Declaring and Diagnosing Research Designs[†]

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

First draft: 5/7/2016
This draft: 10/20/2017 16:09

Abstract

The evaluation of research depends on assessments of the quality of underlying research designs. Such assessments are often stymied by a lack of a common understanding of what consistitutes a design. 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 (M), an inquiry (I), a data strategy (D), and answer strategy (A). Once a design is formally declared in computer code, Monte Carlo techniques can be used 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, Justin 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*a*).

[‡] 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`

Authors and readers of empirical research routinely need to assess the properties of research designs. In doing so, they face two challenges.

First, few tools exist for the comprehensive assessement of the properties of designs. At one extreme, researchers resort to rudimentary power calculators with sometimes hidden assumptions or rely on rules of thumb that may not incorporate important idiosyncratic features of the research setting. At the other extreme, some scholars conduct fully-fledged simulations requiring advanced programming skills beyond the capabilities of many applied researchers. General-use tools for assessing important properties of designs such as statistical power or bias are not available.

Second, surprisingly little attention has been paid to the more fundamental question of what constitutes a design. This lack of clarity carries risks both before and after the implementation of a study. If designs are incompletely specified ex ante, it is difficult for researchers to assess their strengths and improve them. If designs are incompletely specified at the time of analysis, researchers may choose inappropriate analysis procedures or worse, may only report the most attractive model from a set of possible specifications. If designs remain unspecified after analysis, it may be difficult for other scholars to replicate a study or to judge whether a given type of reanalysis is justified.

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).

In addition, we 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 "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 computer code objects and, second, *diagnose* the statistical properties of the design relying on this declaration. We provide an implementation of 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 them as part of a preanalysis plan or a funding request. Whether or not the declaration and diagnosis serves this purpose, the process of generating them 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 replication, and contribute to re-analysis decisions.

Authors can assess the properties of diagnosand-complete research designs in order to improve designs before implementation. Readers and replication authors can diagnose designs before or after implementation in order to select and critique studies based on their designs rather than their results. 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 research communities. A standard set might include power, bias, root mean-squared error, and coverage. Others who 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.

Diagnosis can be executed analytically for simple designs or through Monte Carlo simulation for more complex designs. We provide an algorithm for simulation that produces diagnosand estimates as well as bootstrapped estimates of simulation uncertainty

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

[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.

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 principal 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 a definition of research design put forth by King, Keohane and Verba (1994, KKV), which contains the following four components: 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. As the basis for such a procedure we extend the KKV 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 modelling approach with the potential outcomes framework as presented in Imbens and Rubin (2015), which helps to clarify the inferential quantities – estimands – a research design targets.

1.1 Design elements

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 prob-

abilistic 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 $P_M(a)$.

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_A, 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 $P_M(a|D, A)$.

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

[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.

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.

M and D capture the the analysis-relevant features of a design; they do not describe substantive elements, such as how interventions are implemented or how outcomes are measured. Yet many other aspects of a design that are not explicitly labeled in these features nevertheless 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 potential outcomes function 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 actual average treatment effect (ATE) and the estimated 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.

		Required:			
Diagnosand	**Description**	M	I	D	A
Power	Probability of erroneously failing to reject null hypothesis	✓		✓	✓
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	✓	✓	✓	
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)	✓	✓	✓	
Imbalance	Expected distance of covariates across treatment conditions (Mahalanobis, 1936; Gu and Rosenbaum, 1993)	✓		✓	
Type S Rate	Probability that the replicated estimate has the incorrect sign, if it is statistically significant (Gelman and Carlin, 2014)	✓	✓	✓	✓
Exaggeration Ratio	Expected ratio of absolute value of estimate to estimand, if statistically significant (Gelman and Carlin, 2014)	✓	✓	✓	✓

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

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.

Diagnosands can also be defined for properties that reach beyond classical statistics. For example one might define a study as “conclusive” if some evidence is observed, whether or not formal hypothesis tests are conducted, 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 a study will be considered conclusive, an inference considered robust, or an intervention deemed to have value for money. The three diagnosands depend on observed data only. More subtle analogues can be defined that also make use of potential data, for example one might define a diagnosand as the chances that an intervention is *correctly* considered 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.[3]

In current practice, research designs are commonly declared in terms of the elements required to implement the study. The set of features that are often described include the assignment procedure including sampling and treatments as well as the estimation strategy.

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 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 does not require a statistical test, but it does require the specification of an estimand.

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.

Our notion of diagnosand-completeness does not encompass all of the information relevant to research design, it simply conveys the minimum information required to enable assumption-based diagnosis of a design's inferential properties.[4]

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

[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

2. Existing Methods for Diagnosing Research Designs

While relatively simple, the notion of diagnosand-completeness highlights the inadequacy of many common approaches to inspecting a design's quality.

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); Monte Carlo simulation code, written by researchers to diagnose each study; 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 survey of over 143 computational tools available to social science researchers, considering any software that offered computational methods to diagnose research designs. 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) and Green and MacLeod (2016). Of the 143 tools identified as candidates in this manner, 30 tools were able to diagnose specific inferential properties of designs, such as their power, and so were retained for the survey. Supporting Materials Section S2 provides further details on the methods employed to identify, admit and code the tools included in this survey.

We assess these tools' ability to assist a hypothetical researcher designing a study whose **I**nquiry focuses on the impact of a door-to-door voter mobilization campaign on voter turnout.

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.

Her budget allows her to treat ten households in each of five randomly selected city blocks, which vary in their total size. Turnout is measured for all households based on a voter file. Her **M**odel of the world stipulates that the effectiveness of the mobilization campaign may vary systematically depending on the size of the block. Her **D**ata strategy involves assigning 10 households in blocks of different sizes, and she suspects that she may need to account for the differential assignment probabilities this procedure induces. On the other hand, she is aware that using a probability-weighted estimator as her **A**nswer strategy may reduce efficiency. She is thus unsure whether to pre-register an estimator that simply conditions on block-level fixed effects (BFE), or one that inversely weights observations by their conditional probability of being assigned to their observed treatment condition (IPW-BFE).

Before conducting her study the researcher thus seeks to answer three diagnostic questions:

1. For both estimators what is the power – the probability that the statistical test will lead the researcher to reject the null hypothesis of no effect if the treatment truly (under the model) does affect outcomes?
2. What is the bias of both estimators with respect to the average treatment effect among the population of interest?
3. Given the answers to 1 and 2, is the researcher better off using the BFE or IPW-BFE approach?

Using the tools surveyed for this article, the researcher would be unable to answer any of these questions correctly, because they 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; sampling strategy; exact randomization procedure; formal definition of the estimand as the population average treatment effect; or the use of inverse-probability weighting estimation techniques.[5] 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 show below that these diagnosands shed light on tradeoffs that, while not immediately obvious, are consequential for the inferences provided by the design.

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

[5] The one tool (GLIMMPSE) that was able to account for the fact that the researcher intends to assign exactly m units within blocks of varying size 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*a*). 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 cluster-sample five groups, block-assign ten units in each group to the treatment, reveal the outcomes that obtain under the given random assignment, and then estimate the treatment effects using the three proposed estimation strategies. We then used the parameters from this simulation exercise 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 (13/20) 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. In effect, the tools all assume an unbiased estimator. This is problematic because 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 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 more powered because it produces biased variance estimates that lead to troublingly low coverage probabilities. 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.[6] 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 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*a*) . 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

[6] 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, D)`

	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_2 <- p_Z(d_1)`
		`d <- reveal_outcomes(d_2)`

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 - a_M`

	Design Diagnosis (m draws)	**Code**
	Declare a diagnosand	`bias <- declare_diagnosand(bias = mean(t))`
	Calculate a diagnosand	`diagnose_design(D, diagnosand = bias, sims = m)`

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

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 construct a confidence interval to make a decision 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 this confidence interval reflects both estimation uncertainty (simulation error) and fundamental uncertainty (true variability in the diagnosand, for example across possible population draws).[7]

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 `www.DeclareDesign.org`. The website also contains instructions for implementing this framework in Stata. Section 5 of the supplementary materials provides 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*. Declaring the model, which seemingly it is the aim of the researcher to learn about, may seem to beg the question. Yet declaring a model is unavoidable. In practice it is already familiar to researchers who conduct power calculations, which require effect sizes. Once a diagnosand-complete design is declared, researchers can assess the sensitivity of diagnosis to alternative models and strategies. 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.

Moreover, objects created through a common computer language are directly comparable. Not only can the performance of a design be analyzed with respect to different data, but alternative designs used in different research studies can in principle be compared head to head on the same dataset, if the variables and computer syntax used to create them are common. Having declared a design in code, the computer object can also be reused throughout the lifecycle of a project: the code used to simulate random sampling can be used to implement sampling when the study is launched, and the same functions used to declare the assignment mechanism in a randomized trial can be used to randomize treatment assignment given sample data, and also to implement analysis given outcome data.

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.

Process Tracing. While many qualitative researchers employ frameworks that may seem

[7] 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.

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 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 Supplementary Materials 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 $X \in \{\text{No strong middle class}, \text{Strong middle class}\}$ and the presence or absence of the outcome $Y \in \{\text{No revolution}, \text{Revolution}\}$. A types only have revolutions when there is no middle class, B types only have revolutions when there is a middle class, C types never have revolutions, and D types always have them. A clue $K \in \{\text{Revolution not financed by domestic industry}, \text{Revolution financed by domestic industry}\}$ is generated with probability .2 only if the case is a B type. 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 supporting Materials 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.

Observational Regression-Based Strategies. Many observational studies seek to make causal claims but do not explicitly employ the potential outcomes framework, instead describing inquiries 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 problem if we want to assess the properties of strategies under different assumptions about data generation if 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 $f(x)$).[8] 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.

In Supporting Materials Section S1.8, we declare a design 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. 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

[8] 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.

different probabilities.

Matching. In much observational research, assignment processes are not known. In matching designs, the data strategy typically does not include intervention in assignment nodes by the researcher; they are taken as given. Using observable traits to reconstruct assignment propensities, matching procedures seek to establish conditions under which as-if random assumptions can be made. Diagnosis in such instances can be helpful as a tool to explore the conditions under which such assumptions are justified. In Supplementary Materials Section S1.2 we declare a design with a **M**odel in which three observable random variables are combined in a probit process that assign the treatment variable, Z. The **I**nquiry pertains to the average treatment effect of Z on the outcome Y, which we estimate using an **A**nswer strategy that reconstructs the assignment process (possibly erroneously) 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 group- or sample-level 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 selecting a caliper around this threshold within which to sample units so as to maximize efficiency while reducing bias. The **A**nswer strategy is a simple linear regression in which the assignment and running variables are linearly interacted. In Supplementary Materials 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 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 average treatment effect (PATE).

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 using the R package DeclareDesign in Supplementary Materials Section S1.4. Neither design exhibits bias when the true interaction term is equal to zero (Figure 1 left panel). 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-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

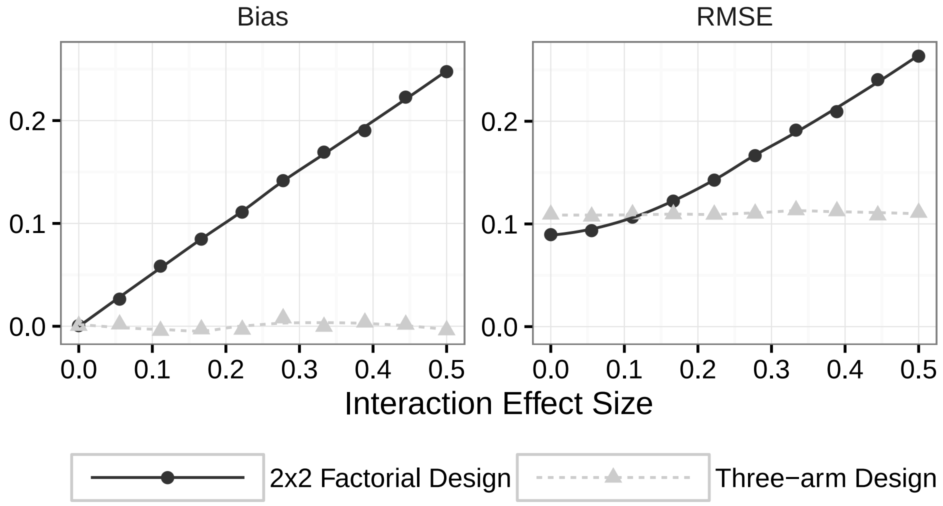

Figure 1: Diagnoses of Designs with Factorial or Three-Arm Assignment Strategies Illustrate a Bias-Variance Tradeoff. Bias (left) and root mean-squared-error (right) are displayed for two assignment strategies, a 2 × 2 treatment arm factorial design (solid lines; circles) and a three-arm design (dark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

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 Designs

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.

Descriptive Inference. In Supplementary Materials Section S1.5, 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 the example, analysts 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 Supporting Materials 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.

In Supplementary Materials Section S1.7, we given an example of a design declaration for 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 (Fafchamps and Labonne, 2016). 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 greater transparency places a premium on considering alternative analysis strategies at early stages of research projects, not only because it reduces researcher discretion, but also 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.

Moreover, as illustrated in section 2, existing power calculators are surprisingly rudimentary: they handle a very small set of special cases, and often do not show how power varies as a result of the many design choices a researcher must make besides sample size. No general tools exist for assessing power or other equally important properties, such as unbiasedness, mean squared error, or coverage. By making the various steps of the prospective design concrete through computer-based simulation, the procedures we have described make clear in advance *which* choices need to be made conditional on data collection, and what the inferential consequences of those choices are.

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.

However, the effectiveness of design registries in reducing the scope for fishing depends on clarity over which elements must be included in a precommitment document. In practice some registries rely on various 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.

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 with — provides incentives for researchers to be clear and transparent in their analysis strategies, and can build confidence in the robustness of findings.[9]

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-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. We provide an example of a design replication of a study for which data is currently not available in Blair et al.

[9] 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	A	B
Alternative **A**nswer strategy	C	D

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.

(2016*b*). In that replication we illustrate how the strategy employed by Björkman and Svensson (2009) could under some reasonable data generating processes give rise to biased results. We emphasize that this exercise does not *demonstrate* bias. Rather, it helps locate possible sources of bias latent in the design.

Table 4 illustrates situations that may arise. In a declared design an author might specify situation A: 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 C are better than for A then this gives grounds to switch to C. 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 B are as good as in A but are better in situation D than in situation C this provides a robustness argument for altering estimation strategies.

- **Model plausibility.** If the diagnostics in situation A are better than in situation B, but the diagnostics in situation D are better than in situation C, 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.

As an illustration of the application of these principles, consider a situation in which a researcher produces an estimate of an average treatment effect. A critic notes that the treatment is highly correlated with a covariate, not included in the original analysis, and that significance is lost once the control is included. The researcher might then counter that although results are sensitive to the inclusion of the control, the new strategy does not satisfy home ground dominance—that is, given prior assumptions about the model, the diagnostics from the new estimation strategy are not better than those from the original strategy. The critic could then describe an alternative model and demonstrate either that the new strategy is more robust to alternative models or that it is preferable on the basis of model plausibility—for example by using the data to demonstrate that the covariate is prognostic of potential outcomes contrary to researcher assumptions. In all cases, transparent arguments can be made by formally comparing the original design to a modified design.

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.[10] 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. 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).

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 assess-

[10] 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).

ment at all. Our method allows researchers to fully characterize, and thus accurately diagnose their designs. 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 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*a*. "DeclareDesign Version 1.0." Software package for R, available at http://declaredesign.org.

Blair, Graeme, Jasper Cooper, Alexander Coppock and Macartan Humphreys. 2016*b*. "Using DeclareDesign to Replicate Bjorkman and Svensson (2009).".
URL: *http://declaredesign.org/replications/bjorkman-svensson.html*

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.).".

Fafchamps, Marcel and Julien Labonne. 2016. Using Split Samples to Improve Inference about Causal Effects. Technical report National Bureau of Economic Research Working Paper No. 21842.

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. Formally p_S is a component of p_Z, though it is given the special attention paid to it in many studies. In the example below each unit generated by p_X is sampled with 10% probability. Again `my_sampling` describes a strategy and not an actual sampling.

```
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.

```
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.[11] 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.

```
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 potential outcomes functions that correspond to

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

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 using '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 regression as well as a set of associated statistics, including the standard error, p-value, and the confidence interval.

```
my_estimator <- declare_estimand(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*a*), we implement it for the widely-used `R` platform.

Declaring and Diagnosing Research Designs

Supplementary Materials

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

```r
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)
```

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

estimand_label	is_B
estimator_label	my_estimator
truth	0.51
mean_guess	0.55
bias	0.04
bias_given_K_seen	-0.5
bias_given_no_K_seen	0.04

S1.2 Matching

```
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	2.39	0.52

S1.3 Regression Discontinuity

```
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	-0.05
rmse	0.89
power	0.05
coverage	0.95
mean_estimate	0.1
sd_estimate	0.89
type_s_rate	0.02
mean_estimand	0.15

S1.4 Experimental Design

```
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	0.023	0.108	0.04	0.000
design_2	main_effect	my_estimator	0.004	0.101	0.01	0.056
design_3	main_effect	my_estimator	-0.007	0.113	0.06	0.111
design_4	main_effect	my_estimator	-0.007	0.104	0.01	0.167
design_5	main_effect	my_estimator	0.018	0.106	0.02	0.222
design_6	main_effect	my_estimator	0.005	0.115	0.08	0.278
design_7	main_effect	my_estimator	-0.003	0.117	0.06	0.333
design_8	main_effect	my_estimator	-0.006	0.114	0.07	0.389
design_9	main_effect	my_estimator	0.017	0.099	0.04	0.444
design_10	main_effect	my_estimator	-0.015	0.104	0.06	0.500
design_11	main_effect	my_estimator	0.003	0.092	0.06	0.000
design_12	main_effect	my_estimator	0.040	0.096	0.06	0.056
design_13	main_effect	my_estimator	0.079	0.129	0.18	0.111
design_14	main_effect	my_estimator	0.087	0.121	0.16	0.167
design_15	main_effect	my_estimator	0.095	0.130	0.18	0.222
design_16	main_effect	my_estimator	0.135	0.158	0.27	0.278
design_17	main_effect	my_estimator	0.153	0.180	0.36	0.333
design_18	main_effect	my_estimator	0.190	0.209	0.58	0.389
design_19	main_effect	my_estimator	0.226	0.244	0.71	0.444
design_20	main_effect	my_estimator	0.260	0.274	0.84	0.500

S1.5 Descriptive Inference

```r
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)))
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, estimand, estimator)
```

```r
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	0.02

S1.6 Bayesian Descriptive Inference

```r
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))
```

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

estimand_label	success_prob	success_prob
estimator_label	flat priors	informative priors
mean_est	0.46	0.48
mean_sd	0.14	0.09
bias	0.00	0.02
mean_shift	-0.04	-0.02
sd_shift	-0.15	-0.02

S1.7 Discovery

```
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	0.00	0.13	0.50
rmse	0.19	0.35	0.50
power	0.77	0.60	1.00
coverage	0.95	0.70	0.00
mean_estimate	0.50	0.63	1.00
sd_estimate	0.19	0.32	0.05
type_s_rate	0	0	0
mean_estimand	0.5	0.5	0.5

S1.8 Model based estimands

```
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	-0.01	0.13
rmse	0.40	0.44
power	0.98	0.97
coverage	0.95	0.96
mean_estimate	1.99	2.13
sd_estimate	0.40	0.42
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

A researcher wants to learn the average treatment effect of door-to-door campaigning on attitudes toward a prominent political issue, taking the entire adult population of the city as the population of interest. Her budget allows her to select five city blocks in which to conduct the campaign, and in each block to randomly assign ten households to the campaign and the rest to the control. Following the intervention, she plans to conduct a survey among randomly selected members of the households in the five study blocks in order to measure attitudes toward the social issue. She compares three estimators: difference-in-means through univariate linear regression (DIM); block-level fixed-effects regression (BFE); and block-level fixed-effects regression with observations inversely weighted by the probability that they were assigned to the treatment condition they are in (IPW-BFE).

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:

```r
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 S13: Bias, RMSE, power and coverage of design in working example.

Table S13 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) and 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.