FE + treatment) and restricted (block FE only) models, and take the average of the f^2. This is perhaps overly generous to the assessed tools, as the f^2 estimated in this way encodes important design information that the tools do not ask for (such as the assignment probabilities).

- *Heterogeneous block sizes:* 1 if tool allows user to specify that units are organized into groups of different sizes, 0 otherwise.
- *Effect sizes correlated with block sizes:* 1 if tool allows user to specify that effects are correlated with group size, 0 otherwise.
- *Non-constant variance control vs. treatment:* 1 if tool allows for different variances in treatment vs. control, 0 otherwise.
- *Estimand:* 1 if tool allows user to formally define estimand as the Population Average Treatment Effect, 0 otherwise.
- *Sampling strategy:* 1 if tool allows user to specify anything about the strategy via which units are selected from the population into the sample, 0 otherwise.
- *Assign m within blocks:* 1 if tool allows users to specify that exactly m units will be assigned to treatment in the j'th block, 0 otherwise.
- *Inverse-probability weights:* 1 if tool allows users to specify that observations will be weighted by the inverse of their conditional assignment probability during estimation of effects, 0 otherwise.
- *Block fixed-effects:* 1 if tool allows users to specify that a block-level fixed-effect will be estimated, 0 otherwise.
- *Covariate adjustment:* 1 if tool allows users to account for conditioning on covariates, 0 otherwise.
- *Power of DIM:* the estimated power of the difference-in-means estimator if the tool is able to estimate it, NA otherwise.
- *Power of BFE:* the estimated power of the block fixed-effects estimator if the tool is able to estimate it, NA otherwise.
- *Power of IPW-BFE:* the estimated power of the inverse probability-weighted block fixed-effects estimator if the tool is able to estimate it, NA otherwise.

- *Bias:* the estimated bias of any of the estimators if the tool is able to estimate it, NA otherwise.
- *Coverage:* the estimated coverage of any of the estimators if the tool is able to estimate it, NA otherwise.

S3. Replicating Bjorkman and Svensson (2009) with DeclareDesign

This document presents a "design replication" of Björkman and Svensson (2009) using `DeclareDesign`, by which we mean an exercise in which we learn about the design of a study that has already been conducted. Note that a design replication requires making assumptions about expected features of the data generation processes as well as treatment effects; researchers can disagree on these features. The design replication provides information on features of the design conditional on these assumptions. This exercise is intended to demonstrate how careful specification of estimands can shed light on – and quantify – otherwise hard to assess limitations of analytic strategies.

The study reports the results of a cluster-randomized trial of the effects of community-based monitoring of health clinics in Uganda. The unit of assignment is the health clinic but measurement takes place at the level of the household. Households are considered treated if they are located within the catchment area (5km radius) of a treated health clinic.

The experiment focuses on improvements in two main health outcomes: reductions in child mortality and increases in child weight. The first outcome is measured as the catchment-area-level under-5 mortality rate, expressed in death rates per 1000 live births. In the control group, this rate was 144, compared with 97 in the treatment group: a 33% reduction in child mortality. The second outcome (measured at the household level) is the weight-for-age of infants, defined as children under 18 months. Weight-for-age is measured in standard units, so the positive 0.14 coefficient estimate implies that the weight-for-age of infants in the treatment group was 0.14 standard deviations higher.

We will now characterize this design using MIDA.

S3.1 Model

The population of interest comprises the households within the catchment areas of the 50 health clinics. When we declare the population, we will create three background covariates, two at the household level and one at the catchment area level.

1. `infant`: indicator that equals one if an infant was born into a household in the 18 months preceding the treatment. This variable is observable.

2. `family_health`: a normally distributed variable that represents the health of the household. This variable is likely to be unobservable. We cannot measure it, but it will be positively correlated with the `weight_for_age` of surviving children.

3. `area_health`: a normally distributed variable that represents the overall health of the community. This variable will be the same for all households living within a catchment area and will ensure that outcomes are correlated within catchment area. This variable is also unobservable.

The data are hierarchical – there are 2500 households in each of 50 clusters. The resulting 125,000 row dataset is the population from which subjects will be sampled.

```r
# Estimated probability of having a child
infant_prob <- (1135 / (1 - 0.1205)) / 5000

pop <- declare_population(
  catchment_area = level(N = 50,
                         area_health = rnorm(N)),
  households = level(N = 2500,
                     infant = rbinom(n = N, 1, prob = infant_prob),
                     family_health = rnorm(N)))
```

The two outcomes of interest are infant mortality and infant weight. We will first build the infant mortality potential outcomes with a custom function. This custom function builds the probability of an infant surviving in terms of a logistic model, then draws from a binomial distribution using the resulting probabilities. We assume that there is a base rate of survival of approximately 86%, and that treatment increases the probability of survival by approximately 5 percentage points. In logits, this is moving from `invlogit(1.81) = 86%` to `invlogit(1.81 + 0.5) = 91%`. The probability of survival is also positively correlated with the latent health of the household and the health of the community. Finally, if a household does not have an infant, then this potential outcome is undefined. We denote treatment status as $Z = 0$ for control and $Z = 1$ for treatment, hence the condition labels `Z0` and `Z1`.

```
alive_po_function <- function(Z, family_health, area_health, infant) {
  alive <- rbinom(n = length(Z),
                  size = 1,
                  prob = invlogit(
                    logit(0.86) + 0.5 * Z + family_health + area_health))
  alive[infant == 0] <- NA
  return(alive)}

pos_alive <- declare_potential_outcomes(
  condition_names = c(0, 1),
  formula =
    infant_alive ~ alive_po_function(Z, family_health, area_health, infant))
```

The second potential outcome is the `weight_for_age` of surviving infants. This potential outcome is equal to the latent health of the household for control units. The treated potential outcome is the sum of the latent health and the 0.14 standard deviation treatment effect. Finally, this outcome is masked if the infant dies or if the household does not have an infant.

```
weight_po_function <-
  function(Z, infant_alive_Z_0, infant_alive_Z_1, family_health, area_health){
    weight <- 0.14 * Z + family_health + area_health
    masked <- infant_alive_Z_1 * Z + infant_alive_Z_0 * (1 - Z)
    weight[(masked) == 0 | is.na(masked)] <- NA
    return(weight)}

pos_weight <- declare_potential_outcomes(
  condition_names = c(0, 1),
  formula =
    weight_for_age ~ weight_po_function(Z, infant_alive_Z_0, infant_alive_Z_1,
                                        family_health, area_health))
```

S3.2 Inquiry

We have two inquiries, the average effect on child mortality (at the cluster level) and the average effect on weight-for-age at the household level. The first estimand is built by first subsetting the data to the households with infants, then aggregating the potential outcomes up to the cluster level. We obtain cluster-level mortality rates under each condition, then take the difference.

Type	Alive (Z = 0)	Alive (Z = 1)	Weight (Z = 0)	Weight (Z = 1)	Estimand
A	1	0	exists	NA	undefined
B	0	1	NA	exists	undefined
C	0	0	NA	NA	undefined
D	1	1	exists	exists	E[Weight(Z=1) - Weight(Z=0)]

```
cl_mortality_estimand <- declare_estimand(estimand_function = function(data){
  cluster_df <-
    aggregate(cbind(infant_alive_Z_0, infant_alive_Z_1) ~ catchment_area,
              FUN  = mean,
              data = subset(data, infant == 1))
  with(cluster_df,(1 - mean(infant_alive_Z_1)) - (1 - mean(infant_alive_Z_0)))},
  label = "Mortality")

hh_weight_estimand <- declare_estimand(estimand_function = function(data){
  with(subset(data, infant_alive_Z_0 == 1 & infant_alive_Z_1 == 1),
       mean(weight_for_age_Z_1 - weight_for_age_Z_0))},
  label = "Weight")
```

The second estimand has a complication – it is only defined for a subset of the population. The table below shows four types of infants: Type A (for "Adverse") is alive if in control, but dies if in treatment. Type B ("Beneficial") is just the reverse: the child dies if untreated, but survives if treated. Type C ("Chronic") would die under either condition and Type D ("Destined") would live under either condition. For the first three types, the child dies under one condition, the other or both. This means that the difference in weight potential outcomes is undefined for those types. The difference in weight due to treatment is only defined for Type D infants, those who would survive under either treatment. We therefore define the estimand as being the difference in outcomes for Type D.

This estimand is not recoverable from this design, as we cannot distinguish type A from type D in the control group and type B from type D in the treatment group.

```
hh_weight_estimand <- declare_estimand(estimand_function = function(data){
  with(subset(data, infant_alive_Z_0 == 1 & infant_alive_Z_1 == 1),
       mean(weight_for_age_Z_1 - weight_for_age_Z_0))},
  label = "Weight")
```

S3.3 Data Strategy

Our data strategy includes both the stratified sampling of households by catchment areas and the random assignment of catchment areas to treatment or control. Since the target is 5,000 total households, the study samples 100 households from each catchment area. Assignment to treatment is straightforward: 25 of the 50 clusters receive treatment.

```r
sampling <- declare_sampling(strata_var = catchment_area, prob = 100/2500)

assignment <- declare_assignment(clust_var = catchment_area)
```

S3.4 The Estimator Functions

The two estimands require different estimation procedures. For the mortality estimand, we first aggregate the data up to the cluster level, then take the difference in cluster means.

```r
cl_mortality_estimator <- declare_estimator(model = function(data){
  cluster_df <- aggregate(cbind(infant_alive, Z) ~ catchment_area,
                          FUN  = mean,
                          data = data)
  lm_robust((1 - infant_alive) ~ Z, data = cluster_df)},
  estimand = cl_mortality_estimand,
  label = "Mortality")
```

The second estimand is at the household level, but we must nevertheless cluster our standard errors by the catchment area. Note that we estimate this quantity among all observed values of `weight_for_age`. In the control group, the observed values are a mixed of types A and D, and in the treatment group, the values are a mixture of types B and D. Ideally, we would subset the estimation to include only Type D households, but this information requires knowledge of both the treated and untreated potential outcomes, which is impossible. If potential outcomes are correlated with type (as they are in this simulation), this estimator is biased.

```r
hh_weight_estimator <- declare_estimator(weight_for_age ~ Z,
                                         model = lm_robust,
                                         clusters = catchment_area,
                                         estimand = hh_weight_estimand,
                                         label = "Weight")
```

S3.5 Diagnosis

We now provide the diagnose_design() function with the declarations we made above. We will draw a large finite population once, then for each simulation, draw a stratified sample, allocate treatments, reveal outcomes, and conduct the estimation.

```
fixed_pop <- pop()
bjorkman_svensson_design <- declare_design(
  fixed_pop,
  pos_alive, pos_weight,
  cl_mortality_estimand, hh_weight_estimand,
  sampling, assignment,
  reveal_outcomes(outcome_variable_names = c("infant_alive", "weight_for_age")),
  cl_mortality_estimator, hh_weight_estimator)

diagnosis <- diagnose_design(
  design = bjorkman_svensson_design, sims = 2000, bootstrap = FALSE)
```

Estimand	Bias	RMSE	Power	Coverage	Mean Estimate	Mean Estimand
Mortality	0.000	0.035	0.390	0.952	-0.060	-0.06
Weight	-0.071	0.229	0.064	0.941	0.069	0.14

The summary of the diagnose output is presented in the table above. Considering the under-5 mortality rate first, we see that the true population average treatment effect is -0.06 percentage points. In our simulations, we estimate the true standard error to be 0.035, which is close to the standard error reported in the original paper of 0.026. The coverage is correct, at 95%. This simulation shows that we are relatively under-powered: only 39.1% of simulations returned a statistically significant result.

Turning next to the weight-for-age analysis, the simulations reveal that our estimator is biased. Because we built into our potential outcomes the assumption that less-health infants were the ones who are most likely to be of type B ("Beneficial"), the treatment group mean is pulled down. Under this assumption, the bias is downwards – our analysis systematically understates the effect on weight-for-age among type D infants, the only type for whom the estimand is defined.

In light of this diagnosis, and given the assumptions made here, how should the study have been designed differently? The main point to arise from this exercise is that the weight-for-age estimand is not defined for all households. A design approach to address this might be to find a pre-treatment covariate that could predict type (A, B, C, or D), so that the weight-for-age estimation would take place only among those predicted to be type B. An example of such a covariate might be age at treatment, as the mortality effects appear to be strongest among younger infants. Another approach would be to define the estimand as the cluster level average weight-for-age, and ignore whether changes in this are due to improvements in weight of particular infants or changes in the composition of infants.